

WORDPRESS SICHERHEIT

INHALT

INHALT	1
Einleitung.....	3
Mögliche Angriffsszenarien	3
<i>ANGRIFFSARTEN</i>	<i>3</i>
SQL-Injection Angriffe.....	3
Cross-Site-Scripting Angriffe.....	3
Wordpress-Spezifische Angriffe	4
<i>Wer sind mögliche Angreifer?</i>	<i>4</i>
Der Spammer.....	4
Die Malware-Mafia.....	4
Das Script-Kiddie.....	5
Der Profi.....	5
<i>Wordpress!?!?</i>	<i>5</i>
<i>Das wichtigste Kurz und Knapp</i>	<i>6</i>
UPDATES.....	6
Gute Passwörter	6
Logs Lesen.....	6
Beigefügte Wordpress Installation aufsetzen	7
Server-Sicherheit.....	9
<i>Apache und PHP</i>	<i>9</i>
<i>mod_security.....</i>	<i>9</i>
Installation	10
<i>Datenbankzugriff.....</i>	<i>10</i>
<i>wp-config.php.....</i>	<i>11</i>
Das Blog Vorbereiten	13
<i>Nutzer einrichten.....</i>	<i>13</i>
Admin Nutzernamen und ID wechseln	13
Einen Nutzer mit eingeschränkten Rechten anlegen	13

Das Blog härten	14
<i>Zugriff auf wp-content und wp-includes einschränken</i>	14
<i>Zugriff auf wp-admin einschränken via .htpasswd.....</i>	14
<i>Unterdrückung von Fehlerhinweisen auf der Login-Seite.....</i>	15
<i>Dateisystem-Rechte setzen und unnötige Dateien entfernen</i>	16
<i>Versionsnummern entfernen.....</i>	16
Wordpress Versionsnummer entfernen.....	16
Von Plugins erzeugte HTML-Kommentare entfernen	17
WPIDS	17
<i>Externe Vulnerability Scanner</i>	18
WP-Scan.....	18
Nessus und Co.	18
SPAM	18
<i>Askimet.....</i>	18
Quellen und weiterführende Links	19

EINLEITUNG

In diesem Dokument sollen Sicherheitsaspekte rund um den Einsatz der Blogsoftware Wordpress diskutiert werden. Darüber hinaus sollen Lösungen aufgezeigt werden, wie die Sicherheit verbessert werden kann.

Sicherheit ist ein weit-gefasster Begriff (1), in diesem Dokument wird es hauptsächlich darum gehen, dass kein unbefugter Dritter in der Lage ist Daten im Blog zu verändern, an nicht-öffentliche Daten zu gelangen oder anderweitig Schaden anzurichten, sei es durch Ausnutzung von Fehlern in der Blog-Software und/oder der Server-Konfiguration.

Es werden Angriffsszenarien diskutiert und mögliche Maßnahmen dagegen erläutert.

MÖGLICHE ANGRIFFSSZENARIEN

Wer will unserem Blog etwas böses und warum? Welcher Mittel bedient er sich dabei?

ANGRIFFSARTEN

Eine kurze Erläuterung der derzeit häufigsten Angriffe – für eine ausführliche Beschreibung und weitergehende Techniken siehe (2)(3) sowie(4).

SQL-INJECTION ANGRIFFE

Ein SQL-Injection Angriff bezeichnet das Ausnutzen einer Sicherheitslücke in Zusammenhang mit SQL-Datenbanken, die durch mangelnde Maskierung oder Überprüfung von Metazeichen in Benutzereingaben entsteht. Der Angreifer versucht dabei, über die Anwendung, die den Zugriff auf die Datenbank bereitstellt, eigene Datenbankbefehle einzuschleusen. Sein Ziel ist es, Daten in seinem Sinne zu verändern oder Kontrolle über den Server zu erhalten. (5)

CROSS-SITE-SCRIPTING ANGRIFFE

Ein Cross-Site Scripting Angriff ist die Übergabe von Parametern an ein serverseitiges Skript, das eine dynamische Webseite erzeugt. Dies kann etwa das Eingabeformular einer Webseite sein, wie in Foren, Blogs und Wikis üblich. Die eingegebenen Daten werden auf der Webseite wieder als Seiteninhalt ausgegeben, wenn die Seite von Benutzern aufgerufen wird. So ist es möglich, manipulierte Daten an alle Benutzer zu senden, sofern das Serverskript dies nicht verhindert. (6)

WORDPRESS-SPEZIFISCHE ANGRIFFE

Angriffe die in Bezug auf Wordpress häufig anzutreffen sind. Für eine ausführliche Diskussion und weitere Beispiele siehe (7).

WÖRTERBUCH-ATTACKEN

Hier wird versucht einen Nutzernamen zu erraten (meist admin) und es werden eine Reihe von oft verwendeten Passwörtern genutzt. Ist dies Erfolgreich wird das Blog verändert um Spam und Malware-Links einzufügen.

