D03-QM Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen der „PersoApp“

Editor: Siniša Đukanovi (Fraunhofer SIT)

Prüfung: Sven Wohlgemuth (TU Darmstadt/CASED)

Typ: [VORGEHENSBESCHREIBUNG]

Projekt: „PersoApp“

Version: 1.0

Datum: 19. Juni 2013

Status: [FREIGABE]

Klasse: [ÖFFENTLICHKEIT]

Datei: D03-QM Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen der PersoApp.docx

Zusammenfassung

Das Dokument legt den Prozess dar, der zur Entwicklung sicherer Open-Source-Software im Rahmen des PersoApp-Projekts Anwendung finden wird. Die Grundlage hierfür bilden die fünf fundamentalen Phasen moderner Softwareentwicklungsprozesse, die um dedizierte Sicherheitsaktivitäten erweitert wurden. Im Fortlauf werden diese Aktivitäten detailliert beschrieben und dienen über den gesamten Projektverlauf als handlungsleitend für den Entwurf und die Entwicklung sicherer Softwaremodule.

Konsortialleitung:
Prof. Dr. Ahmad-Reza Sadeghi und Dr. Sven Wohlgemuth
System Security Lab, TU Darmstadt/CASED, Mornewegstr. 32, 64293 Darmstadt
Tel.: +49-6151-16-75561 Fax: +49-6151-16-72135
E-Mail: persoapp@trust.cased.de Web: https://www.persoapp.de

Nutzungslizenz

Mitglieder des Konsortiums

1. **AGETO Service GmbH**, Deutschland
2. **Center for Advanced Security Research Darmstadt (CASED)**, Deutschland
3. **Fraunhofer Institut für Sichere Informationstechnologie (SIT)**, Deutschland
4. **Technische Universität (TU) Darmstadt**, Deutschland

Versionen

Version	*Datum*	*Beschreibung (Editor)*
0.1	2013-06-14	Erste Version (Fraunhofer SIT)
0.3	2013-06-13	Review von Hr. Kubieziel (AGETO)
1.0	2013-06-19	Finale Version (Fraunhofer SIT)

Autoren

Autoren	*Beiträge*
Philipp Holzinger (Fraunhofer SIT)	Anwendungsbereich, Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen
Stefan Triller (Fraunhofer SIT)	Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen

[1] http://creativecommons.org/licenses/by-sa/3.0/

Inhaltsverzeichnis

1 Ziel und Zweck des Dokumentes

Das Dokument legt den Prozess dar, der zur Entwicklung sicherer Open-Source-Software im Rahmen des „PersoApp“-Projekts Anwendung findet. Die Grundlage hierfür bilden die fünf fundamentalen Phasen moderner Softwareentwicklungsprozesse, die um dedizierte Sicherheitsaktivitäten erweitert wurden. Im Fortlauf werden diese Aktivitäten detailliert beschrieben und dienen über den gesamten Projektverlauf als handlungsleitend für den Entwurf und die Entwicklung sicherer Softwaremodule.

2 Anwendungsbereich

Der in diesem Dokument beschriebene Prozess deckt alle Phasen der Systementwicklung im Rahmen des Open-Source-Community-Projekts zur Entwicklung von Open-Source-Software in „PersoApp“ ab und sollte daher in diesem Rahmen angewendet werden. Konkret handelt es sich hierbei um die Analyse, Modellierung, Implementierung, Validierung und Freigabe von „PersoApp“-Softwaremodulen.

3 Abkürzungen und Begriffsdefinitionen

In der folgenden Tabelle werden die Abkürzungen erläutert, die im Verlauf dieses Dokumentes Verwendung finden. Bei Bedarf kann sie erweitert werden.

Begriff	*Abkürzung*	*Definition*
Datenflussdiagramm	DFD	Das Datenflussdiagramm (engl. data flow diagram) stellt graphisch dar, wie Daten in einem ggf. verteilten IT-System verwendet und bereitgestellt werden. Es besteht grundsätzlich aus den folgenden Elementen: • Data Flows • Data Stores • Processes • Interactors Im Zuge einer Sicherheitsanalyse kann die Darstellung erweitert werden, um Privilegien- und Maschinengrenzen anzuzeigen. Vgl. (Microsoft, 2006)

Notiz: Der Titel des vorliegenden Dokumentes „D03-QM Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen der „AusweisApp“ sowie die Überschrift aus Kapitel 5 sind dem Angebotsschreiben entsprechend übernommen worden. Tatsächlich handelt es sich hier aber um die Entwicklung von „*PersoApp“*-Softwaremodulen, die mit der „AusweisApp“ der OpenLimit SignCubes AG in keinem direkten Zusammenhang stehen. Im Fortlauf des Dokumentes wird daher der korrekte Name „PersoApp“ verwendet. Der Begriff „AusweisApp“ in diesem Dokument bezieht sich nur auf das Softwareprodukt der OpenLimit SignCuges AG, wenn dies explizit angegeben wurde.

4 Zuständigkeiten und Verantwortlichkeiten

Das Fraunhofer SIT als Qualitäts- und Sicherheitsverantwortlicher in diesem Projekt beschreibt in diesem Dokument den Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen der „PersoApp“. Dieser Prozess wird von allen Projektteilnehmern angewendet, die an der Analyse, Modellierung, Implementierung, Validierung und/oder Freigabe beteiligt sind. Das vorliegende Dokument wurde vom Fraunhofer SIT ausgearbeitet, die anderen Konsortialpartner wurden in Form eines Reviews mit der Möglichkeit für Kommentare und Änderungswünsche indirekt daran beteiligt.

