Bundesamt
für Sicherheit in der
Informationstechnik

Studie „Sicherheitsuntersuchung von Content-Management-Systemen“

Nachweis und Ergebnis erfolgter Qualitätssicherungsmaßnahmen bezüglich der erstellten Skriptlösung

Änderungshistorie

Version	Datum	Name	Beschreibung
0.1	04.04.2016	[redacted]	Initialerstellung
0.2	19.04.2016	[redacted]	Einarbeitung der Tests für die PHP-Skript-Lösung
0.3	21.04.2016	[redacted]	Ergänzung Unit-Tests PHP
0.4	03.05.2016	[redacted]	Ergänzung Unit-Tests Python
0.5	09.05.2016	[redacted]	Review
1.0	11.05.2016	[redacted]	Freigabe
2.0	26.05.2016	[redacted]	Finalisierung nach Rückmeldung BSI
3.0	11.08.2016	[redacted]	Ergänzung Pylint Report
4.0	12.08.2016	[redacted]	Aktualisierung Pylint Report

Vorlage:

Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63
53133 Bonn
Internet: https://www.bsi.bund.de

Impressum

Herausgeber:

Test and Integration Center
T-Systems Multimedia Solutions GmbH
Riesaer Str. 5
01129 Dresden

Das Test and Integration Center Dresden der T-Systems Multimedia Solutions GmbH ist ein durch die DAkkS nach DIN EN ISO/IEC 17025 akkreditiertes Prüflaboratorium.

Die Akkreditierung gilt für die in der Urkunde aufgeführten Prüfverfahren.

Registriernummer der Urkunde: D-PL-12109-01-01

Freigabe:

Name	Funktion	Unterschrift
[redacted]	Projektleiter	

Inhaltsverzeichnis

1 Einleitung

Im Rahmen der Studie „Sicherheitsuntersuchung von Content-Management-Systemen" [1] wurden für klar definierte Anwendungsszenarien übersichtliche, dialogbasierte Skripte zur Härtung bereitgestellt, welche eine automatische Konfiguration ermöglichen.

Im vorliegenden Dokument werden die Maßnahmen zur Qualitätssicherung der Skriptlösung sowie die durchgeführten Überprüfungen und Ergebnisse dokumentiert. Die Maßnahmen werden jeweils in die den Skripten zugrundeliegenden Programmiersprachen „PHP" und „Python" eingeteilt.

Generell wurden die Härtungsskripte durch eine zweigeteilte Skript-Lösung realisiert. Ein PHP-basiertes Skript übernimmt die Härtung der PHP-basierten CM-Systeme (WordPress, Joomla, Typo3). Dabei wurden für jedes CMS die Einstellungen realisiert, welche direkt das CMS betreffen, d.h. die CMS-Konfiguration. Darüber hinaus härtet ein Python-basiertes Skript die nicht-PHP-basierten CM-Systeme (Liferay und Plone) und die eingesetzten Systemkomponenten, wie beispielsweise Apache Webserver, SSH und Firewall. Die beiden Skriptlösungen werden unabhängig voneinander ausgeführt und rufen sich nicht gegenseitig auf, da jede der beiden Lösungen eine eigene Transaktionsverwaltung besitzt.

2 Qualitätssicherungsmaßnahmen

Für die Entwicklung wurden Programmierrichtlinien verwendet, deren Einhaltung regelmäßig geprüft wurde. Details sind in Kapitel 2.1 dargestellt.

Die Qualitätssicherung der Skripte erfolgte im Rahmen eines manuellen Funktionstests (siehe Kapitel 2.3) und durch Anwendung von Unit-Tests für einzelne Module (siehe Kapitel 2.2). Dabei wurden folgende Punkte berücksichtigt:

- Die Funktionstests wurden für das jeweils umfangreichste Szenario eines CMS durchgeführt.
- Die Benutzerschnittstelle (UI) wurde auf Verständlichkeit und Bedienbarkeit geprüft, wobei dem Nutzer ein grundlegendes Verständnis hinsichtlich des Umgangs mit der Kommandozeile unterstellt wird.
- An einem Testsystem wurde für jedes CMS jeweils geprüft, ob die in den Checklisten dokumentierten Konfigurationseinstellungen korrekt umgesetzt wurden.
- Alle Skripte wurden auch in Kombination getestet, allerdings erfolgte kein Test von Wechselwirkungen von Skripten für unterschiedliche Komponenten.
- Die Testsysteme wurden nach abgeschlossener Härtung neu gestartet und die Funktionalität der gehärteten Komponenten überprüft.
- Die Unit-Tests decken vorrangig die zur Härtung am häufigsten benötigten Module und Funktionen ab.

2.1 Qualität des Quellcodes

Im Folgenden werden die Maßnahmen beschrieben, um die Codequalität der Skriptlösung zu gewährleisten.

2.1.1 PHP

Für die Prüfung der PHP-Quellcodequalität wurde die in der Entwicklungsumgebung PhpStorm [2] inkludierte Code Inspection genutzt. Im Rahmen dieser Analyse wurde der Code auf folgende Merkmale untersucht:

- Identifikation möglicher Fehler
- Lokalisierung von ungenutztem Code
- Detektion möglicher Performance-Probleme
- Verbesserung der Codestruktur und Wartbarkeit
- Konformität zu Coding Guidelines und Standards

2.1.2 Python

Die Python-Module wurden anhand der Coding Conventions der Python Software Foundation [3] implementiert.

Die Einhaltung der Richtlinien wurde mit dem Programm „Pylint" [4] in Version 1.5.5 geprüft. Hierbei waren folgende Prüfungen inkludiert:

- Prüfung des Coding Standards (entsprechend Style Guide for Python Code [3])

    - Prüfung der Zeilenlängen
    - Analyse der Variablennamen
    - Prüfung der Verwendung importierter Module
- Fehlererkennung
    - Implementierung von Interfaces
    - Nutzung importierter Module

Der nach Entwicklungsende generierte Pylint Report ist in Anhang A Pylint Report beigefügt.