XMLRPC-SCHNITTSTELLE

xmlrpc.php bietet eine automatische Schnittstelle für den Zugriff auf das Blog von ausserhalb über eine API. Programme wie z.B. Windows Live Write nutzen dies. Darüber hinaus ist die xmlrpc.php für Pings und Trackbacks zuständig. Angreifer können über diese Schnittstelle z.B. Wörterbuch-Angriffe durchführen, auch gab es in der Vergangenheit Sicherheitsprobleme.

WER SIND MÖGLICHE ANGREIFER?

DER SPAMMER

Dies ist wohl die häufigste Form von Attacken auf Blogs. Spammer nutzen die gute Positionierung von Blogs im Google-Ranking um ihre Links unterzujubeln und so bessere Suchergebnisse für ihre Seiten zu erreichen. In den meisten Fällen handelt es sich um Kommentarspam, aber auch gezieltere Angriffe, die Sicherheitslücken ausnutzen, oder schwache Passwörter werden immer häufiger. (8)

Meist werden von Spammern eigene Suchmaschinen und Bots eingesetzt.

DIE MALWARE-MAFIA

Häufig werden auch sogenannte IFrames in Blogposts eingeschleust, die eine Seite mit dynamisch generierten Exploit-Codes laden, und so passend zum Browser Trojaner / Viren oder Spyware installieren. (9)

Auch hier wird meist auf Bots und eigene Suchmaschinen oder sogenanntes Google Hacking (10) zurückgegriffen.

DAS SCRIPT-KIDDIE

Die Definition der Wikipedia gibt Aufschluss über dieses Phänomen: Ein Skriptkiddie (von „Skript“ und „Kid“) ist ein Sinnbild für einen stereotypischen Jugendlichen, das sich alltagssprachlich auf den Bereich der Computersicherheit bezieht. Ohne sich mit Computersicherheit intensiver auszukennen, nutzt es vorgefertigte Automatismen, um (meist unter schriftlicher Anleitung) in fremde Computersysteme einzudringen oder sonstigen Schaden anzurichten. Die Bezeichnung hat Anklänge von unreifem Verhalten und Vandalismus.“ (11)

Meist genügt es eine Hinreichend aktuelle Version einzusetzen und Sichere Passwörter zu verwenden um Script-Kiddies scheitern zu lassen.

DER PROFI

Hat sich ein talentierter Hacker auf die Seite eingeschossen, wird es schon schwieriger. Er wird über einen großen Werkzeugkasten an bekannten und unbekanntem Tools verfügen um möglichst viele Informationen über das System zu sammeln und darauf basierend dann einen Angriff konstruieren.

Eventuell besitzt er auch Kenntnis über unveröffentlichte Sicherheitslücken. Hier gilt es die mögliche Angriffsfläche von vornherein gering zu halten dem Angreifer so viele Steine wie möglich in den Weg zu legen. Ein paar spannende Vorträge mit weiteren Informationen gibt es beim CCC: (2), (3)

WORDPRESS!?

Wordpress wird mittlerweile millionenfach als Blogsoftware oder CMS eingesetzt. Sicherheit spielt bei der Entwicklung zunehmend eine Rolle, dennoch gab es und wird es voraussichtlich auch in Zukunft schwere Sicherheitslöcher geben. Es ist davon auszugehen, dass im Kern der Software weniger Sicherheitslöcher zu finden sind, aber Plugins und Themes weiterhin eine große Angriffsfläche darstellen. Dennoch ist Sicherheit bei Wordpress zunehmend ein Thema und es gibt u.a. ein sehr gutes Blog (4) was sich ausschließlich mit Sicherheitsaspekten rund um den Wordpress Einsatz beschäftigt. Dessen Lektüre und regelmäßiges Updaten der Software und der Plugins sollte die Chance Opfer eines Angriffs zu werden auf ein Minimum reduzieren.

DAS WICHTIGSTE KURZ UND KNAPP

Die folgenden Punkte sind essentiell für ein Sicheres Blog. Alle anderen Maßnahmen verfehlen ihre Wirkung wenn diese Punkte nicht befolgt werden.

UPDATES

Ganz einfach: regelmäßig Updaten. Normalerweise werden Sicherheitslücken recht schnell gefixt.

Unter (12) werden Wordpress-Sicherheitslücken bekannt gegeben. Plugins sollten auch aktuell gehalten werden. Normalerweise gibt es einen Feed oder eine Announce-Mailingliste um über Neuigkeiten informiert zu werden.

GUTE PASSWÖRTER

Es mag trivial klingen, aber gute Passwörter sind die wichtigste Grundlage für eine Sichere Webseite. Mindestens 8 Zeichen mit Groß- und Kleinschreibung sowie Sonderzeichen sind Pflicht. Detaillierte Hinweise gibt es unter (13).

