Bundesamt
für Sicherheit in der
Informationstechnik

Handbuch

Skriptlösung zur Härtung von Content Management Systemen

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Tel.: +49 22899 9582
E-Mail: bsi-publikationen@bsi.bund.de
Internet: https://www.bsi.bund.de

Inhaltsverzeichnis

1 Einleitung

Im Zuge einer Sicherheitsuntersuchung durch das BSI wurden ausgewählte Content Management Systeme durch umfassende Sicherheitstests geprüft. Aus den daraus gewonnenen Erkenntnissen über Sicherheit und Angriffspunkte wurden praktische Konfigurationshilfen in Form von Checklisten zur sicheren Konfiguration verschiedener Komponenten erarbeitet. Diese sind auf den Webseiten der ISi-Reihe unter der URL: https://www.bsi.bund.de/dok/6620604 abrufbar. Zur Unterstützung der Anwendung der Checklisten wurde eine dialogbasierte Skriptlösung entwickelt, die die automatisierbaren Konfigurationsänderungen umsetzen kann. Diese Maßnahmen sind in den Checklisten gekennzeichnet. In Verbindung mit einer manuellen Durchführung der nicht automatisierbaren Schritte lässt sich somit ein gutes Sicherheitsniveau erreichen. Die Bedienung der Skriptlösung wird im Folgenden vorgestellt.

Die Härtungsskripte wurden durch eine zwei-geteilte Lösung realisiert. Ein in PHP entwickeltes Skript übernimmt die Härtung der PHP-basierten CM-Systeme (Joomla!, TYPO3, WordPress). Dieses Skript passt im Wesentlichen die Konfigurationsdateien der CM-Systeme an und nimmt keine Änderungen an weiteren Komponenten vor. Darüber hinaus härtet ein Python-Skript die nicht-PHP-basierten CM-Systeme (Liferay und Plone), das Betriebssystem sowie die eingesetzten Systemkomponenten, wie beispielsweise Apache Webserver, SSH und Firewall. Die beiden Skriptlösungen können unabhängig voneinander genutzt werden.

Die Skriptlösung eignet sich sowohl für die Härtung neu eingerichteter als auch bereits bestehender Installationen. Zur umfassenden Härtung eines der unterstützten PHP-basierten CMSe empfiehlt sich im Nachgang zur Härtung des CMS selbst über das PHP-Skript die anschließende Anwendung des Python-Skriptes zur Härtung des Betriebssystems sowie der Systemkomponenten. Die nicht PHP-basierten CMSe werden ausschließlich über das Python-Skript gehärtet.

Im Nachfolgenden wird auf die jeweiligen Skript-Lösungsvarianten eingegangen.

1.1 Unterstützte Komponenten

Die dialogbasierten Skripte zur Härtung unterstützen die im Folgenden aufgeführten Komponenten in der angegebenen Version. Diese orientieren sich an der im Rahmen der Sicherheitstest verwendeten Debian 8-Umgebung. Die Funktionsfähigkeit des Härtungsskripts für andere Betriebssysteme sowie abweichende Versionen der Komponenten wurde nicht getestet und kann deshalb nicht beurteilt werden.

Betriebssystem

- Debian Linux (8.2)
- OpenSSH (6.7p1), Open SSL (1.0.1k)
- Firewall Regeln (iptables)

Content Management Systeme

- Joomla! (3.4.8)
- Typo3 (7.6.2 LTS)
- WordPress (4.4.1)
- Liferay (6.2 CE GA5)
- Plone (5.0)

Laufzeit-Umgebung

- PHP 5.6
- php5-fpm (5.6.14)
- php5-mysql (5.6.14)
- Java 8 (Oracle JDK 8u66)
- Python (2.7.9)

Web- und Application-Server

- Apache 2 Web Server (2.4.10)
- Tomcat Application Server tomcat7 (7.0.56-3)

Datenbanken

- MySQL (5.5.46)

Grundsätzlich ist zu beachten, dass die Anpassungen zur Härtung am Apache-Webserver sowie an der PHP-Laufzeitumgebung nicht durchgeführt werden sollten, sofern Servermanagement-Tools wie Plesk, Confixx, LiveConfig, sysCP o.Ä. installiert sind.