2.2 Unit-Tests

Im Rahmen von Unit-Tests (Komponententests) wurden die einzelnen Module auf funktionale Korrektheit überprüft. Hierzu wurden mit Hilfe von Test-Frameworks für PHP und Python entsprechende Testhilfsprogramme geschrieben. Über die Test-Frameworks werden die einzelnen Testklassen aufgerufen, deren Komponententests ausgeführt und das Ergebnis ausgegeben.

Unit-Tests testen ein Modul isoliert, d. h. weitgehend ohne Interaktion mit anderen Modulen. Aufgrund dieser Funktionsweise können Wechselwirkungen nicht geprüft werden. Zudem können Unit-Tests nur solche Fehler finden, zu deren Entdeckung die programmierten Prüfungen geeignet sind.

2.2.1 PHP

Für das PHP-Skript wurden Unit-Tests auf Basis von PHPUnit [5] geschrieben. Die Datei php/phpunit.xml konfiguriert die Unit-Tests. Folgende Komponenten des PHP-Skripts werden geprüft:

- Configuration: tests/configuration
    - ConfigurationTest.php
- Parser: tests/parser
    - HtAccessParserTest.php
    - JoomlaConfigParserTest.php
    - PHPUserIniParserTest.php
    - Typo3ConfigParserTest.php
    - WordpressConfigParserTest.php
- Transaction: tests/transaction
    - TransactionTest.php
    - FileTransactionTest.php
- Utils: tests/utils/filesystem
    - DeleteTypo3ConfigFileTest.php

Insgesamt sind 51 Unit-Tests mit 110 Assertions implementiert worden.

Die Unit-Tests im Verzeichnis `tests/` samt der phpunit.xml sind Bestandteil des Skriptes und können mit Hilfe von PHPUnit auf jedem System genutzt werden. Hierfür müssen lediglich PHPUnit [5] installiert, in das `php/`-Verzeichnis gewechselt und `phpunit` ausgeführt werden.

2.2.2 Python

Für das Python-Skript wurden Unit-Test mithilfe des Python „Unit testing frameworks" [6] geschrieben. Die folgenden Komponenten werden abgedeckt:

- Utils und Transactions: tests/
    - TestConfigLine.py
    - TestDirectoryTransaction.py
    - TestFileTransaction.py
    - TestPasswdTransaction.py
    - TestTextFileTransaction.py
    - TestTransactionStates.py

Die Unit-Tests im Verzeichnis `tests/` sind Bestandteil des Skriptes und können ohne zusätzliche Module mit einer aktuellen Python-Installation gestartet werden.

2.3 Manuelle Tests

Da die in Kapitel 2.2 beschriebenen Unit-Tests nur isolierte Module betrachten und folglich die Wechselwirkungen von Modulen und zusammenhängende Prozessschritte nicht untersuchen können, wurden darüber hinaus manuelle Tests durchgeführt, in deren Rahmen bestimmte Eigenschaften und Verhaltensweisen der Skriptlösung auf Korrektheit geprüft wurden.

Im Folgenden werden die durchgeführten manuellen Funktionstests, die zugrundeliegende Umgebung sowie das Ergebnis beschrieben.

2.3.1 Testumgebung

Die Tests wurden in einer Testumgebung durchgeführt, die sich an den Vorgaben der Studie orientierte ([1], Kapitel 5.2.1).

Die Einzelkomponenten wurden aus den Debian-Repositories gezogen, um eine möglichst realitätsnahe Umgebung für die Härtung als Grundlage zu nutzen. Diese Testumgebung beinhaltete folgende Komponenten:

- Debian 8.2
- Apache 2.4.10
- MySQL 5.5.44
- PHP-FPM 5.6.19
- Python 2.79
- Java 7 (Open JDK 7u79)
- WordPress 4.3.1
- Joomla 3.4.4
- Typo3 7.5.0
- Plone 5
- Liferay 6.2 GA5

- Tomcat 7.0.68

Hierbei wurden mehrere Instanzen analog der Testumgebung aufgesetzt, d.h. es wurde jeweils eine Instanz für WordPress, Joomla, Typo3, Plone und Liferay mit den erforderlichen Komponenten installiert.

Die Komponenten wurden soweit wie nötig konfiguriert, jedoch nicht gehärtet.

Es erfolgte außerdem ein Test auf den in der Studie [1] verwendeten Systemen im jeweils umfangreichsten Szenario.

2.3.2 PHP

Für die PHP-basierten Skripte wurden die in den folgenden Kapiteln dargestellten Testfälle durchgeführt.

2.3.2.1 Test der Kommandozeilen-Argumente

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf des Skripts mit Parameter -h bzw. --help	Das Skript wird erfolgreich gestartet. Hilfe-Funktion wird angezeigt.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter -c bzw. --checkonly	Das Skript wird erfolgreich gestartet und ausgeführt, jedoch werden Härtungen nur simuliert und nicht durchgeführt.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter -i bzw. --interactive	Das Skript wird erfolgreich gestartet und ausgeführt. Bei jeder Härtungsmaßnahme wird der Benutzer gefragt, ob er diese umsetzen oder überspringen möchte.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter -s bzw. --silent	Das Skript wird erfolgreich gestartet. Dem Benutzer wird ausgegeben, was gehärtet wurde, aber es erfolgt keine Abfrage von Benutzereingaben.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter -l bzw. --lang (de|en)	Das Skript wird erfolgreich in der ausgewählten Sprache gestartet. Die Ausführung entspricht dem Aufruf mit dem Parameter '-i'.	ERFOLGREICH	-
Aufruf des Skripts ohne Parameter	Das Skript wird erfolgreich gestartet. Die Ausführung entspricht dem Aufruf mit dem Parameter '-i'.	ERFOLGREICH	-

