



Bundesamt
für Sicherheit in der
Informationstechnik

Studie „Sicherheitsuntersuchung von Content-Management- Systemen“

Checklisten zur Härtung - Liferay



Änderungshistorie

Version	Datum	Name	Beschreibung
0.1	11.04.2016		Einarbeitung Checklisten
0.2	26.04.2016		Aktualisierung der Status für die Skriptumsetzung
0.3	26.04.2016		Spezifizierung auf eine Liferay-Checkliste
0.4	29.04.2016		Review
1.0	29.04.2016		Freigabe
1.1	19.05.2016		Erweiterung um Hinweise für Kompatibilität
2.0	20.05.2016		Freigabe
3.0	22.06.2016		Finalisierung anhand Skriptlösung
4.0	22.07.2016		Fehlerkorrekturen
5.0	12.08.2016		Finalisierung anhand Skriptlösung

Inhaltsverzeichnis

	Änderungshistorie.....	2
1	Einleitung/Motivation.....	5
2	Content Management Systeme.....	6
2.1	Liferay.....	6
3	Laufzeitumgebungen.....	8
3.1	Java.....	8
4	Web- und Application Server.....	9
4.1	Apache Webserver.....	9
4.2	Tomcat Application Server.....	12
5	Datenbanken.....	17
5.1	MySQL.....	17
6	Betriebssysteme.....	21
6.1	Linux.....	21
6.2	SSH.....	23
6.3	Benutzer und Rechte.....	25
6.4	Firewall.....	27
6.4.1	Beispiel für eine iptables-Konfiguration.....	28
	Literaturverzeichnis.....	29
	Stichwort- und Abkürzungsverzeichnis.....	30
	Erläuterungen zu Kommando-Syntax:.....	31

1 Einleitung/Motivation

Ziel der vorliegenden Checkliste ist es, die Umsetzung der Sicherheitsanforderungen für das CMS Liferay zu ermöglichen. Diese wurden im Rahmen der Studie „Sicherheitsuntersuchung von Content-Management-Systemen“ [1] erarbeitet.

Die zugehörigen Härtungsskripte sollen die automatische Implementierung der Sicherheitsanforderungen sicherstellen. Allerdings können einige Anforderungen nur manuell umgesetzt werden. Aus diesem Grund ist bei den nachfolgenden Sicherheitsanforderungen vermerkt, welche der Anforderungen durch die Härtungsskripte automatisiert umgesetzt werden können. Eine manuelle Umsetzung ist prinzipiell möglich.

Die Einteilung der zu implementierenden Maßnahmen erfolgt anhand der Komponenten des Gesamtsystems. Die Härtungsmaßnahmen ergeben sich aus internen Sicherheitsempfehlungen der Deutschen Telekom AG [2] bzw. den daraus resultierenden technischen Sicherheitsanforderungen [3], den Dokumenten der BSI-Reihe zur Internet-Sicherheit (ISi-Reihe) [4], den Sicherheitsempfehlungen der CM-Systeme und der Laufzeitumgebungen sowie der Ergebnisse aus dem Test der bereits gehärteten CM-Systeme entsprechend der Studie [1].

2 Content Management Systeme

2.1 Liferay

Maßnahme	Im Härtungs- skript umgesetzt	Kommandos zur Härtung
Durch sogenannte Log Forgery Schwachstellen können Angreifer über die Zeichenfolgen "%0a%0d" bzw. "\r\n" neue Zeilenumbrüche in Logausgaben provozieren, um bspw. Angriffe zu verschleiern oder irreführende Log-Einträge in die Anwendung einzuschleusen. Der Logger von Liferay muss mittels „Sanitizer“ (com.liferay.portal.kernel.sanitizer.Sanitizer) entsprechend konfiguriert werden, um die genannten Zeichen zu maskieren.	x	Um die Encodierung der NewLine Zeichen in Logausgaben zu forcieren, muss das Attribut "log.sanitizer.enabled = true" in der system-ext.properties von Liferay gesetzt sein.
Liferay verfügt über einen Portal's Plugin Security Manager, welcher alle API-Aufrufe eines eingebundenen Plugins und deren Zugriffsberechtigung prüft. Die Konfiguration des Plugin Security Managers geschieht über die liferay-plugin-package.properties Datei, die das Plugin unter /webapps/<portlet name>/WEB-INF/liferay-plugin-package.properties zur Verfügung stellt. Versucht das Plugin, auf eine Portlet-Ressource außerhalb der spezifizierten Portal Access Control List (PACL) zuzugreifen, wird der Zugriff über den Plugin Security Manager unterbunden. Jedes Plugin sollte, wenn die liferay-plugin-package.properties Datei vorhanden ist, den Plugin Security Manager aktivieren.		Sollte das eingebundene Plugin über eine liferay-plugin-package.properties Datei unter /var/lib/tomcat7/webapps/<portlet name>/WEB-INF/ verfügen, muss der Security Manager über das Attribut "security-manager-enabled=true" in der liferay-plugin-package.properties aktiviert werden.
Bei der Erstellung des administrativen Nutzers während der Installation sollte kein leicht zu erratener Benutzername gewählt werden.		Bei der Installation wird ein administrativer Benutzer angelegt, wobei Begriffe wie 'admin', 'root' oder 'webmaster' für das Administrator-Konto vermieden werden sollten.
Um die Angriffsfläche für potentielle Angreifer zu minimieren, sollte die Registrierung an der Liferay-Instanz deaktiviert werden.	x	Unter dem Menüpunkt -> Control Panel -> Portal Settings -> Authentication -> General muss der folgende Eintrag gesetzt werden: company.security.strangers=false
Um die Angriffsfläche für potentielle Angreifer zu minimieren, sollten nicht verwendete Funktionen deaktiviert werden,	x	Das folgende Attribut sollte in der /opt/liferay/liferay-home/WEB-INF/classes/portal-ext.properties

<p>wozu unter anderem die JSON Webservices gehören könnten.</p>		<p>gesetzt werden, um die JSON Webservices global zu deaktivieren: "json.web.service.enabled=false".</p>
<p>Nach dem Least-Privilege-Prinzip sollten Nutzern lediglich so viele Rechte zugewiesen werden, wie sie benötigen. In Liferay existieren Omni-Administratoren, welche Zugriff zu Core Funktionalitäten von Liferay besitzen (wie bspw. das Herunterfahren der Liferay Instanz). Die Konfiguration der Omni-Benutzer sollte auf wenige Personen beschränkt werden.</p>		<p>Das folgende Attribut sollte in der /opt/liferay/liferay-home/WEB-INF/classes/portal-ext.properties gesetzt werden, um ausgewählte Administratoren als Omni-Benutzer zu setzen: "omniadmin.users". Wird das Attribut leer gelassen, besitzen alle Administratoren der Liferay-Instanz den Omni-Status.</p>
<p>Die Anwendung sollte so konfiguriert werden, dass der Location-Header des HTTP-Response keinen willkürlichen Wert annehmen kann.</p>		<p>Das folgende Attribut sollte in der /opt/liferay/liferay-home/WEB-INF/classes/portal-ext.properties gesetzt werden, um nur ausgewählte URLs bzw. IPs im Location Header des HTTP-Response zuzulassen: "redirect.url.domains.allowed" und "redirect.url.ips.allowed". Werden die Felder leer gelassen, ist ein willkürlicher Wert zulässig.</p>
<p>Die Anwendung sollte den Schutz vor Phishing-Attacken und somit eine Cookie-Unterstützung vor der Anmeldung unterstützen.</p>		<p>Das folgende Attribut sollte in der /opt/liferay/liferay-home/WEB-INF/classes/portal-ext.properties gesetzt werden, um Phishing Attacken durch Cookie-Validierung zu unterbinden: "session.enable.phishing.protection=true".</p>
<p>Die Anwendung sollte keine Passwörter als Klartext über längere Zeit im Heap Speicher halten.</p>		<p>Das folgende Attribut sollte in der /opt/liferay/liferay-home/WEB-INF/classes/portal-ext.properties gesetzt werden, um Passwörter im Heap Speicher zu verschleiern: "session.store.password=false".</p>
<p>Eine Anmeldung an der Liferay-Instanz sollte immer über die Kombination Benutzername/Passwort erfolgen.</p>		<p>Die folgenden Attribute sollten in der /opt/liferay/liferay-home/WEB-INF/classes/portal-ext.properties gesetzt werden, um die Eingabe eines Passworts auch für LDAP-Nutzer zu erzwingen: "ldap.import.user.password.enabled=true" und "ldap.auth.required=true".</p>
<p>Nach dem Least-Privilege-Prinzip sollten die Standard-Benutzerrechte so angepasst werden, dass Benutzer nur jene Aufgaben durchführen können, die ihnen anvertraut sind. Jedes zusätzliche Recht ermöglicht Missbrauch.</p>		<ul style="list-style-type: none"> • Prüfung aller Rollen auf sinnvolle Rechte. • Prüfung aller Benutzer auf sinnvolle Rollen.