LOGS LESEN

Man sollte wissen was auf dem Server passiert und ab und zu einen Blick in die Logs werfen. PHP sollte so konfiguriert sein, dass die Fehlermeldungen nicht auf der Website erscheinen sondern lediglich geloggt werden. Hinweise zur Konfiguration finden sich in (14) um auffälliges Verhalten zu erkennen. Praktisch, jedoch mit viel anfänglich viel Konfigurationsaufwand verbunden sind Tools wie z.B. logcheck (15).

BEIGEFÜGTE WORDPRESS INSTALLATION AUFSETZEN

Im Archiv wordpress.zip befindet sich eine Aktuelle Wordpress Version mit ein paar nützlichen Plugins.

Damit diese Ihre Arbeit verrichten kann, sind folgende Schritte notwendig:

Hier kurz und knapp zusammengefasst, Die einzelnen Punkte sind im Dokument auch noch einmal ausführlich beschrieben:

INSTALLATION

1. Dateien hochladen. Idealerweise sollte sich Wordpress im public_html Ordner befinden.
2. Die wp-config.php entweder passend zur Datenbank editieren oder umbenennen und den Wordpress Installationsassistenten nutzen, danach jedoch die Optionen aus der Beigelegten wp-config.php übernehmen.
3. Die Pfade in den .htaccess Dateien im Hauptordner und im wp-admin Ordner so anpassen, dass sie auf eine existierende .htpasswd Datei zeigen, Apache hat Probleme mit relativen Pfadangaben, es ist also am besten absolute Pfadangaben zu verwenden.
4. Die .htpasswd Datei einrichten. Einen Generator dafür findet sich z.B. unter (16)

KONFIGURATION

5. Die Blog-Einstellungen aufrufen. <http://<url>/wp-admin>
6. Ein paar Nutzer anlegen, davon der letzte mit Admin-Rechten. Alle anderen Nutzer wieder Löschen auch den von Wordpress standardmäßig angelegten Admin-User löschen.
7. Unter Einstellungen -> Verschiedenes den Upload Pfad auf uploads setzen. (Ausserhalb von wp-content). Damit hat man später eine bessere Kontrolle über den Upload-Ordner.,,
8. Permalinks unter Einstellungen -> Permalinks aktivieren.
9. Unterstützung für XMLRPC-Schnittstelle unter Einstellungen -> Schreiben deaktivieren.

PLUGINS

10. „Secure Wordpress“ aktivieren, bei Einstellungen → Secure WP , überall ein Häkchen setzen.
11. Askimet Plugin für die SPAM Vermeidung aktivieren. sich bei Wordpress.com registrieren und einen API-Key holen. Askimet ist recht effektiv, was die Spam-Erkennung angeht.
12. Das cleanUmlauts Plugin aktivieren. Dies sorgt dafür das Umlaute (äöü..) in den Permalinks umgeschrieben werden und so für Google mehr Sinn ergeben.
13. Google Sitemaps Plugin aktivieren. Dieses Plugin legt eine XML-Datei mit der Struktur des Inhaltes ab, sinnvoll für eine gute Indizierung in Google und anderen Suchmaschinen.
14. izioSEO Plugin aktivieren und entsprechend wie gewünscht konfigurieren. Dies ist ein vollständiges Plugin zur Suchmaschinenoptimierung. Die Standardwerte sind hier schon ganz ordentlich. Darüber hinaus gibt es die Möglichkeit hier die Verifikation für die Google Webmastertools einzutragen, sowie die ID für Google Analytics.
15. wp-IDS wenn gewünscht aktivieren (siehe dazu den Abschnitt über wp-IDS)
16. Bei Bedarf wp-super-cache und wp-widget-cache aktivieren. Dies sind 2 Plugins die eine gecachte Version des Inhalts vorhalten. Und somit die Performance effektiv erhöhen. Eine passende Cache-Zeit auswählen und die entsprechenden Rewrite-Regeln in der .htaccess eintragen oder vom Plugin eintragen lassen. Hier gibt es häufiger mal Probleme. Gut Testen ob es wie gewünscht funktioniert. Aus dem Quelltext ist entnehmbar ob eine Seite gecached ausgeliefert wird.

TESTEN

17. Testen ob alles zufriedenstellend funktioniert.
18. Bloggen 😊

SERVER-SICHERHEIT

Dies ist ein weites Feld und hängt sehr von der individuellen Konfiguration ab. Ein paar gute Tipps gibt es unter (17). Hier soll das Thema nur kurz angerissen werden...

APACHE UND PHP

Es ist Sinnvoll keine Informationen über die Verwendete Version preiszugeben, dies kann über die Direktive `ServerTokens Prod` in der `httpd.conf` realisiert werden. Ansonsten gibt es hilfreiche Hinweise auf der Apache Website (18) und speziell zu PHP ein sehr lesenswertes kostenloses Buchkapitel (14).