2.3.2.2 Test der Sprachdateien

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf des Skripts in Sprache deutsch (--lang de)	Das Skript wird erfolgreich gestartet. Ausgaben erfolgen in deutscher Sprache.	ERFOLGREICH	-

Aufruf des Skripts in Sprache englisch (--lang en)	Das Skript wird erfolgreich gestartet. Ausgaben erfolgen in englischer Sprache.	ERFOLGREICH	-
Aufruf des Skripts ohne Sprachangabe	Das Skript wird erfolgreich gestartet. Ausgaben erfolgen in deutscher Sprache.	ERFOLGREICH	-

2.3.2.3 Test der Konfigurationsdatei

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Prüfung der Korrektheit der Konfigurationsdatei	Die Konfigurationsdatei ist vorhanden, fehlerfrei und syntaktisch korrekt. Die Konfigurationsdatei wird vom Skript erfolgreich erkannt.	ERFOLGREICH	-
Fehlerhafte Konfigurationsdatei	Die Konfigurationsdatei ist fehlerhaft. Das Skript bricht die Ausführung ab.	ERFOLGREICH	-

2.3.2.4 Test der Benutzer-Schnittstelle

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Eingabe korrekter Werte nach Aufforderung durch das Skript	Das Skript wird erfolgreich weiter ausgeführt.	ERFOLGREICH	-
Eingabe falscher Werte nach Aufforderung durch das Skript	Der erforderliche Wert wird erneut abgefragt.	ERFOLGREICH	-
Abbruch des Skriptes (Strg-C)	Das Skript wird abgebrochen. Es wird keine Härtung vorgenommen.	ERFOLGREICH	-

2.3.2.5 Test der Funktionalität der Komponenten - ungehärtetes System

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf auf einem ungehärteten System	Das Skript wird fehlerfrei abgearbeitet. Inhalte der Konfigurationsdateien werden entsprechend der Härtungsmaßnahmen aktualisiert.	ERFOLGREICH	-
Prüfung der Umsetzung der Härtung	Alle Härtungsmaßnahmen wurden erfolgreich umgesetzt.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CM-Systems	Das CM-System arbeitet fehlerfrei. Webseiten des Frontends können vollständig und korrekt abgerufen werden.	ERFOLGREICH	-

Test der Funktionsfähigkeit des CMS-Backends	Ein Login in das Backend ist möglich.	ERFOLGREICH	-

2.3.2.6 Test der Funktionalität der Komponenten - gehärtetes System

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf auf einem bereits nach den Härtungsmaßnahmen aus der Studie [1] gehärteten System	Das Skript wird fehlerfrei abgearbeitet. Es werden keine Einstellungen geändert.	ERFOLGREICH	-
Prüfung der Umsetzung der Härtung	Alle Härtungsmaßnahmen sind weiterhin korrekt umgesetzt.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CM-Systems	Das CM-System arbeitet fehlerfrei. Webseiten des Frontends können vollständig und korrekt abgerufen werden.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CMS-Backends	Ein Login in das Backend ist möglich.	ERFOLGREICH	-

2.3.2.7 Test der Funktionalität der Komponenten - teilweise gehärtetes System

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf auf einem teilweise gehärteten System	Das Skript wird fehlerfrei abgearbeitet. Es werden nur die noch nicht erfolgten Einstellungen geändert.	ERFOLGREICH	-
Prüfung der Umsetzung der Härtung	Alle Härtungsmaßnahmen wurden bzw. sind weiterhin korrekt umgesetzt.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CM-Systems	Das CM-System arbeitet fehlerfrei. Webseiten des Frontends können vollständig und korrekt abgerufen werden.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CMS-Backends	Ein Login in das Backend ist möglich.	ERFOLGREICH	-

2.3.2.8 Test der Backup-Funktion

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Prüfung der Inhalte des Backup-Ordners nach Durchlauf des Skripts	Für jede geänderte Datei ist eine Kopie der originalen Datei im Backup-Ordner vorhanden.	ERFOLGREICH	-

Prüfung der Inhalte der Backup-Dateien	Die Backup-Datei entspricht dem Stand der Datei vor der erfolgten Härtung.	ERFOLGREICH	-

2.3.2.9 Test der Logging-Funktion

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Prüfung der Inhalte der Logging-Datei	Alle Änderungen werden korrekt und vollständig in der Logging-Datei protokolliert.	ERFOLGREICH	-

2.3.2.10 Test der Rollback-Funktion

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf der Rollback-Funktion (durch vorzeitigen Abbruch des Skriptes an unterschiedlichen Stellen)	Die Originaldateien werden erfolgreich aus dem Backup-Ordner wiederhergestellt.	ERFOLGREICH	-

2.3.3 Python

Für die Python-basierten Skripte wurden die in den folgenden Kapiteln dargestellten Testfälle durchgeführt.

2.3.3.1 Test der Kommandozeilen-Argumente

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf des Skripts mit Parameter --help	Das Skript wird erfolgreich gestartet. Die Hilfe-Funktion wird angezeigt.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter --diff	Das Skript wird erfolgreich gestartet und ausgeführt. Nach jeder Gruppe von Härtungsmaßnahmen wird dem Benutzer eine Differenz des aktuellen und neuen Standes angezeigt und er wird gefragt, ob er den neuen Stand übernehmen möchte.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter --checkonly	Das Skript wird erfolgreich gestartet und ausgeführt, jedoch werden Härtungen nur simuliert und nicht durchgeführt.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter --interactive	Das Skript wird erfolgreich gestartet und ausgeführt. Bei jeder Härtungsmaßnahme wird der Benutzer gefragt, ob er diese	ERFOLGREICH	-

	umsetzen oder überspringen möchte.		
Aufruf des Skripts mit Parameter --silent	Das Skript wird erfolgreich gestartet. Dem Benutzer wird ausgegeben, was gehärtet wurde, aber es erfolgt keine Abfrage von Benutzereingaben.	ERFOLGREICH	-
Aufruf des Skripts mit Parameter bzw. --lang (de| en)	Das Skript wird erfolgreich in der ausgewählten Sprache gestartet. Die Ausführung entspricht dem Aufruf mit dem Parameter '--interactive'.	ERFOLGREICH	-
Aufruf des Skripts ohne Parameter	Das Skript wird erfolgreich gestartet. Die Ausführung entspricht dem Aufruf mit dem Parameter '--interactive'.	ERFOLGREICH	-