2 Bezug der Skriptlösung

Die aktuelle Version der Skriptlösung wird auf folgender Webseite zum Download angeboten:

https://github.com/CMS-Garden/cmshardening

Nach dem Download können Sie die Datei mit folgendem Befehl entpacken:

```
$ tar xfvz DATEI
```

Neben einem manuellen Download ist alternativ auch das Klonen über ein Terminal mit folgendem Befehl möglich:

- `$ sudo apt-get install git`
- `$ sudo git clone --recursive https://github.com/CMS-Garden/cmshardening`

Die Härtungsskripte liegen in den Unterverzeichnissen

- `hardening-scripts/php`
- `hardening-scripts/python`

In den folgenden Kapiteln werden beide Härtungsskripte näher beschrieben.

3 PHP-basiertes Skript

Die Härtung PHP-basierter Content-Management-Systeme erfolgt über ein PHP-Skript. Somit kann dieses Skript auch auf einer von Shared-Hosting-Anbietern zur Verfügung gestellten Installation mit sehr eingeschränktem Benutzerrechten verwendet werden. Hierzu ist jedoch ein Shell-Zugang erforderlich, um das Skript auf dem Server ausführen zu können.

Aktuell unterstützt das Skript die Härtung folgender Content-Management-Systeme:

- Joomla! (3.4.8)
- Typo3 (7.6.2 LTS)
- WordPress (4.4.1)

3.1 Voraussetzungen

Die einzige Voraussetzung zum Starten des Skriptes ist die Installation von PHP, welche jedoch durch Verwendung von PHP-basierten CMS bereits erfüllt sein sollte.

Aktuell ist das Skript ausschließlich über Kommandozeile verwendbar, jedoch ist es bereits so konstruiert, dass auch eine Web-Applikation darauf aufsetzten könnte, welche im Browser aufgerufen wird. Eine derartige Implementierung ist jedoch zur Zeit nicht realisiert.

Bei der Ausführung des Skriptes und der Härtung werden mehrere Dateien und Verzeichnisse gelesen bzw. geschrieben. Deshalb ist zu beachten, dass der Account unter dem das Skript gestartet wird entsprechende Zugriffsrechte auf diese Dateien und Verzeichnisse besitzt. Eine Auflistung der betroffenen Dateien und der Zugriffsarten ist unter Kapitel 3.5.4 Dateiberechtigungen übersichtlich dargestellt.

3.2 Starten und Startoptionen

Das PHP Skript kann komplett ohne Parameter gestartet werden. Dabei wird die Konfiguration aus der config.yml-Datei verwendet. Gibt es benötigte Angaben, die in der Konfigurationsdatei nicht gesetzt sind, so wird der Benutzer aufgefordert diese entsprechend zu tätigen.

Sollten weitere Optionen gewünscht sein, so kann über `--help` die Hilfe-Meldung anzeigt werden:

```
$ php php/index.php --help

Dieses Härtungsskript soll dabei unterstützen bestimmte Content-Management Systeme (CMS)
durch entsprechende Konfiguration sicherer zu machen.

Die folgenden Argumente werden unterstützt:

  -h | --help        Diese Hilfe-Seite

  -c | --checkonly   Durchlauf des Skriptes ohne Änderungen durchzuführen

  -i | --interactive Vor jeder Änderung wird der Benutzer gefragt, ob die Änderung
                     durchgeführt werden soll.

  -s | --silent      Es wird nicht mehr vor jeder Änderung gefragt, ob die Änderung
                     durchgeführt werden soll

  -l | --lang        Erlaubt die Änderung der Sprache auf 'en' oder 'de'. (benötigt die
                     PHP-Extension 'classkit' oder 'runkit')

Diese Parametrisierung kann ebenfalls in der config.yml vorgenommen werden. Die Angabe
der obigen Argumente überschreibt temporär die Konfiguration.
```