MOD_SECURITY

`mod_security` (19) ist eine Art Web-Firewall für Apache. Man kann Regeln definieren und beliebige Restriktionen setzen. Die Konfiguration ist nicht ganz einfach und teilweise ist schwierig gute Regeln zu finden, die bestehende Anwendungen nicht stören. Für Wordpress gibt zum Glück vordefinierte Regeln (20), die jedoch teilweise für die aktuelle Version oder spezielle Plugins angepasst werden müssten. `mod_security` erhöht die Sicherheit, wenn es richtig eingesetzt wird ungemein, aber die Konfiguration würde den Rahmen hier sprengen. Auf heise.de (21) gibt es einen guten Artikel zur Einführung.

INSTALLATION

Nun zurück zu Wordpress... Beginnen wir mit der Installation:

DATENBANKZUGRIFF

Bevor man mit der Installation von Wordpress beginnt ist es wichtig die richtigen Berechtigungen für die Datenbank und den Datenbankbenutzer zu setzen.

Wordpress sollte für sich eine eigene Datenbank nutzen.

Es sollte ein eigener Benutzer angelegt werden, der lediglich Zugriff auf diese Datenbank besitzt.

Plesk macht dies für MySQL-Datenbanken automatisch.

Auf keinen Fall für die Verbindung zum Datenbankserver den root-Account verwenden! Dieser darf z.B. auch Systembefehle ausführen und ist fast so mächtig wie der Unix root Benutzer.

Los geht's – zuerst legen wir als MySQL root die Datenbank an:

```
$ mysql -u root
mysql> CREATE database wordpress;
Query OK, 1 row affected (0.00 sec)
```

Dann wird der Benutzer angelegt und die notwendigen Rechte vergeben, sowie ein hoffentlich gutes Passwort festgelegt.

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP -> ON wordpress.* -> TO
'wpuser'@'localhost' -> IDENTIFIED BY 'strongpassword'; Query OK, 0 rows affected (0.01 sec)
```

Idealerweise ist der mysqld nicht von aussen zu erreichen:

```
# /etc/mysql/my.cnf
# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 127.0.0.1
```

WP-CONFIG.PHP

Weiter geht es mit der wp-config.php im Wordpress Verzeichnis. In der Datei sind die Datenbankzugangsdaten hinterlegt und es können eine Reihe von Konfigurationsoptionen gesetzt werden. Alle Optionen sind unter (22) beschrieben.

WP-CONFIG.PHP SICHERN

Seit Version 2.6 ist es möglich die wp-config.php auch außerhalb des Wordpress Ordners zu platzieren.

Dies ist sehr zu empfehlen, z.B. in Falle einer PHP-Fehlkonfiguration gelangt so trotzdem niemand an die sensiblen Datenbankdaten:

```
server:/var/www/wordpressdomain.de/public_html> mv wp-config.php ..
```

Falls das nicht möglich ist, ist es ratsam die wp-config.php per .htacces vor Zugriffen zu schützen:

```
# protect wp-config.php
<files wp-config.php>
Order deny,allow
deny from all
</files>
```

WP-CONFIG.PHP EINSTELLUNGEN

```
// ** MySQL Einstellungen ** //
define('DB_NAME', 'wordpress'); // Der Name der Datenbank, die du benutzt.
define('DB_USER', 'wpuser'); // Dein MySQL-Datenbank-Benutzername.
define('DB_PASSWORD', 'longandgoodpass'); // Dein MySQL-Passwort.
define('DB_HOST', 'localhost'); // 99% Chance, dass du hier nichts ändern musst.
define('DB_CHARSET', 'utf8');
define('DB_COLLATE', 'utf8_general_ci');
```

TABLE PREFIX

Viele Exploits benötigen den Wordpress Datenbank-Prefix es ist daher sehr ratsam diesen auf einen Zufälligen langen Wert zu setzen:

```
// Einen anderen Prefix als wp_ wählen ist sehr zu empfehlen.  
$table_prefix = 'whaligg00w_'; // Zufälligen Prefix wählen
```

SECRET KEYS

Seit Wordpress 2.7 gibt es insgesamt 4 sogenannte Secret-Keys. Diese erhöhen die Sicherheit der generierten Passwörter und Cookies da diese als Salt (23) genutzt werden. Dadurch werden Attacken auf die als Hash gespeicherten Passwörter schwieriger. Diese Werte kann man sich auf der Wordpress-Seite generieren lassen (24).

