WordPress – SICHERHEIT – Experten Tipps

WordPress absichern wie ein Profi – Der komplette Guide

Wie Du WordPress absichern kannst wie ein Profi | Experten Tipps

WordPress absichern 2022 ✔ Professionelle Tipps zur echten WordPress Sicherheit vom Experten ✔ Schritt für Schritt Aktualisiert: 17.05.2023

Es kursieren sehr viele gut gemeinte Tipps im Netz, wie man WordPress absichern kann. Viele von ihnen taugen leider nicht viel. Denn echte WordPress Sicherheit gibt es nicht mit der einfachen Installation eines Plugins. Es ist ein Konzept von Maßnahmen, die aufeinander aufbauen. In diesem Beitrag zeige

ich Dir, wie Du Dein WordPress bombensicher machst.

Inhaltsverzeichnis Anzeigen

Wenn Dir wirklich etwas an der WordPress Sicherheit liegt, dann solltest Du alle existierenden Sicherheitslücken schließen. Das kannst Du jedoch nur, wenn Dir bewusst ist, über welche Wege Dein WordPress angegriffen werden kann.

Erst dann leuchten die Maßnahmen ein und erst dann wird Dir bewusst, dass es keine Sicherheit mittels Plugin-Installation geben kann. Als **langjährige Experten** in der WordPress Sicherheit geben wir Dir heute Hintergrundwissen und eine Anleitung zur Absicherung Deines WordPress. Übrigens: Bis heute wurde keine Website gehackt, die wir abgesichert haben.

Dieses Tutorial ist nur für fortgeschrittene Anwender gedacht und **nicht für Anfänger**. Du musst Dich auskennen mit FTP und der functions.php.

Auch interessant:

Cloud Sicherheit – Wie Du Dropbox und Co absichern kannst

WordPress Sicherheitslücken

Klären wir doch mal die wichtige Frage, über welche Wege WordPress überwiegend gehackt wird (und gehackt werden kann).

1. **Sehr leicht zu merkende und viel zu kurze Passwörter** (!)
2. **Veraltete WordPress-Versionen** – Mit jeder neuen Version werden die Sicherheitslücken der alten bekannt
3. **Veraltete Plugin-Versionen** – Auch Plugins haben eklatante Sicherheitslücken.
4. **Brute-Force Angriffe** gegen den Admin-Zugang
5. **Brute-Force Angriffe** gegen die xmlrpc.php Datei
6. **SQL-Injektionen** über Formulare
7. **Von außen zugängliche** WordPress-Dateien

8. Sicherheitslücke **WordPress REST-API (Update 26.05.2022)**

Zu 1: – WordPress Sicherheit fängt mit Deinem Passwort an

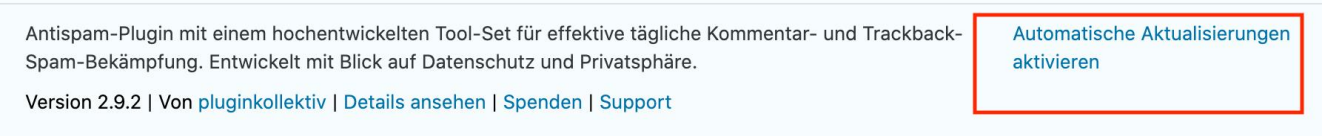
WordPress absichern ohne ein richtig gutes und wirklich sicheres Passwort hat leider überhaupt keinen Zweck. Alles, was leicht zu merken ist, ist auch leicht zu knacken. Und das wäre fatal. Deshalb sorge für ein anständiges Passwort aus Buchstaben, Zahlen, Sonderzeichen und Groß- und Kleinschreibung.

Ein gutes Passwort sollte schon 30stellig sein. Merken kann man sich das nicht mehr, aber es gibt ja Passwortmanager oder die entsprechenden Funktionen im Webbrowser.

Passwort-Generator aufrufen (externer Link)

Zu 2 + 3: – Die Updates

Das Du **WordPress und die Plugins aktuell halten** solltest und die Updates so schnell wie möglich ausführen solltest, hast Du bestimmt schon gelesen. Aber lesen bringt nichts. **Du musst es tun!** Ansonsten bettelst Du darum, gehackt zu werden. Zudem werden gern Plugins eingesetzt, die als beständig unsicher gelten – zum Beispiel der Revolution Slider. Übrigens kannst Du ab WordPress 5.5 Deine Plugins automatisch aktualisieren lassen.



Zu 4 + 5: – Brute-Force Angriffe

Hier versucht man mit der Brechstange Deine Zugangsdaten zu bekommen. Es werden zum Teil Tausende Variationen von Benutzernamen und Passwort ausprobiert. Diese Angriffe haben

immer wieder Erfolg, weil der Benutzername meistens Admin ist und das Passwort kurz und gut zu merken ist.

Gern wird auch ein Angriff gegen die xmlrpc.php Datei ausgeführt, die zum Beispiel dazu dient, Beiträge per E-Mail veröffentlichen zu können. Auch über diese Datei kann man einen Vollzugriff auf die Website bekommen.

Was ist ein Brute-Force Angriff? (externer Link)

Zu 6: – SQL-Injektionen über Formulare

In ungeschützte Formulare (und auch direkt in der Adresszeile des Browsers) wird gern versucht Schadcode einzubringen. Hat das Erfolg, werden die Besucher Deiner Seite bereits durch einen einfachen Aufruf der Website mit Viren und Trojanern verseucht. Du wirst es erst merken, wenn Dein Webhoster die Website abschaltet oder Google die Seite aus dem Index nimmt.

Was ist eine SQL-Injektion? (externer Link)

Zu 7: – Von außen zugängliche WordPress-Dateien

Nicht jeder Webhoster hat eine sichere Konfiguration seiner Hosting-Pakete oder Server. Manchmal sind WordPress-Dateien von außen zugänglich. Beliebte Angriffsziele sind hier zum Beispiel die install.php und die wp-config.php

Zu 8: – Die WordPress REST-API

Die REST-API bietet viele Möglichkeiten Inhalte auszulesen und diese können dann an externe Apps oder Websites übergeben werden. Dazu stellt die API strukturierte Daten (JSON) öffentlich zur Verfügung. Dazu gehören jedoch auch Daten, die man nicht gern für jedermann öffentlich abrufbar sehen möchte. Dazu solltet Ihr den vollständigen Artikel lesen, es gibt erstens noch viel mehr Informationen dazu und zweitens ein

umfangreicheres Code-Beispiel.

Als kleines Goodie habe ich Dir noch ein Plugin geschrieben, das Du im Artikel herunterladen kannst.

WordPress REST-API Sicherheitslücke deaktivieren

Ein Code-Beispiel, das die REST-API für externe Besucher abschaltet

```
<?php
/* Ab hier kopieren */
/**
 * REST-API fuer extere User abschalten
 */
add_filter('rest_authentication_errors', function($result) {
if ( ! is_user_logged_in() ) {
return new WP_Error( 'rest_API_cannot_access', array( 'status'
=> rest_authorization_required_code() ) );
}
return $result;
});
```

PHP

Copy

WordPress absichern. Echte WordPress Sicherheit!

Du solltest die folgenden Arbeiten immer mit einem **FTP-Zugang** erledigen, niemals in den Editoren von WordPress. Diese gehören abgeschaltet, weil sie ein extremes Sicherheitsrisiko darstellen.

Wie das geht, erfährst Du weiter unten.

