ITSS-E

IT-Sicherheitsstandard des FMG-Konzerns

Entwicklung von IT-Systemen

V 2.1

	Datum	Organisations-einheit	Name
erstellt	18.09.2017	IT	A. Cmarits
geprüft	25.10.2017	PEGO	C. Reiser
freigegeben	07.11.2017	GFB	Dr. M. Kerkloh, A. Gebbeken, T. Weyer

Inhalt

1 Einleitung

Der IT-Sicherheitsstandard des FMG-Konzerns (ITSS) definiert verbindliche Sicherheits-Mindestvorgaben für die Entwicklung, die Einführung, den Betrieb, die Außerbetriebnahme und die Entsorgung von informationsverarbeitenden Systemen im FMG-Konzern. Bei kritischen Geschäftsprozessen sind gegebenenfalls zusätzliche Maßnahmen zu ergreifen.

Der ITSS ist Bestandteil des FMG Informationssicherheits-Rahmenwerk (IS-Framework), aus dessen Vorgaben Sicherheitskonzepte und -maßnahmen so abgeleitet werden, dass stets ein angemessener Schutz gewährleistet ist.

Der ITSS beschreibt nicht die Umsetzung der Sicherheits-Mindestvorgaben – diese muss jeweils projekt- bzw. bereichsspezifisch erarbeitet werden.

Neben dem vorliegenden „ITSS-B: Einführung & Betrieb von IT-Systemen" existiert ein ergänzender Teil „ITSS-E: Entwicklung von IT-Systemen" als Sicherheitsstandard für Software-Entwicklung.

Der vorliegende „ITSS-E: „Entwicklung von /*IT-System*en" (ITSS-E) ergänzt den „ITSS-B: Einführung & Betrieb von /*IT-System*en" (ITSS-B) und gilt als Sicherheitsstandard für Software-Entwicklung. Die Kenntnis des ITSS-B wird vorausgesetzt. Die darin enthaltenen Vorgaben sind - sofern relevant - umzusetzen, im Konfliktfall gelten die Regeln des ITSS-E.

1.1 Geltungsbereich

Der ITSS gilt ortsunabhägig für alle Bereiche und Beteiligungsgesellschaften des FMG-Konzerns und ist für alle Auftraggeber sowie deren interne und externe Mitarbeiter, die direkt oder indirekt bei der Entwicklung, der Einführung, dem Betrieb sowie der Außerbetriebnahme und Entsorgung von Anwendungen und informationsverarbeitenden Systemen im FMG-Konzern beteiligt sind, verbindlich.

Als Entwicklung ist die systematische Herstellung von Computerprogrammen (Software) definiert. Im Gegensatz zur reinen Programmierung beinhaltet die Entwicklung den gesamten Softwareentwicklungsprozess. Neben der eigentlichen Programmierarbeit gehört dazu auch das Erarbeiten der Anforderungen an die Software sowie das Erstellen einer sicheren Softwarearchitektur und die Planung der Umsetzung.

Der ITSS-E trat am 01. April 2007 in Kraft und gilt für Neuentwicklung und Änderung von Software-Anwendungen und informationsverarbeitenden Systemen, die im Auftrag des FMG-Konzerns sowie dessen Beteiligungsgesellschaften entwickelt werden. Dies gilt analog auch für die Anpassung von Standard-Softwareprodukten. Für zum Stichtag bereits im Einsatz bzw. in Realisierung befindliche Anwendungen sind die Vorgaben nicht verpflichtend, sollten aber soweit wie möglich berücksichtigt werden.

1.2 Rollen & Verantwortlichkeiten

FMG-Informationssicherheits-Management

Der ITSS wird vom FMG-Informationssicherheits-Management (IS-Management) erarbeitet und in Kraft gesetzt. Er wird im Intranet der FMG veröffentlicht. Der ITSS wird jährlich oder bei Bedarf überprüft und den aktuellen organisatorischen Bedingungen, neuen IT-Entwicklungen und Bedrohungen der Informationssicherheit angepasst.

Die Rollen des IS-Managements der FMG sind in der Informationssicherheitsleitlinie beschrieben.

Auftraggeber und Betreiber

Zusätzliche Rollen innerhalb des ITSS sind die des Auftraggebers und des Betreibers. Mit dem Begriff „Betreiber" sind alle Betreiber (operativer Betrieb) von Informationssystemen gemeint. Mit dem Begriff „Auftraggeber" ist der für ein System Verantwortliche gemeint. Im Sinne des ITSS ist dies der Unternehmensteil, der das System in Auftrag gegeben hat, selbst betreibt oder durch einen „Betreiber" betreiben lässt.

Der Auftraggeber hat dafür zu sorgen, dass die Einhaltung des ITSS durch ihn oder die von ihm beauftragten Betreiber sichergestellt ist. Das beinhaltet, dass die vom Betreiber zu gewährleisteten ITSS-Regelungen vertraglich klar fixiert sind.