```
define('AUTH_KEY', ' :dr+%/5V4sAUG-gg%aS*v;&xGhd%{YKC^Z7KKGh j>k[.Nf$y7iGKdJ3c*[Kr5Bg'] );  
define('SECURE_AUTH_KEY', 'Tufw0u>#+hA?^|3RfGTm>@*+S=8\"'\'+\"'}<m#+}V)p:Qi?jXLq,<h\\`39m_(' );  
define('LOGGED_IN_KEY', 'S~AACm4h1;T^\"qW3_8Zv!Ji=y|)~5i63JI |Al[(<YS<2V^$T])=8Xh2a:b:}U_E');  
define('NONCE_KEY', 'k1+E0c-&w?hG8j84>6L9v\"6C89NH?ui{*3\\(t09mumL/ffp_!K$JCEkLuy ={x{0'});
```

LOGIN UND ADMIN-BEREICH NUR ÜBER SSL

Darüber hinaus sollte man bei dieser Gelegenheit gleich den Zugang zum Admin-Bereich automatisch über SSL-Laufen lassen:

```
define('FORCE_SSL_ADMIN', true);  
define('FORCE_SSL_LOGIN', true);
```

Weitere Informationen gibt es im Wordpress-Codex (22).

DAS BLOG VORBEREITEN

Jetzt haben wir schon eine halbwegs sichere Grundlage geschaffen. Zeit weiter zu gehen und uns um die Benutzer zu kümmern. Der Standardmäßige Admin Benutzer sollte geändert werden und ein Nutzer mit eingeschränkten Rechten angelegt werden für die tägliche Nutzung des Blogs

NUTZER EINRICHTEN

Wordpress ist anfällig ist für eine User-Enumeration Attacke. Das heißt es ist bekannt wie der Admin Nutzer heißt und wie welche interne ID er hat. Ideal für SQL-Injection-Angriffe (5).

ADMIN NUTZERNAME UND ID WECHSELN

Darum ist es angebracht den standardmäßig aktivierten Admin zu löschen. Am Einfachsten ist es man legt ein paar Nutzer an und löscht diese dann wieder. Dem letzten angelegten dieser Nutzer gibt man Admin Rechte. Der Nutzername sollte logischerweise nicht unbedingt admin lauten ☺

Folgende Schritte sind dafür notwendig:

1. Im Admin-Bereich einen zusätzlichen Administrator hinzufügen.
2. Logout
3. Sich mit den Zugangsdaten des eben angelegten Administrators anmelden.
4. Den alten *admin* aus der Benutzerliste streichen.

Das hat den Vorteil, das die User-Id 1 in der Datenbank nicht mehr vergeben ist und der Admin-Nutzer nicht mehr existiert. Damit werden Brute-Force Attacken (25) auf das Passwort sowie SQL-Injection Attacken (5) erschwert.

EINEN NUTZER MIT EINGESCHRÄNKTEN RECHTEN ANLEGEN

Für alle täglichen Arbeiten am Blog sollte ein Nutzer mit eingeschränkten Rechten angelegt werden. Nur wirklich notwendige Dinge sollten ihm erlaubt sein. z.B. Kommentare moderieren, Posts schreiben. Jedoch nicht Themes oder Plugins editieren. Da hilft nur Probieren. Wordpress bietet jedoch ein paar vordefinierte Rollen (26).

DAS BLOG HÄRTEN

Um mögliche Angriffe weiter einzuschränken ist es sinnvoll den direkten Zugriff auf php Dateien in den Wordpress Verzeichnissen zu begrenzen.

Eine Benutzer-Registrierung ist normalerweise nicht notwendig. Kommentare können Seitenbesucher auch so abfassen und die nachträgliche Korrektur von Kommentaren ist m.E. nicht Relevant im Vergleich zum Sicherheitsgewinn durch die nachfolgend dargestellten Maßnahmen.

ZUGRIFF AUF WP-CONTENT UND WP-INCLUDES EINSCHRÄNKEN

Innerhalb des wp-content Verzeichnisses befinden sich die Themes und die Plugins der Wordpress Installation. Es macht Sinn den direkten Zugriff von außerhalb auf die PHP-Dateien einzuschränken. Normalerweise werden Wordpress-Intern diese Dateien per PHP Included, d.h. ein externer Zugriff ist nicht nötig. Lediglich Ausnahmen für Dateien mit Medien-Inhalten wie CSS/JavaScript-Dateien für das Theme oder Bilder sind hier zu dulden.

.htaccess für das wp-content Verzeichnis: Der direkte Zugriff auf php Dateien ist verboten.

```
Options -Indexes
Order Allow,Deny
Deny from all
<Files ~ "\.(css|jpe?g|png|gif|js|mp3|pdf|swf|html|ico|xsl)$">
  Allow from all
</Files>
```

.htaccess für wp-includes.