5 Entwurfs- und Entwicklungsprozess von sicheren Open-Source-Softwaremodulen der „PersoApp“

Im Rahmen des Open-Source-Projekts „PersoApp“ nehmen Sicherheit und Transparenz eine entscheidende Rolle ein. Die Entwicklung sicherer Software ist jedoch keinesfalls trivial und bedarf der Erweiterung gängiger Softwareentwicklungsprozesse. Zu diesem Zweck werden in diesem Kapitel sicherheitsrelevante Aktivitäten vorgestellt, die im Rahmen der fünf grundlegenden Phasen der Softwareentwicklung durchzuführen sind. Hierdurch soll über den gesamten Projektverlauf und darüber hinaus durch eine systematische Vorgehensweise ein akzeptables Sicherheitsniveau gewährleistet werden können.

Nachfolgend wird zunächst ein Überblick über den gesamten Entwicklungsprozess gegeben, um zu verdeutlichen, wie sich die Sicherheitsaktivitäten in einen typischen Softwareentwicklungsprozess integrieren lassen. Anschließend werden die einzelnen Aktivitäten detailliert vorgestellt und erläutert.

Der Prozess im Überblick

In Abbildung 1 wird der gesamte Entwurfs- und Entwicklungsprozess im Überblick gezeigt. Hierbei wurde sich an den fünf grundlegenden Phasen der Softwareentwicklung orientiert: Analyse, Modellierung/Entwurf, Implementierung, Validierung/Review und Freigabe.

Neben gängigen Entwurfs- und Entwicklungstätigkeiten, wie z.B. Softwaredesign und Unit-Testing, wurde der Prozess um sicherheitsspezifische Prozessschritte erweitert: Bedrohungsanalyse, Risikoanalyse, Sicherheitsmodellierung, Sicherheitsarchitektur und Sicherheitstests.

Jede Prozessphase und jeder sicherheitsspezifische Prozessschritt umfassen mehrere Sicherheitsaktivitäten, die in den nachfolgenden Unterkapiteln detailliert vorgestellt werden.

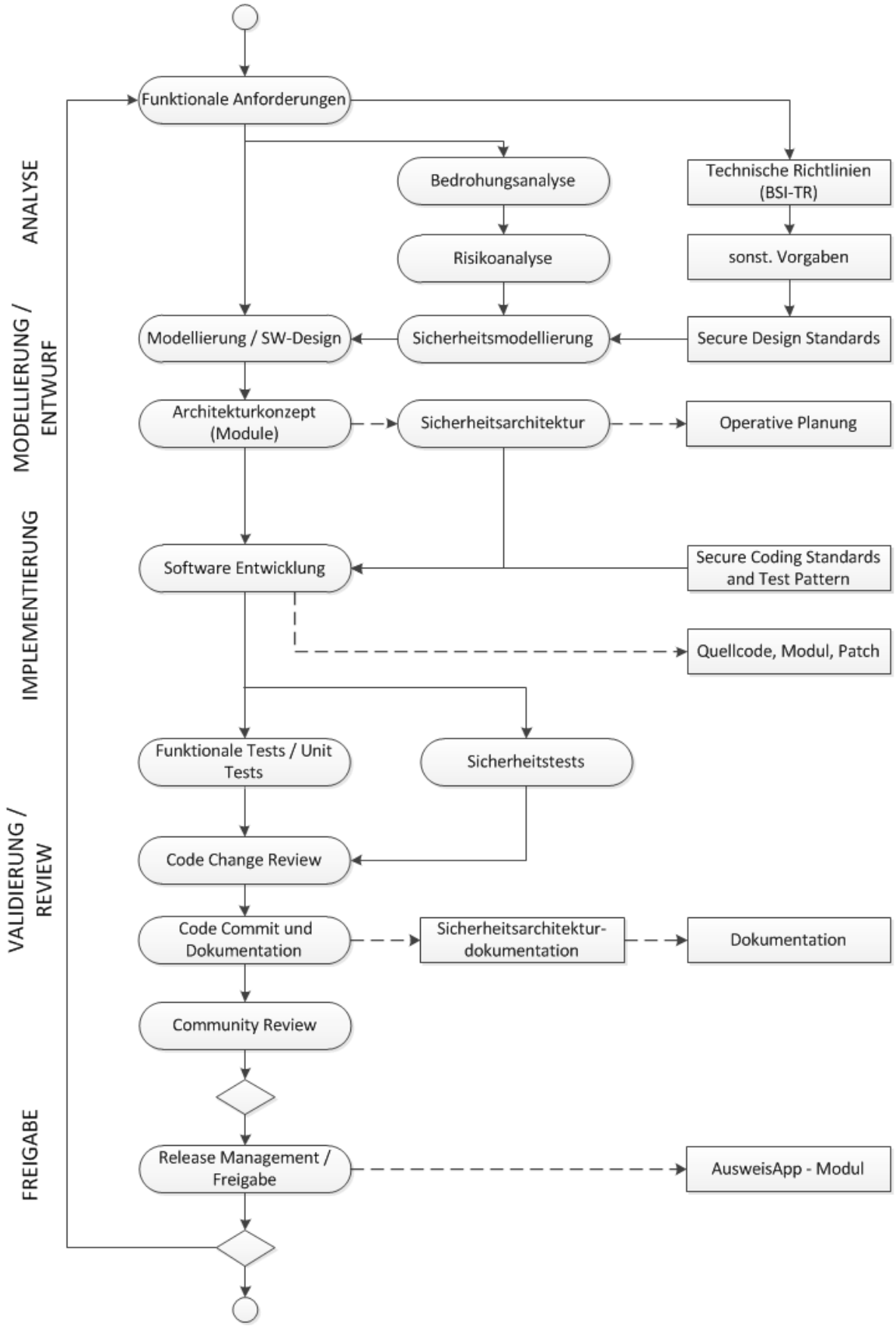

Abbildung 1 - Der gesamte Entwurfs- und Entwicklungsprozess im Überblick.