Der einfachste Aufruf des Skripts erfolgt über:

```
$  php ./php/index.php
```

Hierbei sind folgende Optionen aktiv:

−−checkonly

--lang=de

Soll also z.B. das Skript in englischer Sprache im interaktiven Mode (Abfrage vor jeder Änderung) gestartet werden, wird folgender Befehl verwendet:

```
$ php ./php/index.php --lang=en --interactive
```

3.3 Ablauf des Skriptes

Das Skript führt die folgenden Schritte durch:

1. Erläuterungen für den Benutzer zu Risiken und zur Handhabung des Skriptes ausgeben.
2. Es können generelle Informationen vom Benutzer abgefragt werden, z.B. ob der Server SSL/TLS unterstützt.
3. Eine Auswahl von unterstützten Modulen/CM-Systemen wird aufgelistet, als Auswahl für den Benutzer.
4. Anschließend ermittelt das Skript vom Benutzer die benötigten Informationen speziell für das Modul/CM-System, wie z.B. den Pfad zur Konfigurationsdatei.
5. Anlegen einer Sicherungskopie durch das Skript für jede anzupassende Datei (meist Konfigurationsdateien), standardmäßig im `./php/backups/`-Verzeichnis.
6. In diesem Schritt werden die tatsächlichen Härtungsschritte (sogenannte Utils) durchgeführt, wobei der Benutzer zunächst sieht, welche Konfiguration aktuell eingestellt ist, wie sie sicher gestaltet sein sollte und welche Datei betroffen ist. Gibt es Abweichungen zwischen dem Soll und Ist, wird der Benutzer gefragt, ob die Änderung durchgeführt werden soll. Diese Rückfragen können durch die Konfiguration `Dialog.Interactive = false` in der `config.yml` deaktiviert werden.
7. Da es auch Härtungsrichtlinien in den Checklisten gibt, welche nicht automatisch durch das Skript umgesetzt werden können, werden nun noch diese Richtlinien in Form einer Checkliste angezeigt, so dass der Benutzer manuell eine Umsetzung vornehmen kann.
8. Nachdem alle Utils abgearbeitet wurden, wird der Benutzer gebeten, sein System auf Funktionstüchtigkeit zu prüfen. Hier kann er alle Änderungen noch Rückgängig machen, um den Originalzustand wiederherzustellen.
9. Nachdem in einem kurzen Hinweistext auf die Backups verwiesen wurde, beendet sich das Skript.

3.4 Konfiguration des Skripts

Sollte der Nutzer Änderungswünsche zu den voreingestellten Konfiguration haben, so gibt es zwei Arten von Konfigurationen, die angepasst werden können:

- Generelle Konfiguration in `config.yml`, mit Einstellungen die den Skriptablauf beeinflussen
- Modul-Konfiguration unter `modules/<modul>.yml`, welche Modul-spezifische Einstellungen enthalten.

Die Konfigurationsdateien sind im YAML-Format erstellt, so dass auch komplexere Spezifikationen wie Listen und verschachtelte Angaben möglich sind.

3.4.1 Generelle Konfiguration

Der Inhalt der YAML-Datei config.yml beschreibt einige Einstellungen, welche die Ausführung der Skripte im Allgemeinen bestimmen. Die wesentlichen Einstellungen sind:

- `Dialog.Language` um die Sprache des PHP-Skripts einzustellen. Mögliche Werte sind aktuell `'de'` und `'en'`. Alternativ kann dies über den Kommandozeilenparameter `--lang` eingestellt werden.
- `HardeningSettings.CheckOnly` um den Checkonly-Mode zu aktivieren, in welchem das Skript durchlaufen wird, ohne dabei tatsächlich Härtungen durchzuführen. Alternativ kann dies über den Kommandozeilenparameter `--checkonly` eingestellt werden.

3.4.2 Modul-Konfiguration

Für jedes Content-Management-System (Modul) gibt es im Verzeichnis `modules/` eine YAML-Datei. Welche dieser Dateien durch das Skript geladen wird, ist in der `config.yml` unter Hardening.IncludeModules beschrieben.