```
Options -Indexes
Order Allow,Deny
Deny from all
<Files ~ "\.(css|jpe?g|png|gif|js|swf|mp3|pdf)$">
  Allow from all
</Files>
```

ZUGRIFF AUF WP-ADMIN EINSCHRÄNKEN VIA .HTPASSWD

Ähnliches gilt für den wp-admin Ordner, hier ist noch zu beachten, dass die Möglichkeit Dateien in die Mediathek hochzuladen nicht eingeschränkt werden soll.

Folgende .htaccess wird dem gerecht:

```
<FilesMatch ~"(async-upload|admin-ajax)\.php$">
AuthName "datensicherheit login"
AuthType Basic
AuthUserFile /pfad/zum/blog/.htpasswd
require valid-user
</filesMatch>
<FilesMatch "(async-upload|admin-ajax)\.php$">
  <IfModule mod_security.c>
```

```
SecFilterEngine Off
</IfModule>
Allow from All
</FilesMatch>
```

Darüber hinaus ist in der .htaccess im Wordpress root Verzeichnis noch der Zugriff auf die Login-Seite zu sperren:

```
<files wp-login.php>
AuthName "login"
AuthType Basic
AuthUserFile /HIER_SERVER_PFAD_EINTRAGEN/.htpasswd
require valid-user
</files>
```

UNTERDRÜCKUNG VON FEHLERHINWEISEN AUF DER LOGIN-SEITE

Wordpress ist, wenn keine .htaccess für den Login vorgesehen ist, anfällig für eine User-Enumeration-Attacke (25) . D.h. es ist ermittelbar welcher Nutzeraccount existiert und welcher nicht.

Das Plugin Secure Wordpress (27) nimmt sich dieser Problematik an.

Secure WordPress Hilf

Akismet ist fast fertig. Du mußt deinen [WordPress.com-API-Schlüssel](#) eingeben, damit es funktioniert.

Einstellungen

Fehler-Meldungen	<input checked="" type="checkbox"/> deaktiviert die Hinweis- und Fehlermeldung beim Login von WordPress
WordPress Version	<input checked="" type="checkbox"/> Entfernen der Version von WordPress in allen Bereichen, inkl. Feed, nicht im Admin-Bereich
index.html	<input checked="" type="checkbox"/> hinterlegt eine <code>index.html</code> in <code>/plugins/</code> um das Auslesen des Verzeichnis zu vermeiden
Really Simple Discovery	<input checked="" type="checkbox"/> Entfernt den link für Really Simple Discovery im head des Frontend
Windows Live Writer	<input checked="" type="checkbox"/> Entfernt den link für Windows Live Writer im head des Frontend
Core Update	<input checked="" type="checkbox"/> Deaktiviert das Core-Update für Nicht-Admin's. Die Mitteilung einer neuen Version von Plugins wird ausschließlich Nutzern gezeigt, die die Rechte zum Editieren von Plugins haben.
Plugin Update	<input checked="" type="checkbox"/> Deaktiviert das Plugin-Update für Nicht-Admin's. Die Mitteilung einer neuen Version von WordPress wird ausschließlich Nutzern gezeigt, die die Rechte zum Editieren von Plugins haben.

Einstellungen aktualisieren »

DATEISYSTEM-RECHTE SETZEN UND UNNÖTIGE DATEIEN ENTFERNEN

Hier gilt, dass der Webserver nur minimale Rechte haben sollte, d.h. nur was unbedingt notwendig ist sollte Schreibrechte besitzen. z.B. der upload-Ordner. Dies ist abhängig von der Serverkonfiguration.

In unserem Fall benötigen die Verzeichnisse upload und – wenn aktiviert – wp-contents/cache Schreibrechte.

VERSIONSNUMMERN ENTFERNEN

WORDPRESS VERSIONSNUMMER ENTFERNEN

Einen großen Angriffsvektor stellen die Wordpress-Plugins und Wordpress selbst dar, indem Versionsinformationen als Kommentar im Quelltext dargestellt werden. Diese zu verstecken ändert natürlich nichts an der Sicherheit an sich, allerdings vermeidet man so unnötig Informationen preiszugeben, die für einen Angriff genutzt werden könnten.

```
<script type='text/javascript' src='http://www.toomuchcookies.net/wp
balupton-edition/js/jquery.lightbox.packed.js?ver=1.3.1'></script>
<meta name="generator" content="WordPress 2.7.2-alpha" />

<script language="javascript1.4" type="text/javascript" src="http://
/plugins/audio-player/audio-player.js"></script>
<link rel="stylesheet" href="http://www.toomuchcookies.net/wp-conten
/css/syntax hilite css.css" type="text/css" media="all" />
  <script language="javascript" type="text/javascript" src="ht
/plugins/ig syntax hilite/js/syntax hilite js.js"></script>
  <script language="javascript" type="text/javascript">
    var arrCode = new Array();
  </script>
    <style type="text/css">
        ol.footnotes li {lis
        ol.footnotes{font-si

</style>
    <script type="text/javascript">jQuery(document).read
jQuery(".gallery").each(function(index, obj){ jQuery(obj).find("a").