3 Laufzeitumgebungen

3.1 Java

Maßnahme	Im Härtings- skript umgesetzt	Kommandos zur Härtung
Die eingesetzte Version muss auf Aktualität geprüft werden.	x	user@host:~\$ java -version

4 Web- und Application Server

4.1 Apache Webserver

Da die Dateien der Apache-Konfiguration stark von der Linux-Distribution und der Apache-Version abhängen, können sich die Pfade und Dateinamen unterscheiden, z.B. wird alternativ zu "apache2" oft auch "httpd" verwendet.

Maßnahme	Im Härtungs-skript umgesetzt	Kommandos zur Härtung
Nicht benötigte Apache-Module müssen deaktiviert werden. Aktive Module erhöhen die Komplexität der Installation und damit auch die Angriffsfläche. Die meisten gefunden Schwachstellen im Produkt Apache HTTPd finden sich in Modulen. Schwachstellen in deaktivierten Module sind somit nicht direkt ausnutzbar.	x	<pre> root@host:~\$ editor /etc/apache2/apache2.conf # Prüfung der LoadModule Direktiven # und/oder root@host:~\$ ls /etc/apache2/mods_enabled # Deaktivierung unnötiger Module: root@host:~\$ a2dismod <modname> # z.B.: sudo a2dismod autoindex # Insbesondere folgende Module sollten deaktiviert werden: # actions, autoindex, access_compat, suexec, include </pre>
Falls mod_suexec nicht verwendet wird, sollten die suexec helper binary entfernt werden.	x	Entfernen/Deinstallieren des apache2-suexec/ apache2-suexec-custom Paketes
Nicht benötigte HTTP-Methoden müssen deaktiviert werden. In den meisten Installationen genügen die Methoden GET und POST. Bei einzelnen Einsatzszenarien können darüber hinaus auch weitere Methoden notwendig sein.	x	<pre> root@host:~\$ editor /etc/apache2/apache2.conf # Folgende Zeile muss in allen Location, File und Directory-Bereichen vorkommen: AllowMethods GET POST # Zusätzlich muss das Modul allowmethods aktiviert sein sudo a2enmod allowmethods </pre>
Informationen über den Webserver in HTTP-Headern müssen auf ein Mindestmaß beschränkt werden.	x	<pre> root@host:~\$ editor /etc/apache2/conf- enabled/security.conf ServerTokens Prod </pre>
Informationen über den Webserver in Fehlerseiten, die durch den Webserver ausgeliefert werden, müssen entfernt werden.	x	<pre> root@host:~\$ editor /etc/apache2/conf- enabled/security.conf ServerSignature Off </pre>
Standardmäßig vorhandene Inhalte müssen entfernt werden.		<pre> # Entfernen von default content Inhalten in folgenden Verzeichnissen /opt/apache2/ /var/www/html /usr/lib/cgi-bin/test-cgi root@host:~\$ sudo rm <file> root@host:~\$ sudo rmdir <directory> </pre>

Die Erstellung von Verzeichnislisten (Indizierung) muss deaktiviert werden.		<pre>root@host:~\$ editor /etc/apache2/apache2.conf # Es darf in der Options-Direktive nicht "Indexes" vorkommen <Directory /> ... Options None ... </Directory></pre>
Alle Webserverprozesse dürfen nicht mit Systemprivilegien laufen.		<pre>root@host:~\$ editor /etc/apache2/apache2.conf User \${APACHE_RUN_USER} Group \${APACHE_RUN_GROUP} # Die Platzhalter sind unter /etc/apache2/env definiert. APACHE_RUN_USER darf hierbei nicht 'root' entsprechen, sondern muss ein dedizierter Nutzer sein.</pre>
Der Webserver darf nur Dateien ausliefern, die für die Auslieferung bestimmt sind. Es kann durch die Directory-, File*- und Location*-Direktiven bestimmt werden, welche Dateien/Pfade aufgerufen werden dürfen und welche verboten sind. Es sollte explizit definiert werden, welche Dateiendungen erlaubt sind.	x	<pre>root@host:~\$ editor /etc/apache2/apache2.conf <Directory /> AllowMethods GET POST Require all denied Options None AllowOverride None </Directory> <Directory /var/www/> AllowMethods GET POST Require all granted Options SymLinksIfOwnerMatch AllowOverride None </Directory> <FilesMatch "[^].*\.(?!css html js pdf txt xml xsl gif ico jpe?g png mp4 mp3 mkv svg))[0-9a- z]{2,4}\$"> Require all denied </FilesMatch></pre>
Für die Verschlüsselung mit HTTPS muss das TLS-Protokoll verwendet werden.	x	<pre># In der Konfigurationsdatei für SSL muss folgende Direktive eingestellt werden: SSLProtocol All -SSLv2 -SSLv3 # Bei Bedarf auch -TLSv1, so dass nur TLSv1.1 und TLSv1.2 unterstützt werden.</pre>
Es muss eine OpenSSL-Version >= 1.0.1 eingesetzt werden, damit das Protokoll TLSv1.2 unterstützt wird. Regelmäßige Updates von OpenSSL erlauben die Verwendung von neueren und sichereren TLS-Protokollen.		<pre># Prüfung der installierten OpenSSL-Version: root@host:~\$ openssl version</pre>
Die TLS-Konfiguration darf keine unsicheren Cipher-Suites verwenden.	x	<pre># In der Konfigurationsdatei für SSL muss folgende Direktive eingestellt werden: SSLCipherSuite 'ECDHE-ECDSA-AES256- GCM-SHA384:ECDHE-RSA-AES256-GCM-</pre>