Innerhalb des betreffenden Unternehmensteils (Auftraggeber) tragen die personalverantwortlichen Führungskräfte die Verantwortung für die Einhaltung des ITSS.

Informationsverantwortlicher

Der Informationsverantwortliche ist verantwortlich für die Klassifizierung von Informationen in seinem Verantwortungsbereich. Typischerweise gehört er der ersten oder zweiten Führungsebene an oder hat bereichsübergreifende Aufgaben (z. B. Revision, Beihilfe, IS, Arbeitsschutz, Datenschutz). Bei Aufgaben- und Funktionsänderungen passt er die Berechtigungen entsprechend an. Die Verantwortlichkeit zur Klassifizierung von Informationen kann durch ihn auch an andere Mitarbeiter delegiert werden.

Projekt-/Produktverantwortlichen

Der Projekt-/Produktverantwortliche ist verantwortlich für die Umsetzung der Vorgaben des ITSS bei der Entwicklung, Einführung, Änderung sowie Aussonderung und Entsorgung von *IT-System*en sowie Anwendungen / Software.

1.3 Ausnahmeregelung

Abweichungen vom ITSS-E sind durch Auftraggeber bzw. durch den Projekt- /Produktverantwortlichen, soweit vertraglich in seiner Verantwortung, beim FMG ISec-Beauftragten mit Angabe von Gründen schriftlich mit dem bereitgestellten Standard-Formular zu melden.

Können einzelne Regeln des ITSS nachweislich technisch nicht realisiert werden, so haben hierfür keine Meldungen zu erfolgen.

1.4 Hinweis zur Nutzung

Aus dem Hauptdokument kann man durch Anklicken der Pfeile auf ggf. vorhandene Erläuterungen oder Beispiele im Anhang springen. Der Rücksprung erfolgt durch die entsprechende Funktion im Acrobat Reader.

Im Text findet sich das Symbol  ; es steht für „soweit die technischen Voraussetzungen für die Umsetzung gegeben sind“. Dies wird von den Bereichs-ISec-Beauftragten festgelegt. Sofern die Voraussetzungen nicht gegeben sind, muss dieser Sachverhalt an den FMG ISec-Beauftragten gemeldet werden.

Desweitern sind diverse technische Begriffe im Schriftsatz hervorgehoben, wenn es im Dokument befindlichen Glossar dafür ein nähere Erklärung gibt. Die entsprechende Formatierung sieht dann wie folgt aus *: kanonische Form.*

1.5 Überwachung

Die Umsetzung des ITSS wird stichprobenartig durch den FMG ISec-Beauftragten überprüft.

2 Sicherheitsarchitektur

Moderne Software-Anwendungen sind in der Regel drei- bzw. vierschichtig (bei Webanwendungen). Nicht nur das Netz und die entsprechenden Server müssen „gehärtet" sein (ITSS-B), auch die Applikationen müssen eine Absicherung erfahren, um einem ganzheitlichen Sicherheitskonzept zu folgen (ITSS-E).