  <!-- Generated by Simple Tags 1.6.4 - http://wordpress.org/e
  <meta name="keywords" content="Open_Source, Software, Progra
  </script>
```

Das Plugin Secure Wordpress (27) stellt sicher, dass Wordpress keine Informationen über seine Versionsnummer öffentlich macht.

VON PLUGINS ERZEUGTE HTML-KOMMENTARE ENTFERNEN

Bei Plugins sieht es etwas schwieriger aus. Hier ist die Information über die Versionsnummer meist im Plugin-Quelltext selbst fest eingestellt.

Dies lässt sich allerdings recht einfach beseitigen indem man den Plugin-Quelltext nach HTML-Kommentaren (<!--) absucht und diese entfernt.

WPIDS

WP-IDS (28) ist ein sogenanntes Intrusion-Detection-System (29). D.h. es werden gefährliche Abfragen erkannt und bei Bedarf verschiedene Schritte unternommen z.B. die Abfrage geblockt oder eine E-Mail versendet. Es gibt für WP-IDS aktuelle Signaturen die man z.B. auch per cronjob Updaten könnte.

Log blocked intrusions to database:

Use JSON filter rules (may increase performance a little bit):

[Save Settings](#)

WP Lockdown:

Disable forgotten password features of Wordpress:

Enable Lockdown Framebreaker (Prevents evil sites from including your site on their domain):

Disable XML-RPC ability (if you don't use Offline Writers it's recommend to disable XML-RPC requests):

[Save Settings](#)

Last Blocked Bad Requests:

ID	Value	Tag	Page	IP	Impact	Time
1322		HTTP_X_FORWARDED_FOR	/index.php...	unknown	-	2009-03-0 21:20:21
1321	/wp-content/plugins/download.html?path=download.html	SERVER_URI	/index.php...	217.81.57.122	4	2009-03-0 21:17:18
1320	<script>alert("445028851543");</script>	start	/index.php...	217.81.57.122	39	2009-03-0 21:14:09
1319	/wp-content/plugins/ViewDay.html?start=scriptalert445028851543	REQUEST_URI	/index.php...	217.81.57.122	-	2009-03-0 21:14:08
1318	/wp-content/plugins/ViewDay.html?start=scriptalert(445028851543);	REQUEST_URI	/index.php...	217.81.57.122	-	2009-03-0 21:14:08
1317	start=scriptalert445028851543/script	QUERY_STRING	/index.php...	217.81.57.122	-	2009-03-0 21:14:08
1316	start=scriptalert(445028851543);/script	QUERY_STRING	/index.php...	217.81.57.122	-	2009-03-0 21:14:08
1315	/wp-content/download.html?path=download.html	SERVER_URI	/index.php...	217.81.57.122	4	2009-03-0 21:13:07

Der Einsatz ist zu empfehlen, allerdings ist die Geschwindigkeit des Blogs dadurch etwas langsamer.

EXTERNE VULNERABILITY SCANNER

Es gibt viele Programme die automatisch eine Seite auf bekannte Sicherheitslücken prüfen. Natürlich sind diese Tools nicht in der Lage alle Probleme aufzudecken, aber dennoch hilfreich um die Sicherheit der eigenen Seite zu überprüfen.

WP-SCAN

WP-Scan (30) ist ein Online-Security Scanner für Wordpress . Entwickelt von den Personen rund um blogsecurity.net (4), prüft er auf gängige Fehlkonfiguration der Installation. Dazu muss man lediglich eine Textdatei namens wpscan.txt mit dem Inhalt `<!--wp-scan -->` anlegen. Nicht vergessen, diese nach dem Scannen wieder zu entfernen.

NESSUS UND CO.

Es gibt eine Vielzahl an weiteren Vulnerability-Scannern um die Server-Sicherheit zu prüfen. Empfehlenswert sind der kostenlose Scanner Nessus (31) und speziell für Webseiten der Acunetix Web Security Scanner (32). Dieser scannt in der kostenfreien Version jedoch nur gegen Cross-Site-Scripting Angriffe.

SPAM

Spam ist ein großes Problem in Bezug auf Blogs. Kommentar und Trackback-Spam tritt nach einer gewissen Zeit auf jeder neuen Blog-Installation auf. Was kann man dagegen tun?

ASKIMET

Als effektives Mittel gegen Kommentar und Trackback-Spam hat sich Askimet (33) herausgestellt. Dies ist ein Wordpress Plugin was jeden neuen eingehenden Kommentar auf eigenen Servern nach vielen Kriterien auf Spam untersucht. Dabei werden die Daten allerdings die USA übermittelt, was bei einigen Personen Datenschutzbedenken hervorgerufen hat. Da Kommentare aber meist sowieso öffentlich zugänglich sind, stellt dies für die meisten Nutzer kein Problem dar. Dies scheint aber ein Der größte Vorteil von Askimet ist seine gute Erkennungsrate und die Barrierefreiheit, da auf Captchas verzichtet werden kann.