Die Snippets sind geeignet für:

- **WordPress-Version:** Ab 4.5 – inklusive 5.5.xx

- **PHP-Version:** inkl. PHP 7.4.xx

Am Ende dieses Artikels hast Du alle Sicherheitslücken geschlossen und kannst dich an einer sicheren Website erfreuen. Als spezialisierte SEO Agentur wissen wir, wovon wir sprechen. Wir führen Dich Schritt für Schritt durch die einzelnen Punkte.

Die Basis der Sicherheit. Eine perfekte .htaccess Datei

Seit mittlerweile 9 Jahren entwickle ich eine .htaccess Datei und habe sie jedes Jahr stets verbessert und überarbeitet. Sie ist die Grundlage einer guten Sicherheitsstrategie und sorgt zudem noch für einen enormen Performance-Schub für Dein WordPress.

Folgendes wird abgesichert:

- Alle wichtigen WordPress-Dateien und Ordner gegen Zugriff von außen
- Dank ausgeklügelter Firewall Schutz vor SQL-Injektionen
- Schutz gegen die Ausnutzung von eventuellen Sicherheitslücken in Plugins
- Schutz gegen die Einschleusung von Schadsoftware jeder Art
- Schutz gegen Brute-Force Angriffe auf Uploads-Ziele
- Setzt HTTP-Response Header für Browser-Sicherheit
- Sperrt die xmlrpc.php Datei gegen jeden Zugriff

Den Adminbereich von WordPress absichern

Der Adminbereich ist das Herz Deiner Website und sollte so sicher wie nur möglich sein. Das erreichen wir durch drei wichtige Schritte. Alle drei Maßnahmen sorgen dafür, dass sich Hacker die Zähne ausbeissen und keine Chance mehr haben, über

diesen Weg in Deine Website einzudringen.

1

Teil 1: Eine zusätzliche Passwortabfrage – HTTP Authentifikation

Eine HTTP Authentifikation ist eine sehr wirkungsvolle Sache. Bevor man nicht die korrekten Zugangsdaten eingegeben hat, kommt man nicht an den Adminbereich von WordPress und kann sich demzufolge auch nicht einloggen. Diese zusätzliche Passwortabfrage ist schnell eingerichtet.

Du benötigst dafür **einen FTP-Zugang** zu Deinem Webhosting und ein **FTP-Programm** wie zum Beispiel FileZilla.

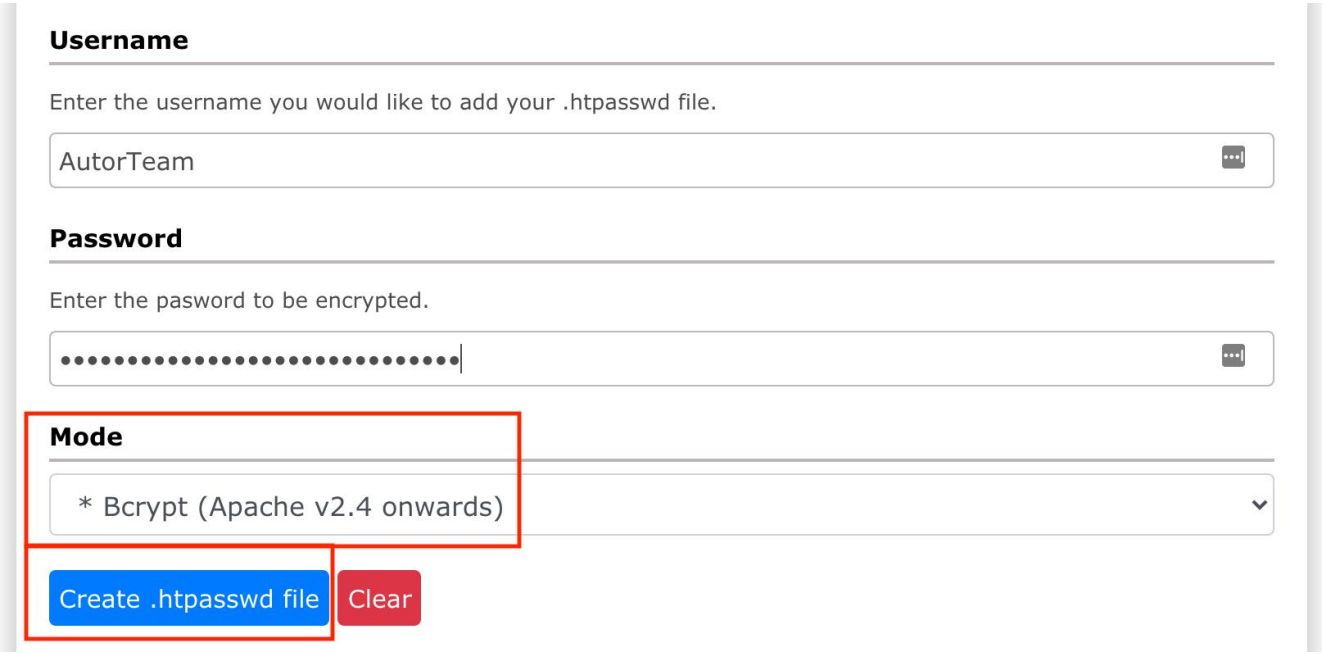
.htpasswd erstellen

Um diese Abfrage einzurichten benötigst Du erstens die obige .htaccess Datei und eine Datei namens .htpasswd, die Du erstellen musst. Beide Dateien sind versteckte – oder Systemdateien – die normalerweise nicht angezeigt werden. Du musst die Anzeige von versteckten Dateien also aktivieren.

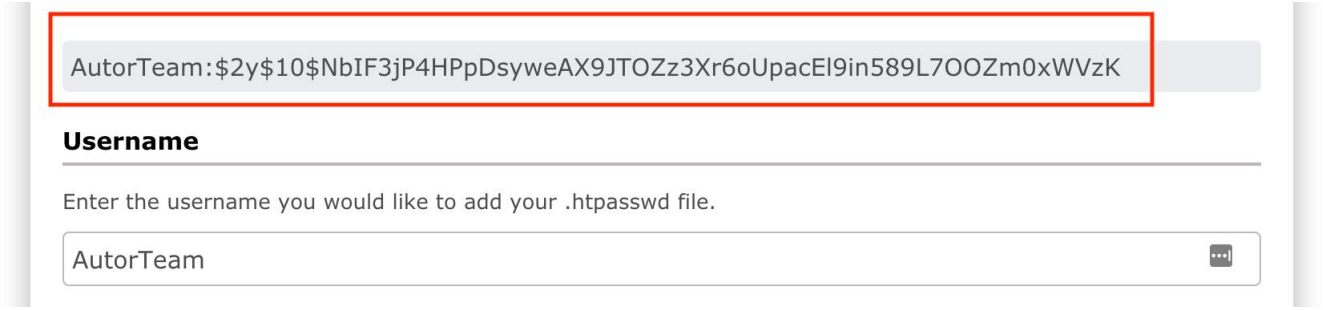
Lege nun mit dem Editor von Windows oder TextEdit von macOS eine reine Textdatei mit dem Namen .htpasswd an.

Erzeuge jetzt mit dem Passwort-Generator ein sicheres Passwort. Es sollte mindestens 25stellig sein. Notiere Dir das Passwort und rufe jetzt den .htpasswd Generator auf. Gib einen Benutzernamen Deiner Wahl ein und das soeben generierte Passwort.

Stelle bei »Mode« **Bcrypt** ein. Siehe Screenshot. Das sorgt für eine ziemlich gute Verschlüsselung des Passworts. Danach klicke auf den blauen Button.



Die dadurch erstellten Zugangsdaten findest Du oberhalb von Username.