2.3.3.2 Test der Sprachdateien

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf des Skripts in Sprache deutsch (--lang de)	Das Skript wird erfolgreich gestartet. Ausgaben erfolgen in deutscher Sprache.	ERFOLGREICH	-
Aufruf des Skripts in Sprache englisch (--lang en)	Das Skript wird erfolgreich gestartet. Ausgaben erfolgen in englischer Sprache.	ERFOLGREICH	-
Aufruf des Skripts ohne Sprachangabe	Das Skript wird erfolgreich gestartet. Ausgaben erfolgen in deutscher Sprache.	ERFOLGREICH	-

2.3.3.3 Test der Konfigurationsdatei

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Prüfung der Korrektheit der Konfigurationsdatei	Die Konfigurationsdatei ist vorhanden, fehlerfrei und syntaktisch korrekt. Die Konfigurationsdatei wird vom Skript erfolgreich erkannt.	ERFOLGREICH	-
Fehlerhafte Konfigurationsdatei	Die Konfigurationsdatei ist fehlerhaft. Das Skript bricht die Ausführung ab.	ERFOLGREICH	-

2.3.3.4 Test der Benutzer-Schnittstelle

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Eingabe korrekter Werte	Das Skript wird erfolgreich weiter	ERFOLGREICH	-

nach Aufforderung durch das Skript	ausgeführt.		
Eingabe falscher Werte nach Aufforderung durch das Skript	Der erforderliche Wert wird erneut abgefragt.	ERFOLGREICH	-
Abbruch des Skriptes (Strg-C)	Das Skript wird abgebrochen. Es wird keine Härtung vorgenommen.	ERFOLGREICH	-
Fehlende Python-Module oder Systemkomponenten	Das Skript wird abgebrochen. Der Benutzer wird auf die fehlende(n) Komponente(n) hingewiesen.	ERFOLGREICH	-

2.3.3.5 Test der Funktionalität der Komponenten - ungehärtetes System

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf auf einem ungehärteten System	Das Skript wird fehlerfrei abgearbeitet. Inhalte der Konfigurationsdateien werden entsprechend der Härtungsmaßnahmen aktualisiert.	ERFOLGREICH	-
Prüfung der Umsetzung der Härtung	Alle Härtungsmaßnahmen wurden erfolgreich umgesetzt.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CM-Systems	Das CM-System arbeitet fehlerfrei. Webseiten des Frontends können vollständig und korrekt abgerufen werden.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CMS-Backends	Ein Login in das Backend ist möglich.	ERFOLGREICH	-

2.3.3.6 Test der Funktionalität der Komponenten - gehärtetes System

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf auf einem bereits nach den Härtungsmaßnahmen aus der Studie [1] gehärteten System	Das Skript wird fehlerfrei abgearbeitet. Es werden keine Einstellungen geändert.	ERFOLGREICH	-
Prüfung der Umsetzung der Härtung	Alle Härtungsmaßnahmen sind weiterhin korrekt umgesetzt.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CM-Systems	Das CM-System arbeitet fehlerfrei. Webseiten des Frontends können vollständig und korrekt abgerufen werden.	ERFOLGREICH	-
Test der	Ein Login in das Backend ist	ERFOLGREICH	-

Funktionsfähigkeit des CMS-Backends	möglich.		

2.3.3.7 Test der Funktionalität der Komponenten - teilweise gehärtetes System

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Aufruf auf einem teilweise gehärteten System	Das Skript wird fehlerfrei abgearbeitet. Es werden nur die noch nicht erfolgten Einstellungen geändert.	ERFOLGREICH	-
Prüfung der Umsetzung der Härtung	Alle Härtungsmaßnahmen wurden bzw. sind weiterhin korrekt umgesetzt.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CM-Systems	Das CM-System arbeitet fehlerfrei. Webseiten des Frontends können vollständig und korrekt abgerufen werden.	ERFOLGREICH	-
Test der Funktionsfähigkeit des CMS-Backends	Ein Login in das Backend ist möglich.	ERFOLGREICH	-

2.3.3.8 Test der Backup-Funktion

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Prüfung der Inhalte des Backup-Ordners nach Durchlauf des Skripts	Für jede geänderte Datei ist eine Kopie der originalen Datei im Backup-Ordner vorhanden.	ERFOLGREICH	-
Prüfung der Inhalte der Backup-Dateien	Die Backup-Datei entspricht dem Stand der Datei vor der erfolgten Härtung.	ERFOLGREICH	-
Prüfung der Berechtigungen der Backup-Dateien	Die Berechtigungen (User, Group, Others) der Kopie sind identisch zur Originaldatei.	ERFOLGREICH	User- und Group-Zuordnung selbst werden nicht übernommen, da die verwendete Funktion shutil.copytree diese Informationen nicht kopieren kann (Linux-Standard).