		SHA384:ECDHE-RSA-AES256-SHA384:DHE-RSA-AES256-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:DHE-RSA-AES128-SHA256' # Nähere und immer aktuelle Informationen können unter https://wiki.mozilla.org/Security/Server_Side_TLS eingesehen werden.
Die TLS-Konfiguration muss vorsehen, dass beim Verbindungsaufbau die als am sichersten einzustufenden Cipher-Suites mit oberster Priorität ausgewählt werden.	x	# In der Konfigurationsdatei für SSL muss folgende Direktive eingestellt werden: SSLHonorCipherOrder On
SSL-Komprimierung muss deaktiviert werden.	x	# In der Konfigurationsdatei für SSL muss folgende Direktive eingestellt werden: SSLCompression off
Der Zugriff auf den Webserver und Server-Fehler müssen protokolliert werden.	x	root@host:~\$ editor /etc/apache2/apache2.conf LogLevel notice ErrorLog \${APACHE_LOG_DIR}/error.log LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined LogFormat "%h %l %u %t \"%r\" %>s %O" common LogFormat "%{Referer}i -> %U" referer LogFormat "%{User-agent}i" agent
Das unautorisierte Überschreiben der Webserver-Konfiguration muss verhindert werden.	x	root@host:~\$ chown -R root:root /etc/apache2/ root@host:~\$ chmod -R 644 /etc/apache2/
Verschiedene Instanzen des Apache-Webserver auf einem System müssen jeweils als dedizierter Nutzer ausgeführt werden.		root@host:~\$ editor /etc/apache2/apache2.conf User \${APACHE_RUN_USER} Group \${APACHE_RUN_GROUP}
Die Webanwendung muss gegen Clickjacking per X-FRAME-OPTIONS-Header abgesichert werden. Siehe https://de.wikipedia.org/wiki/Clickjacking .	x	root@host:~\$ editor /etc/apache2/conf-enabled/security.conf Header set X-Frame-Options: "sameorigin" # Hierfür muss das Modul mod_headers aktiviert sein: root@host:~\$ a2enmod headers
Die Funktion "TraceEnabled" muss deaktiviert werden, so dass die HTTP-Methode TRACE nicht unterstützt wird.	x	root@host:~\$ editor /etc/apache2/conf-enabled/security.conf TraceEnable Off
Apache kann über die LimitRequestBody Direktive die max. Uploadgröße von Dateien festlegen und dadurch einen	x	root@host:~\$ editor /etc/apache2/apache2.conf # Die Direktive LimitRequestBody muss für

zusätzlichen Schutz vor DoS-Attacken bieten.		das entsprechende Upload-Verzeichnis gesetzt werden: <Directory "/usr/local/apache2/uploads"> LimitRequestBody 10485760 </Directory>
Wenn nicht unbedingt benötigt, muss die Unterstützung für "Server Side Includes" deaktiviert werden.		root@host:~\$ editor /etc/apache2/apache2.conf # Es darf in der Options-Direktive nicht "Includes" vorkommen <Directory /> Options None # (Nicht '+Includes') </Directory> # Das Modul mod_include muss deaktiviert sein: root@host:~\$ a2dismod include
Der Logs-Ordner sollte nur im Besitz und beschreibbar von root sein, so dass keine unbefugten Änderungen möglich sind.	x	root@host:~\$ chmod 750 /var/log/apache2 root@host:~\$ chmod 640 /var/log/apache2/* root@host:~\$ chown -R root:adm /var/log/apache2
Der Apache vhost muss dediziert für die Domain konfiguriert sein. Andernfalls wäre es denkbar, dass eine Anwendung durch den Request-Header 'Host:' angreifbar wäre.		# Für jeden Domain und jeden Port wird eine Datei unter /etc/apache2/sites-available/ angelegt: <VirtualHost <domain-name>:80> -> CMS-Konfiguration <VirtualHost <domain-name>:443> -> CMS-Konfiguration SSL <VirtualHost *:80> -> zeigt auf leeren document root # Anschließend wird jede dieser Sites aktiviert: root@host:~\$ a2ensite
Der Standard-vhost sollte entfernt werden.	x	root@host:~\$ /etc/apache2/sites-enabled/999-default.conf root@host:~\$ /etc/apache2/sites-available/999-default.conf
Anwendungen liefern manchmal einen HTTP-Response-Header aus, der es sehr leicht macht, die Anwendung samt Version zu identifizieren. Diese Header sollten über eine Apache Direktive gelöscht werden.	x	root@host:~\$ editor /etc/apache2/conf-enabled/security.conf # Löschen eines Headers, hier für Liferay (Header-Name "Liferay-Portal") Header unset Liferay-Portal

Hinweis: Die folgenden Module müssen aufgrund der getroffenen Härtungsmaßnahmen aktiviert werden:

- headers
- rewrite
- allowmethods

4.2 Tomcat Application Server

Maßnahme	Im Härtungs-	Kommandos zur Härtung
----------	--------------	-----------------------

	skript umgesetzt	
Nicht benötigte HTTP-Methoden müssen über „security constraints“ verboten werden.	x	In der Datei "/etc/tomcat7/web.xml": Setzen eines Tags <security-constraint> und Aufführen aller unerwünschten HTTP- Methoden. Siehe: http://www.techstacks.com/howto/disable-http-methods-in-tomcat.html
Informationen über den Webserver in HTTP-Headern müssen auf ein Mindestmaß beschränkt werden.	x	Version-String in den Headern: Suche in /etc/tomcat7/server.xml nach String: <Connector Hinzufügen des Attributes: Server="Apache" Ändern des Attributes: xpoweredBy="false" oder Löschen dieses Attributs
Informationen über den Webserver in Fehlerseiten, die durch den Webserver ausgeliefert werden, müssen entfernt werden.		Fehlerseiten sollten komplett ersetzt werden, so dass im Falle einer Exception oder anderen HTTP-Fehlercodes eine dedizierte Fehlerseite ausgeliefert wird. Editieren der "/etc/tomcat/web.xml", Einfügen eines Tags: <error-page> <exception-type>java.lang.Throwable</exception-type> <location>/error.html</location> </error-page> <error-page> <error-code>*</error-code> <location>/error.html</location> </error-page> Jede Anwendung muss darüber hinaus selbst eigene Error-Pages definieren. Dies ist in der Struktur von Tomcat so angelegt. Durch Einsatz eines Reverse-Proxy-Servers können feinere Filter implementiert werden. Siehe: http://geekflare.com/apache-tomcat-hardening-and-security-guide/ https://www.owasp.org/index.php/Securing_tomcat
Nicht benötigte Standard-Inhalte müssen entfernt werden.		Es werden mehrere Default-Web-Anwendungen mit Tomcat ausgeliefert. Folgende dieser Anwendungen sollten deaktiviert oder gelöscht werden: ROOT (zeigt die Versionsnummer des Tomcat) Documentation (zeigt die Versionsnummer des Tomcat) Examples (könnte Schwachstellen enthalten und bietet Möglichkeiten, Cookie-Informationen auszulesen)

		Manager (erlaubt nach Authentifizierung die Konfiguration des Tomcat inklusive Deployment von eigenen Webanwendungen) Host Manager (nach Authentifizierung können virtuelle Hosts konfiguriert werden)
Die Erstellung von Verzeichnislisten (Indizierung) muss deaktiviert werden.	x	Standardmäßig ist das Listing bei Tomcat 7 aus den Debian Repositories deaktiviert. In der "/etc/tomcat7/web.xml" muss folgender Eintrag stehen: <servlet> [...] <init-param> <param-name>listings</param-name> <param-value>>false</param-value> </init-param> [...] </servlet>
Der Webserver muss der einzige extern zugreifbare Service eines Systems sein, soweit der Webserver nicht ausschließlich für eine Administrations-Schnittstelle genutzt wird.		Wenn ein Reverse Proxy eingesetzt wird, sollte der Tomcat-Port mittels Firewalls so eingeschränkt werden, dass ausschließlich der Reverse Proxy darauf zugreifen kann. Andere Ports am Server sollten nur sehr restriktiv freigegeben werden.
Alle Webserverprozesse dürfen nicht mit Systemprivilegien laufen.		Prüfen des Laufzeitbenutzers: user@host:~\$ ps -ef grep tomcat
Der Zugriff auf den Webserver muss protokolliert werden.		Prüfung ,ob das standardmäßige Logging aktiv ist, indem die Dateien unter \$CATALINA_HOME/logs untersucht werden.
Das automatische Deployment von Anwendungen muss abgeschaltet werden.	x	Suche in der Datei \$CATALINA_HOME/conf/server.xml nach: <Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true"> und Setzen von autoDeploy="false"
Für jede Anwendung muss innerhalb des Application Servers ein eigener Service konfiguriert werden.		Werden mehrere Anwendungen auf dem Tomcat Server betrieben, so muss für jeden Server in der server.xml ein eigener <service>-Block erstellt werden, um die Anwendungen voneinander zu separieren.
Falls der Shutdown-Port nicht benötigt wird, muss er deaktiviert werden.	x	Suche in /etc/tomcat7/server.xml nach <Server port="8005" shutdown="SHUTDOWN"> Server port muss auf "-1" geändert werden, damit der Port deaktiviert wird.
Falls der Shutdown-Port benötigt wird, muss er mit einem starken Passwort abgesichert werden.	x	Suche in /etc/tomcat7/server.xml nach <Server port="8005" shutdown="SHUTDOWN"> "SHUTDOWN" muss durch einen langen zufälligen Wert, welcher als eine Art Passwort