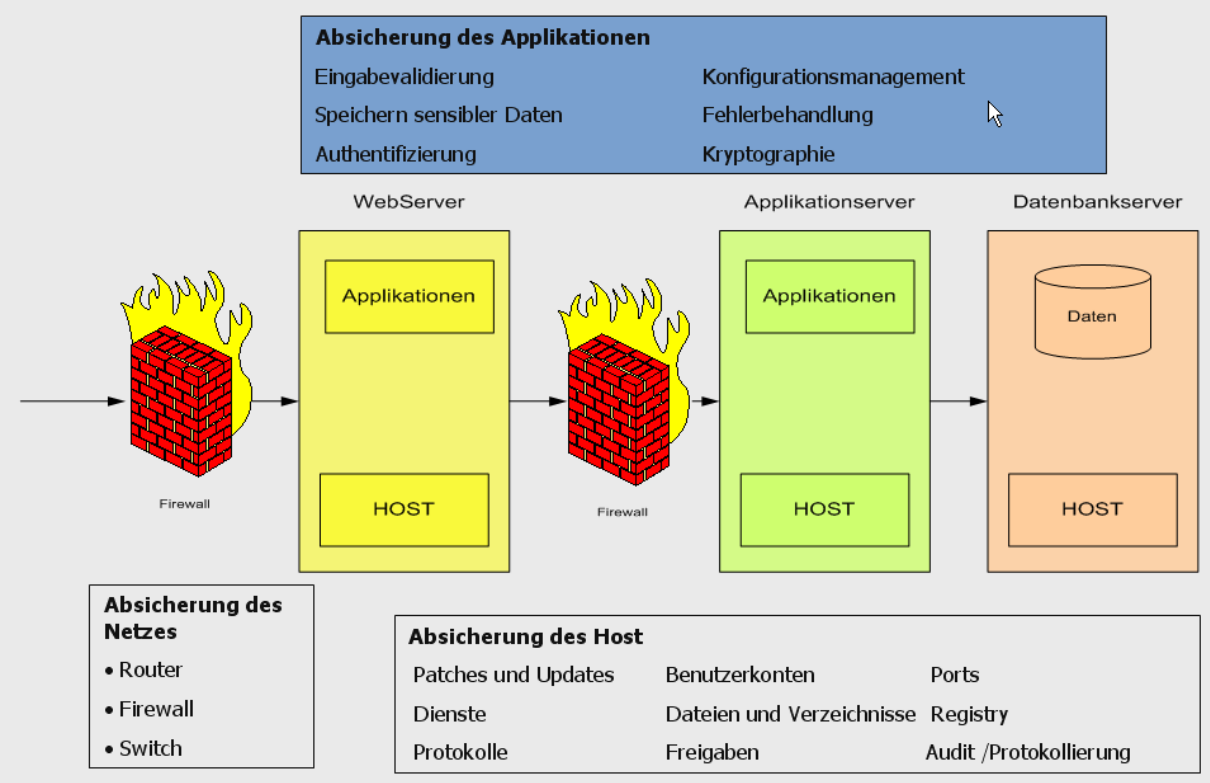

Abbildung 1: Sichere Softwarearchitektur

3 Checkliste für Sicherheitsmaßnahmen

Der Umfang von notwendigen Schutzmaßnahmen hängt vorrangig von der *Vertraulichkeit* ab, wie folgender Checkliste zu entnehmen ist:

	Vertraulichkeit			
Maßnahme	**streng vertraulich**	**vertraulich**	**dienstlich**	**öffentlich**
Authentisierung	Pflicht	Pflicht	Pflicht	Möglich
Einsatz von Kryptografie	Pflicht	Möglich	Möglich	Kein Bedarf
Eingabeprüfung				
- *Fremdsysteme*	Pflicht	Pflicht	[*]	[*]
- Benutzereingaben	Pflicht	Pflicht	Pflicht	Pflicht
- Konfigurations*daten*	Pflicht	Pflicht	[*]	[*]
Fehlerbehandlung	Pflicht	Pflicht	Pflicht	Pflicht
Berechtigungen	Pflicht	Pflicht	Möglich	Kein Bedarf
Speicherung und Übertragung Zugangsdaten (Zugangsdaten sind immer geheim)	Pflicht	N/A	N/A	N/A
Protokollierung	Pflicht	Pflicht	Pflicht	Möglich
Prozesse	Pflicht	Pflicht	Pflicht	Pflicht
Sicherheitsschulungen für Entwickler	Pflicht	Pflicht	Pflicht	Pflicht

[*] Pflicht, wenn Datenintegrität hoch sein muss

Abbildung 1: Checkliste für Sicherheitsmaßnahmen

4 Vorgaben zu den Sicherheitsmaßnahmen im Bereich Technologien

4.1 Authentisierung

Sicherheitsziel: Nur solche Authentisierungsverfahren werden angewendet, die garantieren, dass ein Benutzer oder Prozess eindeutig derjenige ist, für den er sich ausgibt.

- Bei Einsatz eines zentralen Identity-Managements auf Basis einer PKI muss dieses verwendet werden.
- Neben dieser Regelung finden sich im ITSS-B unter Punkt 2.1 weitere grundsätzliche Vorgaben.

4.2 Einsatz von Kryptografie

Sicherheitsziel: Verbesserung des Schutzes von Anwendungen, durch den Einsatz von kryptografischen Verfahren

- Bewährte Kryptografieverfahren müssen verwendet werden
  - Für die Nutzung von Kryptografie müssen die bewährten Kryptografieservices oder Bibliotheken, die im Betriebssystem oder in der Laufzeitumgebung enthalten sind, verwendet werden. Es dürfen keine eigenen Verfahren entwickeln werden.
  - *Daten* müssen so lange wie möglich verschlüsselt belassen werden.
  - Klartextdaten sind in so wenig Variablen wie möglich abzuspeichern.
  - Passende Algorithmen und erforderliche Schlüssellängen müssen entsprechend den Sicherheitsanforderungen gewählt werden. Sicherheit steigt mit der Schlüssellänge. Empfohlen ist eine *AES* basierte Verschlüsselung.

4.3 Eingabeprüfung

Sicherheitsziel: Verhindern von *SQL Injection*s, *Buffer Overflow*s und anderer *Angriffe*, die auf Benutzereingaben basieren, durch die Überprüfung aller eingegebenen *Daten*.

- *Fremdsysteme*
  Eine Validierung der *Daten*eingabe ist zu implementieren, wenn die *Daten* vertraulich sind bzw. die *Verfügbarkeit* der Anwendung betroffen ist.