2.3.3.9 Test der Logging-Funktion

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Prüfung der Inhalte der Logging-Ausgabe	Log-Nachrichten werden korrekt und vollständig ausgegeben.	ERFOLGREICH	-

Prüfung des Log-Levels	Es werden nur Log-Meldungen ausgegeben, die dem angegebenen Log-Level oder einem in der Kritikalität höherwertigen Log-Level entsprechen.	ERFOLGREICH	-

2.3.3.10 Test der Abbruch-Funktion

Testfall	Erwartetes Ergebnis	Status	Bemerkung
Vorzeitiges Abbrechen des Skriptes (an unterschiedlichen Stellen)	Die Originaldateien werden nicht verändert.	ERFOLGREICH	-

2.3.4 Abgrenzung

Die für den Test ausgewählten Module und Funktionalitäten orientieren sich an den Anforderungen an die Härtungsskripte sowie an die innerhalb der Studie [1] abgegrenzten Systeme und Komponenten.

Nicht getestet wurde deshalb das Verhalten der Skripte auf anderen Betriebssystemen (z.B. CentOS, Ubuntu oder Linux-Mint) oder beim Einsatz anderer Systemkomponenten für Befehlsinterpreter (Ruby, Perl), Webserver (Nginx, Google Web Server), Datenbanksystem (SQLite, OracleDB) oder Anwendungsserver (JBoss- WildFly, Glassfish).

Ebenfalls nicht getestet wurde das Verhalten des Skriptes auf Umgebungen mit eingeschränkten Benutzerrechten, wie zum Beispiel bei Shared-Hosting-Systemen.