Diese Modul-Konfigurationsdateien enthalten Modul-spezifische Einstellungen. Sowohl Pfade zu Dateien oder Verzeichnissen des CMS, als auch die Härtungsrichtlinien des Moduls sind hier definiert.

Will der Nutzer z.B. den Pfad der CMS-Installation nicht bei jedem Aufruf des Skriptes eingeben, kann dieser in der Konfigurationsdatei hinterlegt werden. Dabei ist die Syntax von YAML zu beachten.

Die Härtungsrichtlinien müssen in der Regel nicht vom Benutzer angepasst werden. Sollte jedoch der Wunsch bestehen, einen anderen Wert für eine Konfiguration zu setzen, kann dies hier eingestellt werden.. Auch können weitere Härtungsrichtlinien der selben Art hier ohne Programmieraufwand hinzugefügt werden.

3.5 Zusätzliche Informationen

3.5.1 Sicherungen

Für alle automatischen Änderungen des Skriptes an Dateien wird eine Sicherungskopie im Verzeichnis `backups/` angelegt. Der Dateiname der Backup-Datei enthält neben dem originalen Dateinamen auch einen Zeitstempel.

Die manuell durchgeführten Änderungen werden nicht durch das Skript gesichert.

3.5.2 Log Dateien

Während der Ausführung des Skriptes werden Log-Einträge erstellt, so dass der Verlauf des Skriptes auch im Nachhinein nachvollziehbar ist. Dabei werden ähnliche Informationen geloggt, wie dem Benutzer präsentiert werden. Für jeden Start des Skriptes wird eine neue Log-Datei im Verzeichnis `logs/` mit einem Zeitstempel im Dateinamen erstellt.

3.5.3 Sprachen

Aktuell werden die Sprachen Deutsch (`'de'`) und Englisch (`'en'`) unterstützt. Diese Sprachen können in der `config.yml` oder temporär über das Kommandozeilen-Argument `--lang=xx` bzw. `-l xx` konfiguriert werden. Weitere Sprachen können analog zu den Dateien im Verzeichnis `lang/` erstellt werden.

3.5.4 Dateiberechtigungen

Das Skript greift auf einige Dateien und Verzeichnisse mit verschiedenen Aktionen zu, so dass die Dateiberechtigungen vor der Ausführung geprüft werden sollten (siehe Tabelle 1).

Datei/Verzeichnis	Zugriff	Beschreibung
`php/`	lesen	Das Skript muss die Skript-Dateien lesen können.
`php/backups/`	schreiben	Das Skript erstellt Backup-Dateien. Falls das Verzeichnis nicht existiert, wird das Skript dieses mit entsprechenden Rechten erstellen.
`php/logs/`	schreiben	Das Skript erstellt Log-Dateien. Falls das Verzeichnis nicht existiert, wird das Skript dieses mit entsprechenden Rechten erstellen.
`php/lang/langcache/`	lesen/ schreiben	Das Skript muss Sprach-Cache-Dateien anlegen können. Falls das Verzeichnis nicht existiert, wird das Skript dieses mit entsprechenden Rechten erstellen.
`$WordPress/wp-config.php`	lesen/ schreiben	Zum Härten der WordPress Konfiguration
`$TYPO3/typo3conf/LocalConfiguration.php`	lesen/ schreiben	Zum Härten der TYPO3 Konfiguration
`$JOOMLA/configuration.php`	lesen/ schreiben	Zum Härten der Joomla! Konfiguration
`$TYPO3/ENABLE_INSTALL_TOOL`	löschen	Zum Löschen des Markers für die Aktivierung des Install Tools

Tabelle 1: Dateiberechtigungen

Diese Liste ist nicht zwangsweise vollständig, da durch weitere Härtungsmaßnahmen auch weitere Dateien und Verzeichnisse betroffen sein könnten. Platzhalter wie z.B. $JOOMLA beschreiben die Root-Verzeichnisse der CMS-Installationen, welche vom Benutzer angegeben werden.