Kopiere diese Zeile und füge sie in Deine .htpasswd Datei ein. Speichere die Datei ab und lade sie mit dem FTP-Programm in das Hauptverzeichnis von WordPress.

Jetzt muss der korrekte und vollständige Server-Pfad zur .htpasswd ermittelt werden.

Server-Pfad ermitteln

Um den vollständigen Server-Pfad zur Datei zu ermitteln, nutzen wir eine kleine PHP-Datei. Erstelle mit einem Text-Editor eine Datei namens dir.php und kopiere folgendes hinein:

```
<?php
$dir = dirname(__FILE__);
echo "<p>Der vollständige Pfad zur .htpasswd Datei in diesem Verzeichnis: " . $dir . "/.htpasswd" . "</p>";
```

PHP

Copy

Lade diese Datei nun in das Hauptverzeichnis von WordPress und rufe die Datei im Browser auf:

https://deine-website.de/dir.php

HTTP

Copy

Kopiere den angezeigten Pfad und notiere Ihn. Er sieht so aus:

/usr/local/www/apache24/noexec/deinewebseite/.htpasswd

HTTP

Copy

Dieser Pfad muss nun in die .htaccess eingetragen werden. Wenn Du meine Datei nutzt, ist der betreffende Block relativ weit unten zu finden. Du musst vor dem Code die Rauten # entfernen, um ihn nutzen zu können.

So muss es nachher aussehen:

```
# ----------------------------------------------------------
----------
#   Protect your WordPress Login with HTTP Authentification
# ----------------------------------------------------------
----------

# If you want to use it, comment it out and set your path to
.htpasswd
<Files wp-login.php>
 AuthName "Admin-Bereich"
 AuthType Basic
                                                AuthUserFile
/usr/local/www/apache24/noexec/deinewebseite/.htpasswd
 require valid-user
</Files>
```

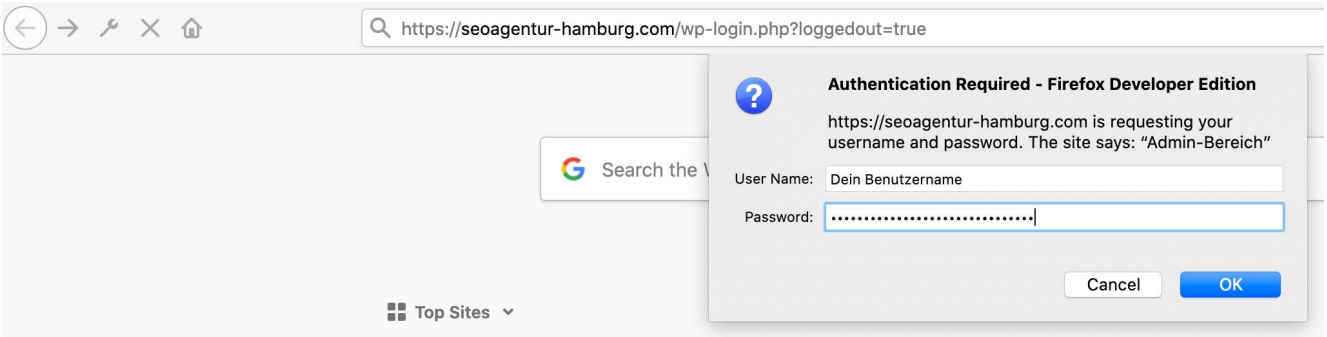
Apache Configuration

Copy

WICHTIG: Lösche jetzt die dir.php wieder vom Server. Sie stellt ein Sicherheitsrisiko dar.

Lade jetzt die .htaccess Datei wieder auf Deinen Server hoch. Jetzt sollten sich beide Dateien (.htaccess und .htpasswd) im Hauptverzeichnis von WordPress befinden.

Wenn Du jetzt Deinen Adminbereich aufrufst – egal ob mit wp-login.php oder wp-admin – kommt die folgende Passwortabfrage:



Übrigens musst Du die Zugangsdaten nur einmal eingeben, danach befindet sich die Abfrage im Browser-Cache. Erst wenn dieser gelöscht wird, kommt die Abfrage erneut.

2

Teil 2: WordPress absichern: Zugang nur noch mit E-Mail-Adresse

Ein kleines Code-Snippet mit großer Wirkung. Hacker probieren allen möglichen und unmöglichen Benutzernamen aus, bevorzugt natürlich »Admin«, weil er so weit verbreitet ist. Hat ein Hacker Deinen Benutzernamen, braucht er nur noch Dein Passwort.

Daher sorgen wir dafür, dass er garantiert nicht Deinen Benutzernamen bekommt. Weil es ihn nicht mehr gibt. Denn statt dem Benutzernamen kannst Du Dich nur noch mit Deiner E-Mail-

Adresse und dem Passwort einloggen.

Kopiere den folgenden Code in die functions.php Deines (Child-) Themes. Du kannst für die Snippets auch ein eigenes Plugin anlegen.

```
<?php

// Ab hier kopieren

/**
 * Sicherheit: Anmeldung nur noch mit E-Mail-Adresse, anstatt Benutzernamen
 *
 * @author Andreas Hecht
 */

//WordPress Authentifikation löschen
remove_filter('authenticate', 'wp_authenticate_username_password', 20);

// Neue Authentifikation setzen - Anmelden nur mit E-Mail und Passwort
add_filter('authenticate', function($user, $email, $password){

    //Check for empty fields
        if(empty($email) || empty ($password)){
            //create new error object and add errors to it.
            $error = new WP_Error();

            if(empty($email)){ //No email
                        $error->add('empty_username', __('<strong>FEHLER</strong>: Das E-Mail Feld ist leer.'));
            }
                        else  if(!filter_var($email, FILTER_VALIDATE_EMAIL)){ //Invalid Email
                    $error->add('invalid_username', __('<strong>FEHLER</strong>:  Die  E-Mail-Adresse  ist ungültig'));
            }
```

```
            if(empty($password)){ //No password
                        $error->add('empty_password',
__('<strong>FEHLER</strong>: Das Passwort-Feld ist leer.'));
            }

            return $error;
        }

        //Check if user exists in WordPress database
        $user = get_user_by('email', $email);

        //bad email
        if(!$user){
            $error = new WP_Error();
                                $error->add('invalid',
__('<strong>FEHLER</strong>: Deine Eingaben sind ungültig.'));
            return $error;
        }
        else{ //check password
            if(!wp_check_password($password, $user->user_pass,
$user->ID)){ //bad password
                $error = new WP_Error();
                                $error->add('invalid',
__('<strong>FEHLER</strong>: Deine Eingaben sind ungültig.'));
                return $error;
            }else{
                return $user; //passed
            }
        }
}, 20, 3);
```

PHP

Copy

3

Teil 3: Redirect auf Google nach

falscher Eingabe der Zugangsdaten

Mit diesem Code-Snippet wirst Du garantiert jeden Hacker verblüffen, der es doch bis zum Adminbereich geschafft hat. Einmal die Zugangsdaten falsch eingegeben, und schon ist Google Dein bester Freund.

```
<?php

// Ab hier kopieren
if ( ! function_exists( 'ah_redirect_after_login_errors' ) ) :
/**
 * Redirect auf Google nach falscher Eingabe der WP-Zugangsdaten
 */
function ah_redirect_after_login_errors() {
  wp_redirect( 'https://www.google.de' );
  exit;
}
add_filter( 'login_errors', 'ah_redirect_after_login_errors' );
endif;
```

PHP

Copy

WordPress absichern mit den richtigen Einstellungen für die wp-config.php

Die wp-config.php Datei an sich haben wir ja schon mit der .htaccess abgesichert. Jetzt kommen noch wichtige Einstellungen in diese WordPress-Steuerungsdatei hinein.