		fungiert, ersetzt werden.
Falls die Manager-Anwendung genutzt wird, muss diese vor unautorisierter Nutzung geschützt werden.		Es muss ein Benutzer mit einem starken Passwort (sha1-Hash, erzeugt mit digest.sh) eingerichtet werden, tomcat-users.xml: <pre><user username="Admin1" password="fbb574b6ce19ae918a63e255f13904d b4976ce30" roles="manager-gui"/></pre> Damit sha1-Hashes verwendet werden können, muss in der \$CATALINA_HOME/conf/server.xml folgender Eintrag vorgenommen werden: <pre><Realm className="org.apache.catalina.realm.UserDat abaseRealm" digest="sha" resourceName="UserDatabase"/></pre> Der passende Konnektor für die Manager-Anwendung darf nur mit HTTPS betrieben werden, so dass die Anmeldedaten nicht unverschlüsselt übertragen werden.
Nicht benötigte Konnektoren sollten deaktiviert werden. Üblicherweise wird entweder der HTTP- oder der AJP-Konnektor benötigt. Der jeweils andere Konnektor sollte aus der server.xml gelöscht oder auskommentiert werden.		/etc/tomcat7/server.xml: <pre><!-- Define an AJP 1.3 Connector on port 8009 --> <Connector address="127.0.0.1" port="8009" protocol="AJP/1.3" redirectPort="8443" /></pre> Der nicht benötigte Konnektor muss gelöscht bzw. auskommentiert werden.
Das Ausliefern von Standard-Fehlerseiten muss unterbunden werden.		Es können alternative Fehlerseiten konfiguriert werden. Für alle webapps \$CATALINA_HOME/webapps/<APP_NAME>/WEB-INF/web.xml und /var/lib/tomcat7/webapps/ROOT/web.xml: <pre><error-page> <error-code>404</error-code> <location>/errors/code.jsp</location> </error-page> <error-page> <error-code>500</error-code> <location>/errors/code.jsp</location> </error-page> [...]</pre>
Die Serverinfo.properties muss verschleiert werden, so dass keine Informationen über den Webserver veröffentlicht werden.		Entpacken: jar xf catalina.jar org/apache/catalina/util/ServerInfo.properties Setzen: server.info=<SomeWebServer> Setzen: server.number=<SomeVersion> Setzen: server.built=<BuildDate> Packen: jar uf catalina.jar org/apache/catalina/util/ServerInfo.properties
Die Größe der Log-Dateien kann schnell enorm werden, so dass das LogLevel mit einem sinnvollen Wert versehen und ein	x	Das LogLevel des FileHandler unter /var/lib/tomcat7/conf/logging.properties sollte in produktiven System nicht auf "FINE"

Limit an Einträgen definiert werden sollte.		oder höher gesetzt werden. Zusätzlich muss ein Limit für die Dateigröße der Logs festgelegt werden: java.util.logging.FileHandler.limit=10000
Alternative Pfad-Trennzeichen müssen deaktiviert werden, so dass Requests, die durch einen vorgeschalteten Reverse Proxy blockiert wurden, nicht so umgeschrieben werden können, dass sie diese Blockierregel umgehen können und weiterhin vom Tomcat Server interpretiert werden.		\$CATALINA_HOME\bin\catalina.sh: - Dorg.apache.catalina.connector.CoyoteAdapter.ALLOW_BACKSLASH=false - Dorg.apache.tomcat.util.buf.UDecoder.ALLOW_ENCODED_SLASH=false
„Facade Recycling“ muss deaktiviert werden, so dass für jeden Request eine neue Facade erzeugt wird. Dies verhindert, dass zwischen Requests versehentlich Informationen anderer Nutzer ausgegeben werden.		\$CATALINA_HOME\bin\catalina.sh: - Dorg.apache.catalina.connector.RECYCLE_FACADES=false

Empfehlung für Quellen:

- Tomcat SSL Konfiguration: <https://tomcat.apache.org/tomcat-7.0-doc/config/http.html>
- Tomcat Härtung: https://www.owasp.org/index.php/Securing_tomcat
- Tomcat Härtung: <http://geekflare.com/apache-tomcat-hardening-and-security-guide/>
- Tomcat HTTP-Methoden: <http://www.techstacks.com/howto/disable-http-methods-in-tomcat.html>

5 Datenbanken

5.1 MySQL

Maßnahme	Im Härtungs- skript umgesetzt	Kommandos zur Härtung
Nicht benötigte (Default-)Datenbanken auf dem Datenbanksystem müssen gelöscht werden.		<pre> user@host:~\$ mysql -u root -p # Identifizieren der vorhandenen Datenbanken: mysql> SHOW DATABASES; # Wechsel in eine dieser Datenbanken: mysql> USE <database_name>; # Auflistung der Tabellen in dieser Datenbank: mysql> SHOW TABLES; # Löschen einer ganzen Datenbank: mysql> DROP DATABASE <database_name>; # Löschen einer einzelnen Tabelle innerhalb der Datenbank: mysql> DROP TABLE <table_name>; </pre>
Nicht benötigte (Default-) Benutzer und (Default-)Rollen müssen gelöscht werden.		<pre> user@host:~\$ mysql -u root -p # Auflistung aller Benutzer mysql> SELECT user, host FROM mysql.user; # Löschen eines Benutzers mysql> DELETE FROM mysql.user WHERE user='<default user name>; </pre>
Erweiterte Privilegien dürfen nur dem Super-User zur Verfügung stehen.		<pre> user@host:~\$ mysql -u root -p mysql> SELECT user, host FROM mysql.user WHERE (Select_priv = 'Y') OR (Insert_priv = 'Y') OR (Update_priv = 'Y') OR (Delete_priv = 'Y') OR (Create_priv = 'Y') OR (Drop_priv = 'Y') OR (File_priv = 'Y') OR (Process_priv = 'Y') OR (Super_priv = 'Y') OR (Shutdown_priv = 'Y') OR (Create_user_priv = 'Y') OR (Grant_priv = 'Y') OR (Repl_slave_priv = 'Y'); # darf nur root oder debian-sys-maint zurückgeben mysql> SELECT user, host FROM mysql.db WHERE db = 'mysql' AND ((Select_priv = 'Y') OR (Insert_priv = 'Y') OR (Update_priv = 'Y') OR (Delete_priv = 'Y') OR (Create_priv = 'Y') OR (Drop_priv = 'Y') OR (Grant_priv = 'Y')); # darf nur root oder nichts zurückgeben # sollten auch weitere Benutzer zurückgegeben werden, so muss geprüft werden warum diese die erweiterten Privilegien (markiert mit 'Y') besitzen und sie sollten entsprechend angepasst werden. </pre>
Die Privilegien SELECT, INSERT, UPDATE, DELETE, CREATE, DROP und ALTER		<pre> user@host:~\$ mysql -u root -p mysql> SELECT user,host,Db FROM mysql.db </pre>

