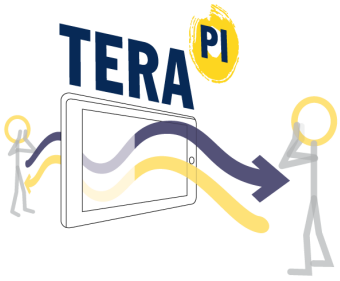

Magnus Executive Consulting GbR

Tera-Pi, ein Produkt der Terapon GmbH

IT-Sicherheitsarchitektur

Dateiname	sicherheitsarchitektur-1.3.odt
Autoren	Nils Magnus, Magnus Executive Consulting GbR, magnus@linuxtag.org
Klassifikation	vertraulich

Version	Datum	Beschreibung	Autor	Status
0.1	20.05.2014	Erste Fassung	Nils Magnus	Entwurf
0.2	28.07.2014	Initialisierung des Dokuments	Nils Magnus	Entwurf
0.3	08.08.2014	Anforderungen	Nils Magnus	Prerelease
0.4	12.08.2014	Risikobetrachtung	Nils Magnus	Prerelease
0.5	18.08.2014	Detailspezifikation von Sicherungsmaßnahmen	Nils Magnus	Prerelease
0.8	19.08.2014	Zum Review an Techcast	Nils Magnus	Review
0.9	26.08.2014	Feedback von Entwicklung	Nils Magnus	Review
1.2	05.09.2014	Einarbeitung des Feedbacks	Nils Magnus	RC
1.3	01.10.2014	Feedback vom Auftraggeber	Nils Magnus	Release

Inhalt

1 Executive Summary

Die Tera-Pi-Anwendung verarbeitet Daten von Teilnehmern, die den sensiblen gesundheitlichen Bereich betreffen. Ein wesentlicher Faktor für den Erfolg der Plattform ist das Vertrauen seiner Anwender in den sicheren und datenschutzkonformen Umgang mit diesen Daten.

Der Betrieb der Tera-Pi-Anwendung erfordert daher eine Reihe von Sicherheitsmaßnahmen zur Sicherung von Vertraulichkeit, Integrität und Verfügbarkeit der Anwendung. Dieses Dokument beschreibt Maßnahmen und architektonische Rahmenbedingungen, die diese Sicherheitsziele umsetzen. Tabelle 1 fasst die Sicherheitsfunktionen zusammen.

Die in diesem Dokument definierten Festlegungen sind Teil der Applikationsspezifikation und wurden im Rahmen der Entwicklung der Plattform berücksichtigt. Das konkrete Vorhandensein und die Wirksamkeit der gelisteten Sicherheitsfunktionen sollte bei Abnahme und Überführung in den Regelbetrieb durch ein Sicherheitsaudit überprüft und testiert werden.

Sicherheits-funktion	Bereich	Kurzbeschreibung der Funktion
1	Integrität	Zur Registrieren ist ein Lizenzkey notwendig.
2	Vertraulichkeit	Die Anmeldedaten enthalten nur wenige personenbezogene Daten. Die Zuordnung zu einer konkreten Person erfolgt über einen Pseudonymisierungsserver.
3	Vertraulichkeit	Im Rahmen der Anwendung werden keine personenbezogenen Daten direkt verarbeitet.
4	Vertraulichkeit	Zur Datenübertragung nutzt die Anwendung eine verschlüsselte Transportsicherung.
5	Verfügbarkeit	Die Verfügbarkeit wird durch redundante Applikations- und Datenbankserver erhöht (erweiterte Sicherheitsfunktion, optional).
6	Integrität	Backups implementieren einen Schutz gegen Datenverlust und -veränderung.
7	Vertraulichkeit	Ein Löschkonzept regelt den Umgang mit Daten nach Beendigung der Anwendungsnutzung.
8	Vertraulichkeit	Statistiken und Analysen personenbezogener Daten werden nicht angefertigt.
9	Vertraulichkeit	Logfiles speichert die Anwendung nur für betriebliche Zwecke und für eine kurze Zeit.
10	Vertraulichkeit	Durch Nondisclosure-Agreements ist eine Weitergabe von Daten vertraglich ausgeschlossen.
11	Vertraulichkeit	Es gelten besondere Vertraulichkeit und Sorgfaltsverpflichtungen für Mitarbeiter und beauftragte dritte Unternehmen.