Der korrekte Platz für unsere Eintragungen ist **oberhalb** der define('WP_DEBUG', false); Konstante.

Nutze die Sicherheitsschlüssel!

Die Sicherheitsschlüssel sorgen für eine Verschlüsselung Deiner Zugangsdaten während des Logins. Nutzt Du keine, werden die Zugangsdaten unverschlüsselt übertragen.

```
<?php


/**#@+
 * Sicherheitsschlüssel
 *
 * Ändere jeden untenstehenden Platzhaltertext in eine beliebige,
 * möglichst einmalig genutzte Zeichenkette.
 * Auf der Seite {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}
 * kannst du dir alle Schlüssel generieren lassen.
 * Du kannst die Schlüssel jederzeit wieder ändern, alle angemeldeten
 * Benutzer müssen sich danach erneut anmelden.
 *
 * @since 2.6.0
 */
define('AUTH_KEY',                                ')tr/o>x!>CD+@VV4EH}TAmm+i[!]f4|r.>K@MCo/,wDkBq^`c_0t9>fkgPnO?;g');
define('SECURE_AUTH_KEY',
'Zw!x0qEni%?OdHHs*s[kRF3ULD~xw*iCWO9F6oyzdL}}8%e2>+{Cd@a~`2>wQ-S|');
define('LOGGED_IN_KEY',
'W<De;xTff~PE?^xXlE{vkN{O$mOlSIz`4za`cYk/;-<<&/hC>a.Q1!k`mK>HE6bQ');
define('NONCE_KEY',                        'qH_9<.w&fC6$YON~WK`zge#iuc3~<WPLD5nF;Bdl8:+G)2+s_vzk&bVC79C2>?b');
define('AUTH_SALT',
'X*2OOu?q)JhQ3=NUumf[(I^u?|sH|>vY?r^:XPJLW+w7JCYeakqAjtjnI{_h~1a');
define('SECURE_AUTH_SALT',
'0wyeDI|N[]8}U<m[>g{]MhVA@WA|*<h}=j9i2vM)3m%`a/gtVSoH7>mb|cN2VL/');
```

```
define('LOGGED_IN_SALT',                'U]y/VEz<pP$-
+r0Iv^.CGBSh$.zI;~HSp:p0xtb9YMN%46${^F>?Bd!xrm$y}^bq');
define('NONCE_SALT',                '-|~?0 Hs%`,Ce$d+Oo#.mw
D5MW<7aI`Of]:gkp`r6S}tJfumjn2jvQsJqz-vgvM');
```

PHP

Copy

Die folgende Website generiert Dir die Schlüssel:

https://api.wordpress.org/secret-key/1.1/salt/

2

Schalte die Editoren für Theme und Plugins ab

> *In jeder WordPress-Installation kann man Theme- und Plugin-Dateien direkt im Adminbereich bearbeiten. Unter den Menüpunkten »Design« und »Plugins« findet man auch jeweils den Editor für die betreffenden Dateien. Dieser Editor ist sehr gefährlich, wenn er in die Hände eines Hackers gerät.*

```
<?php


/**
 *
 * Files Editoren abschalten
 *
 */
define('DISALLOW_FILE_EDIT', true);
```

PHP

Copy

3

Login in den Adminbereich nur über HTTPS

Sollte selbsterklärend sein. Wenn Deine Website HTTPS nutzt, sollte auch kein HTTP-Login in den Adminbereich möglich sein.

```
<?php

// Forciere das Anmelden mit SSL
define('FORCE_SSL_LOGIN', true);

// Adminbereich nur Nutzbar mit SSL
define('FORCE_SSL_ADMIN', true);
```

PHP

Copy

4

Datenübertragung nur mit FTPS

Die Datenübertragung von Deinem Rechner zum FTP-Zugang Deiner Website sollte ausschliesslich mit FTPS erfolgen. Tut es das nicht, werden Deine Zugangsdaten unverschlüsselt an den Server übertragen. Das wäre ein enormes Sicherheitsrisiko.

```
<?php

//FTP nur über SSL
define('FTP_SSL', true);
```

PHP

Copy

Extra: Du hast einen Blog mit mehreren

Autoren?

Dann solltest du Deine Autoren daran hindern, einfache Passwörter zu verwenden. Hier kommt ein Code-Snippet, dass Deine Autoren daran hindert, ihre Passwörter zu ändern.

```
<?php

//Ab hier kopieren
/**
 * Sicherheit: User davon abhalten, ihre Passwörter zu ändern
 *
 * @author Andreas Hecht
 */
class Password_Reset_Removed
{

  function __construct()
  {
      add_filter( 'show_password_fields', array( $this, 'disable' ) );
      add_filter( 'allow_password_reset', array( $this, 'disable' ) );
  }

  function disable()
  {
    if ( is_admin() ) {
      $userdata = wp_get_current_user();
      $user = new WP_User($userdata->ID);
      if ( !empty( $user->roles ) && is_array( $user->roles ) && $user->roles[0] == 'administrator' )
        return true;
    }
    return false;
  }

}

$pass_reset_removed = new Password_Reset_Removed();
```

PHP

Copy

Entfernen der XML-RPC Schnittstelle aus dem HTML-Header der Website

Die gefährliche Datei xmlrpc.php haben wir ja bereits mit der .htaccess Datei gesperrt, jetzt entfernen wir diese Schnittstelle noch aus dem HTTP-Response Header. Der Code kommt in die functions.php.

```
<?php

//Ab hier kopieren
if ( ! function_exists( 'AH_remove_x_pingback' ) ) :
/**
 * Entfernen der XML-RPC Schnittstelle aus dem HTML-Header der Website
 */

 function AH_remove_x_pingback( $headers )
 {
 unset( $headers['X-Pingback'] );
 return $headers;
 }
add_filter( 'wp_headers', 'AH_remove_x_pingback' );
endif;
```

PHP

Copy

Kleines FAQ zur WordPress

Sicherheit

Bringt es was, wenn ich die wp-config.php verschiebe?

Nein. Außer das Du Deine Website fehleranfälliger gemacht hast nicht. Hacker finden die Datei, auch wenn Du sie verschiebst. Das bringt absolut nichts.

Was bringen Sicherheitsplugins wie WordFence, Sucuri etc.

Absolut nichts. Sie gaukeln Dir eine Sicherheit vor, die sie nicht erfüllen können. Diese Plugins versprechen Sicherheit, weil sie Deine WP-, Theme- und Plugin-Dateien auf Schadsoftware scannen. Wenn Du gehackt wurdest, manipuliert der Hacker zuerst diese Plugins. Denn er will ja, dass der Hack möglichst lange unentdeckt bleibt. Zudem sorgen diese Plugins noch dafür, dass Deine Website deutlich langsamer wird. Wenn Du Sicherheit willst, dieser Artikel ist die Anleitung dazu.

Ich brauche keine WordPress Absicherung. Ich habe Limit Login Attemps!

Klasse. Ehrlich. Einen Hacker im ersten Lehrjahr kannst Du damit erschrecken. Profis werden vor Lachen auf dem Fußboden liegen. Warum? Das Plugin limitiert die Loginversuche von EINER bestimmten IP-Adresse. Profis hingegen greifen Dich mit einem Botnetz an. Da prasseln dann Tausende von Anfragen an Deinen Adminbereich von Tausenden von IP-Adressen ein. Wenn von jeder IP nur ein Hackversuch kommt, kann das Plugin nichts stoppen. Im Grunde ist es vollkommen wirkungslos.