dürfen nur Benutzern gegeben werden, die diese auch für die entsprechenden Datenbanken benötigen.		<pre>WHERE Select_priv='Y' OR Insert_priv='Y' OR Update_priv='Y' OR Delete_priv='Y' OR Create_priv='Y' OR Drop_priv='Y' OR Alter_priv='Y'; # Für jeden Eintrag ist zu überprüfen, ob der Benutzer die Rechte für die Datenbanken benötigt.</pre>
Es muss sichergestellt sein, dass kein Benutzer mit leerem Benutzernamen existiert.		<pre>user@host:~\$ mysql -u root -p mysql> SELECT user,host FROM mysql.user WHERE user = ""; # darf nichts zurückgeben.</pre>
Es muss sichergestellt sein, dass bei der Benutzer-Authentifizierung keine Wildcards (“%“) im Hostnamen verwendet werden. Durch Wildcards ist es möglich, dass weitaus mehr Clients den Benutzer zur Authentifizierung verwenden können.		<pre>user@host:~\$ mysql -u root -p mysql> SELECT user, host FROM mysql.user WHERE host = '%'; # darf nichts zurückgeben. # In manchen Fällen ist der Einsatz von Wildcards sinnvoll, der Einsatzzweck sollte jedoch geprüft werden.</pre>
Es muss sichergestellt sein, dass ein Login des Super-Users nur an Localhost erfolgen kann.		<pre>user@host:~\$ mysql -u root -p mysql> SELECT User,Host,Password FROM mysql.user where user='root'; # Nur <hostname>; localhost; 127.0.0.1; ::1 darf zurückgegeben werden</pre>
Falls Passwörter als Authentisierungsmerkmal genutzt werden, müssen diese mindestens 8 Zeichen lang sein und drei der folgenden Zeichentypen beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.		<pre># Ab Version 5.6 kann eine Passwort-Policy über das Plugin validate_password eingestellt werden: root@host:~\$ editor /etc/mysql/my.cnf # Folgender Inhalt sollte in der my.cnf hinterlegt werden um das Plugin zu aktivieren: [mysqld] plugin_dir = /usr/lib/mysql/plugin plugin-load=validate_password.so validate- password=FORCE_PLUS_PERMANENT # Anschließend muss die Passwort-Policy konfiguriert werden: user@host:~\$ mysql -u root -p mysql> SHOW VARIABLES LIKE 'validate_password%'; validate_password_dictionary_file validate_password_length 8 validate_password_mixed_case_count 1 validate_password_number_count 1 validate_password_policy MEDIUM validate_password_special_char_count 1 </pre>
Es ist sicherzustellen, dass alle Accounts ein Passwort haben.		<pre>user@host:~\$ mysql -u root -p mysql> SELECT User,host FROM mysql.user WHERE (plugin IN('mysql_native_password', 'mysql_old_password') AND</pre>

		(LENGTH(Password) = 0 OR Password IS NULL)) OR (plugin='sha256_password' AND LENGTH(authentication_string) = 0); # darf nichts zurückgeben.
Zugriffe auf Datenbanksysteme sowie kritische Datenbank-Procedures und Datenbankinhalte müssen geloggt werden.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] log-error slow_query_log_file = /var/lib/mysql/slow-query.log slow_query_log = 1 log-raw = OFF # erst seit 5.7 log-warnings = 2
Die Betriebssystemrechte für die Dateien und Verzeichnisse der Datenbank (Programm-, Control-, Trace- und Logdateien) müssen exklusiv dem Betriebssystemkonto des Datenbanksystems zugeordnet werden.		# zum Überprüfen: sudo ls -l <Verzeichnis/Datei> # Identifizieren des Datenverzeichnis: user@host:~\$ mysql -u root -p mysql> show variables where variable_name = 'datadir'; root@host:~\$ chmod 700 <zurückgegebenes Verzeichnis/Datei> root@host:~\$ chown mysql:mysql <zurückgegebenes Verzeichnis/Datei> # Identifizieren der Log-Verzeichnisse und Dateien: user@host:~\$ mysql -u root -p mysql> show variables like 'log_bin_basename'; mysql> show variables like 'log_error'; mysql> show variables like 'slow_query_log_file'; mysql> show variables like 'relay_log_basename'; mysql> show variables like 'general_log_file'; root@host:~\$ chmod 660 <zurückgegebenes Verzeichnis/Datei> root@host:~\$ chown mysql:mysql <zurückgegebenes Verzeichnis/Datei> # Identifizieren der Datei mit dem SSL-Key: user@host:~\$ mysql -u root -p mysql> show variables where variable_name = 'ssl_key'; root@host:~\$ chmod 400 <zurückgegebenes Verzeichnis/Datei> root@host:~\$ chown mysql:mysql <zurückgegebenes Verzeichnis/Datei> # Identifizieren des Plugin-Verzeichnis: user@host:~\$ mysql -u root -p mysql> show variables where variable_name = 'plugin_dir' root@host:~\$ chmod 755 <zurückgegebenes Verzeichnis/Datei> root@host:~\$ chown mysql:mysql <zurückgegebenes Verzeichnis/Datei>

Der Datenbank-Daemon muss mit der Option <code>safe-user-create</code> gestartet werden.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] safe-user-create = 1
Die Startoption <code>old-passwords</code> muss deaktiviert werden, um sichere Passwort-Hash-Funktionen zu verwenden.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] old-passwords = 0
Die Startoption <code>secure-auth</code> muss verwendet werden, damit keine Accounts mehr verwendet werden können, die noch alte Passwort-Hashes haben.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] secure_auth=ON
Sofern nicht benötigt, muss die Symlink-Funktion deaktiviert werden.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] skip_symbolic_links=yes
Die Datenbank muss mit der Option <code>local-infile=0</code> gestartet werden.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] local-infile=0
Die Startoption <code>allow-suspicious-udfs</code> muss deaktiviert sein.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] allow-suspicious-udfs = false
Die Startoption <code>skip-grant-tables</code> muss deaktiviert sein.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] skip-grant-tables = false
Der MySQL-Listener ist an Localhost zu binden (anstelle von <code>skip-networking</code>).	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] bind-address = 127.0.0.1
Der SQL-Mode <code>NO_AUTO_CREATE_USER</code> muss aktiviert sein.		user@host:~\$ mysql -u root -p mysql> SET GLOBAL sql_mode = 'modes + NO_AUTO_CREATE_USER'; SET SESSION sql_mode = 'modes + NO_AUTO_CREATE_USER'; # Wirksamkeit testen: mysql> SELECT @@GLOBAL.sql_mode; SELECT @@SESSION.sql_mode;
Der Datenbank-Dämon muss mit einem dedizierten, nicht-administrativen Unix/Linux-Konto laufen, das auf dem System nur für die MySQL-DB verwendet wird.	x	root@host:~\$ editor /etc/mysql/my.cnf [mysqld] user = <mysql-user> # Wirksamkeit testen: user@host:~\$ ps -ef grep "^<mysql-user>.*\$" # darf nur den Mysql-Deamon "mysqld" zurückgeben
Standardmäßig loggt Unix alle MySQL-Kommandozeilen-Queries in die <code>.mysql_history</code> Datei im eigenem Home-Verzeichnis. Die History-Datei sollte regelmäßig gelöscht oder gegen <code>/dev/null</code> gelinkt werden. Optional: Betrifft die <code>.bash_history</code> ebenso.		Die Datei <code>~/mysql_history</code> darf keine sensiblen Daten/Queries enthalten (optional: <code>~/bash_history</code>). Als Softlink Richtung <code>/dev/null</code> : user@host:~\$ ln -s /dev/null \$HOME/mysql_history