Anhang A Pylint Report

```
************* Module hardening.io.ConsoleWriter
R:181, 4: Too many arguments (7/5) (too-many-arguments)
************* Module hardening.storage.FileAndDirectoryTransaction
R: 47, 4: Too many arguments (6/5) (too-many-arguments)
R:248, 4: Too many arguments (6/5) (too-many-arguments)
************* Module hardening.utils.HardeningUtil
R: 29, 4: Too many branches (15/12) (too-many-branches)
************* Module hardening.utils.configfile.ApacheConfigEntry
R:119, 4: Too many arguments (6/5) (too-many-arguments)
************* Module hardening.utils.configfile.ConfigLine
R: 28, 4: Too many arguments (8/5) (too-many-arguments)
************* Module hardening.utils.os.UserItem
R:  1, 0: Cyclic import (hardening -> hardening.core.RuntimeOptions) (cyclic-
import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.NetworkInterfacesTransaction) (cyclic-import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.DirectoryTransaction) (cyclic-import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.MysqlTransaction) (cyclic-import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.ServiceTransaction) (cyclic-import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.CommandTransaction) (cyclic-import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.PasswdTransaction) (cyclic-import)
R:  1, 0: Cyclic import (hardening.storage -> hardening.storage.XmlTransaction)
(cyclic-import)
R:  1, 0: Cyclic import (hardening.storage -> hardening.storage.NoTransaction)
(cyclic-import)
R:  1, 0: Cyclic import (hardening.storage ->
hardening.storage.NetworkInterfacesTransaction ->
hardening.storage.TextFileTransaction) (cyclic-import)
R:  1, 0: Similar lines in 2 files
==hardening.utils.HardeningUtil:200
==hardening.utils.module.EnterModule:53
        if meta is None:
            return None

        if isinstance(meta, dict):
            if opt in meta.keys():
                return meta[opt]
            else:
                return None

        assert isinstance(meta, list)

        # complete match (e.g. en_US)
        for localized_meta in [m for m in meta if isinstance(m, dict)]:
            language = core.RuntimeOptions().get_locale()
            if language == localized_meta[constants.CONFIG_LANGUAGE] and \
                    opt in localized_meta.keys():
                return localized_meta[opt]

        # language match (e.g. en)
        for localized_meta in [m for m in meta if isinstance(m, dict)]:
```



```
            language = core.RuntimeOptions().get_locale().split("_")[0]
            if language == localized_meta[constants.CONFIG_LANGUAGE] and \
                    opt in localized_meta.keys():
                return localized_meta[opt]

        return None
 (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.utils.configfile.ApacheConfigEntry:17
==hardening.utils.configfile.IniFileEntry:17
from hardening.utils.configfile.ConfigFileEntry import ConfigFileEntry


# pylint: disable=duplicate-code
@utils.ModuleSettings(
    options=[
        utils.UtilOption(constants.OPTION_TRANSACTION, required=True,
                         docstring="path of the apache config file"),
        utils.UtilOption(constants.OPTION_KEY, required=True,
                         docstring="name of the configuration setting"),
        utils.UtilOption(constants.OPTION_VALUE, required=True,
                         docstring="value to be set for the configuration
setting"),
        utils.UtilOption(constants.OPTION_SECTION, required=False,
                         docstring="section inside the config file where "
                                   "setting must be placed in"),
        utils.UtilOption(constants.OPTION_LISTSEPARATOR, required=False,
                         docstring="if the value is a list then this options
specifies "
                                   "how the distinct values must be separated"),
        utils.UtilOption(constants.OPTION_SEPARATOR, required=False, (duplicate-
code)
R:  1, 0: Similar lines in 2 files
==hardening.utils.configfile.ApacheConfigEntry:162
==hardening.utils.configfile.IniFileEntry:117
    def __section_start_index(self):
        return self.__find_section()[0]

    def __section_end_index(self):
        return self.__find_section()[1]

    def get_minimum_index(self):
        if self.__section is None:
            return 0
        else:
            return self.__section_start_index()

    def get_maximum_index(self):
        if self.__section is None:
            return self.transaction().lines_count() - 1
        else:
            return self.__section_end_index()
 (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.misc.UpdateCopyright:43
==hardening.misc.UpdateCopyrightPHP:44
            with open(_file, 'r') as code_file:
                newtext = transform(code_file.read())
            if newtext:
```

```
                # pylint: disable=superfluous-parens
                print("updating %s" % _file)
                with open(_file, 'w') as code_file:
                    code_file.write(newtext)
            else:
                # pylint: disable=superfluous-parens
                print("omitting %s" % _file)


if __name__ == '__main__':
    update_copyright() (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.misc.UpdateCopyright:28
==hardening.misc.UpdateCopyrightPHP:29
        parts = text.split(os.linesep)
        shebang = parts[0] + "\n"
        text = os.linesep.join(parts[1:])
    else:
        shebang = ""

    copyright_info = COPYRIGHT_TEXT.format(program_name=PROGRAM_NAME).strip()

    return shebang + copyright_info + os.linesep + os.linesep + text


def update_copyright():
    for root, _, files in os.walk(os.getcwd()):
        for _file in [os.path.join(root, x) (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.utils.configfile.ApacheConfigEntry:72
==hardening.utils.configfile.IniFileEntry:70
            if self.__section['name'] is None or len(
                    self.__section['name'].strip()) == 0:
                self.__section = None

    def __find_section(self):
        section_stack = list()

        section_start = None
        section_end = None

        idx = 0
        while idx < self.transaction().lines_count():
 (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.utils.configfile.ApacheConfigEntry:143
==hardening.utils.configfile.IniFileEntry:103
    def current_sctn_is_target_sctn(self, section_stack):
        if self.__section is None and len(section_stack) > 0:
            return False
        if self.__section is not None and len(section_stack) == 0:
            return False
        if self.__section is not None and self.__section != section_stack[-1]:
            return False
        return True

    def __create_section_begin(self):
        if self.__section is None:
            return None (duplicate-code)
```



```
R:  1, 0: Similar lines in 2 files
==hardening.utils.configfile.ApacheConfigEntry:38
==hardening.utils.configfile.IniFileEntry:38
        utils.UtilOption(constants.OPTION_COMMENTCHAR, required=False,
                         docstring="all characters in a line after the comment "
                                   "character are ignored"),
        utils.UtilOption(constants.OPTION_MULTIPLE, required=False,
                         docstring="if this is set to true then existing
settings with the "
                                   "same key and different value are not
overwritten, "
                                   "but a new line is being inserted"),
        utils.UtilOption(constants.OPTION_BEFOREKEY, required=False,
                         docstring="describes a key in the config file BEFORE
which the "
                                   "current setting will be inserted.")
    ],
    required_transaction=storage.TextFileTransaction) (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.storage.MysqlTransaction:29
==hardening.storage.NoTransaction:38
    def __commit__(self):
        pass

    def __prepare_commit__(self):
        pass

    def __rollback__(self):
        pass
 (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.info.lib.JavaConfig:52
==hardening.info.lib.PHPConfig:45
        except OSError:
            return

        if process.returncode != 0:
            core.LogManager().get_logger().fatal(stderr.strip())
            raise subprocess.CalledProcessError(
                process.returncode, cmd, output=stderr)
 (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.utils.DisabledSection:29
==hardening.utils.DisabledUtil:30
        try:
            cause = " " + \
                str(self.get_runtimeinfo(constants.OPTION_CAUSE,
interpolate=False))
        # pylint: disable=bare-except
        except:
            pass
        core.LogManager().get_logger().info( (duplicate-code)
R:  1, 0: Similar lines in 2 files
==hardening.utils.filesystem.ChangeOwner:61
==hardening.utils.filesystem.ChangePermissions:48
    def __run__(self):
        if os.path.isfile(self.transaction().url()):
            apply_to_files = True
        else:
```

```
            apply_to_files = self.get_option(constants.OPTION_APPLY_TO_FILES)
(duplicate-code)
R:  1, 0: Similar lines in 3 files
==hardening.utils.configfile.ApacheConfigEntry:45
==hardening.utils.configfile.ConfigFileEntry:44
==hardening.utils.configfile.IniFileEntry:45
        utils.UtilOption(constants.OPTION_BEFOREKEY, required=False,
                         docstring="describes a key in the config file BEFORE