4 Python-basiertes Skript

Das python-basierte Skript dient der Härtung des Betriebssystems, allgemeiner Systemkomponenten sowie nicht PHP-basierter Content-Management-Systeme. In der aktuellen Version unterstützt es folgende Komponenten:

Betriebssystem

- Debian Linux (8.2)
- OpenSSH (6.7p1), Open SSL (1.0.1k)
- Netzwerk, Firewall Regeln (iptables)

Content Management Systeme

- Liferay (6.2 CE GA5)
- Plone (5.0)

Laufzeit-Umgebung

- PHP 5.6-FPM
  - Besonderheiten für Wordpress
  - Besonderheiten für Joomla
  - Besonderheiten für Typo3
- Java 8 (Oracle JDK 8u66)
- Python (2.7.9)

Web- und Application-Server

- Apache 2 Web Server (2.4.10)
  - Besonderheiten für Wordpress
  - Besonderheiten für Joomla
  - Besonderheiten für Typo3
- Tomcat Application Server tomcat7 (7.0.56-3)

Datenbanken

- MySQL (5.5.46)

Wenn auf dem zu härtenden System Servermanagement-Tools, wie zum Beispiel Plesk oder Confixx eingesetzt werden, sollten die Anpassungen an PHP und Apache mit besonderer Sorgfalt vorgenommen werden, da diese zu Einschränkungen der Funktionalität der Servermanagement-Tools führen können.

4.1 Voraussetzungen

Die Skriptlösung eignet sich sowohl für die Härtung neu eingerichteter als auch bereits bestehender Installationen. Das Skript wird über die Kommandozeile gestartet und benötigt zwingend „sudo“-Rechte, da bei Ausführung und Härtung des Skriptes mehrere Dateien und Verzeichnisse gelesen, geschrieben oder gelöscht werden, beziehungsweise Rechte und Benutzer von Dateien und Verzeichnissen geändert werden. Es ist somit zur Anwendung auf Webservern bestimmt, auf denen ein Vollzugriff möglich ist. Aus diesem Grund eignet sich dieses Skript nicht zur Verwendung in Shared-Hosting-Umgebungen.

Das Ausführen des Skriptes setzt eine aktuelle Python-Installation der Major-Version 2 oder 3 voraus und benötigt folgende zusätzlichen Debian-Pakete:

- python-yaml
- python-netifaces
- python-six
- python-lxml
- netfilter-persistent
- iptables-persistent

Diese können mit dem folgenden Befehl installiert werden:

```
$ sudo apt-get install python-yaml python-netifaces python-six python-lxml
netfilter-persistent iptables-persistent
```

Die Skripte sind ausschließlich zur lokalen Härtung bestimmt und unterstützen keine Remote-Härtung.

4.2 Starten und Startoptionen

Das Skript wurde so entwickelt, dass es ohne vorherige Konfiguration gestartet werden kann. Notwendige Konfigurationen werden entweder selbst durch das Skript ermittelt oder der Benutzer wird aufgefordert entsprechende Angaben zur Laufzeit des Skriptes zu machen.

Es ist zwingend erforderlich, dass der Aufruf des Skriptes über den sudo-Befehl erfolgt und nicht als Benutzer root. Dies ist notwendig, da bei der Härtung von OpenSSH die Benutzer eingeschränkt werden, die sich per SSH am Server anmelden dürfen. Dabei wird der aktuelle Benutzer vom Skript ermittelt und dieser Liste hinzugefügt.