Tabelle 1: Übersicht der Sicherheitsfunktionen

2 Motivation und Ziele

Die T*echcast GmbH* (im Folgenden *Entwickler* oder *Betreiber*) konzipiert, entwickelt und betreibt für die *Terapon GmbH*, ein Unternehmen der KÖTTER Unternehmensgruppe (im Folgenden *Anbieter*) das Portal *Tera-Pi*. Dieses Portal bietet *Lizenznehmern* und ihren Mitarbeiter eine Plattform, um sich über verschiedene Gesundheitsthemen zu informieren und eventuell die Wartezeit bis zu einer klassischen Therapie im jeweiligen Feld durch interaktive, computer- und netzgestützte Maßnahmen zu überbrücken und zu ergänzen.

Aufgrund der besonderen Natur dieses Angebotes und dem Umgang mit personenbezogenen Daten identifiziert diese **IT-Sicherheitsarchitektur** allfällige IT-Bedrohungen für die **Sicherheitsziele** Vertraulichkeit, Integrität und Verfügbarkeit des Angebots. Es orientiert sich dabei im Aufbau an typischen Standardwerken wie den BSI-Grundschutzkatalogen oder dem Regelwerk von ISO 2700x. Daraus leitet es Anforderungen und Spezifikationen ab, die dafür sorgen, dass die Sicherheitsziele für die Anwender und Betreiber der Anwendung erreicht werden.

2.1 Aufbau und Gliederung

Diese Sicherheitsarchitektur gliedert sich in zwei Teile: Der erste beschreibt in Kapitel 3 insgesamt elf Sicherheitsfunktionen, die dazu diesen, die Sicherheitsziele zu erreichen. Ein realisiertes Sicherheitsziel sorgt jeweils für die Umsetzung einer Kategorie aus Vertraulichkeit, Integrität und Verfügbarkeit. Aus unterschiedlichen Rollen und in verschiedenen Kontexten heraus sind diese Maßnahmen gruppiert. Abschnitt 3.1 beschreibt die Registrierung neuer Teilnehmer, der Kernabschnitt 3.2 die eigentliche Nutzung der Plattform und Abschnitt 3.3 Statistik sowie Abrechnung zwischen Anbietern und Lizenznehmern. Die mit dem Betrieb und der Wartung der Plattform verbundenen Ziele beschreibt Abschnitt 3.4.

Die Sicherheitsfunktionen sind technologie-agnostisch beschrieben, sodass sie auch bei einer Erweiterung des Angebotes oder bei der Aktualisierung einzelner Teilkomponenten weiterhin Gültigkeit besitzt.

Kapitel 4 legt technische Details und konkrete Parameter fest, die die Sicherheitsfunktionen bei Stand der Anlage dieses Dokumentes erfüllten. In regelmäßigen Abständen sollten diese Angaben einer Revision unterzogen und ggf. angepasst werden. Abschnitt 4.1 widmet sich dem Pseudonymisierungsdienst, Abschnitt 4.2 dem administrativen Zugang zu den Systemen. Die Verwaltung von Lizenzschlüsseln regelt Abschnitt 4.3. Maßnahmen zur technischen Härtung der eingesetzten Software spezifiziert Abschnitt 4.4 und die allgemeine Hostsicherheit beschreibt Abschnitt 4.5.

Kapitel 5 beschreibt optionale Maßnahmen, die das Sicherheitsniveau der Anwendung zusätzlich erhöhen.

