Bundesamt
für Sicherheit in der
Informationstechnik

Sicheres Bereitstellen von Web-Angeboten mit WordPress

BSI-Checkliste zur Internet-Sicherheit (ISi-Check)

Bundesamt für Sicherheit in der Informationstechnik
ISi-Projektgruppe
Postfach 20 03 63
53133 Bonn
Tel. +49 (0) 228 99 9582-0
E-Mail: isi@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

1 Einleitung

Dieses Dokument ist überwiegend an Administratoren und Revisoren gerichtet, die für die Konfiguration und den sicheren Betrieb der Komponenten für das Bereitstellen von Web-Angeboten zuständig sind. Ziel der vorliegenden Checkliste ist es, die Umsetzung der Sicherheitsanforderungen für ein Wordpress-System auf LAMP-Basis (Linux, Apache, MySQL, PHP) zu ermöglichen. Diese wurden im Rahmen einer Sicherheitsuntersuchung erarbeitet. Diese Liste baut auf die Härtungsmaßnahmen zur Absicherung eines LAMP-Systems [1] auf und erweitert diese um WordPress-spezifische Maßnahmen. Zunächst muss eine Härtung des Systems nach [1] erfolgen, bevor die weiterführenden Maßnahmen dieser Checkliste umgesetzt werden.

Ergänzend zu den Checklisten wird eine skriptbasierte Lösung zum Download bereitgestellt [2], um einzelne Härtungsschritte teilweise zu automatisieren. Da jedoch einige Anforderungen nur manuell umgesetzt werden können, ist bei den nachfolgenden Sicherheitsanforderungen deutlich gemacht, welche der Anforderungen durch die Härtungsskripte automatisiert umgesetzt werden können. Diese sind in der Spalte H (“Härtungsskript“) mit einem “X“ gekennzeichnet. Die Einteilung der zu implementierenden Maßnahmen erfolgt anhand der Komponenten des Gesamtsystems.

1.1 Unterstützte Komponenten

Die Härtungsmaßnahmen dieser Checkliste orientieren sich an dem in [1] beschriebenen LAMP-System sowie der WordPress-Version 4.4.1. Die Funktionsfähigkeit für andere Betriebssysteme sowie abweichende Versionen der Komponenten wurde nicht getestet und kann deshalb nicht beurteilt werden.

1.2 Erläuterungen zur Kommando-Syntax

In den Spalten „Kommandos zur Härtung“ werden teilweise Präfixes verwendet, um anzudeuten, in welchem Kontext der Befehl ausgeführt werden muss. Diese Präfixes und weitere Syntax-Elemente sollen an dieser Stelle beschrieben werden:

Präfix / Syntax	***Beispiel***	***Bedeutung***
<datei>:	**php.ini:**	Hier wird auf den Inhalt einer Datei verwiesen. Ist kein absoluter Pfad zu der Datei angegeben, so ist davon auszugehen, dass die Datei an keinem festen Ort im Dateisystem zu finden ist. Der absolute Pfad der Datei ist abhängig vom Zielsystem.
#	**# Prüfung der LoadModule Direktiven**	Zeilen, welche mit einem # beginnen, sind Kommentare innerhalb einer Reihe von Anweisungen. Sie dienen lediglich der Dokumentation der Anweisungen.
`root@host:~$`	`root@host:~$ chmod 644 /pfad/datei`	Ein Shell-Befehl, ausgeführt mit Root-Rechten. Alternativ zu dem Root-Nutzer kann hier auch `sudo` verwendet werden.