Eine Auflistung aller Optionen kann über `--help` anzeigt werden:

```
$ sudo python hardening.py --help
optional arguments:
  -h, --help            show this help message and exit
  --mode {interactive,diff,silent,check-only}
                        Select the mode in which this script will run. Default
                        is check-only.
  --lang LANG           Select language for this tool. Default is your system
                        locale or english. Valid options are de_DE (or only
                        'de') and en_US (or only 'en').
  --log logfile         Display a log of applied changes; you can use '-' as
                        filename to log to stdout
```

```
  --version             Display the version of this tool.
  --documentation       Display the full documentation in markdown format.
```

Der einfachste Aufruf des Skripts erfolgt über:

```
$ sudo python hardening.py
```

Hierbei sind folgende Optionen aktiv:

--mode=check-only

--lang=de_DE

--log=/root/TODO

Die möglichen Modi werden im Folgenden erläutert:

interactive:

Im Interactive-Modus fragt das Skript zu Beginn nach den auszuführenden Modulen. Sind diese ausgewählt, beginnt entsprechend die Härtung, wobei vor jeder Maßnahme die vorzunehmenden Änderungen angezeigt werden. Anschließend erfolgt eine Abfrage, ob der Benutzer die Härtungsmaßnahme durchführen möchte. Er hat die Möglichkeit, die Härtungsmaßnahme durchzuführen zu lassen, zu überspringen oder die gesamte Härtung abzubrechen.

Bei Abbruch der Härtung werden keine Änderungen vorgenommen, auch keine die vorher zugelassen wurden. Damit findet ein komplettes Roll-back statt.

diff:

Im Differenz-Modus wird der Anwender ebenfalls in einem interaktiven Menü nach den auszuführenden Modulen befragt. Die Maßnahmen werden nach Inhalt gruppiert - zum Beispiel alle Änderung an einer bestimmten Datei - und dem Benutzer als Abweichung zur bisherigen Konfiguration angezeigt. Der Benutzer hat auch hier die Möglichkeit, die Änderungen durchführen zu lassen, zu überspringen oder das Skript abzubrechen, jedoch wird durch die Gruppierung die Häufigkeit dieser Abfragen reduziert.

Bei Abbruch der Härtungen werden ebenfalls keine Änderung geschrieben.

silent:

Beim Silent-Modus wird dem Benutzer ausgegeben, welche Härtung gerade vorgenommen wird, es erfolgt jedoch keine Nachfrage. Die Auswahl der auszuführenden Härtungsmodule erfolgt in der Datei „hardening-scripts/python/config.yml“ (mehr dazu in Kapitel 4.4).

Das Skript kann per <Strg-C> abgebrochen werden, wobei Härtungen, welche sich rückgängig machen lassen, rückgängig gemacht werden.

checkonly:

Der Checkonly-Modus ist der „Default“-Modus für das Härtungsskript. Der Checkonly-Modus ist gleich dem Interactive-Modus, mit dem Unterschied, dass keine Änderungen durchgeführt werden, sondern lediglich eine textuelle Ausgabe erfolgt.

4.3 Ablauf des Skriptes

Das Skript wird in den folgenden Schritten durchlaufen:

1. Erläuterungen für den Benutzer zu Risiken und zur Handhabung des Skriptes ausgeben
2. Eine Auswahl von unterstützten Modulen wird aufgelistet, als Auswahl für den Benutzer. (Im Silent-Modus werden die Module aus der generellen Konfiguration geladen.)

3. Es ist möglich, dass generelle Informationen vom Benutzer abgefragt werden, z.B. welches Netzwerk-Interface verwendet werden soll.
4. In diesem Schritt werden die tatsächlichen Härtungsschritte (sogenannte Utils) durchgeführt, wobei dem Benutzer angezeigt wird, was durch den Härtungsschritt verändert wird. Je nach gewähltem Modus kann der Benutzer bestimmen, ob er den jeweiligen Härtungsschritt durchführen möchte. Dabei legt das Skript im Verzeichnis „/root/hardening-backup/“ für jede geänderte Datei oder jedes geändertes Verzeichnis eine Kopie des Originals an.
5. Nachdem in einem kurzen Hinweistext auf den nötigen Neustart des System hingewiesen wurde, beendet sich das Skript.

4.4 Konfiguration des Skripts

Sollte der Nutzer Änderungswünsche zu den voreingestellten Werten haben, so gibt es zwei Arten von Konfigurationen, die angepasst werden können:

- Generelle Konfiguration in `config.yml`, mit Einstellungen die den Skriptablauf beeinflussen.
- Modul-Konfiguration unter modules/<modul>.yml, welche Modul-spezifische Einstellungen enthalten.