5.1 Analyse

Diese erste Phase des Entwurfs- und Entwicklungsprozesses widmet sich typischerweise der Erfassung von Anforderungen. Im Kontext der sichereren Softwareentwicklung wurden die Prozessschritte „Bedrohungsanalyse“ und „Risikoanalyse“ ergänzt, die aus den folgenden Aktivitäten bestehen:

5.1.1 Beschreibung der Software

Zuordnung zu Prozessschritt: Bedrohungsanalyse, Risikoanalyse

Das Fundament sowohl der Bedrohungsanalyse, als auch der Risikoanalyse, stellt eine kurze Beschreibung der zu betrachtenden Software dar. Dies umfasst die folgenden Punkte:

- Namensbezeichnung
- Versionsnummer
- Beschreibung der Domäne, innerhalb derer das System eingesetzt werden soll
- Beschreibung des Bestimmungszwecks der Software (ggf. anhand von Use-Case-Diagrammen)
- Erläuterung der grundlegenden Systemarchitektur
- Beschreibung der eingesetzten Technologien
- Erstellung und Erläuterung eines Kontextdiagramms (ggf. in Form eines Datenflussdiagramms, DFD)

5.1.2 Beschreibung der Systemnutzer

Zuordnung zu Prozessschritt: Bedrohungsanalyse, Risikoanalyse

Das Ziel dieser Aktivität ist die Beschreibung aller Nutzergruppen, die die „PersoApp“ unmittelbar verwenden. Dies umfasst sowohl die lokale Verwendung (etwa auf einem PC), als auch den entfernten Zugriff über das Internet (etwa durch einen eID-Service-Provider). Durch die Beschreibung der Nutzergruppe soll dargelegt werden:

- welcher Funktionalität sich der jeweilige Nutzer bedient
- welche Zugriffsrechte der jeweilige Nutzer hat

5.1.3 Beschreibung der Stakeholder

Zuordnung zu Prozessschritt: Bedrohungsanalyse, Risikoanalyse

Um die schützenswerten Assets der PersoApp identifizieren und bewerten zu können, bedarf es einer vollständigen Auflistung aller Stakeholder, die ein berechtigtes Interesse an dem System haben. Typische Stakeholder können beispielsweise sein:

- Benutzergruppen
- Dienstleister (im vorliegenden Kontext z.B. eID-Service-Provider)
- Projektpartner
- Auftraggeber

- etc ...

5.1.4 Beschreibung der Assets

Zuordnung zu Prozessschritt: Risikoanalyse

Auf Grundlage der Ergebnisse von 5.1.1, 5.1.2 und 5.1.3 gilt es in dieser Aktivität alle Assets zu identifizieren, die im Zuge der Entwicklung einer Sicherheitsarchitektur zu berücksichtigen sind. Als Asset gilt hierbei alles, das für mindestens einen Stakeholder eine signifikante Wertigkeit hat. Es kann sich sowohl um physikalische Objekte, als auch um abstrakte Werte wie z.B. Daten, oder Glaubwürdigkeit handeln.

Die Identifikation relevanter Assets kann in Zusammenarbeit mit den Stakeholdern erfolgen. Für die Dokumentation der Ergebnisse bietet sich eine tabellarische Auflistung der Assets an, die für jeden Eintrag eine kurze textuelle Beschreibung enthält sowie eine Zuordnung zu den entsprechenden Stakeholdern.

Beispielhaft könnte dies wie folgt aussehen:

Asset-ID	*Asset-Bezeichner*	*Beschreibung*	*Zuordnung: Stakeholder*
A1	PIN	Individuelle, geheime PIN des Ausweisinhabers (…)	Benutzer, (…)
…	…	…	…

Abbildung 2 - Beispielhafte Darstellung einer Tabelle für die Erfassung von Assets.

5.1.5 Festlegung von Sicherheitszielen für Assets

Zuordnung zu Prozessschritt: Risikoanalyse

Auf Grundlage von 5.1.4 werden in dieser Aktivität für jedes Asset spezifische Sicherheitsziele festgelegt. Bei den Sicherheitszielen handelt es sich um (National Institute of Standards and Technology, 2004):

- **Vertraulichkeit** (engl.: Confidentiality): Daten dürfen nur von berechtigten Akteuren eingesehen werden können.
- **Integrität** (engl.: Integrity): Daten dürfen nur von berechtigten Akteuren verändert werden, sie müssen nichtabstreitbar und authentisch sein.
- **Verfügbarkeit** (engl.: Availability): Daten müssen zeitnah und zuverlässig zur Verfügung stehen.

Die Festlegung der jeweiligen Sicherheitsziele kann in Zusammenarbeit mit den Stakeholdern erfolgen. Hierbei ist es möglich, einem Asset mehrere Schutzziele zuzuordnen, mindestens jedoch eins. Für die Dokumentation der Ergebnisse kann die Tabelle aus 5.1.4 entsprechend erweitert werden.

5.1.6 Priorisierung von Assets und ihrer Sicherheitsziele

Zuordnung zu Prozessschritt: Risikoanalyse

Um Sicherheitsinvestitionen effektiv einsetzen und sicherheitsrelevante Entscheidungen fundiert treffen zu können, bedarf es einer Priorisierung der identifizierten Assets und deren Sicherheitsziele. Diese Priorisierung erfolgt auf Grundlage des erwarteten Schadens im Falle der Verletzung eines oder mehrerer Sicherheitsziele. Die Schätzung des zu erwartenden Schadens kann in Zusammenarbeit mit den jeweiligen Stakeholdern erfolgen und sich der Betrachtung unterschiedlicher Schadensszenarien bedienen. Die Dokumentation der Ergebnisse kann eine Ergänzung der Tabelle aus 5.1.4, bzw. 5.1.5 darstellen.