- Benutzereingaben
  Diese müssen validiert werden.
  Befindet sich der Client im Vertrauensbereich des Servers reicht eine Validierung auf dem Client nach erfolgreicher Autorisierung am Server aus.
  Sollte sich der Client in einem anderen *Vertrauensbereich* befinden oder handelt es sich um eine Webanwendung, muss die Validierung prinzipiell auch am Server durchgeführt werden.
  Beispielsweise kann die Validierung über geeignete reguläre Ausdrücke erfolgen.
- Konfigurationsdaten
  Konfigurationsdaten müssen gegen ein Schema validiert werden. Sofern sie streng vertrauliche *Daten* enthalten – dies ist nach Möglichkeit zu vermeiden - ist nicht nur die Struktur sondern auch der Inhalt zu prüfen.

4.4 Fehlerbehandlung

Sicherheitsziel: Vermeiden der Ausgabe von sicherheitsrelevanten *Informationen* (implementierungsspezifische Infos, z. B. Webserver, *Daten*banken etc.) – Filterung von Fehlermeldungen.

- Benutzer dürfen nur die absolut notwendigen technischen *Informationen* in der Fehlermeldung erhalten. Inkonsistenzen, die zur unbeabsichtigten Offenlegung von *Informationen* führen oder *Denial-of-Service*-Angriffe begünstigen könnten, sind durch das Abdecken aller Ausnahmen, die in der Anwendung bzw. dem Framework und den APIs (*Application Programming Interfaces*) vorkommen, zu vermeiden.
- Ausführliche Fehlermeldungen sind im Fehlerprotokoll festzuhalten und sollten eine Fehler-ID enthalten. Mit dieser Fehler-ID kann eine Fehlermeldung, welche dem Benutzer angezeigt wird die detaillierte (technische) Fehlermeldung referenzieren.
- Vertrauliche *Daten*, wie z. B. Passwörter, dürfen nicht protokolliert werden.
- Die Behandlung und Verbreitung von Fehlern und Ausnahmen innerhalb der Anwendung ist sorgfältig zu planen.

4.5 Berechtigungen

Sicherheitsziel: Sicherstellen, dass über Berechtigungen festgelegt ist, was ein authentisierter Benutzer tun kann und auf welche Ressourcen er zugreifen darf.

- Es müssen Benutzerkonten mit den geringsten möglichen Privilegien und Zugriffsrechten auf Systemebene verwendet werden. Dazu gehören beispielsweise Zugriffsrechte auf Dateien, Ordner, Geräte im Netz, Datenbankobjekte oder Ereignisprotokolle.
- Ressourcen müssen unter Berücksichtigung von Performanceaspekten schnellstmöglich wieder freigegeben werden.
- Besonders *anonyme Internet-Benutzerkonten* sind mit Zugriffsrechtsbeschränkungen, die den Zugriff speziell für anonyme Benutzer einschränken, zu sichern.

4.6 Speicherung und Übertragung von Zugangsdaten

Sicherheitsziel: Treffen von besonderen Schutzmaßnahmen, wenn in einer Anwendung geheime *Daten*, etwa Passwörter, Anmeldeinformationen, Datenbankverbindungsstrings oder andere Konfigurationsdaten gespeichert, übertragen oder verarbeitet werden.

Jeder, der Zugriff auf den Server hat, kann die *Informationen* lesen. Um zu prüfen, ob ein Benutzer vertrauliche bzw. streng vertrauliche *Daten*, beispielsweise sein *Passwort*, kennt, kann man den *Hash-Wert* des *Passwort*s speichern, den *Hash-Wert* aus der Benutzereingabe des *Passwort*s berechnen und die Werte vergleichen.

- Eine Speicherung von vertraulichen bzw. streng vertraulichen *Daten* (z. B. Datenbank-Kennwörtern) im Klartext im Code bzw. Konfigurationsdateien ist nicht zulässig.

Ein Angreifer, der Zugriff auf die ausführbaren Dateien hat, kann String-Konstanten aus den kompilierten Dateien extrahieren, auch wenn der Quellcode nicht auf dem Anwendungsserver sichtbar ist.

- Vertrauliche bzw. streng vertrauliche *Daten* dürfen nicht auf Clientrechnern abgelegt werden. Z.B. bei Verlust oder Diebstahl hat ein potentieller Angreifer beliebig viel Zeit, die sensiblen *Daten* zu extrahieren. Für mobile Clients muss eine Festplattenverschlüsselung implementiert werden.

- Eine Übermittelung von vertraulichen bzw. streng vertraulichen *Daten* ist über sichere Transportwege durchzuführen (Beispiel: Eine Übermittelung über Email ist nur gestattet, wenn diese Mails mit *PGP* verschlüsselt werden). Entweder die *Daten* oder der Übertragungsweg ist zu verschlüsseln.
  Ein verbreitetes Verfahren ist die Verwendung von *SSL* zwischen Client und Web-Server.
- Für private Schlüssel kann eventuell dedizierte Hardware für Key-Management in Betracht gezogen werden. Dies ist insbesondere im Rahmen einer *PKI-Struktur* sinnvoll.