3 Dynamische Anwendungsbeschreibung und Sicherheitsfunktionen

Aus Sicht der Anwender besteht die Nutzung der Anwendung aus mehreren Teilprozessen, die jeweils eigene **Sicherheitsanforderungen (Ziele)** besitzen. Dazu implementiert die Anwendung eine Reihe von **Sicherheitsfunktionen**. Dieser Abschnitt beschreibt die Abläufe und die in diesem Zusammenhang getroffenen **Maßnahmen zur Sicherung** der jeweiligen Ziele.

3.1 Registrieren

Sicherheitsfunktion 1: Wahrung der Integrität der Benutzerberechtigungen

Ein Lizenznehmer, der die Nutzung von Tera-Pi lizenziert, erhält vom Betreiber eine Anzahl von Lizenzen , die aus einem mindestens 48 Bit langen Schlüssel bestehen und die beispielsweise aus einem zehn-stelligen Passwort bestehen. Maximal 2^{32} Lizenzen dürfen vergeben werden, sodass eine Brute-Force-Attacke per Enumeration auf den Lizenzkey 65.535 Versuchen standhält. Die Applikation verhindert, dass binnen zehn Minuten mehr als 100 Eingabeversuche auf einen Lizenzkey erfolgen.

Der Lizenznehmer verteilt die Lizenzschlüssel unter eigener Verantwortung an seine Mitarbeiter (im Folgenden *Teilnehmer*) und teilt ihnen den Link zur Registrationsseite des Portals mit.

Sicherheitsfunktion 2: Wahrung der Vertraulichkeit der Anmeldedaten

Auf der Registrationsseite registrieren sich die Teilnehmer mit folgenden Angaben:

- **Name**, mit dem die Teilnehmer von der Anwendung angesprochen werden. Dieser Name soll explizit ein Pseudonym sein. Der Lizenznehmer sollte die Teilnehmer darauf hinweisen, dass diese Möglichkeit besteht. Ein Beispiel für den Namen ist „Erika Mustermann“ oder „josef456“.
- **Lizenzschlüssel**, den der Teilnehmer vom Lizenznehmer erhalten hat, um seine Berechtigung nachzuweisen. Ein Beispiel für den Lizenzschlüssel ist „IOMVQHNSKY“.
- **E-Mail-Adresse**, an die über eine Anonymisierungsfunktion Nachrichten an den Teilnehmer zugestellt werden können. Der Lizenznehmer kann seine Teilnehmer darauf hinweisen, dass auch für die E-Mail-Adresse auf Wunsch eine anonyme Adresse verwendet werden kann, sofern der Teilnehmer unter dieser Adresse Nachrichten empfangen kann. Beispiele dafür sind zusätzliche E-Mail-Accounts von Dienstleistern wie GMX oder Gmail. Diesen E-Mail-Account benötigt die Plattform etwa für eine initiale Bestätigung des Zugangs.

  Alternativ steht es Teilnehmern jedoch auch frei, aus Komfortgründen einen bestehenden E-Mail-Zugang zu nutzen. Die Anwendung ersetzt ohnehin direkt nach Eingabe der Adresse diese durch eine pseudonymes Pendant, sodass zu keiner Zeit eine Verknüpfung von Benutzernamen, Lizenzschlüssel und E-Mail-Adresse auf dem System der Anwendung gespeichert sind. Details zum Pseudonymisierungsdienst beschreibt Abschnitt 4.1. Ein Beispiel für die E-Mail-Adresse ist „erika.mustermann47@gmail.com“ oder „abc@mailinator.com“.

Nach der Registration erhält der Teilnehmer einen Aktivierungslink an die hinterlegte E-Mail-Adresse, der einmal gültig ist. Sobald er diesen Link aufruft, wird er aufgefordert, ein Passwort zu wählen. Mit seinem Benutzernamen und dem gewählten Passwort kann er sich fortan im Portal einwählen.