Die Konfigurationsdateien sind im YAML-Format erstellt, so dass auch komplexere Spezifikationen wie Listen und verschachtelte Angaben möglich sind.

4.4.1 Generelle Konfiguration

Der Inhalt der YAML-Datei config.yml beschreibt einige Einstellungen, welche die Ausführung der Skripte im Allgemeinen bestimmen. Diese sind in der Datei näher erläutert. Die wichtigsten Einstellungen sind:

- `RunModules` und `CMS`, um die auszuführenden Module auszuwählen und CMS-spezifische Einstellungen für die Module einzuschalten. Valide Werte sind:
  - `RunModules: [apache, debian, java, mysql, network, php, python, sshd, tomcat]`
  - `CMSModules: [joomla, liferay, plone, typo3, wordpress]`

  Die Einstellungen werden nur im silent-Modus verwendet. Bei allen anderen Modi werden diese Werte vorher vom Benutzer erfragt.
- `HardeningSettings`, um sehr spezifische und vom Nutzer zu konfigurierende Informationen zu definieren. Änderungen an diesen Einstellungen sind standardmäßig nicht nötig, erlauben jedoch erfahrenen Benutzern eine einfache Konfiguration häufig in der Standardkonfiguration angepasster Werte (wie zum Beispiel dem Port für den SSH-Dienst).

4.4.2 Modul-Konfiguration

Für jedes Modul gibt es im Verzeichnis `modules/` eine YAML-Datei. Welche dieser Dateien durch das Skript geladen wird bestimmt der Benutzer zur Laufzeit. Nur für den silent-Mode ist dies in der `config.yml` unter RunModules beschrieben.

Diese Modul-Konfigurationsdateien enthalten Modul-spezifische Einstellungen. Sowohl Pfade zu Dateien oder Verzeichnissen des CMS als auch die Härtungsrichtlinien des Moduls sind hier definiert. Alle Parameter sind in den jeweiligen Dateien näher erläutert.

Die Härtungsrichtlinien müssen in der Regel nicht vom Benutzer angepasst werden. Sollte jedoch der Wunsch bestehen, einen anderen Wert für eine Konfiguration zu setzen, kann dies hier eingestellt werden. Ebenfalls können weitere Härtungsrichtlinien der selben Art ohne Programmieraufwand hinzugefügt werden.

4.5 Zusätzliche Informationen

4.5.1 Sicherungen

Für alle automatischen Änderungen des Skriptes an Dateien wird eine Sicherungskopie im in der globalen Konfigurationsdatei angegebenen Verzeichnis, standardmäßig `/root/hardening-backups/`, angelegt. Der Dateiname der Backup-Datei enthält neben dem originalen Dateinamen auch einen Zeitstempel.

4.5.2 Log Dateien

Über den Parameter `-log` kann ein Pfad für die Log-Ausgabe angegeben werden. Das Log beschreibt die Änderungen welche am System durchgeführt wurden. Wird der Parameter `-log` nicht angegeben, so wir das Log im Home-Verzeichnis des Superusers abgelegt. Mit '-' als Wert kann die Ausgabe des Logs in der Konsole erfolgen.

4.5.3 Sprachen

Aktuell werden die Sprachen Deutsch (`'de_DE'`) und Englisch (`'en_US'`) unterstützt. Diese Sprachen können über das Kommandozeilen-Argument `--lang=xx` konfiguriert werden. Weitere Sprachen können analog zu den Dateien im Verzeichnis locale/ erstellt werden. Die Standard-Sprache ist Deutsch.

4.5.4 Farbige Ausgaben

Es wird empfohlen, das Paket `coloredlogs` zu installieren, um farbige Konsolen-Ausgaben zu erhalten.

Die Installation kann per:

```
sudo apt-get install python-colorlog
```

oder bei Verwendung von Python3 mit:

```
sudo apt-get install python3-colorlog
```

erfolgen.