- Wenn eine Anwendung abstürzt, dürfen keine vertraulichen *Daten* preisgegeben werden, die einem Angreifer eventuelle Schwachstellen in der Anwendung aufzeigen würden (z. B. detaillierte Versionsnummern, Patchstände etc.)

4.7 Protokollierung sicherheitsrelevanter Aktivitäten (Revision)

Sicherheitsziel: Protokollierung sicherheitsrelevanter Aktivitäten auf jedem System über alle Schichten hinweg zum Feststellen und Nachweis von Missbrauch oder aufgetretenen Ereignissen.

- Alle Aktivitäten und Transaktionen der folgenden Auflistung sind über das gesamte System hinweg zu protokollieren.
  - Anmeldeversuche

- Verwendung administrativer Funktionen (z. B. *Impersonation*)

- Es sind Protokollierungsmechanismen für eigenen Code zu verwenden und die entsprechenden Funktionen der Produkte, die eingesetzt werden (Web-Server, Anwendungsserver, Datenbank, Betriebssystem usw.), zu aktivieren.
- Alle Protokolle sind mit Zeitstempeln – wenn möglich zumindest sekundengenau - zu versehen, um die spätere Analyse zu vereinfachen und die Synchronisation mehrerer Protokolle, z. B. für Datenbank und Anwendung, zu ermöglichen.

4.8 Kennzeichnung vertraulicher und streng vertraulicher Informationen

Sicherheitsziel: Sichere Behandlung von vertraulichen Informationen durch den Empfänger

- Beim Ausdrucken von Informationen muss eine explizite Kennzeichnung möglich sein.
- Beim Datenexport müssen die exportierten Daten explizit gekennzeichnet werden können.

5 Vorgaben im Bereich Prozesse

5.1 Vorphase (SLM)

Die Key-Account-Manager stellen mit dem Kunden den Schutzbedarf fest. Das ISM führt im Rahmen des Risikomanagementprozesses die formale Prüfung durch.

5.2 Projektinitierung/Projektplanung

Der Projektleiter führt eine *Gefährdungsanalyse* mit *Bedrohung*en und Schwachstellen durch und entwirft eine darauf abgestimmte Architektur. Die IS-Spezialisten und Bereichs-ISec-Beauftragten unterstützen ihn dabei.

5.3 Realisierung

Der Projektleiter bzw. der Entwickler führt ein Self-Assessment bzgl. ITSS für *Betreiber* und Entwickler durch und stellt, falls notwendig, Ausnahmeanträge zum ITSS. Die IS-Spezialisten und Bereichs-ISec-Beauftragten unterstützen ihn dabei. Der Projektleiter berücksichtigt die Sicherheitsaspekte im Benutzer- und Betriebs-Handbuch. Der *Betreiber* führt die Prüfung durch.

5.4 Projektabschluss

Der Projektleiter berücksichtigt die Sicherheitsaspekte im Abschlussbericht. Der Projektservice führt die formale Prüfung durch.

6 Vorgaben im Bereich Menschen

6.1 Sicherheitsschulungen für Software-Entwickler

Sicherheitsziel: Durchführen von Software-Entwicklungsaufgaben nach aktuellem technischen und sicherheitstechnischen Standard.

- Anforderung an den Prozess Personalentwicklung
  - Entwickler bzw. Projektleiter müssen die nötige Fachkunde zur Ausführung ihrer Tätigkeiten besitzen und sind gemäß aktuellen technischen Standards im Bereich Secure Coding zu schulen.
  - Entwickler bzw. Projektleiter sind zum FMG-ITSS und der FMG IT-Nutzungsrichtlinie zu schulen.

Anhang A – Allgemeine Beispiele

A 4.2 Einsatz von Kryptografie

Algorithmus - Schlüssellänge - Einsatz

- AES 128-256 Bit (16-32 Bytes): bevorzugt zu verwenden
  - Langsame, aber starke Verschlüsselung großer Datensätze
- TripleDES 128-Bit oder 192-Bit (16 oder 24 Bytes)
  - Verschlüsselung großer *Daten*sätze
- Rivest, Shamir and Adleman (RSA) oder Digital Signature Algorithm (DSA) 384-16, 384 Bit (48-2,048 Bytes)
  - Digitale Signaturen
  - Secure Hash Algorithm (SHA)1.0 Hashing
  - Hash-based Message Authentication
  - Code (HMAC) SHA1.0
  - Keyed Hashing

A 4.3 Eingabeprüfung

Grundsätzlich müssen Eingabe*daten* in kanonischer Form vorliegen, bevor sie validiert werden.

- Beispiel: Der folgende String enthält Datei- und Pfadnamen in ihrer Grundform:
  - c:\data\anyfile.htm

Erlaubt die Applikation auch Pfadangaben, bedeutet dies ein potentielles *Risiko*.