3.2 Nutzung der Anwendung

Sicherheitsfunktion 3: Wahrung der Vertraulichkeit während der Anwendungsnutzung

Während der Nutzung der Anwendung sind nur wenige Sicherheits- und Datenschutzfragen relevant, da die Anwendung selbst mit Pseudonymen arbeitet und dort keine Verbindungen vom pseudonymen Benutzernamen zu einem personenbezogenen Merkmal gespeichert sind. Möchte die Anwendung selbst oder der Betreuer des Teilnehmer diesem eine Nachricht schicken, so sendet die Anwendung diese an einen gesonderten Pseudonymisierungsserver, der seinerseits die Nachricht erst zustellt.

Sicherheitsfunktion 4: Wahrung der Vertraulichkeit der Datenübertragung

Die Übertragung von Daten zur Anwendung und die Antworten daraufhin sind auch ohne direkten Bezug zu einer realen Identität schutzwürdig und sind daher vor dem Abhören geschützt. Außer der Startseite werden alle Abrufe der Anwendung durch das Transportprotokoll SSL/TLS verschlüsselt.

Zum Einsatz kommen mindestens SSLv3 und TLS1.0, TLS1.2 wird jedoch ebenfalls angeboten werden. Der öffentliche Schlüssel des Zertifikats hat mindestens 2048 Bit. Als symmetrischer Cipher kommt AES mit mindestens 128 Bit zum Einsatz, als Hashfunktion SHA-2 mit 256 Bit. Grundsätzlich orientiert sich die Konfiguration der Transportsicherung den Empfehlungen der Technischen Richtlinie TR-02102-2 des Bundesamtes für Sicherheit in der Informationstechnik.

Sicherheitsfunktion 5: Wahrung der Verfügbarkeit

Damit sich Teilnehmer auf den Beratungserfolg der Anwendung einlassen, erwarten Sie eine dauerhafte und ununterbrochene Verfügbarkeit der Anwendung. Regelmäßige Kontrolle und Wartung der Software- und Hardware-Komponenten verringert das Risiko eines Ausfalls.

Schutz vor technischem Versagen einzelner Komponenten (Webserver, Datenbank) leistet diese Maßnahme nicht. Eine über den genannten Grundschutz hinausgehende Implementierung der Sicherheitsfunktion 5 beschreibt Abschnitt 5.1.

Sicherheitsfunktion 6: Wahrung der Integrität durch Backups

Zur Vorbeugung von Datenverlusten bzw. zum Zweck der Wiederherstellung der Datenintegrität werden sowohl von Datenbank wie Applikationsserver täglich inkrementelle und wöchentliche Vollbackups angefertigt. Jeweils vier Vollbackups werden vorgehalten

Sicherheitsfunktion 7: Wahrung der Vertraulichkeit nach Löschung

Wenn sich ein Teilnehmer dazu entscheidet, fortan nicht mehr das Angebot der Anwendung zu nutzen und sich explizit davon abmeldet, werden seine personenbezogenen Merkmale komplett aus der aktiven Datenbank gelöscht.

In Verlaufsdaten wie Logfiles und Backups sind die personenbezogenen Daten nach Ablauf der Rentention-Zeit nicht mehr enthalten.

3.3 Abrechnung und Auswertung

Sicherheitsfunktion 8: Wahrung der Vertraulichkeit durch Verzicht auf Auswertungen

Zu Abrechnungs- und Statistikzwecken analysiert weder der Betreiber noch der Anbieter personenbezogene Daten und stellt solche auch nicht dem Lizenznehmer zur Verfügung.

Sicherheitsfunktion 9: Wahrung der Vertraulichkeit der Zugriffe

Beim Zugriff auf die Anwendung verwaltet der Webserver prinzipbedingt die IP-Adresse des aufrufenden Teilnehmers. Über diese Adressen ist es je nach Legislation mitunter möglich, den aufrufenden Teilnehmer persönlich zu identifizieren (oder zumindest den angemeldeten Telekommunikationsteilnehmer). Im deutschen Rechtsraum ist dazu eine richterliche Anordnung notwendig, in anderen Rechtsräumen können jedoch andere Regelungen gelten.

Der Webserver speichert die Logfiles für maximal 14 Tage, um daraus betriebstechnische Auswertungen anzufertigen. Diese Auswertungen gruppieren einzelne Aufrufe zu Sessions, ordnen ihnen jedoch keine IP-Adressen oder damit unmittelbar assoziierten Daten zu.