5.1.7 Erfassung bekannter Sicherheitsvorfälle

Zuordnung zu Prozessschritt: Bedrohungsanalyse

Für die zukünftige Weiterentwicklung der „PersoApp“ sollten bekannte Sicherheitsvorfälle der Vergangenheit Berücksichtigung finden, die einen Bezug zu der Software erkennen lassen. Dabei kann es sich zum Beispiel um die Bekanntmachung sicherheitskritischer Design- oder Implementierungsfehler handeln. Im Kontext dieses Projektes gilt insbesondere:

- Die Entwicklung der PersoApp beruht auf einem Produkt der AGETO Service GmbH. Bekannte Sicherheitsvorfälle beim Einsatz dieses Produktes sind für die Weiterentwicklung der Software von Interesse.
- Die Software verwendet gegebenenfalls Komponenten externer Parteien (z.B. Kryptographiebibliotheken, oder Ähnliches), für die es wiederum bekannte Sicherheitsvorfälle gibt. Auch dies sollte berücksichtigt werden.
- Vergleichbare Produkte, wie etwa die „AusweisApp“ der OpenLimit SignCubes AG, oder Ähnliche, könnten aufgrund ihrer Gemeinsamkeiten mit der PersoApp durch ähnliche Bedrohungen und Schwachstellen betroffen sein.

5.1.8 Erstellung von Misuse-Cases

Zuordnung zu Prozessschritt: Bedrohungsanalyse

Um die Absichten potentieller Angreifer zu untersuchen und mögliche Sicherheitsanforderungen an die Software ableiten zu können, wird in dieser Aktivität ein sogenanntes Misuse-Case-Diagramm angefertigt. Hierfür kann das Use-Case-Diagramm, sofern es denn in 5.1.1 bereits angefertigt wurde, um unerwünschte Aktionen potentieller Angreifer ergänzt werden (Opdahl, 2000).

Bedrohungsanalyse

Die Bedrohungsanalyse widmet sich vordergründig der Identifikation und Dokumentation potentieller Bedrohungen, die mittelbar und unmittelbar auf die PersoApp einwirken können. Da dies als fundamental für die Entwicklung sicherer Softwaremodule angesehen wird, wurde der Entwicklungsprozess um diesen Prozessschritt ergänzt. Wie oben bereits dokumentiert wurde, umfasst dies die folgenden Aktivitäten:

- 5.1.1 Beschreibung der Software

- 5.1.2 Beschreibung der Systemnutzer
- 5.1.3 Beschreibung der Stakeholder
- 5.1.7 Erfassung bekannter Sicherheitsvorfälle
- 5.1.8 Erstellung von Misuse-Cases

Risikoanalyse

Die Risikoanalyse zielt darauf ab, mögliche Bedrohungen und potentielle Schwachstellen zu bewerten und zu priorisieren. Dementsprechend wurden die folgenden Aktivitäten beschrieben:

- 5.1.1 Beschreibung der Software
- 5.1.2 Beschreibung der Systemnutzer
- 5.1.3 Beschreibung der Stakeholder
- 5.1.4 Beschreibung der Assets
- 5.1.5 Festlegung von Sicherheitszielen für Assets
- 5.1.6 Priorisierung von Assets und ihrer Sicherheitsziele

5.2 Modellierung

Im Anschluss an die Analysephase widmet sich die Modellierungsphase der konkreten Ausgestaltung der zu entwickelnden Software(-komponente). Im Kontext des „PersoApp“-Projektes wurde dieser Prozessabschnitt um zwei weitere sicherheitsrelevante Prozessschritte ergänzt: Sicherheitsmodellierung und Sicherheitsarchitektur. Entsprechend werden die folgenden Aktivitäten eingeführt:

5.2.1 Beschreibung der Systemarchitektur, der Zielplattform(en) und der Technologien

Zuordnung zu Prozessschritt: Sicherheitsarchitektur

Um sicherheitsspezifische Mechanismen und Funktionalitäten bewerten und einführen zu können, bedarf es zunächst einer sorgfältigen Dokumentation des bestehenden Modells. Hierfür gilt es die verwendeten Architektur- und Designparadigmen geeignet zu beschreiben, die eingesetzten Technologien darzulegen sowie die Zielplattformen und ihre Spezifika festzuhalten. Jede Architektur- und Designentscheidung birgt Vor- und Nachteile im Hinblick auf die Sicherheitseigenschaften der Software.

5.2.2 Auswahl relevanter Schwachstellen aus der CWE-Datenbank

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Die CWE-Datenbank (CWE: Common Weakness Enumeration) ist ein von der MITRE Corporation öffentlich zur Verfügung gestelltes Nachschlagewerk für gängige Schwachstellen in Softwareprodukten (vgl. Abbildung 3). Für die „PersoApp“ kann auf Basis ihrer Systemarchitektur und technischen Abhängigkeiten eine Recherche gemacht werden, um potentiell relevante Datenbankeinträge zu identifizieren.

Siehe: http://cwe.mitre.org/

Abbildung 3 - Das Onlineportal "Common Weakness Enumeration" der MITRE Corporation, CWE 20: Improper Input Validation.