6 Betriebssysteme

6.1 Linux

Maßnahme	Im Härtingskript umgesetzt	Kommandos zur Härtung
<p>Damit während des Boot-Vorgangs die Konfiguration von Grub nicht genutzt werden kann, um direkt in eine Root-Shell zu booten, sollten Teile des Grub-Menüs mit einem Passwort-Schutz versehen werden. Hierfür muss eine Anweisungsdatei für <code>{update-grub}</code> unter <code>/etc/grub.d/</code> angelegt werden, in der ein entsprechendes Passwort gesetzt wird, welches mittels <code>grub-mkpasswd-pbkdf2</code> erstellt wurde.</p>		<pre># Erstellung eines Passworts mittels grub- mkpasswd-pbkdf2: root@host:~\$ grub-mkpasswd-pbkdf2 # Erstellung einer Anweisung für update-grub: root@host:~\$ editor /etc/grub.d/01_boot- password # Inhalt der Datei inklusive des Passwort- Hashes aus grub-mkpasswd-pbkdf2: #!/bin/sh cat <<EOF ##### # Set a boot config protection password ##### set superusers="root" # The password is: toor -- Generate a new one running grub-mkpasswd-pbkdf2 password_pbkdf2 root grub.pbkdf2.sha512.10000.746EB9982AC[...] # ##### EOF # Ausführen von update-grub, damit die Änderungen wirksam werden: root@host:~\$ update-grub</pre>
<p>Viele Dienste und Protokolle öffnen Ports am Server, über die ein Angreifer potentiell ins System eingreifen kann. Aus diesem Grund müssen alle nicht benötigten Dienste und Protokolle deaktiviert werden.</p>		<pre># Prüfung des laufenden Dienste mittels systemctl: root@host:~\$ systemctl status <daemon> # Deaktivierung unnötiger Dienste: root@host:~\$ systemctl disable xxx.service # Prüfung der offenen Ports: root@host:~\$ ss -tulpn root@host:~\$ netstat -tulpen</pre>
<p>Bei der Installation von Software und Hardware werden oftmals Funktionen aktiviert, die nicht für den Betrieb und die Funktionalität des Systems notwendig sind. Funktionen der Software sind meistens ein fester Bestandteil, der nicht einzeln gelöscht oder deinstalliert werden kann. Solche Funktionen müssen über die Konfiguration oder Einstellungen dauerhaft deaktiviert werden.</p>		
<p>Damit einem Nameserver nicht erlaubt</p>	x	<pre>root@host:~\$ editor /etc/host.conf</pre>

wird, Hostnamen falsch aufzulösen (spoofen), wird die Einstellung des Resolvers so angepasst, dass eine Validierung des Hostnames durchgeführt wird. Dies schützt vor Hostname-Spoofing.		# Inhalt der Datei (Wesentlich ist "nospoof on"): order hosts, bind multi on nospoof on spooalert on
Der Linux-Kernel kann zum Schutz vor Auswirkungen von Buffer-Overflows angewiesen werden, die Speicherstruktur von Programmen zufällig zu gestalten.	x	root@host:~\$ echo 'kernel.randomize_va_space = 2' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ sysctl --system
Falls vorhanden, muss die Funktion für „rp_filter“ (Reverse Path Filter) bzw. eine entsprechende Funktion des verwendeten Derivates gesetzt sein. Diese Einstellung bewirkt, dass eingehende Netzwerk-Pakete gefiltert werden, für deren Quell-Adresse keine Rückroute bekannt ist.	x	root@host:~\$ echo 'net.ipv4.conf.all.rp_filter = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ sysctl --system
ARP-Pakete sollen nur von den Interfaces mit der korrekten Adresse beantwortet werden. Referenz: http://wiki.openvz.org/Multiple_network_interfaces_and_ARP_flux	x	root@host:~\$ echo 'net.ipv4.conf.all.arp_filter = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ echo 'net.ipv4.conf.default.arp_filter = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ echo 'net.ipv4.conf.all.arp_ignore = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ echo 'net.ipv4.conf.default.arp_ignore = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ echo 'net.ipv4.conf.all.arp_announce = 2' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ echo 'net.ipv4.conf.default.arp_announce = 2' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ sysctl --system
Die IPv4- und IPv6-Adressen aller Schnittstellen eines Servers müssen statisch konfiguriert werden.	x	root@host:~\$ editor /etc/network/interfaces: # Je Interface (hier eth0) muss eine statische IP-Adresse vergeben werden: iface eth0 inet static address 10.0.2.15/24 gateway 10.0.2.1 # Jedes umkonfigurierte Interface muss neu gestartet werden: ip addr flush dev eth0 && ifdown -a && ifup -a
Die Verarbeitung von ICMPv4- und ICMPv6-Paketen, die für den Betrieb nicht benötigt werden, muss deaktiviert werden.	x	root@host:~\$ echo 'net.ipv4.icmp_echo_ignore_broadcasts = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ echo 'net.ipv4.icmp_ignore_bogus_error_responses = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ sysctl --system

IP-Pakete mit nicht benötigten Optionen oder Erweiterung-Headern dürfen nicht bearbeitet werden. Bestimmte Optionen und Erweiterungen von Paketen könnten ausgenutzt werden, um einen Denial-of-Service-Angriff durchzuführen.	x	# Siehe Firewall-Maßnahmen.
Auf dem System gespeicherte Passwörter müssen vor unbefugtem Zugriff geschützt gespeichert werden. Grundsätzlich sollten nur Passwort-Hashes (bcrypt, scrypt) statt Passwörtern gespeichert werden. Dateien, die Hashes von Passwörtern enthalten, müssen vor unbefugtem Zugriff geschützt werden.	x	# Prüfung, dass Passwörter nicht in der /etc/passwd hinterlegt sind: root@host:~\$ cat /etc/passwd # Einschränkung des Zugriffs auf die /etc/shadow: root@host:~\$ chown root:shadow /etc/shadow root@host:~\$ chmod 640 /etc/shadow # Suche nach Dateien mit potentiellen enthaltenen Passwörtern: root@host:~\$ find -P / -nowarn -exec grep "passwd" {} --print \; 2>/dev/null
Sicherheitsrelevante Ereignisse müssen abhängig vom Verwendungszweck des Systems mit genauem Zeitstempel und einer eindeutigen Systembezeichnung protokolliert werden.	x	# Falls rsyslog als Log-Daemon verwendet wird: root@host:~\$ echo 'auth,authpriv.* /var/log/auth.log' >>/etc/rsyslog.conf root@host:~\$ echo '\$ActionFileDefaultTemplate RSYSLOG_SyslogProtocol23Format' >>/etc/rsyslog.conf
Der Schutz vor SYN-Flood Attacken muss aktiviert werden.	x	root@host:~\$ echo 'net.ipv4.tcp_syncookies = 1' >> /etc/sysctl.d/99-hardened.conf root@host:~\$ sysctl --system

6.2 SSH

Maßnahme	Im Härtingskript umgesetzt	Kommandos zur Härtung
Secure Shell (SSH) muss der einzige Dienst sein, der eine Fernwartung/Remote-Anmeldung mit privilegierten Rechten ermöglicht.	x	root@host:~\$ update-rc.d telnet disable root@host:~\$ update-rc.d rsh disable
Es muss ausschließlich die Protokollversion SSH-2 eingesetzt werden.	x	root@host:~\$ editor /etc/ssh/sshd_config # Einstellung in der sshd_config Datei, dass nur Protokoll 2 verwendet wird, was üblicherweise Standard ist: Protocol 2
Der SSH-Serverdienst muss als Stand-Alone-Daemon ausgeführt werden.	x	root@host:~\$ echo 'SSHD_OPTS=' > /etc/default/ssh root@host:~\$ update-rc.d xinetd disable
Es müssen starke Krypto-Algorithmen verwendet werden.	x	root@host:~\$ editor /etc/ssh/sshd_config # Folgende Einstellung zu den Direktiven Cipher, MAC und KexAlgorithms sollten