Sicherheitsfunktion 10: Wahrung der Vertraulichkeit durch Nondisclosure

Der Betreiber und der Anbieter erklären, dass sie Einblick in Daten aus der laufenden Anwendung oder den gespeicherten Verlaufsdaten nur im Rahmen ihrer Rechtspflichten und im Rahmen richterlicher Anordnung gewährend. Ferner lassen sie sich im Rahmen der Lizenzvereinbarung vom Lizenznehmer eine Selbstverpflichtungserklärung ausstellen, dass dieser niemals Einblick in die Daten der laufenden Anwendung und den gespeicherten Verlaufsdaten verlangen wird.

3.4 Wartung

Sicherheitsfunktion 11: Wahrung der Vertraulichkeit während des Betriebs

Der Betreiber sorgt dafür, dass alle seine Mitarbeiter auf Vertraulichkeit und Sorgfalt verpflichtet werden, die im Rahmen von betrieblicher Wartung, Aktualisierung, Weiterentwicklung und Aktualisierung Zugriff auf die Anwendung, der dort verarbeiteten Daten sowie der zugrunde liegenden Datenverbindungen und Systeme haben. Dies gilt auch für alle dritten Unternehmen, die Aufgaben im Rahmen einer Auftragsdatenverarbeitung (ADV) i. S. d. §11 BDSG übernehmen, sowie für ihre Erfüllungsgehilfen.

4 Spezifikation wichtiger Teilkomponenten

Dieser Abschnitt beschreibt und dokumentiert technische Details und Anforderungen an Teilkomponenten, die eine wichtige Rolle für die Sicherheitsfunktionen spielen.

4.1 Pseudonymisierungsserver

Der Pseudonymisierungsdienst ist ein kritisches System für die Sicherheitsziele. Gleichzeitig ist er notwendig, um die Funktion der Benachrichtigung über eine existierende E-Mail-Adresse zu realisieren. Um die Sicherheitsziele zu erreichen, bedarf dieses System besonderer Absicherung und Isolation. Es läuft auf einem eigenem Host bzw. eigener virtuellen Maschine (VM). Es ist der einzige Dienst auf diesem System. Ausschließlich Mitarbeiter des Betriebsteams haben Zugriff auf das System (sowohl Host wie Datenbank).

Während der Registration füllt der Teilnehmer ein Formular aus (siehe Abschnitt 3.1) und trägt dort auch die Klartext-E-Mail-Adresse ein. Das Formular sendet dieses Datum jedoch nicht an den Applikationsserver, sondern fängt die Übertragung mittels Javascript ab, das sich per AJAX-Call zum Pseudonymisierungsserver verbindet:

```
<script
  src="http://devel.techcast.com/d7_terapi/sites/all/modules/terapi/js/register.js">
</script>
```

Vom Pseudonymisierungsserver erhält die Registrationsseite den pseudonymen E-Mail-Alias:

```
$ wget -SqO- 'http://devel.techcast.com/terapi-mailservice/anomail/alias.jsonp?
_suppress_status_code=1&callback=obj&email=abcxxx%4f.de&_=1409064985464'
  HTTP/1.0 200 OK
  Date: Tue, 26 Aug 2014 15:18:50 GMT
  Content-Length: 94
  Vary: Accept-Encoding
  Content-Type: application/javascript; charset=utf-8
  Connection: keep-alive

obj({
    "status": 200,
    "data": "cc0d8f76d9de8eca66fb8ca5e8a28b71@devel.techcast.com"
});
```

Der abgebildete Aufruf verdeutlicht die Funktionsweise und die Übertragenen Daten. Im Produktionssystem erfolgt die Verbindung zu einer separaten VM und ist weiterhin per SSL/TLS abgesichert.