5.2.3 Erstellung eines Angreifermodells

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Um die Wirksamkeit und Notwendigkeit von Schutzmaßnahmen bewerten zu können, muss eine grundlegende Analyse der potentiellen Angreifer durchgeführt werden. Hierfür eignet sich die Erstellung eines sogenannten Angreifermodells, in dem einzelne Klassen von Angreifern (z.B. Scriptkiddie, Hacktivist, Organisiertes Verbrechen, etc.) hinsichtlich mehrerer Eigenschaften beschrieben werden. Zu diesen Eigenschaften zählen unter anderem:

- Bezeichner der Angreiferklasse
- Beschreibung der Angreiferklasse
- Technische Fähigkeiten
- Angriffsziel(e)

Bei Bedarf können die Eigenschaften, die das Modell beinhalten soll, erweitert werden.

5.2.4 Erstellung eines Datenflussdiagramms (DFD)

Zuordnung zu Prozessschritt: Sicherheitsarchitektur

Die tatsächliche Sicherheitsarchitektur der PersoApp, mitsamt ihrer sicherheitsrelevanten Abhängigkeiten und Systemeigenschaften, wird in Form eines oder mehrerer Datenflussdiagramme dokumentiert (Torr, 2005). Ausgangspunkt hierfür kann ein eventuell in 5.1.1 erstelltes Kontextdiagramm darstellen, das hier (iterativ) weiterentwickelt und präzisiert wird. Für die Erstellung des Diagramms bietet sich die Verwendung einer geeigneten Werkzeugunterstützung an, wie etwa das SDL Threat Modeling Tool von Microsoft (Siehe http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx).

Diese Aktivität bildet unmittelbar die Grundlage für die folgenden Aktivitäten 5.2.5 und 5.2.6.

5.2.5 Identifikation von Trust-Boundaries

Zuordnung zu Prozessschritt: Sicherheitsarchitektur

Auf Basis des Datenflussdiagramms aus 5.2.4 werden sogenannte Trust-Boundaries in das Modell eingefügt (Microsoft, 2006). Hierdurch soll verdeutlicht werden, an welcher Stelle Daten über eine Privilegien- und/oder Maschinengrenzen hinweg übertragen werden, da dies für eine Sicherheitsbetrachtung von besonderem Interesse sein kann. Beispielsweise betrifft dies die kabellose (z.B. WLAN, Bluetooth) oder kabelgebundene (z.B. LAN, Internet) Datenübertragung über Netzwerke.

5.2.6 Anwendung von STRIDE

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Für jedes Element des Datenflussdiagramms aus 5.2.4, bzw. 5.2.5 wird anhand der STRIDE-Kategorien ermittelt, welche Bedrohungen grundsätzlich bestehen können. STRIDE steht hierbei für (Torr, 2005):

- **S**poofing: Vorgabe einer falschen Identität/gefälschter Daten
- **T**ampering: Unberechtigte Manipulation von Daten
- **R**epudiation: Durchführung abstreitbarer Handlungen
- **I**nformation disclosure: Unberechtigter Zugriff auf Daten
- **D**enial of service: Beeinträchtigung der Verfügbarkeit eines Systems
- **E**levation of privilege: Unerlaubte Ausweitung der Benutzerrechte

Für das Mapping zwischen Diagrammelement und Bedrohungskategorie ist die folgende Tabelle zu Hilfe zu nehmen (Microsoft, 2006):

Element	*Spoofing*	*Tampering*	*Repudiation*	*Information Disclosure*	*Denial of Service*	*Elevation of Privilege*
Data Flows		x		x	x	
Data Stores		x		x	x	
Processes	x	x	x	x	x	x
Interactors	x		x			

Tabelle 1 - Mapping zwischen den Elementen des DFD und den Bedrohungskategorien gemäß STRIDE.

Die Zuordnung kann entweder manuell durchgeführt werden, oder aber durch eine geeignete Werkzeugunterstützung (wie beispielsweise das SDL Threat Modeling Tool von Microsoft, siehe http://www.microsoft.com/security/sdl/adopt/threatmodeling.aspx) automatisiert erfolgen.

Das Ergebnis dieser Aktivität ist letztlich eine umfangreiche Liste potentieller Bedrohungen, die aus dem Systemdesign der PersoApp abgeleitet worden sind.

5.2.7 Priorisierung potentieller Schwachstellen

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Diese Aktivität ist der Priorisierung der potentiellen Schwachstellen der PersoApp gewidmet, die in den bisherigen Analyseaktivitäten identifiziert werden konnten. Hierfür wird jede Schwachstelle all jenen Assets zugeordnet, die durch eine erfolgreiche Ausnutzung der Lücke betroffen wären. Da in 5.1.6 bereits die identifizierten Assets mitsamt ihrer Sicherheitsziele priorisiert wurden, erfolgt hierdurch implizit eine Priorisierung der potentiellen Schwachstellen.

Bei Bedarf kann eine weitergehende Differenzierung anhand der in 5.2.5 ermittelten Trust-Boundaries erfolgen. Dabei gelten all jene potentiellen Bedrohungen als höher priorisiert, die mit einer Überschreitung einer Maschinen- bzw. Privilegiengrenze einhergehen.

5.2.8 Peer-Review des Systemdesigns

Zuordnung zu Prozessschritt: Sicherheitsarchitektur

Bevor das Systemdesign in die Implementierungsphase übergeht, wird es in dieser Aktivität von einem ausgewählten Personenkreis einem Review unterzogen. Das Ziel dieses Reviews ist die Identifikation grober Sicherheitsmängel, die die Entwicklung sicherer Softwaremodule nachhaltig gefährden würde. Das Systemdesign gilt solange als inakzeptabel, bis es das Peer-Review bestanden hat. Unter Umständen kann es sich hierbei um eine iterative Vorgehensweise handeln, bei der Modellierung

und Bewertung wiederholt durchgeführt werden müssen, um ein allgemein akzeptiertes Ergebnis zu erzielen.