		<p>gesetzt werden: Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,aes256-ctr,aes192-ctr,aes128-ctr MACs hmac-sha2-512,hmac-sha2-256,hmac-ripemd160 KexAlgorithms curve25519-sha256@libssh.org,ecdh-sha2-nistp521,ecdh-sha2-nistp384,ecdh-sha2-nistp256,diffie-hellman-group-exchange-sha256</p>
TCP/IP-Port-Weiterleitungen („Port Forwarding“) müssen kontrolliert eingesetzt oder deaktiviert werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config GatewayPorts no # Standardwert</pre>
Gateway-Ports dürfen nicht aktiviert werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config GatewayPorts no # Standardwert</pre>
X11-Weiterleitungen müssen kontrolliert eingesetzt oder deaktiviert werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config X11Forwarding no # Falls X11Forwarding notwendig sein sollte, so sollte es an das Loopback Interface gebunden werden: X11UseLocalhost yes</pre>
SSH-„Agent Forwarding“ muss serverseitig deaktiviert werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config AllowAgentForwarding no</pre>
Tunnel Devices dürfen nicht benutzt werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config PermitTunnel no # Standardwert</pre>
Es muss unterbunden werden, dass der Root-Account direkt über SSH angemeldet werden kann. Vorab muss geprüft werden, dass es einen alternativen Nutzer gibt, da sonst eine Aussperrung erfolgt.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config PermitRootLogin no</pre>
SSH-Logins müssen aufgezeichnet werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config SyslogFacility AUTH #Standardwert LogLevel VERBOSE</pre>
Host-basierte Authentifizierung (via rhosts/shosts) darf nicht verwendet werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config IgnoreRhosts yes # Standardwert RhostsRSAAuthentication no # Standardwert HostbasedAuthentication no # Standardwert IgnoreUserKnownHosts yes</pre>
Ein leeres Passwortfeld darf nicht akzeptiert werden und es sollte der Zeitpunkt der letzten Anmeldung angezeigt werden.	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config PrintLastLog yes #Standardwert PermitEmptyPasswords no #Standardwert</pre>
Administrative Dienste müssen an genau eine Schnittstelle gebunden werden. Wenn der Server an mehreren Netzen erreichbar ist, ist es sinnvoll, administrative und fachliche Netze zu	x	<pre>root@host:~\$ editor /etc/ssh/sshd_config # Der SSH-Dienst sollte bei Verwendung mehrerer Netze auf die IP-Adresse des administrativen Netzes gebunden sein, so dass er nicht im fachlichen Netz erreichbar ist.</pre>

trennen, so dass z.B. SSH ausschließlich aus dem administrativen Netz erreichbar ist.		ListenAddress \$IPADDRESS # IP-Adresse des administrativen Netzwerkes (0.0.0.0 = alle Interfaces)
Administrative Dienste müssen an einen definierten Port gebunden werden.	x	root@host:~\$ editor /etc/ssh/sshd_config Port 22 #Standardwert
Falls IPv4 und IPv6 nicht gleichzeitig verwendet werden, sollte das zu nutzende Protokoll angegeben werden.	x	root@host:~\$ editor /etc/ssh/sshd_config # Angabe, ob IPv4 ("inet") oder IPv6 ("inet6") unterstützt wird. AddressFamily inet
Pluggable Authentication Modules (PAM) sollte nur verwendet werden, falls unbedingt erforderlich. Die Hinweise in der manpage für sshd_config sind zu beachten.	x	root@host:~\$ editor /etc/ssh/sshd_config # Angabe, ob PAM genutzt ("yes") oder nicht verwendet wird ("no"). UsePAM no
Die Nutzung des SSH-Dienstes muss auf Gruppen (bzw. Benutzer) beschränkt werden.		root@host:~\$ editor /etc/ssh/sshd_config # Angabe von erlaubten Benutzern in der Form user1[@host] mit Trennung durch Leerzeichen: AllowUsers \$USERS # Angabe von erlaubten Benutzergruppen mit Trennung durch Leerzeichen: AllowGroups \$GROUPS

6.3 Benutzer und Rechte

Maßnahme	Im Härtingskript umgesetzt	Kommandos zur Härtung
Alle Prozesse müssen mit den minimal zur Funktion des Prozesses notwendigen Rechten gestartet werden. Insbesondere sollten Prozesse nach Möglichkeit nicht unter dem Benutzer root betrieben werden. Eine Wiederverwendung von Laufzeitbenutzern für mehrere Prozesse sollte vermieden werden, so dass z.B. der Apache httpd und der Tomcat Server nicht mit demselben Benutzer betrieben werden.		
Konfigurationsdateien, die von Prozessen mit Root-Rechten ausgelesen werden können, sowie ausführbare Dateien, die mit Root-Rechten aufgerufen werden (beispielsweise über cron-, Init-Skripte usw.), und die Verzeichnisse, in denen sich diese befinden, müssen mittels Dateisystem-Berechtigungen derart geschützt sein, dass sie nur mit Root-Rechten veränderbar sind.		

Bei der Erstellung von Dateien müssen die Dateiberechtigungen so gesetzt sein, dass das Verändern nur dem Ersteller der Datei bzw. der entsprechenden Gruppe möglich ist.	x	# In hardening.sh in '/etc/profile.d': umask 0022
Die Mount Points für Dateisysteme, die von Benutzern ohne root-Rechte gemountet werden können, müssen mit der Option „nodev“ und „nosuid“ versehen werden.		# Suchen in /etc/fstab nach Einträgen mit der user-Option und Hinzufügen von nodev und nosuid.
Konten müssen gegen unautorisierte Nutzung durch Verwendung mindestens eines Authentisierungsmerkmals geschützt werden.		# Prüfung der /etc/shadow auf Benutzer ohne Passwort: gawk -i inplace -v INPLACE_SUFFIX=.bak 'BEGIN{FS=OFS=":"} \$2=="{"\$2="!"} {print \$0}' /etc/shadow
Die Berechtigungen von Konten und Anwendungen müssen auf ein für deren Aufgaben notwendiges Minimum reduziert werden.		
Vordefinierte Konten müssen deaktiviert werden.		
Accounts (außer Root), die für die Administration bestimmter Dienste eingerichtet werden (beispielsweise ein Account für die Wartung der Datenbank oder des Webservers), müssen so eingerichtet werden, dass damit ausschließlich diejenigen Arbeiten durchgeführt werden können, die für diese Rolle notwendig sind (Prinzip der minimalen Rechte – Least Privilege).		
Jeder System-Account muss eine eindeutige UID aufweisen.		# Prüfung in der /etc/passwd.
Die PATH-Umgebungsvariable aller Accounts darf nicht den Pfad „.“ (aktuelles Verzeichnis) enthalten.	x	# Prüfung, ob in der PATH-Umgebungsvariable ein . enthalten ist: # In hardening.sh in '/etc/profile.d/' PATH=\$(echo -n \${PATH} awk 'BEGIN{ORS=RS=":"} \$0!="."{print \$0}') export PATH
Vordefinierte Authentisierungsmerkmale müssen geändert werden. Oft werden Software-Komponenten mit einem vordefinierten Passwort installiert. Diese Standard-Passwörter müssen geändert werden, da diese ansonsten zur Authentifizierung von Angreifern genutzt werden können.		
Falls Passwörter als Authentisierungsmerkmal genutzt		# Die Konfiguration einer Passwort-Richtlinie für Betriebssystemnutzer kann in folgenden

werden, müssen diese mindestens 8 Zeichen lang sein und drei der folgenden Zeichentypen beinhalten: Kleinbuchstaben, Großbuchstaben, Ziffern und Sonderzeichen.		Dateien vorgenommen werden: /etc/login.defs /etc/security/pwquality.conf
Falls Passwörter als Authentisierungsmerkmal genutzt werden, muss ein Schutz gegen Wörterbuch- und Brute-Force-Angriffe vorhanden sein, der das Erraten von Passwörtern stark erschwert.		# Es kann das Paket fail2ban verwendet werden, um Accounts zu sperren, welche mehrmals mit einem falschen Passwort authentifiziert werden sollten.
Falls Passwörter als Authentisierungsmerkmal genutzt werden, darf deren Darstellung nicht im Klartext erfolgen.		root@host:~\$ editor /etc/pam.d/common-password # insbesondere die Direktive obscure bewirkt das Ausblenden von Passwörtern: password [success=1 default=ignore] pam_unix.so obscure use_authok try_first_pass sha512
Die Identifizierung von Benutzern an Systemaccounts (root, oracle, wlsadmin,...) muss eindeutig sein und darf nur nach einer individuellen, starken Authentifizierung erfolgen. Es dürfen keine gemeinsamen Passwörter für die Remoteanmeldung verwendet werden.		