Für die Nutzung ist jedoch ein Wordpress-API Key notwendig, den mal durch eine Anmeldung bei Wordpress.com bekommt (34).

QUELLEN UND WEITERFÜHRENDE LINKS

1. **Chaosradio.** Podcast: Sicherheit. [Online] <http://chaosradio.ccc.de/cre046.html>.
2. **24C3.** Unusual Web Bugs. [Online] <http://events.ccc.de/congress/2007/Fahrplan/events/2212.en.html>.
3. **25C3.** Attacking Rich Internet Applications. [Online] <http://events.ccc.de/congress/2008/Fahrplan/events/2893.en.html>.
4. **blogsecurity.net.** Wordpress Security Blog. [Online] <http://blogsecurity.net>.
5. **Wikipedia.** SQL-Injection. [Online] <http://de.wikipedia.org/wiki/SQL-Injection>.
6. —. Cross-Site-Scripting. [Online] http://de.wikipedia.org/wiki/Cross-Site_Scripting.
7. **Eckert, Frank.** Unser täglich Spam. [Online] <http://spam.weltretter.de/2008/03/24/aktuelle-angriffe-auf-wordpress-blogs/>.
8. **Müller, Sergjej.** [Online] <http://playground.ebiene.de/1530/wordpress-wurm-unterwegs/>.
9. **Gulli-News.** iFrames manipulieren Suchergebnis-Seiten und linken auf illegale Inhalte. [Online] <http://www.gulli.com/news/malware-hack-iframes-2008-03-07/>.
10. **Wikipedia.** Google Hacking. [Online] http://en.wikipedia.org/wiki/Google_Hacking.
11. —. Script-Kiddie. [Online] <http://de.wikipedia.org/wiki/Skriptkiddie>.
12. **Wordpress.** Security Announcements. [Online] <http://codex.wordpress.org/CVEs>.
13. **RRZN Hannover.** RRZN - gute Passwörter. [Online] http://www.rrzn.uni-hannover.de/pw_used.html.
14. **Security, Apache.** Chapter 3 - PHP. [Online] <http://www.apachesecurity.net/download/apachesecurity-ch03.pdf>.
15. **Logcheck.** Logfile Scanner. [Online] <http://logcheck.org/>.
16. **Homepage-Kosten.** .htaccess Generator. [Online] <http://www.homepage-kosten.de/htaccess/>.
17. **Labs, Samhain.** Securing a VPS or dedicated server. [Online] <http://la-samhna.de/library/serversec.html>.
18. **Server, Apache HTTP.** Security Tips. [Online] http://httpd.apache.org/docs/2.2/misc/security_tips.html.
19. **mod_security.** Open Source Web Application Firewall. [Online] <http://www.modsecurity.org/>.

20. **blogsecurity.net**. *ModSecurity and Wordpress: Defense in Depth*. [Online]
<http://blogsecurity.net/wordpress/modsecurity-and-wordpress-defense-in-depth>.
21. **security, heise**. Die Apache-Firewall. *Web-Server mit mod_security absichern*. [Online]
<http://www.heise.de/security/artikel/print/69070>.
22. **WordPress**. Codex: Editing wp-config.php. [Online]
http://codex.wordpress.org/Editing_wp-config.php.
23. **Wikipedia**. Salt (Kryptologie). [Online] [http://de.wikipedia.org/wiki/Salt_\(Kryptologie\)](http://de.wikipedia.org/wiki/Salt_(Kryptologie)).
24. **WordPress**. Key Generation. [Online] <https://api.wordpress.org/secret-key/1.1/>.
25. **Moro, Alberto**. Bruteforcing Web Applications. [Online]
http://www.fistconference.org/files/bruteforcing_webapps_v2.pdf.
26. **WordPress**. Codex: Roles and Capabilities. [Online]
http://codex.wordpress.org/Roles_and_Capabilities.
27. **buelte.de**. WordPress Login Sicherheit - Secure WP. [Online]
<http://buelte.de/wordpress-login-sicherheit-plugin/652/>.
28. **PHP-IDS**. WP-IDS. [Online] <http://php-ids.org/category/wpids/>.
29. **Wikipedia**. IDS. [Online] http://de.wikipedia.org/wiki/Intrusion_Detection_System.
30. **blogsecurity.net**. wp-scan. [Online] <http://blogsecurity.net/wordpress/tools/wp-scanner>.
31. **Nessus**. Security Scanner. [Online] <http://nessus.org>.
32. **Acunetix**. Web-Security Scanner. [Online] <http://www.acunetix.com>.
33. **WordPress**. Codex: Editing wp-config.php. [Online]
http://codex.wordpress.org/Editing_wp-config.php.