2 WordPress

Maßnahme	*H*	*Umsetzung der Härtung*
Der Name des Administrator-Kontos sollte nicht leicht zu erraten sein.		Begriffe wie 'admin', 'root' oder 'webmaster' sollten für das Administrator-Konto vermieden werden.
Die Standard-Benutzerrechte sind nach dem Least-Privilege-Prinzip anzupassen, so dass Benutzer nur jene Aufgaben durchführen können, die ihnen anvertraut sind. Jedes zusätzliche Recht ermöglicht Missbrauch.		Administratoren sollen keine Beiträge erstellen, besitzen oder bearbeiten können. Andere Nutzerrollen sind je nach Bedarf zu vergeben. Auch für Single-User Installationen sollte es mehrere Accounts geben, so dass der Administrator-Account nur für administrative Tätigkeiten und nicht für redaktionelle verwendet wird.
Der Zugriff auf die Datei „wp-config.php“ ist am Webserver zu deaktivieren. Es muss verhindert werden, dass der Inhalt der Datei wp-config.php von außen erreichbar gemacht wird.		Die Datei `wp-config.php` sollte außerhalb des DocumentRoot abgelegt werden. Wird die Datei eine Ebene höher als die WordPress-Installation abgelegt, so findet WordPress die Datei weiterhin. Beispiel: • DocumentRoot: `/var/www/html/wordpress/` • Ort für `wp-config.php`: `/var/www/html/`
Sind alle benötigten Plugins und Themes installiert, sollte verboten werden, diese zu bearbeiten oder weitere Plugins/Themes zuzulassen.	x	In die Datei `wp-config.php` die folgende Zeile eintragen: `define('DISALLOW_FILE_EDIT', true);`

Es müssen eigene Salts und Keys für die wp-config.php erzeugt werden.		Über den Aufruf der Webseite https://api.wordpress.org/secret-key/1.1/salt/ werden zufällige Werte für folgende Parameter erzeugt:
		`define('AUTH_KEY', 'xxx');`
		`define('SECURE_AUTH_KEY', 'xxx');`
		`define('LOGGED_IN_KEY', 'xxx');`
		`define('NONCE_KEY', 'xxx');`
		`define('AUTH_SALT', 'xxx');`
		`define('SECURE_AUTH_SALT', 'xxx');`
		`define('LOGGED_IN_SALT', 'xxx');`
		`define('NONCE_SALT', 'xxx');`
		Die erzeugten Zeilen müssen in die Datei `wp-config.php` eingefügt werden. Falls entsprechende Einträge bereits vorhanden sind, müssen diese ersetzt werden.

Der WordPress-Table-Prefix für Datenbanktabellen muss zufällig festgelegt werden. Dies hat in erster Linie den		In der Datei `wp-config.php` muss die folgende Zeile angepasst werden: `$table_prefix = '<prefix>_';`

Vorteil, dass eine Datenbank mehrere CMS-Installationen beinhalten kann, aber es erschwert auch gewisse Datenbank-Angriffe.		Hierbei ist `<prefix>` durch einen zufälligen Wert zu ersetzen. Beispiel: `$table_prefix = 'ksei37_';`
Der Zugriff auf das Backend darf nur via SSL erlaubt werden.	x	In die Datei `wp-config.php` die folgende Zeile eintragen: `define('FORCE_SSL_ADMIN', true);`
Plugins aus der Installationsroutine müssen gelöscht werden, falls sie nicht verwendet werden.		`root@host:~$ rm -rf /var/www/<dirname>/wordpress/wp-content/plugins/akismet` `root@host:~$ rm -f /var/www/<dirname>/wordpress/hello.php`
Ungenutzte Standard-Themes müssen deinstalliert und gelöscht werden. Vorab ist zu prüfen, dass diese tatsächlich nicht verwendet werden.		`root@host:~$ rm -rf /var/www/<dirname>/wordpress/wp-content/themes/twenthythirteen` `root@host:~$ rm -rf /var/www/<dirname>/wordpress/wp-content/themes/twentyfourteen`

Quellenverzeichnis

[1] Bundesamt für Sicherheit in der Informationstechnik, BSI-Standards zur Internet-Sicherheit (ISi-Reihe), Sicheres Bereitstellen von Web-Angeboten (ISi-Web-Server), URL:

https://www.bsi.bund.de/dok/6620604, Stand 03.04.2017.

[2] CMS Garden, Skriptlösung zur Härtung von Webservern, URL: https://github.com/CMS-Garden/cmshardening, Stand 03.04.2017.