6.4 Firewall

Maßnahme	Im Härtingskript umgesetzt	Kommandos zur Härtung
Die Erreichbarkeit von Diensten muss eingeschränkt werden. Es ist jedoch die Einhaltung von RFC 1122 zu beachten, insbesondere beim Filtern bzw. Zulassen von ICMP Paketen. Falls sich Dienste nicht durch Konfiguration auf die minimal erforderlichen Schnittstellen (z.B. auf localhost) beschränken lassen, muss ein lokaler Paketfilter die Erreichbarkeit reglementieren.	x	# Einstellungen der iptables-Firewall müssen gezielt nach Einsatzzweck vorgenommen werden. Es wird empfohlen, die INPUT und OUTPUT Chains generell zu verwerfen (DROP) und mit einem Whitelist-Ansatz einzelne Ports/Protokolle freizugeben (ACCEPT): root@host:~\$ iptables -P INPUT DROP root@host:~\$ iptables -P OUTPUT DROP root@host:~\$ iptables -P FORWARD DROP # Weiterführend müssen alle ein- und ausgehenden Protokolle und Ports mit ACCEPT definiert werden. # Nach einem Test der Auswirkungen muss iptables persistiert werden, damit es nach einem Neustart weiterhin sicher konfiguriert ist.

6.4.1 Beispiel für eine iptables-Konfiguration

Die IP-Adressen müssen angepasst werden. Eventuell sind weitere Freischaltungen notwendig.

```
# Generated by iptables-save v1.4.21
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT DROP [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -m conntrack --ctstate INVALID -j DROP
-A INPUT -p tcp -m tcp --dport 22 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT
-A INPUT -s x.x.x.x/32 -p udp -m udp --sport 53 -m comment --comment "first DNS" -j ACCEPT
-A INPUT -s x.x.x.x/32 -p udp -m udp --sport 53 -m comment --comment "second DNS" -j ACCEPT
-A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
-A INPUT -p icmp --icmp-type source-quench -j ACCEPT
-A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT
-A INPUT -p icmp --icmp-type parameter-problem -j ACCEPT
-A INPUT -p icmp --icmp-type echo-reply -j ACCEPT
-A OUTPUT -o lo -j ACCEPT
-A OUTPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A OUTPUT -m conntrack --ctstate INVALID -j DROP
-A OUTPUT -d 192.168.x.x/32 -p tcp -m tcp --dport 8080 -m comment --comment "proxy" -j ACCEPT
-A OUTPUT -d x.x.x.x/32 -p udp -m udp --dport 53 -m comment --comment "first DNS" -j ACCEPT
-A OUTPUT -d x.x.x.x/32 -p udp -m udp --dport 53 -m comment --comment "second DNS" -j ACCEPT
-A OUTPUT -d 192.168.x.x/32 -p tcp -m tcp --dport 25 -m comment --comment "mail" -j ACCEPT
-A OUTPUT -s 192.168.x.x/32 -p tcp -m tcp --dport 443 -m conntrack --ctstate NEW -j ACCEPT
-A OUTPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
-A OUTPUT -p icmp --icmp-type source-quench -j ACCEPT
-A OUTPUT -p icmp --icmp-type time-exceeded -j ACCEPT
-A OUTPUT -p icmp --icmp-type parameter-problem -j ACCEPT
-A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT
COMMIT
```

Literaturverzeichnis

- [1] Bundesamt für Sicherheit in der Informationstechnik, Studie „Sicherheitsuntersuchung von Content-Management-Systemen“, Stand 01.04.2016.
 - [2] Deutsche Telekom AG, Privacy and Security Assessment Verfahren, <http://www.telekom.com/psa>, Stand 28.04.2016.
 - [3] Deutsche Telekom AG, Technische Sicherheitsanforderungen, <http://www.telekom.com/static/-/155996/21/technische-sicherheitsanforderungen-si>, Stand 28.04.2016.
 - [4] Bundesamt für Sicherheit in der Informationstechnik, BSI-Standards zur Internet-Sicherheit (Isi-Reihe), https://www.bsi.bund.de/DE/Themen/Cyber-Sicherheit/Isi-Reihe/Isi-Reihe_node.html, Stand 29.10.2015.
-

Stichwort- und Abkürzungsverzeichnis

Begriff	Beschreibung
Application Programming Interface (API)	Schnittstelle zur Anwendungsprogrammierung
BSI-Standards zur Internet-Sicherheit (ISi)	Standard zur Information von Behörden und Unternehmen zur Absicherung von Web-Angeboten
Cascading Style Sheets (CSS)	Computersprache für die Gestaltung digitaler, vorwiegend Web-basierter Dokumente
Common Gateway Interface (CGI)	Standard für den Datenaustausch zwischen einem Webserver und dritter Software
Content-Management-System (CMS)	Software zur Erstellung und Pflege von Webinhalten
Datenbank (DB)	System zur elektronischen Datenverwaltung
Denial of Service (DDoS)	Angriff zur Einschränkung der Verfügbarkeit einer Anwendung oder eines Dienstes
End of Life (EOL)	Bezeichnung für nicht mehr produzierte bzw. nicht mehr lieferbare Produkte
Extensible Markup Language (XML)	Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten in Form von Textdateien
Hypertext Markup Language (HTML)	Textbasierte Auszeichnungssprache zur Strukturierung digitaler Dokumente
Internet Adresse (IP)	Adresse in Computernetzen
JavaScript Object Notation (JSON)	Auf JavaScript basierendes Datenformat in Textform zum Zweck des Datenaustauschs zwischen Anwendungen
Lightweight Directory Access Protocol (LDAP)	Netzwerkprotokoll zum Lesen und Bearbeiten von Verzeichnissen, welche in verteilten Netzen/Anwendungen zur Verfügung stehen
Pluggable Authentication Modules (PAM)	Authentifizierungsframework für Anwendungen und Dienste eines Linux-Systems
Secure Shell (SSH)	Bezeichnet sowohl ein Netzwerkprotokoll als auch entsprechende Programme, mit deren Hilfe man auf eine sichere Art und Weise eine verschlüsselte Netzwerkverbindung mit einem entfernten Gerät herstellen kann
Structured Query Language (SQL)	Datenbanksprache zur Definition von Datenstrukturen in relationalen Datenbanken sowie zum Bearbeiten und Abfragen von darauf basierenden Datenbeständen

Erläuterungen zu 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
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.
user@host:~\$	user@host:~\$ ps -ef grep root	Ein Shell-Befehl, ausgeführt ohne spezielle Rechte.
mysql>	mysql> SELECT * from mysql.user;	Ein MySQL-Befehl. Kann entweder in der Kommandozeile oder in einem MySQL-Client verwendet werden.
<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.