Soll ich explizite Dateiberechtigungen auf dem Server setzen?

Hmm, kannst Du schon machen. Aber ob das wirklich praktikabel ist, ist die zweite Sache. Ab und an brauchen Plugins bestimmte Berechtigungen, um zu funktionieren. Auch Updates müssen ohne Probleme laufen. Natürlich kann man sagen, dass man durch Dateiberechtigungen die Manipulation der Dateien von Außen unterbindet.

Im Prinzip wäre das nützlich. Aber Du hast durch meine .htaccess ja schon den Zugriff auf die wichtigsten Dateien gesperrt. Wenn ich auf die Dateien nicht zugreifen kann, kann ich sie auch nicht manipulieren.

WordPress absichern durch das Abändern des Benutzernamens?

Auch erfahrene WordPress Webworker wie Perun empfehlen Dir, den Standard »Administrator« oder »Admin« in einen anderen Benutzernamen abzuändern. Manche gehen einen Schritt weiter und empfehlen Dir, den ersten Admin zu löschen und vorher einen weiteren Admin mit eigenem Benutzernamen anzulegen, um die #ID 1 gegen eine #ID 2 auszutauschen.

Kann dieser Tipp meine Website sicherer machen?

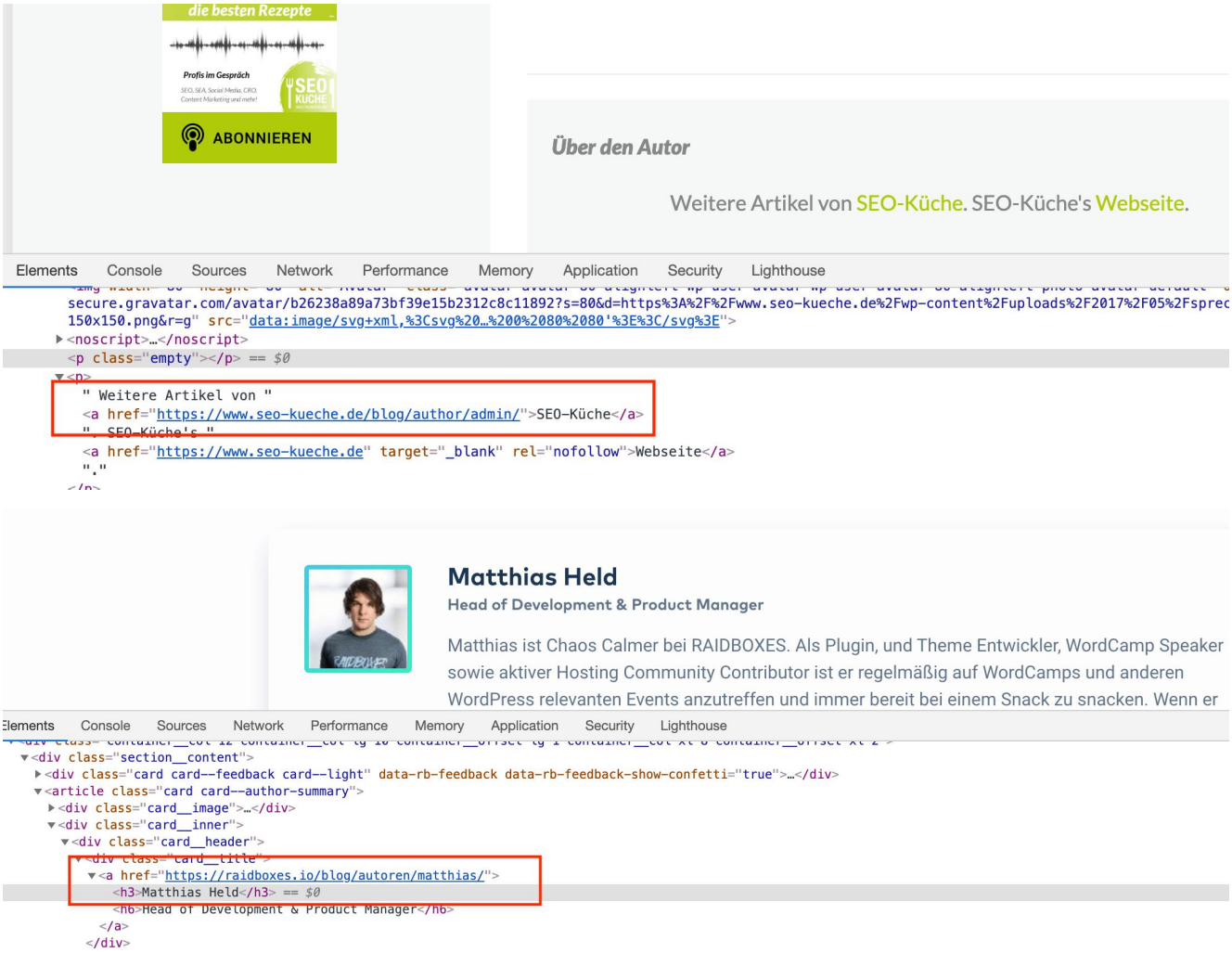
Nein. Der Tipp zeugt von absolut fehlender Sachkenntnis oder von nicht durchdachter Problemstellung. Der Benutzername des Administrators kann innerhalb von Sekunden herausgefunden werden.

Denn jede Autor-Box unter den Beiträgen und jedes Autoren-Archiv in WordPress gibt den Benutzernamen preis. Solltest Du also mit einem Admin-Account Beiträge schreiben, geben alle zwei Möglichkeiten Deinen Admin-Benutzernamen preis.

Wenn alles nichts bringt, kann der Benutzername auch im

Quelltext der Kommentare gefunden werden. Ups…

Zwei Beispiele:



https://gist.github.com/seoagentur-hamburg/c96bc796764baaa64d43b70731013f8a#file-htaccess

eine automatische Aktualisierung durchführen

„wp-config-php“ lässt sich durch den Eintrag

```
define( ‚WP_AUTO_UPDATE_CORE‘, true );
```

Plugins automatisch aktualisieren

```
add_filter( 'auto_update_plugin', '__return_true' );
```

Themes automatisch aktualisieren

```
add_filter( 'auto_update_theme', '__return_true' );
```

Du bist auf der Suche nach einer seriösen SEO Agentur?

Dir hat unser Artikel gefallen und Du möchtest unsere Hilfe in Anspruch nehmen? Dann melde dich bei unverbindlich bei uns. Wir freuen uns auf Deine Anfrage!