5.2.9 Dokumentation aller Sicherheitsmechanismen

Zuordnung zu Prozessschritt: Sicherheitsarchitektur

Um über den gesamten Projektverlauf sicherheitsrelevante Architektur- und Designentscheidungen nachvollziehbar zu gestalten, wird in dieser Aktivität eine Dokumentation aller vorgesehenen Sicherheitsmechanismen angefertigt. Die Dokumentation sollte für jeden Mechanismus, neben technischen Einzelheiten, festhalten, welche Motivation seiner Einführung zugrunde lag (d.h. welcher potentiellen Schwachstelle er sich widmet) und welche Grundannahmen getroffen wurden (z.B. dass Mitarbeiter grundsätzlich vertrauenswürdig sind).

5.2.10 Bewertung der Sicherheitsmechanismen

Zuordnung zu Prozessschritt: Sicherheitsarchitektur

Auf Basis von 5.2.9 wird hier eine Bewertung der dokumentierten Sicherheitsmechanismen vorgenommen. Letztlich soll hierdurch ermittelt werden, ob die eingesetzten Verfahren grundsätzlich geeignet sind und ob die zugrundeliegenden Annahmen gültig sind. Da sich Anforderungen und der Systemkontext über den Projektverlauf wandeln können, sollte diese Aktivität in regelmäßigen Abständen wiederholt werden.

5.2.11 Bestimmung der Angriffsfläche

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Im Zuge dieser Aktivität werden alle möglichen Angriffspunkte identifiziert, über die ein Angreifer potentiell Zugriff auf eine Instanz der PersoApp erhalten könnte. Hierbei handelt es sich beispielsweise um Eingabefelder, Netzwerkschnittstellen und Softwareschnittstellen. Die Gesamtheit dieser Angriffspunkte wird in diesem Kontext als Angriffsfläche verstanden. Die detaillierte Bestimmung dieser Angriffsfläche ist notwendig, um eine systematische Analyse und Verbesserung der Sicherheitseigenschaften der PersoApp zu ermöglichen.

5.2.12 Reduzierung der Angriffsfläche

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Auf der Grundlage von 5.2.11 wird hier angestrebt, die Angriffsfläche zu minimieren. Hierfür werden alle identifizierten Angriffspunkte im Einzelnen betrachtet und mit Hinblick darauf bewertet, ob ihre Verfügbarkeit gemindert werden kann, oder ob sie gar ganz entbehrlich sind. Die Minderung der Verfügbarkeit eines Angriffspunktes kann z.B. erreicht werden durch:

- Entfernen der Funktionalität
- Entfernen von Schnittstellen
- Einschränkung der Verfügbarkeit von Schnittstellen auf einen eingeschränkten Nutzerkreis

- Zeitliche Einschränkung der Verfügbarkeit von Schnittstellen/Funktionalität

Bei der Minimierung der Angriffsfläche gilt es jedoch die funktionalen und nichtfunktionalen Anforderungen der PersoApp zu berücksichtigen. Ein eventueller Trade-off zwischen Benutzerfreundlichkeit und Sicherheit, bzw. Funktionalität und Sicherheit sollte nicht implizit erfolgen.

5.2.13 Erfassung der Sicherheitsaspekte für das Deployment

Zuordnung zu Prozessschritt: Sicherheitsmodellierung

Die „PersoApp“ soll durch eine breite Entwickler- bzw. Nutzerbasis auf möglichst sichere Weise genutzt werden können, ohne spezielle Kenntnisse im Bereich der IT-Sicherheit vorauszusetzen. Um dies zu ermöglichen, müssen grundlegende Sicherheitsaspekte erfasst werden, die bei der Verteilung und Installation der PersoApp relevant sind. Hierbei kann die Beantwortung der folgenden Fragen als Leitlinie dienen:

- Welche Aktivitäten (z.B. Erstellung von Zertifikaten) umfasst der Build eines „PersoApp“-Releases?
- Welche Konfiguration (z.B. Compiler, Linker, etc.) erfordert der Build eines „PersoApp“-Releases?
- Welche Aktivitäten (z.B. Setzen eines Passwortes) sind bei der Installation der „PersoApp“ durchzuführen?

Beim Build eines „PersoApp“-Releases gilt es insbesondere darauf zu achten, dass keine unnötigen Informationen und Artefakte in die allgemeine Verteilung einfließen. Konkret handelt es sich hierbei beispielsweise um Debug- und Loginformationen, oder etwa um temporäre Dateien, die für die Installation und den Betrieb der PersoApp nicht vonnöten sind.

Sicherheitsmodellierung

Im Kontext dieses Projektes umfasst die Sicherheitsmodellierung all jene Aktivitäten, deren Ergebnisse die Entwicklung einer Sicherheitsarchitektur beeinflussen können. Hierbei handelt es sich vordergründig um die Identifikation und Bewertung von potentiellen Schwachstellen, sowie um die Dokumentation und Optimierung von Sicherheitseigenschaften.

Im Einzelnen sind hierfür die folgenden Aktivitäten durchzuführen:

- 5.2.2 Auswahl relevanter Schwachstellen aus der CWE-Datenbank
- 5.2.3 Erstellung eines Angreifermodells
- 5.2.6 Anwendung von STRIDE
- 5.2.7 Priorisierung potentieller Schwachstellen
- 5.2.11 Bestimmung der Angriffsfläche
- 5.2.12 Reduzierung der Angriffsfläche
- 5.2.13 Erfassung der Sicherheitsaspekte für das Deployment

Sicherheitsarchitektur

Die Erstellung einer Sicherheitsarchitektur beruht letztlich auf den Ergebnissen der Sicherheitsmodellierung. Die Einführung und Bewertung geeigneter Sicherheitsmechanismen in die „PersoApp“ bezieht sich unmittelbar auf konkrete Bedrohungen und unterliegt einer sorgfältigen Prüfung.