Der Applikationsserver lässt sich vom Pseudonymisierungsdienst die Gültigkeit des Alias bestätigen, bevor er den Account anlegt. Nicht aktivierte Aliase und die ihnen zugeordneten realen E-Mail-Adressen löscht der Pseudonymisierungsdienst einmal pro Stunde.

Die Zustellung von Nachrichten des Applikationsservers erfolgt per SMTP zum Pseudonymisierungsserver, der dazu eine Schnittstelle anbietet. Der so aufgerufene Mail Transfer Agent (MTA) leitet die Nachricht ebenfalls per SMTP an den realen E-Mail-Empfänger weiter.

Der Pseudonymisierungsdienst nimmt nur Nachrichten vom Applikationsserver entgegen.

4.2 Zugang zu den Systemen

Der physikalische Zugang zum System inkl. einem möglichen Hostsystem und allen zur Datenspeicherung und zu Backupzwecken genutzten weiteren Systemen ist nur dem Betriebsteam möglich. Lässt der Betreiber seine Systeme extern bei einem Dienstleister hosten, verpflichtet er diesen im Rahmen einer Auftragsdatenverarbeitung (ADV) i. S. d. §11 BDSG zu den gleichen Sorgfaltspflichten, die auch für ihn selbst gelten.

Es gibt keinen lokalen Zugang zum laufenden System, es sei denn, er ist per Passwort geschützt und hat danach die gleichen Sicherheitskonfigurationen wie ein SSH-Zugang.

Der Zugang zum eigentlichen System erfolgt in der Regel nur per SSH. Login auf dem System nur per persönlichem Account des Betriebsmitarbeiters. Aktivitäten mit Rootrechten benötigen expliziter Nutzung von „sudo“. Aktivitäten mit Rootrechten werden protokolliert. Der Betreiber informiert die Betriebsmitarbeiter über diese Maßnahmen und verpflichtet sie auf Vertraulichkeit und besondere Sorgfalt.

4.3 Ausgabe von Lizenzschlüsseln

Der Betreiber erzeugt Lizenzschlüssel, die der Anbieter an die Lizenznehmer zum Zweck der Abrechnung ausgibt und die er zum Zweck der Authentisierung bei Anmeldung im Portal in seiner Datenbank hinterlegt.

Diese Schritte wirken sich nicht auf die Teilnehmer aus, da Vertraulichkeit und Integrität ihrer personenbezogenen Daten davon nicht betroffen sind – Lizenzschlüssel finden nur für das Anlegen neuer Accounts Anwendung. Diese Schritte sind jedoch für den Anbieter geschäftskritisch, da das Fehlen oder die Kompromittierung der Lizenzschlüssel unberechtigten Zugang Dritter zum Portal ermöglicht, ohne dass eine Abrechnung erfolgt.

Zusätzliche Vorkehrungen, die das Risiko der Kompromittierung von Lizenzschlüsseln mindern, beschreibt Abschnitt 5.2.

4.4 Konfiguration des Applikationsservers

Der Applikationsserver ist die direkte Schnittstelle des Portals zum öffentlichen Internet und damit diversen Angriffsvektoren ausgesetzt. Der Applikationsserver läuft auf einem eigenen Host bzw. einer eigenen virtuellen Maschine (VM). Die Anwendung ist der einzige Dienst auf diesem System. Der Applikationsserver ist nach jeweils aktualisierten Common Best Practises betrieben, die jeweils fortgeschrieben werden.

Zum Mindestmaß an implementierten Schutzmaßnahmen des Apache-Webservers im Applikationsserver gehören:

- Deaktiviertes Index-Browsing in der Vhost-Definition der »httpd.conf«:

```
<Directory />
  Options -Indexes
  Order allow,deny
  Allow from all
</Directory>
```

- Whitelisting der auszuliefernden Dateitypen (Bilder, Javascript, Stylesheets) und explizites Ausführen der auszuführenden Dateitypen (PHP),
- explizite Freigabe der auszuliefernden Verzeichnisse,
- Weiterleitung aller HTTP-Anfragen auf einen HTTPS-Port,