+49 40 – 209 659 47info@seoagentur-hamburg.com

Jetzt weitere interessante Beiträge lesen

WordPress

Google Fonts Download: Den Google Font lokal laden

vor 1 Jahr

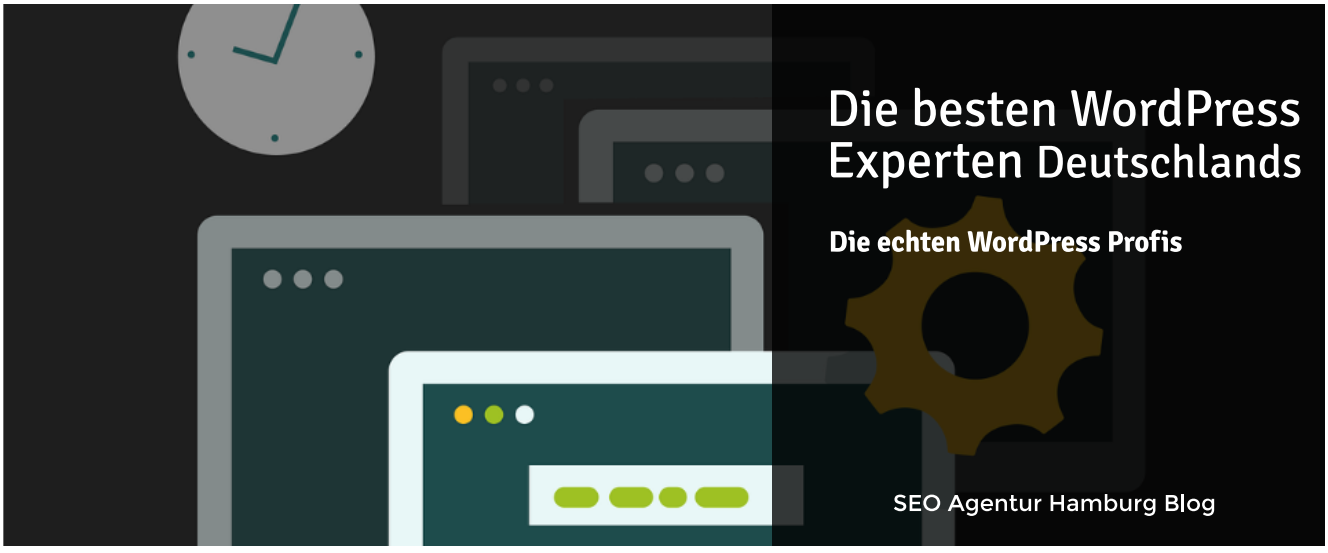
Google Schriften zu verwenden ist sehr beliebt. Doch ein Google Font verursacht erhebliche DSGVO Probleme, da Daten in die USA…

WordPress

Wie Du ein WordPress Child Theme erstellen kannst für Anfänger

vor 3 Jahren

Um zu vermeiden, dass ein Theme-Update eigene Änderungen überschreibt, lohnt es sich, ein WordPress-Child-Theme zu erstellen. Denn ein Child Theme…



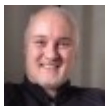
WordPress

Die 10 besten WordPress Spezialisten Deutschlands?

vor 4 Jahren

Ich wurde vor einiger Zeit im Rahmen einer Spezialistenempfehlung als einer der zehn besten WordPress Spezialisten Deutschlands von der Website…

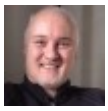
39 Kommentare. Hinterlasse eine Antwort

- Markus16. Februar 2023 10:34 Hallo Andreas, nochmals herzlichen Dank für diese vielen Infos hier.
  Ich habe festgestellt, dass nach Entfernen der XML-RPC Schnittstelle aus dem HTML-Header der Website die Seite nicht mehr erreichbar ist. Als ich den Eintrag aus der wp-config entfernt hatte, lief es wieder. Kennst du das Phänomen bzw. hast du eine Idee, woran es liegen könnte?
  Antworten
  -  Andreas Hecht16. Februar 2023 14:32 Hi Markus,ich habe im Artikel nichts davon geschrieben, dass der Code zum Entfernen der XML-RPC Schnittstelle in die wp-config.php hinein soll. Lesen hilft in solchen Fällen ungemein.
    Antworten

- Isa1. Februar 2023 19:39 Hallo Andreas,
  Danke für diese Anleitung. Ich habe diese umgesetzt und alles Funktioniert bis auf eine Kleinigkeit: Ich benutze deine htaccess Datei und habe anschließend die zusätzliche Passwortabfrage (HTTP Authentifikation) mit reingenommen. Sobald ich mich anmelde komme ich rein,

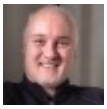
allerdings sobald ich den Browser neustarte muss ich die Daten erneut eingeben (die Login Daten speichern sich anscheinend nicht im Cache ab. Kann es sein das es was mit der htaccess Datei zu tun hat, da diese ja den Cache komprimiert? Den Cache vom Browser löschen hat nichts gebracht. Übrigens nutze ich All-Inkl als Hoster. Ich finde den Fehler nicht. Hast du da einen Tipp? Antworten

-  Andreas Hecht1. Februar 2023 20:43 Hi Isa,das hört sich für mich an, als ob der Cache des Browsers beim beenden geleert wird. Antworten
  -  Isa4. Februar 2023 9:40 Danke für die Schnelle Antwort. Allerdings ist es nicht nur auf einem Gerät und Browser so sondern bei allen die ich jetzt ausprobiert habe. Eine Lösung habe ich dazu noch nicht gefunden. Antworten

-  Frank19. September 2022 1:09 Toller Artikel! Funktionieren die Snippets auch mit WordPress 6.0.2. ? Antworten
  -  Andreas Hecht25. September 2022 15:11 Hallo Frank,ja, das tun sie. Antworten

-  Frank16. September 2022 1:28 I’m Snippet „REST-API fuer extere User abschalten“ hat sich eine fehlerhafte Klammersetzung eingeschlichen.
„return $result;“ wird nie ausgeführt und der Filter

liefert demzufolge nichts zurück, wenn die Bedingung nicht zutrifft. Antworten

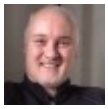
-  Andreas Hecht16. September 2022 14:27 Hi Frank,bei meinem Test wird genau das gewünschte Ergebnis erreicht. Was genau sollte da nicht funktionieren? Antworten

  -  Frank19. September 2022 23:54 Hallo Andreas,also wenn du zweimal hintereinander ein „return“ laufen lassen willst wie in deinem Beispiel oben, dann kann das 2. return doch nie erreicht werden, weil das erste return die gesamte Funktion verlässt und alles danach schlicht nicht mehr ausgeführt wird.Die Funktion macht formal in folgender Form weitaus mehr Sinn:

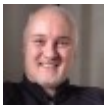
    ```
    add_filter('rest_authentication_errors', function($result) {
    if ( ! is_user_logged_in() ) {
    return new WP_Error( 'rest_API_cannot_access', array( 'status' => rest_authorization_required_code() ) );
    }
    return $result;
    });
    ```

    … und zwar darum, weil add_filter() sich von add_action() in WP dadurch unterscheidet, dass add_filter einen Wert entgegennimmt, ihn modifiziert (oder auch nicht) und dann wieder zurückgibt. Deine Funktion oben gibt aber praktisch immer dann gar nichts zurück, wenn die Bedingung nicht greift, also im konkreten Fall: wenn du eingeloggt bist. Ein

eingeloggter Benutzer wird daher NIE eine REST-Error zu sehen bekommen, auch dann nicht, wenn es einen gibt. Mag schon sein, dass dann alles funktional erscheint, aber wenn Fehler, die auftreten, nicht rückgemeldet werden, muss noch lange nicht alles in Ordnung sein …
Ich hoffe, das hilft. Die Klammer ist einfach verrutscht – keine große Sache. Antworten

-  Andreas Hecht25. September 2022 15:12 Hi Frank,okay, das hatte ich nicht bedacht. Danke für Deine Mühe, ich ändere das Snippet ab. Antworten

-  Joachim2. September 2022 10:21 Hallo Andreas, ich hoffe, du kannst mir helfen: Es geht um Teil 1, eine zusätzliche Passwortabfrage. Die Schritte habe ich alle ausgeführt und die Eingabemaske erscheint auch bei mir. Allerdings geht es nach der Eingabe der Zugangsdaten nicht weiter, sondern die Maske erscheint einfach erneut und es geht nicht weiter. Verhindert eventuell die Ninja-Firewall die Ausführung? Ich würde mich freuen, wenn du helfen kannst. Mein Hoster ist All-Inkl. Antworten