which the "
                                   "current setting will be inserted.")
    ],
    required_transaction=storage.TextFileTransaction) (duplicate-code)


Report
======
3510 statements analysed.

Statistics by type
------------------

+---------+-------+-----------+-----------+------------+---------+
|type     |number |old number |difference |%documented |%badname |
+=========+=======+===========+===========+============+=========+
|module   |85     |85         |=          |9.41        |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|class    |77     |77         |=          |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|method   |499    |499        |=          |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+
|function |17     |17         |=          |100.00      |0.00     |
+---------+-------+-----------+-----------+------------+---------+


External dependencies
---------------------
::

    coloredlogs (hardening.core.LogManager)
    distro (hardening.info.lib.OSInfo)
    hardening (hardening.core.RuntimeOptions)
      \-constants
(hardening.utils.configfile.ApacheConfigEntry,hardening.utils.DisabledSection,ha
rdening.info,hardening.core.LogManager,hardening.utils.net.StaticIP,hardening.ut
ils.module.EnterModule,hardening.utils.module.LeaveModule,hardening.utils.config
file.XmlFileEntry,hardening.utils.net.Firewall,hardening.utils.DisabledUtil,hard
ening.utils.configfile.IniFileEntry,hardening.utils.os.CheckVersion,hardening.ut
ils.apache.ApacheDismod,hardening.utils.filesystem.Directory,hardening.utils.fil
esystem.ChangePermissions,hardening.utils.filesystem.ChangeOwner,hardening.utils
.HardeningUtil,hardening.utils,hardening.utils.os.UserItem,hardening.utils.confi
gfile.ConfigFileEntry,hardening.utils.apache.ApacheEnmod,hardening.utils.os.Conf
igureService)
      \-core
(hardening.info.lib.ApacheConfig,hardening.storage,hardening.utils.configfile.Ap
acheConfigEntry,hardening.storage.DirectoryTransaction,hardening.utils.DisabledS
ection,hardening.storage.ServiceTransaction,hardening.info,hardening.scheduler,h
ardening.storage.TransactionInfo,hardening.utils.module.EnterModule,hardening.in
```



```
fo.PhpInfo,hardening.storage.Transaction,hardening.info.PloneInfo,hardening.io.C
onsoleWriter,hardening.utils.DisabledUtil,hardening.utils.configfile.IniFileEntr
y,hardening.info.LiferayInfo,hardening.info.lib.PackageManager,hardening.storage
.PasswdTransaction,hardening.storage.CommandTransaction,hardening.info.lib.Netwo
rkConfiguration,hardening.info.lib.OSInfo,hardening.utils.CommandUtil,hardening.
storage.FileTransaction,hardening.utils.configfile.ConfigFileEntry,hardening.sto
rage.FileAndDirectoryTransaction,hardening.utils.HardeningUtil,hardening.info.li
b.PHPConfig,hardening.info.Condition,hardening.storage.NetworkInterfacesTransact
ion,hardening.utils.configfile.ConfigLine,hardening.storage.XmlTransaction,harde
ning.utils.filesystem.ChangeOwner,hardening.info.lib.JavaConfig,hardening.storag
e.NoTransaction,hardening.storage.TextFileTransaction)
      | \-ChangeLog (hardening.core)
      | \-ExecutionState (hardening.core)
      | \-HardeningFailure (hardening.core)
      | \-LogManager (hardening.core)
      | \-RuntimeOptions
(hardening,hardening.storage.TransactionInfo,hardening.core.Documentation,harden
ing.core.ChangeLog,hardening.core,hardening.core.LogManager)
      | \-Singleton
(hardening.core.ExecutionState,hardening.core.RuntimeOptions,hardening.core.Docu
mentation,hardening.core.ChangeLog,hardening.core,hardening.core.LogManager)
      | \-caller_name (hardening.core.LogManager)
      \-info
(hardening.info.ApacheInfo,hardening.info.EnvironmentInfo,hardening.storage.Tran
sactionInfo,hardening.info.ResolveKey,hardening.info.Property,hardening.info.Use
rInfo,hardening.info.NetInfo,hardening.storage.DirectoryTransaction,hardening.in
fo.PythonInfo,hardening.info.PhpInfo,hardening.utils.configfile.ConfigFileEntry,
hardening.utils.HardeningUtil,hardening.info.PloneInfo,hardening.info.Condition,
hardening.scheduler,hardening.info.JavaInfo,hardening.info.LiferayInfo)
      | \-lib
(hardening.utils.apache.ApacheDismod,hardening.info.Condition,hardening.utils.fi
lesystem.ChangeOwner,hardening.storage.PasswdTransaction,hardening.utils.os.Chec
kVersion)
      |   \-ApacheConfig (hardening.info.ApacheInfo,hardening.info.lib)
      |   \-Group (hardening.info.lib)
      |   \-JavaConfig (hardening.info.lib)
      |   \-NetworkConfiguration
(hardening.utils.net.Firewall,hardening.info.NetInfo,hardening.info.lib)
      |   \-OSInfo (hardening.info.lib,hardening.info.ResolveKey)
      |   \-PHPConfig (hardening.info.lib)
      |   \-PackageManager
(hardening.info.lib,hardening.info.PythonInfo,hardening.info.PhpInfo,hardening.i
nfo.lib.JavaConfig,hardening.utils.HardeningUtil,hardening.info.lib.PHPConfig,ha
rdening.info.lib.PythonConfig,hardening.info.JavaInfo)
      |   \-Passwd (hardening.info.lib)
      |   \-PythonConfig (hardening.info.lib)
      \-io
(hardening.utils.module.EnterModule,hardening.utils.HardeningUtil,hardening.info
.lib.NetworkConfiguration,hardening.utils.CommandUtil,hardening.storage.XmlTrans
action,hardening.storage.TextFileTransaction,hardening.info.PhpInfo,hardening.ut
ils.module.LeaveModule,hardening.storage.Transaction,hardening.utils.os.CheckVer
sion,hardening.info.PloneInfo,hardening.info,hardening.io.ConsoleWriter,hardenin
g.info.LiferayInfo)
      \-storage
(hardening.utils.configfile.ApacheConfigEntry,hardening.storage.DirectoryTransac
tion,hardening.storage.MysqlTransaction,hardening.storage.ServiceTransaction,har
dening.scheduler,hardening.utils.filesystem.DeleteFile,hardening.utils.net.Stati
cIP,hardening.utils.configfile.XmlFileEntry,hardening.utils.net.Firewall,hardeni
ng.utils.configfile.IniFileEntry,hardening.storage.PasswdTransaction,hardening.s
```

```
torage.CommandTransaction,hardening.storage.XmlTransaction,hardening.utils.files
ystem.ChangePermissions,hardening.utils.filesystem.ChangeOwner,hardening.utils.H
ardeningUtil,hardening.storage.NetworkInterfacesTransaction,hardening.utils.os.U
serItem,hardening.utils.configfile.ConfigFileEntry,hardening.storage.NoTransacti
on,hardening.utils.os.ConfigureService,hardening.storage.TextFileTransaction)
      | \-CommandTransaction
(hardening.storage.ServiceTransaction,hardening.storage,hardening.storage.Passwd
Transaction)
      | \-DirectoryTransaction (hardening.storage)
      | \-FileAndDirectoryTransaction
(hardening.storage,hardening.storage.FileTransaction,hardening.storage.Directory
Transaction)
      | \-FileTransaction
(hardening.storage,hardening.storage.TextFileTransaction,hardening.storage.XmlTr
ansaction)
      | \-MysqlTransaction (hardening.storage)
      | \-NetworkInterfacesTransaction (hardening.storage)
      | \-NoTransaction (hardening.storage)
      | \-PasswdTransaction (hardening.storage)
      | \-ServiceTransaction (hardening.storage)
      | \-TextFileTransaction
(hardening.storage,hardening.storage.NetworkInterfacesTransaction)
      | \-Transaction
(hardening.storage,hardening.storage.MysqlTransaction,hardening.storage.FileAndD
irectoryTransaction,hardening.storage.CommandTransaction,hardening.storage.NoTra
nsaction)
      | \-TransactionInfo (hardening.storage,hardening.storage.FileTransaction)
      | \-XmlTransaction (hardening.storage)
      \-utils
(hardening.utils.filesystem.DeleteFile,hardening.utils.misc.DisplayMessage,harde
ning.utils.apache.ApacheDismod,hardening.utils.filesystem.Directory,hardening.ut
ils.os.ConfigureService,hardening.utils.filesystem.ChangeOwner,hardening.utils.c
onfigfile.XmlFileEntry,hardening.utils.configfile.ApacheConfigEntry,hardening.ut
ils.module.EnterModule,hardening.utils.filesystem.ChangePermissions,hardening.ut
ils.net.StaticIP,hardening.utils.module.LeaveModule,hardening.utils.configfile.C
onfigFileEntry,hardening.utils.DisabledSection,hardening.utils.apache.ApacheEnmo
d,hardening.utils.net.Firewall,hardening.utils.DisabledUtil,hardening.utils.conf
igfile.IniFileEntry,hardening.utils.os.UserItem,hardening.utils.os.CheckVersion)
        \-CommandUtil
(hardening.utils.apache.ApacheEnmod,hardening.utils.apache.ApacheDismod)
        \-DisabledSection (hardening.scheduler)
        \-DisabledUtil (hardening.scheduler)
        \-HardeningUtil (hardening.utils,hardening.utils.CommandUtil)
        \-configfile
        | \-ApacheConfigEntry (hardening.utils.configfile)
        | \-ConfigFileEntry
(hardening.utils.configfile.IniFileEntry,hardening.utils.configfile,hardening.ut
ils.configfile.ApacheConfigEntry)
        | \-ConfigLine
(hardening.utils.configfile.ConfigFileEntry,hardening.utils.configfile)
        | \-IniFileEntry (hardening.utils.configfile)
        | \-XmlFileEntry (hardening.utils.configfile)
        \-module
          \-EnterModule (hardening.scheduler)
          \-LeaveModule (hardening.scheduler)
    lxml
      \-etree (hardening.storage.XmlTransaction)
      \-objectify (hardening.storage.XmlTransaction)
    netifaces (hardening.info.lib.NetworkConfiguration)
```



```
    network_interfaces (hardening.storage.NetworkInterfacesTransaction)
    psutil (hardening.storage.PasswdTransaction)
    six
(hardening.info.lib.PackageManager,hardening.io,hardening.storage.FileAndDirecto
ryTransaction,hardening.utils.configfile.ConfigFileEntry,hardening.storage.Trans
action,hardening.utils.HardeningUtil,hardening.io.ConsoleWriter,hardening.schedu
ler)
    yaml (hardening.info,hardening.core.LogManager)
```