- Deaktivieren unnötiger Methoden und der Server-Signatur:

```
TraceEnable off
ServerSignature Off
ServerTokens Prod
```

- Die in Abschnitt 3.2 bei Sicherheitsfunktion 4 beschriebenen Cipher sind in der gleichen Konfigurationsdatei spezifiziert:

```
SSLProtocol -ALL +SSLv3 +TLSv1
SSLCipherSuite ALL:!aNULL:!ADH:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM
```

Der PHP-Interpreter des Applikationsservers ist durch folgende Maßnahmen gegen typische Angriffe gesichert:

- Deaktivieren der Interpreter-Signatur in der »/etc/php.d/secutity.ini«:

```
expose_php          = Off
```

- Fehler zeigt der Interpreter nicht über den öffentlichen Zugang an, sondern loggt sie in ein regelmäßig auszuwertendes Logfile:

```
display_errors      = Off
log_errors          = On
error_log           = /var/log/httpd/php_scripts_error.log
```

- Übermäßige Ressourcennutzungen sind limitiert bzw. verboten (Zeitangaben in Sekunden):

```
max_execution_time = 30
max_input_time     = 30
memory_limit       = 128M
```

- Nicht benötigte Funktionen sind deaktiviert:

```
allow_url_fopen   = Off
allow_url_include = Off
disable_functions = exec,passthru,shell_exec,system,proc_open,popen,
curl_multi_exec,parse_ini_file,show_source
```

4.5 Hostsicherheit

Eine hostbasierte Firewall blockiert alle ein- und ausgehenden Ports bis auf tcp/22, tcp/80 und tcp/443 sowie ausgehendes udp/53.

Unix-Filepermissions sorgen dafür, dass andere Prozesse mit normalen Benutzerrechten keine Dateien der Tera-Pi-Applikation und der Datenbank lesen oder schreiben bzw. verändern können. Das gilt auch wechselseitig für die beiden Teilkomponenten. Es gibt getrennte Accounts für die Laufzeitumgebung (Webserver, PHP) und die installierte Software. Der Laufzeitaccount hat keinen schreibenden Zugriff auf die Softwareinstallation.

Kernel-Parameter schützen zusätzlich vor einer Reihe von allfälligen netzbasierten Angriffen. Die folgenden Parameter, hinterlegt in »/etc/sysctl.conf« sichern gegen diese ab:

```
kernel.exec-shield=1
kernel.randomize_va_space=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.conf.all.accept_source_route=0
net.ipv4.icmp_echo_ignore_broadcasts=1
net.ipv4.icmp_ignore_bogus_error_messages=1
net.ipv4.conf.all.log_martians = 1
```

Updates aller verwendeten Softwarekomponenten und -pakete werden regelmäßig nach Bereitstellen durch die jeweiligen Hersteller eingespielt oder durch einen Vertretungsberechtigten des Anbieters freigezeichnet. Die Freizeichnung ist für Lizenznehmer und Teilnehmer einsehbar.

Der Betreiber gewährleistet für den Fall einer Störung eine Erreichbarkeit per Telefon oder E-Mail über dediziert für diesen Zweck eingerichtete Kanäle (Telefonnummer oder E-Mail-Adresse). In der Zeit von Montag bis Freitag zwischen 9 Uhr und 17 Uhr gewährleistet er eine Reaktionszeit von 60 Minuten auf solche Störungsmeldungen, klassifiziert und bestätigt ihren Eingang. Außerhalb dieser Zeiten erfolgt eine Reaktion bis 10 Uhr am nächsten Arbeitstag. Der Betreiber behebt die bestätigte Störung umgehend innerhalb der genannten Arbeitszeiten.

5 Ergänzende Maßnahmen

Folgende Maßnahmen sind optional und nicht Teil der für einen Grundschutz unerlässlichen Maßnahmen. Sie erhöhen jedoch die Sicherheit der Anwendung um zwei Aspekte: Ihrer Verfügbarkeit für den Teilnehmer und das Verfahren zur Lizenzschlüsselvergabe durch den Betreiber.