  -  Andreas Hecht2. September 2022 14:14 Hi Joachim,schalte mal diese Firewall komplett ab. Diesen Mist brauchst Du nicht mehr. Wenn es dann noch nicht funktioniert, stimmt etwas mit dem Pfad

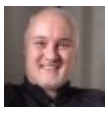
zur .htpasswd nicht. Antworten

-  Heiko26. Mai 2022 10:04 Hallo Andreas, diese Seite wurde am 25.05.2022 aktualisiert, hat sich inhaltlich was geändert? Bei der Gelegenheit möchte ich mal DANKE sagen für das Know-How, das du hier kostenlos mit uns teilst. Antworten
  -  Andreas Hecht26. Mai 2022 15:11 Hallo Heiko,ja, da ist die Absicherung der WordPress REST-API dazugekommen. Ich habe das heute noch einmal deutlicher herausgestellt. Antworten
    -  Heiko7. Juni 2022 15:09 Hallo Andreas, ich habe die Absicherung der REST-API ausprobiert, sowohl per Code als auch mit deinem Plugin. In beiden Fällen kann ich keine Beiträge mehr bearbeiten – es erscheint nur eine weiße Seite… Theme TwentyTwenty mit Twentig… Antworten
      -  Andreas Hecht7. Juni 2022 15:19 Hi Heiko,dann ist eines Deiner Plugins schlecht programmiert und benötigt die (komplette) Schnittstelle. Da kann man am Code nichts ändern. Antworten
        -  Frank21. September 2022 13:36 Doch, kann man. Man könnte den Fehler im Snippet beseitigen, so: https://www.kuketz-blog.de/wordpress-rest-api-unter-

wordpress-4-7-deaktivieren/ (https://www.kuketz-blog.de/wordpress-rest-api-unter-wordpress-4-7-deaktivieren/)Das Snippet kursiert in unzählige Male in falsch abgeschriebener Fassung im Netz, sogar beim Kulturbanausen. Auf dieser Seite einmal mehr.

-  Frank21. September 2022 13:28 Hi Heiko, ich würde das REST-Snippet einmal auf die folgende (richtige) Variante abändern und schauen, ob es damit geht (dennwenn du als authentifizierter Benutzer REST-Fehler hast, produziert das Snippet selbst Fehler, weil es keinen Rückgabewert hat):
https://www.kuketz-blog.de/wordpress-rest-api-unter-wordpress-4-7-deaktivieren/ (https://www.kuketz-blog.de/wordpress-rest-api-unter-wordpress-4-7-deaktivieren/)
Schöne Grüße! Antworten

-  Leon17. Mai 2021 8:29 Kann man die Einträge in der functions.php des Child-Themens nicht auch in ein gesondertes Plugin schieben? Antworten

- die schreibmaus22. April 2021 13:04 hallo andreas,auf der suche nach weiteren sicherheits-features für wordpress im netz bin ich auf eine andere seite gestoßen: „https://kinsta.com/de/blog/wordpress-url-loggst/“. dort empfehlen sie unter anderem, mithilfe des wps hide plugins die normale url, unter der üblicherweise der login stattfindet, umzubiegen auf eine beliebig selbstgewählte url, die der angreifer nicht kennen kann. das plugin funktioniert soweit, allerdings natürlich nicht in kombination mit der zusätzlichen Passwortabfrage (HTTP Authentifikation), die du am anfang deines artikels beschreibst. frage an dich als experten: würde es sich lohnen, die login-url zusätzlich zu „verbiegen“, um hackerangriffe weiter zu erschweren? könnte man das mit deiner zusätzlichen passwortabfrage kombinieren? vermutlich müsste man deine .htaccess-datei noch mal anpassen, aber dafür bin ich nicht profi genug. es wäre toll, wenn du das machen könntest, sofern du es für sinnvoll hältst.liebe grüße,die schreibmaus Antworten
  -  Andreas Hecht22. April 2021 13:12 Das verstecken der Login-URL hilft nicht, das können Hacker schnell herausfinden. Antworten
    - die schreibmaus28. April 2021 13:53 vielen dank für deine einschätzung! Antworten
- schreibmaus21. April 2021 19:53 hallo

andreas,vielen dank für dein tolles tutorial! die beschriebenen dinge haben gut funktioniert, bis auf eines: der redirect auf die google-startseite bei eingabe eines falschen logins funktioniert bei mir so nicht. jedenfalls bekomme ich da genauso eine weiße seite mit wordpress-logo angezeigt, wir der andere andreas, der die schrieb. dabei bin ich kein anfänger und habe das gesamte tutorial bestimmt 5 mal gelesen. hast du eine idee, was ich eventuell doch falsch mache?danke dir für eine rückmeldung,
die schreibmaus Antworten

-  schreibmaus19. April 2021 20:59 hallo andreas,herzlichen dank auch von mir für diesen tollen beitrag. was die weiterleitung zu google angeht, geht es mir allerdings wie dem anderen andreas hier, zitat:„Das mit dem Redirect beim falscher Eingabe der Zugangsdaten passt bei mir leider auch noch nicht. Statt Redirect bekomme ich eine leere Seite mit dem WP-Logo (befinde mich da noch immer auf meiner Domain).“das geht mir leider auch so. habe das snippet direkt nach dem „Anmeldung nur noch mit E-Mail-Adresse“-snippet am beginn der functions.php-datei des child-themes eingefügt. ich bin zwar kein anfänger, aber trotzdem unsicher, ob ich das entsprechend richtig gemacht habe, weil es – wie gesagt – nicht funktioniert.vielleicht hast du eine idee, was ich falsch mache.die schreibmaus Antworten

-  Konstantin3. März 2021 15:33 Moin Moin,
kann sein das: Teil 3: Redirect auf Google nach falscher Eingabe der Zugangsdaten
mit dem aktuellen WordPress nicht mehr funktioniert? Antworten

-  Frank15. Dezember 2020 17:25 Hallo Andreas,
einfach mal ein herzliches Dankeschön für deine tolle Arbeit, die unglaublich viel Zeit spart und WP deutlich sicherer macht. Bleib gesund und herzliche Grüße
Frank Antworten
-  Tilo10. Dezember 2020 16:37 Hallo,1.000 Dank für die hervorragende Anleitung.Ich habe ein paar Fragen und Probleme, die evtl. beantwortet und gelöst werden könnten.Zu Punkt Teil 2: WordPress absichern: Zugang nur noch mit E-Mail-Adresse:
Mit der Benutzeranmeldung (normale Kundenanmeldung) hinter einem woocommerceshop, können sich die Kunden nun auch alle nur noch mit der E-Mail anmelden? Oder gilt dies nur für den Admin?…lese ich am Code zumindest nicht heraus. Das würde evtl. für Probleme sorgen, da nicht alle Kunden so firm drinnen sind.Zu Punkt htaccess und Firewall:
ich habe seit der Umstellung auf die/Ihre htaccess auf experten-kredite.de Probleme mit dem PlugIn CalculateFiledsForm (Button „Direkt anfragen“). Da ich dort Daten abfrage und via Clickevent weitergebe denke ich das dies an einer Firewallregel liegt, da er mir im Anschluss einen 403 ausgibt.Gibt es dafür evtl. eine Lösung?Vielen Dank
Tilo Antworten
  -  Andreas Hecht10. Dezember 2020 17:58 Hallo Tilo,Punkt 2: Ja, das dürfte sich auch auf die WooCommerce-User auswirken. Zur .htaccess: Der 403 sollte durch die 7G-Firewall ausgelöst werden. Da bitte die 7G-Firewall in der Datei gegen die 6G-Firewall austauschen. Siehe: https://seoagentur-hamburg.com/die-perfekte-htacce

ss-fuer-wordpress/ (https://seoagentur-hamburg.com/die-perfekte-htaccess-fuer-wordpress/) Antworten