Raw metrics

type	number	%	previous	difference
code	4587	56.88	4587	=
docstring	914	11.33	914	=
comment	1240	15.38	1240	=
empty	1323	16.41	1323	=

Duplication

	now	previous	difference
nb duplicated lines	168	168	=
percent duplicated lines	2.106	2.106	=

Messages by category

type	number	previous	difference
convention	0	0	=
refactor	29	29	=
warning	0	0	=
error	0	0	=

```
% errors / warnings by module
-----------------------------

+-----------------------------------------------------+------+--------+---------
+-----------+
|module                                               |error |warning |refactor |
convention |
+=====================================================+======+========+==========+======
=====+
|hardening.utils.os.UserItem                          |0.00  |0.00    |79.31    |0.00
|
+-----------------------------------------------------+------+--------+---------
+-----------+
|hardening.storage.FileAndDirectoryTransaction        |0.00  |0.00    |6.90     |0.00
|
+-----------------------------------------------------+------+--------+---------
+-----------+
|hardening.utils.configfile.ConfigLine                |0.00  |0.00    |3.45     |0.00
|
+-----------------------------------------------------+------+--------+---------
+-----------+
|hardening.utils.configfile.ApacheConfigEntry         |0.00  |0.00    |3.45     |0.00
|
+-----------------------------------------------------+------+--------+---------
+-----------+
|hardening.utils.HardeningUtil                        |0.00  |0.00    |3.45     |0.00
|
+-----------------------------------------------------+------+--------+---------
+-----------+
|hardening.io.ConsoleWriter                           |0.00  |0.00    |3.45     |0.00
|
+-----------------------------------------------------+------+--------+---------
+-----------+


Messages
--------

+-------------------+------------+
|message id         |occurrences |
+===================+============+
|duplicate-code     |13          |
+-------------------+------------+
|cyclic-import      |10          |
+-------------------+------------+
|too-many-arguments |5           |
+-------------------+------------+
|too-many-branches  |1           |
+-------------------+------------+


Global evaluation
-----------------
Your code has been rated at 9.92/10 (previous run: 9.92/10, +0.00)
```


Literaturverzeichnis

[1] Bundesamt für Sicherheit in der Informationstechnik, Studie „Sicherheitsuntersuchung von Content-Management-Systemen“, Stand 27.05.2016.

[2] JetBrains, PhpStorm IDE, https://www.jetbrains.com/phpstorm/, Stand 11.05.2016.

[3] Python: PEP 8 -- Style Guide for Python Code, https://www.python.org/dev/peps/pep-0008/, Stand 11.05.2016.

[4] Pylint, https://www.pylint.org/, Stand 11.05.2016.

[5] Sebastian Bergmann, PHPUnit, https://phpunit.de/, Stand 11.05.2016.

[6] Python, unittest — Unit testing framework, https://docs.python.org/2/library/unittest.html, Stand 11.05.2016.