Der Anwender kann durch unten dargestellte Eingaben dieselbe Datei öffnen, aber auch das erlaubte Dateiverzeichnis verlassen.

- Folgende Eingabe wären auch gültig:
  - c:\data\sub\..\anyfile.htm
  - c:\ data\ anyfile.htm
  - ..\anyfile.htm c%3A%5Cdata%5Csub%5C%2E%2E%5-Canyfile.htm

Besser wäre es, keine Pfadangaben zuzulassen.

Anmerkung: Dieses Bespiel ist insbesondere im Webumfeld relevant.

A 4.5 Berechtigungen

Folgende Möglichkeiten gibt es, ein Berechtigungskonzept umzusetzen. Je nach Anwendungsfall muss entschieden werden, welches Konzept umgesetzt wird.

- Zugriff über Rollen
  - Die Entscheidung über das Zugriffsrecht hat auf der Rolle des Aufrufenden, der ein Benutzer oder ein System sein kann, zu basieren.
  - Aufrufer sind auf Anwendungsebene (Middle-Tier) auf Rollen abzubilden.
  - Der Zugriff auf Programm-Module hat von der Zugehörigkeit zu bestimmten Rollen abzuhängen.
  - Der nachfolgende Zugriff auf Ressourcen hat mit Hilfe einer vorgegebenen Anzahl von Identitäten, die in Verbindung mit der Rollenzugehörigkeit des Aufrufers stehen zu erfolgen.
    Der Vorteil dieser Methode liegt darin, dass die Berechtigungen, die jeder Rolle zugeordnet sind, einer festen Anzahl von Datenbankkonten zugeordnet werden können. Connection Pooling kann nach wie vor genutzt werden.

- Zugriff über „*Impersonation*“
  Ein feingranularer Ansatz kann über das Prinzip der „*Impersonation*“ erzielt werden. Ressourcen werden mit Hilfe des Sicherheitskontexts einem Betriebssystembenutzer zugeordnet. Dieser Ansatz kann sinnvoll sein, wenn die Anwendung Zugriffsrechte primär auf benutzerspezifische Ressourcen vergibt. Der Vorteil des Verfahrens besteht darin, dass (abhängig vom Host-Betriebssystem und der Entwicklungs-/Laufzeitumgebung) eine Revision auf Betriebssystemebene möglich ist, weil der Sicherheitskontext hier bekannt ist. Ein Nachteil liegt in der geringen Skalierbarkeit, weil ein effektives Connection Pooling für den *Daten*bankzugriff nicht möglich ist.

  - *Impersonation* wird daher meist in Intranet-basierten Anwendungen, die in ihrer Reichweite begrenzt sind, eingesetzt.

- Zugriff über eine Pool-Identität.
  Die am wenigsten granulare, aber am besten skalierbare Methode ist die Steuerung des Ressourcenzugriffs über eine Pool-Identität. Unabhängig vom tatsächlichen Benutzer sind der Pool-Identität immer dieselben Berechtigungen zugewiesen. Das primäre Berechtigungsverfahren läuft im Middle-Tier des Systems auf der Basis von Rollen ab.
  - Hier werden Benutzer mit gleichen Privilegien in einer Sammelrolle zusammengefasst.
  - Dieses Verfahren unterstützt Datenbank-Connection-Pooling.

7 Glossar

Angriff	Bewusster oder absichtlicher Versuch, eine Verwundbarkeit in einem System auszunutzen oder zu suchen.
Anonyme InternetBenutzerkonten	Dies sind anonyme Benutzer, welche nicht der internen Benutzerverwaltung des Betreibers unterliegen, und damit nicht authentifiziert werden können.
Application ProgrammingInterface (API)	Eine Programmierschnittstelle ist eine Schnittstelle, die von einem Softwaresystem anderen Programmen zur Anbindung an das System zur Verfügung gestellt wird. Oft wird dafür die Abkürzung API (für engl. application programming interface, deutsch: Schnittstelle zur Anwendungsprogrammierung) verwendet.
Authentifizierung	Überprüfen einer Identität
Authentizität	Überprüfte und bestätigte zweifelsfreie Herkunft bzw. Identität
Autorisierung	Erteilte Berechtigung
Bedrohung	Umstand, der direkt oder indirekt zu einem Schaden oder Sicherheitsverlust führen kann
Betreiber	Anbieter von IT-Dienstleistungen, IT-Diensten, IT-Systemen oder IT-Bestandteilen technischer Systeme, die im Auftrag des FMG-Konzerns Leistungen erbringen
Buffer Overflows	Buffer Overflows gehören zu den häufigsten Sicherheitslücken in einer Software, die sich u. a. über das Internet ausnutzen lassen. Im Wesentlichen werden bei einem Pufferüberlauf durch Fehler im Programm zu große Datenmengen in einen dafür zu kleinen Speicherbereich geschrieben, wodurch dem Ziel-Speicherbereich nachfolgende Informationen im Speicher überschrieben werden.
Daten	Gebilde aus Zeichen zur Abbildung von Informationen und Medien (Sprache, Bilder), die gespeichert oder verarbeitet werden.
Denial-of-Service	Als Denial of Service (DoS, etwa Dienstverweigerung) bezeichnet man einen Angriff auf einen Host (Server)