Hierfür sind die folgenden Aktivitäten durchzuführen:

- 5.2.1 Beschreibung der Systemarchitektur, der Zielplattform(en) und der Technologien
- 5.2.4 Erstellung eines Datenflussdiagramms (DFD)
- 5.2.5 Identifikation von Trust-Boundaries
- 5.2.8 Peer-Review des Systemdesigns
- 5.2.9 Dokumentation aller Sicherheitsmechanismen
- 5.2.10 Bewertung der Sicherheitsmechanismen

5.3 Implementierung

In der Implementierungsphase werden die Spezifikationen aus den vorhergehenden Phasen „Modellierung“ und „Analyse“ in Quelltext überführt. Neben dem funktionalen Quelltext, der das lauffähige Endprodukt bildet, werden auch automatisierte Tests (z.B. J-Unit-Tests) erzeugt. Mit Hilfe dieser Tests können Teile des Quelltextes auf ihre Funktion getestet werden. Um die zu entwickelnde Software sicherer zu machen, wird diese Phase um die nachfolgenden Aktivitäten erweitert, dem Erstellen von einer sicheren Defaultkonfiguration und der Anwendung von „Secure Coding Guidelines“.

5.3.1 Entwicklung einer sicheren Defaultkonfiguration

Nahezu jede Software ermöglicht es dem Benutzer bestimmte Einstellungen selbst zu konfigurieren. Diese Einstellungen werden in Konfigurationsdateien gespeichert, initiale Konfigurationsdateien werden zusammen mit der Software ausgeliefert. Ziel dieser Aktivität ist es dafür zu sorgen, dass alle Einstellungen die beim Erstellen der Software variabel bleiben sollen, um später vom Benutzer geändert werden zu können, mit sinnvollen Standardwerten innerhalb der Konfigurationsdateien belegt werden. Zudem sollten diese Einstellungen so dokumentiert werden, dass ein Benutzer nachvollziehen kann, welche Konsequenzen die Änderung eines Parameters zur Folge hat. In dieser Dokumentation sollten alle Parameter mit Namen, Standardwert, nicht erlaubten Werten und Abhängigkeiten zu anderen Parametern aufgeführt sein. Eine sicherheitskritische Abhängigkeit ist z.B. das Verschlüsseln einer Netzwerkverbindung, welche dann auch ein Zertifikat/Passwort für den Verschlüsselungsalgorithmus benötigt.

5.3.2 Anwendung von „Secure Coding Guidelines“

Die „Secure Coding Guidelines“ hängen stark von den eingesetzten Technologien, Plattformen und Architekturparadigmen ab. Somit liefert 5.2.1 die Grundlage für diese Aktivität. Jede Technologie, Plattform oder Architekturparadigma hat gängige

Angriffspunkte, Schwachstellen oder erfordert bestimmte Gegenmaßnahmen. Daher ist es nötig den Entwicklern Leitfäden an die Hand zu geben wie sie mit ihnen umzugehen haben. Dies umfasst insbesondere die Vorschrift zur Verwendung bestimmter APIs/Bibliotheken und wie sie einzusetzen sind oder die Vermeidung von gängigen Fehlern die häufig gemacht werden. Oft besitzen API-Dokumentationen auch einen dedizierten Abschnitt über Sicherheit, auf den besonderen Wert gelegt werden sollte. Allgemein sei hier auf die Leitfäden des CERT verwiesen, und da die PersoApp Java als Programmiersprache einsetzt, auf den Leitfaden von Oracle.

https://www.securecoding.cert.org/confluence/display/seccode/CERT+Secure+Coding+Standards

http://www.oracle.com/technetwork/java/seccodeguide-139067.html

5.4 Validierung

In dieser Phase werden Funktions- und Integrationstests durchgeführt. Funktionstests dienen zum Auffinden von Fehlern, welche die in den ersten Phasen des Entwicklungsprozesses festgelegten Spezifikationen verletzen. Integrationstests stellen sicher, dass die zu entwickelnde Software auf den Zielsystemen problemlos installiert werden kann. Zu diesen Tests kommen die nachfolgenden Aktivitäten hinzu um auch die Sicherheit der zu entwickelnden Software zu testen.

5.4.1 Anwendung von Codescannern

Statische Codescanner lesen den Quelltext einer Software ein und versuchen an Hand von bestimmten Regeln Muster zu erkennen die wahrscheinlich die Sicherheit der Software beeinträchtigen können. Manuelle Codereviews sind recht aufwendig und oft ist es gut einen Startpunkt im Quelltext zu haben bei dem man genauer hinschauen sollte, diesen können Codescanner oft liefern. Im Rahmen des Projektes werden wir mit mindestens einem Codescanner (z.B. Findbugs für Java) den Quelltext der PersoApp scannen. Eine Aussage darüber ob der Quelltext „sicher“ ist können Codescanner jedoch nicht liefern, dennoch ist es sinnvoll den Quelltext zumindest mit einem Codescanner zu scannen, da es weitestgehend automatisierbar ist und kleinere Fehler gefunden werden können.