5.1 Weitergehende Vorkehrungen zur Wahrung der Verfügbarkeit

Ergänzend zu Abschnitt 3.2 Nutzung der Anwendung, Sicherheitsfunktion 4 „Wahrung der Verfügbarkeit“ lässt sich optional eine hochverfügbare Umgebung anlegen (High Availability, HA).

Die Anwendung liefe dann auf zwei unabhängigen Web-Applikationsservern mit einem HA-Failover-Lastverteiler. Wenn der aktive Server einen Ausfall hat, leitet der Lastverteiler die Anfragen auf den jeweils anderen Server um. Dieser Betriebsmodus heißt Hot-Standby-Betrieb.

Ein Teilnehmer müsste sich im Failover-Fall neu anmelden (keine Session-Integrität). Der Teilnehmer kann jedoch ohne Datenverlust weiterarbeiten. Session-Integrität, sodass für den Teilnehmer im Failover-Fall kein neuerliches Anmelden erforderlich ist, ließe sich mit zusätzlichen Maßnahmen implementieren.

Neben dem Web-Applikationsserver wären seine Datenbanken mittels eines Master-Slave-Betriebs ebenfalls redundant ausgelegt. Der aktive Master würde alle Datenänderungen, die die Web-Anwendung per INSERT- und UPDATE-Anweisungen auslöst, an einen Slave-Knoten weiterleiten, welcher die Daten noch einmal speichert. So läge auch bei einem vollständigen Ausfall des Masters ein Backup der Daten vor.

Der maximale Datenverlust beschränkte sich so auf Daten, die der Master noch nicht vollständig an den Slave übermittelt hat. Die zeitliche Differenz im Datenbestand („Seconds behind Master“) beträgt nicht mehr als 60 Sekunden, was eine Monitoring-Software überwacht.

5.2 Weitergehende Vorkehrungen zum Schutz der Lizenzschlüssel

Ergänzend zu Abschnitt 4.3 Ausgabe von Lizenzschlüsseln lassen sich optional die Verfahren zur Erzeugung, Speicherung und Überprüfung von Lizenzschlüsseln weiter absichern.

Dies geschähe erstens, indem das Programm, dass die Schlüssel erzeugt, nur in einem Pfad les- und ausführbar ist, auf den der Funktionsbenutzer der Portalsoftware keinen Zugriff besitzt. Das verhindert, dass jemand mit Zugriff auf die Ablaufumgebung des Programms unerlaubt Lizenzschlüssel erzeugt und hinterlegt.

Zweitens bekäme derjenige Datenbankbenutzer, der auf die Datenbanktabelle mit den Lizenzschlüsseln zugreift, nur SELECT-Rechte, aber keine Berechtigung für UPDATE- oder INSERT-Operationen. Das verhindert, dass jemand im Besitz des Datenbankzugangs für die Anwendung unerlaubt erzeugte Lizenzschlüssel hinterlegt.

Drittens würde die Datenbank nur die Hashwerte des Lizenzschlüssels speichern und mit der gehashten Benutzereingabe vergleichen, anstatt die Lizenzschlüssel im Klartext auszulesen. Denn sind die Lizenzschlüssel im Klartext hinterlegt, so erfolgt die Verifikation durch einen einfachen Vergleich des vom Teilnehmer eingegebenen Lizenzschlüssels mit dem in der Datenbank hinterlegten Wert. Das würde aber bedeuten, dass ein Benutzer mit den Datenbank-Zugangsrechten der Applikation auf sämtliche Lizenzschlüssel Zugriff besitzt. Ist hingegen nur der Hash-Wert der Lizenzschlüssel in der Datenbank hinterlegt, berechnet die Web-Anwendung den Hash-Wert des vom Teilnehmer eingegebenen Lizenzschlüssels und vergleicht diesen dann mit dem in der Datenbank hinterlegten Hash-Wert. Diese Maßnahme verhindert also, dass jemand auf Seiten des Betreibers unerlaubt die Lizenzschlüssel einsieht.