	mit dem Ziel, einen oder mehrere seiner Dienste arbeitsunfähig zu machen. In der Regel geschieht dies durch Überlastung. Erfolgt der Angriff koordiniert von einer größeren Anzahl anderer Systeme aus, so spricht man von Verteilte Dienstblockade bzw. DDoS (Distributed Denial of Service).
Digitale Zertifikate	Digital signierte elektronische Daten, die sich zum Nachweis der Echtheit von Objekten verwenden lassen.
Entwickler	Entwickler von Anwendungen und informationsverarbeiteten Systemen die im Auftrag des FMG-Konzerns Leistungen erbringen.
Externe Systeme	Systeme die nicht innerhalb der Betreiber-IT verwendet werden, z. B. Systeme in der DMZ, Partnersysteme
Gefährdungsanalyse	Jedes System bietet durch Schnittstellen eine potentielle Angriffsfläche für Angriffe von außen. Die Schnittstellen müssen auf Schwachstellen überprüft werden. Dabei wird auch bewertet, wie wahrscheinlich ein bestimmter Angriffstyp ist und wie hoch der eventuelle Schaden sein kann.
Fremdsysteme	Systeme, die nicht unter der Verwaltung des Betreibers stehen, z.B. private PCs, Rechner von Wartungstechnikern etc.
Hash-Wert	Ein Hash-Wert bzw. Streuwert ist ein skalarer Wert, der aus einer komplexeren Datenstruktur (Zeichenketten, Objekte, ...) mittels einer Hash-Funktion berechnet wird. Ein Hash-Wert wird auch als Fingerprint bezeichnet. Denn wie ein Fingerabdruck einen Menschen nahezu eindeutig identifiziert, ist ein Hashwert eine nahezu eindeutige Kennzeichnung einer übergeordneten Menge. Hash-Werte dienen beispielsweise als Schlüssel für Tabellen, um assoziative Arrays (Hashtabellen) zu implementieren.
Identität	repräsentiert ein Objekt/Subjekt (Person, IT-System) mit verschiedenen Attributen.
Impersonation	Impersonation ermöglicht aus dem Quellcode heraus ein Programm unter einem anderem Benutzerkontext ausführen zu lassen.

Informationen	Im Rahmen von Geschäftsabläufen interpretierte Daten
Integrität	Eigenschaft, dass IT-Systeme und Daten, die genutzt bzw. gespeichert, verarbeitetet oder übertragen werden, ausschließlich zulässigen Veränderungen unterliegen
IT	Abkürzung für Informationstechnologie
IT-Sicherheit	Fachgebiet, das sich mit der Sicherheit von IT-Prozessen befasst. IT-Sicherheit entsteht aus einem sinnvollen Zusammenspiel von technischen und organisatorischen Maßnahmen
IT-Sicherheitsmanagement	Managementaufgabe, die sich mit der Sicherheit von IT-Prozessen und der Erfassung und Minimierung von Risiken befasst
IT-Systeme	Informationstechnische Systeme: Mit Informatikmitteln (Computer, Datenbanken, Programmen etc.) realisiertes Informationssystem zur Unterstützung eines Geschäftssystems.
IT-Verbund	Nach BSI-Grundschutz: Gesamtheit von infrastrukturellen, organisatorischen, personellen und technischen Komponenten, die der Aufgabenerfüllung in einem bestimmten Anwendungsbereich der Informationsverarbeitung dienen. Ein IT-Verbund kann dabei als Ausprägung die gesamte IT einer Institution oder auch einzelne Bereiche, die durch organisatorische Strukturen (z. B. Abteilungsnetz) oder gemeinsame IT-Anwendungen (z. B. Personalinformationssystem) gegliedert sind, umfassen. (FMG-spezifisch, z. B. Gepäckförderanlage, E-Mail-System).
Kanonische Form	Das kanonische Format der Informatik bezeichnet eine allgemeingültige und eindeutige Bezeichnung eines Datensatzes.
	Beispielsweise führen Sicherheitsentscheidungen, die auf einen Dateinamen basieren, potentiell zu Schwierigkeiten, da es mehrere mögliche Repräsentationen des Namens geben kann. • 8.3 Konvention • NTFS Alternate Data Streams • Directory Traversal