5.4.2 Durchführung von Codereviews

Da statische Codescanner nur bedingt dazu geeignet sind Fehler in der Software zu finden, sind manuelle Codereviews sinnvoll. Diese sind jedoch mit wesentlich mehr Aufwand verbunden und sollten daher auf kritische Bereiche der Software fokussiert werden. Die Ergebnisse von Codescannern können hierfür als Startpunkt genutzt werden um tiefer im Quellcode zu schauen ob es ein „false positive“ Fehler war oder wirklich eine kritische Lücke darstellt. Einen weiteren Ausgangspunkt stellt die Architekturbeschreibung der Software dar, sowie die Auflistung der verwendeten Technologien (vgl. Aktivität 5.2.1). In ihr werden sicherheitskritische Teile der Software identifiziert und anschließend im Quelltext begutachtet. Mit Hilfe von Versionskontrollsystemen lassen sich die Autoren bestimmter Codezeilen ermitteln um nachfragen zu können warum die Autoren bestimmte

Implementierungsentscheidungen getroffen haben. Hat ein Reviewer einen potentiellen Fehler gefunden und ist mit der Erklärung des Autors nicht zufrieden, wendet er sich an das Kernentwicklerteam um eine Lösung zu finden. Neben sicherheitskritischen Stellen wird der Quellcode ebenso auf die Einhaltung der Programmierrichtlinien aus dem noch zu erstellenden Dokument „D04-QM Programmierrichtlinien zur Erstellung von PersoApp-Softwaremodulen“ geprüft. Patches, die im Verlauf des Projektes eingereicht werden, werden von dem jeweiligen Modulverantwortlichen, für das der Patch gedacht ist, begutachtet und erst danach übernommen.

5.5 Freigabe

Nach erfolgreicher Validierung wird die Software für die Benutzer bereitgestellt. Hierbei werden letzte Funktionstests durchgeführt und den Benutzern eine sichere Standardkonfiguration angeboten.

5.5.1 Durchführung eines finalen Codereviews

Das finale Codereview entspricht in weiten Teilen dem Codereview aus Aktivität 5.4.2 mit dem Unterschied, dass hierbei auch Funktionstests und Installationstests durchgeführt werden.

5.5.2 Bereitstellung eines sicheren Setups

Die Konfigurationsdateien aus Aktivität 5.3.1 sollten bei Auslieferung der Software so konfiguriert sein, dass sie selbst für einen unbedarften Benutzer, der nichts an ihnen verändert, eine „sichere“ Standardkonfiguration darstellen. Dies bedeutet, dass insbesondere dort wo der Benutzer explizit Sicherheitsoptionen ein- oder ausschalten kann, diese standardmäßig aktiviert sind. Zudem sollte die Software so konfiguriert sein, dass sie mit den Daten des Benutzers sparsam umgeht und sie möglichst nicht zwischenspeichert (z.B. die PIN des Personalausweises sollte nicht auf dem PC gespeichert werden). Weiterhin sollte die Software nach der Installation mit den niedrigst-möglichen Rechten ausgeführt werden um die Angriffsfläche (vgl. Aktivität 5.2.12) niedrig zu halten.

6 Interne und externe Anforderungen

Eine Vielzahl *interner Anforderungen* hat Einfluss auf die Durchführbarkeit einzelner Aktivitäten, wie sie in diesem Dokument beschrieben worden sind. Daher ist es den Projektverantwortlichen freigestellt, situationsbedingt die Entscheidung zu treffen, einzelne Aktivitäten hinzuzufügen, zu modifizieren, oder gänzlich auszulassen. Mögliche Ursachen hierfür können in der Zeit- und Ressourcenplanung liegen, oder technischen und organisatorischen Gegebenheiten geschuldet sein, die sich im Projektverlauf ergeben. In jedem Fall sind Abweichungen von dem im vorliegenden Dokument beschriebenen Prozess in nachvollziehbarer Weise zu dokumentieren.

Externe Anforderungen umfassen den Einsatz von Hardware, Software und ggf. Dokumentationen externer Parteien, die ausschließlich gemäß ihrer vorliegenden Lizenz verwendet werden dürfen.

7 Abläufe

Der Entwurfs- und Entwicklungsprozess von „PersoApp“-Softwaremodulen erfolgt gemäß der oben festgelegten Vorgehensweise. Für eine detailliertere Beschreibung einzelner Vorgänge sind auch die nachfolgenden Dokumente zu beachten:

- D08-QM Review-Konzept „PersoApp“
- D08-QM-2 Operative Planung und Durchführung von Reviews und Release-Updates
- D08-QM-3 Prozessbeschreibung zur Durchführung von Code-Reviews und Sicherstellung der Dokumentationsqualität
- D09-QM-2 Release Management von Software-Modulen und Dokumente der Open-Source-„PersoApp“

8 Mitgeltende Dokumente

Die nachfolgenden Dokumente beinhalten zu berücksichtigende prozessrelevante Beschreibungen:

- D04-QM Programmierrichtlinien zur Erstellung von „PersoApp“-Softwaremodulen
- D08-QM Review-Konzept „PersoApp“
- D08-QM-2 Operative Planung und Durchführung von Reviews und Release-Updates
- D08-QM-3 Prozessbeschreibung zur Durchführung von Code-Reviews und Sicherstellung der Dokumentationsqualität
- D09-QM-2 Release Management von Software-Modulen und Dokumente der Open-Source-„PersoApp“

Literatur

Microsoft. (2006). *Uncover Security Design Flaws Using The STRIDE Approach*. Retrieved 5 23, 2013, from Uncover Security Design Flaws Using The STRIDE Approach: http://msdn.microsoft.com/en-us/magazine/cc163519.aspx

National Institute of Standards and Technology. (2004). Standards for Security Categorization of Federal Information and Information Systems. *FIPS PUB 199* .

Opdahl, G. S. (2000). Eliciting Security Requirements by Misuse Cases. *Proceedings of the 37th International Conference on Technology of Object-Oriented Languages and Systems* .

Torr, P. (2005). Demystifying the threat modeling process. *Security & Privacy, IEEE (Volume 3, issue 5)* .