-  Tilo11. Dezember 2020 7:05 Hallo Andreas,Vielen Dank für deine schnelle Antwort…Mit der Anmeldung mit einem woocommerceshop gibt es da sicher einige Probleme mit Kunden, da ja da auch steht „Benutzername oder E-Mail-Adresse“ (hier mal am Kundenbeispiel https://viewegerback.de/mein-konto/ (https://viewegerback.de/mein-konto/)). Gibt der Nutzer nicht die/seine E-Mail ein, wird er auf Google weitergeleitet. Hier ist die Frage, an dich als Profi, ob es dafür einen anderen Weg gibt. Einfachster Weg, die Zeile ändern in nur „E-Mail-Adresse“. Hier wäre die Frage, wo ich dies ändern muss?Mit der Firewall hatte ich die 6G getestet, aber da war gar nichts zu machen, sondern gleich alles dicht. Ich habe jetzt bei den Filtereinstellungen den einen Wert (null) rausgenommen…und es geht. Ich denke, das wird nicht gleich die Sicherheit auf den Kopf stellen. ;opDanke dir
  Tilo Antworten

  -  Tilo11. Dezember 2020 8:30 …wichtig für alle die WordPress 5.6 und die .htaccess https://gist.github.com/seoagentur-hamburg/c96bc796764baaa64d43b70731013f8a#file-htaccess (https://gist.github.com/seoagentur-hamburg/c96bc796764baaa64d43b70731013f8a

#file-htaccess)
verwenden, sei noch mitgeteilt, dass die Permalinkstruktur evtl. neu gesetzt (wegen der Authorization) werden muss
https://de.wordpress.org/support/topic/nach-update-auf-5-6-keine-bearbeitung-moeglich/
(https://de.wordpress.org/support/topic/nach-update-auf-5-6-keine-bearbeitung-moeglich/) Antworten

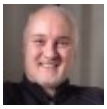
-  Iva29. Oktober 2020 15:21 Hallo Andreas,
vielen Dank für deinen hilfreichen Beitrag. Ich habe eine Frage noch: was würdest du empfehlen für die Absicherung der functions.php-Datei. Besonders sensibel sind z.B. die Zugangsdaten zu dem SMTP-Server, da Username und Passwort im Klartext stehen? Kann man sie verschlüsseln?
Besten Dank! Antworten
  -  Andreas Hecht13. November 2020 14:55 Sorry für die späte Antwort. Seit Corona habe ich mehr Arbeit als ich bewältigen kann. Warum willst Du einen E-Mail-Server (SMTP) in die functions.php eintragen? Antworten

-  Matthias28. Oktober 2020 3:06 Hey Andreas … interessanter Beitrag! Ich hab 2 Fragen.
1. Wenn ich das alles so umsetze, brauch ich dann noch

Wordfence?
2. Das mit Child Theme hab ich nicht ganz verstanden .. Ich nutze kein Child Theme, kann ich dann das trotzdem anwenden? Danke Antworten

-  Andi3. November 2020 0:20 Wenn du die 7G Firewall in deiner htaccess implementiert hast, ein 32 Zeichen langes Passwort und eine 2 Faktor-Authentifizierung, sowie zusätzlichen htacces-Schutz für wp-admin verwendest, dann brauchst du das Plugin „Wordfence“ nicht. Nutze am besten die htaccess-Datei, welche Adresse hier im Beitrag zur Verfügung stellt.Du könntest die von Andreas aufgeführten Anpassungen auch direkt in der functions.php einfügen. Problem: Nach einem WordPress Update werden all deine Einträge überschrieben. Daher ist ein Child-Theme zu empfehlen.Andreas Hecht hat hier in seinem Blog eine Anleitung dazu. Antworten

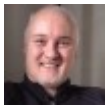
-  Andreas2. September 2020 12:24 Hi Andreas,
toller Beitrag – aber gleich zwei Fragen.1) Umleitung auf Google nach falscher Eingabe der Zugangsdaten funktioniert bei mir nicht.
2) Login via E-Mails ist möglich, aber auch weiterhin mit dem Standard-Benutzernamen.Deinen Code habe ich in die funtions.php unter /wp-includes eingefügt. Ist das evtl. der Fehler? Oder muss ich deine Code Snippets direkt am Anfang oder am Ende der php-Datei einfügen? Antworten

-  Andreas Hecht2. September 2020 12:31 Hi Andreas,genau deshalb schrieb ich, dass die Maßnahmen des Artikels nicht für Anfänger geeignet

sind. Du kannst nicht einfach **irgendwo** etwas hinein kopieren und dann sagen, dass es nicht funktioniert. Der Code gehört in die functions.php **des verwendeten Themes!** Und da solltest Du vorher ein Child-Theme erstellt und aktiviert haben, ansonsten sind die Änderungen nach dem nächsten Theme-Update weg. Das steht da auch ganz deutlich, wo das hin muss. Man muss nur **LESEN**. Ich zitiere mich mal:Kopiere den folgenden Code in die functions.php Deines (Child-) Themes: Antworten

-  Andreas2. September 2020 17:30
  Trotzdem danke Andreas. Den Hinweis hatte ich auch gelesen, dachte mir aber, weil du Child in Klammern gesetzt hast, dass es auch in die Haupt-Functions.php eingefügt werden kann. Sorry, Anfängerfehler.Aber wenn man als Anfänger keine Fragen stellt, kann sich an dem Status auch nichts ändern.
  Und ich habe nicht behauptet, dass dein Code nicht funktioniert. Ich habe lediglich als Anfänger einen Fehler gemacht und hatte keine Erklärung dafür. Ich möchte ja den gesamten Code verstehen, bevor ich einfach nur ‚Copy and Paste' mache.Leider bin ich kein gelernter Informatiker und muss mir die Materie hier selbst erarbeiten und beibringen – nicht immer einfach.
  Bevor ich Dir also weitere unnötige Fragen stelle, kannst du mir evtl. einen Tipp geben, welche Quellen ich für das Verstehen des Codes nutzen kann?Das mit dem Redirect beim falscher Eingabe der Zugangsdaten passt bei mir leider auch noch nicht. Statt Redirect bekomme ich eine leere Seite mit dem WP-Logo (befinde mich da noch immer auf

meiner Domain). Entweder habe ich den Snippet an der falschen Stelle eingefügt (‚functions.php' im Hauptverzeichnis, da du hier ja nicht auf die functions.php des Child-Themes verwiesen hast oder?) oder es fehlt noch eine andere Voraussetzung, die ich übersehen habe.Deine Arbeit und Ratschläge weiß ich sehr wohl zu schätzen. Nochmals vielen Dank. Antworten

-  Andreas Hecht2. September 2020 17:55 Andreas,selbst wenn Du das (Child-) einfach mal streichst, bleibt noch »**in die functions.php Deines Themes**« über. Dort, und nur dort kommt Code hinein. Und wenn Du willst, dass sich der Code auch noch nach einem Theme-Update dort befindet, dann erstellst Du von Deinem aktiven Theme ein Child-Theme, dass Du dann aktivierst. In dieses Child-Theme kommt ebenfalls eine functions.php hinein, in die dann jeder Code-Schnipsel hineinkommt. Übrigens muss man dafür kein Informatiker sein. Aber erstens sehr genau lesen und zweitens **VORHER fragen**, bevor man einfach irgendetwas macht, was man nicht versteht.

https://gist.github.com/seoagentur-hamburg/c96bc796764baaa64d43b70731013f8a#file-htaccess