	• Case-Insensitive Dateinamen
Nachweisbarkeit	Verbindlicher Nachweis, so dass an Veränderungen Beteiligte über keinerlei Mittel verfügen, ihre Beteiligung zu bestreiten
Obfuskation	Unkenntlichmachung von Quellcode in den ausgelieferten Programmdateien, um eine eventuelle Rückentwicklung des Quellcodes zu verhindern.
Passwort	Zeichenkette, die als Authentifizierungsinformation dient
PGP	Pretty Good Privacy (PGP) ist ein von Phil Zimmermann entwickeltes Programm zur Verschlüsselung von Daten. Es benutzt das sog. Public Key-Verfahren, d.h. es gibt ein eindeutig zugeordnetes Schlüsselpaar: Einen öffentlichen, mit dem jeder die Daten für den Empfänger verschlüsseln kann, und einen geheimen privaten Schlüssel, den nur der Empfänger besitzt und der durch ein Kennwort geschützt ist. Nachrichten an einen Empfänger werden mit seinem öffentlichen Schlüssel verschlüsselt und können dann nur durch den privaten Schlüssel des Empfängers entschlüsselt werden. Diese Verfahren werden auch asymmetrische Verfahren genannt, da Sender und Empfänger zwei unterschiedliche Schlüssel verwenden.
PKI-Struktur	Als Öffentlicher-Schlüssel-Infrastruktur bzw. Public-Key-Infrastruktur bezeichnet man in der Kryptologie und Kryptografie ein System, welches es ermöglicht, digitale Zertifikate auszustellen, zu verteilen und zu prüfen. Die innerhalb einer PKI ausgestellten Zertifikate sind meist auf Personen oder Maschinen festgelegt und werden zur Absicherung computergestützter Kommunikation verwendet. Als wesentliche Bestandteile einer (minimalen) PKI sind sicherzustellen: • Digitale Zertifikate • Zertifizierungsstelle (Certificate Authority, CA) • Registrierungsstelle (Registration Authority, RA) • Zertifikatsperrliste (Certificate Revocation List) • Verzeichnisdienst

	• Validierungsdienst:
Registrierungsstelle	Organisation, bei der Personen, Maschinen oder auch untergeordnete Zertifizierungsstellen Zertifikate beantragen können. Diese prüft die Richtigkeit der Daten im gewünschten Zertifikat und genehmigt den Zertifikatsantrag, der dann durch die Zertifizierungsstelle signiert wird
Restrisiko	Risiko, das zwar erkannt, aus technischen, organisatorischen oder finanziellen Gründen nicht, oder nur mit unverhältnismäßigem Aufwand, beseitigt werden kann
Risiko	Möglichkeit, einen Schaden zu erleiden. Im IT-Verbund ergeben sich Risiken aufgrund der Tatsache, dass real existierende Bedrohungen auf Verwundbarkeiten treffen können.
Sicherheit	Zustand des Geschütztseins vor Gefahr oder Schaden
SQL Injections	SQL-Injektion (engl. SQL Injection) bezeichnet das Ausnutzen einer Sicherheitslücke in Zusammenhang mit SQL-Datenbanken. Diese entsteht bei mangelnder Maskierung oder Überprüfung von Funktionszeichen. Der Angreifer versucht über die Anwendung, die den Zugriff auf die Datenbank bereitstellt, eigene Datenbankbefehle einzuschleusen. Sein Ziel ist es dabei, Kontrolle über die Datenbank oder den Server zu erhalten
SSL	Secure Sockets Layer (SSL) ist ein Verschlüsselungsprotokoll für Datenübertragungen im Internet.
Validierungsdienst	Ein Dienst, der die Überprüfung von Zertifikaten in Echtzeit ermöglicht
Vertrauensbereich	Unterliegt das System oder eine Software-Anwendung nicht mehr der Kontrolle eines Betreibers, ist es außerhalb seines Vertrauensbereiches.
Verfügbarkeit	Gewährleistung der Durchführung genehmigter Zugriffe und Veränderungen auf Daten und Systemen innerhalb einer definierten Zeit
Vertraulichkeit	Gewährleistung, dass nur berechtigten Nutzern der Zugang zu einem definierten Zweck möglich ist

Verzeichnisdienst	Ein durchsuchbares Verzeichnis, welches ausgestellte Zertifikate enthält, meist ein LDAP-Server, seltener ein X.500-Server
Zertifikatsperrliste	Listen mit zurückgezogenen, abgelaufenen und für ungültig erklärten Zertifikaten
Zertifizierungsstelle	Organisation, welche ein CA-Zertifikat bereitstellt und die Signatur von Zertifikatsanträgen übernimmt.

Änderungshistorie

Org.-einheit	Bearbeiter	Datum	Änderung	Alte Versionsnr.
ITS	A. Cmarits	30.11.2006	• Finale Version	
ITS	A. Cmarits	06.11.2008	• Entfernung von Inkonsistenzen zum ITSS-B; Einfügen Kapitel 4.8	1.0
IT	A. Cmarits	18.09.2017	• Anpassung an FMG Layout	2.03