Secure Software Development Lifecycle

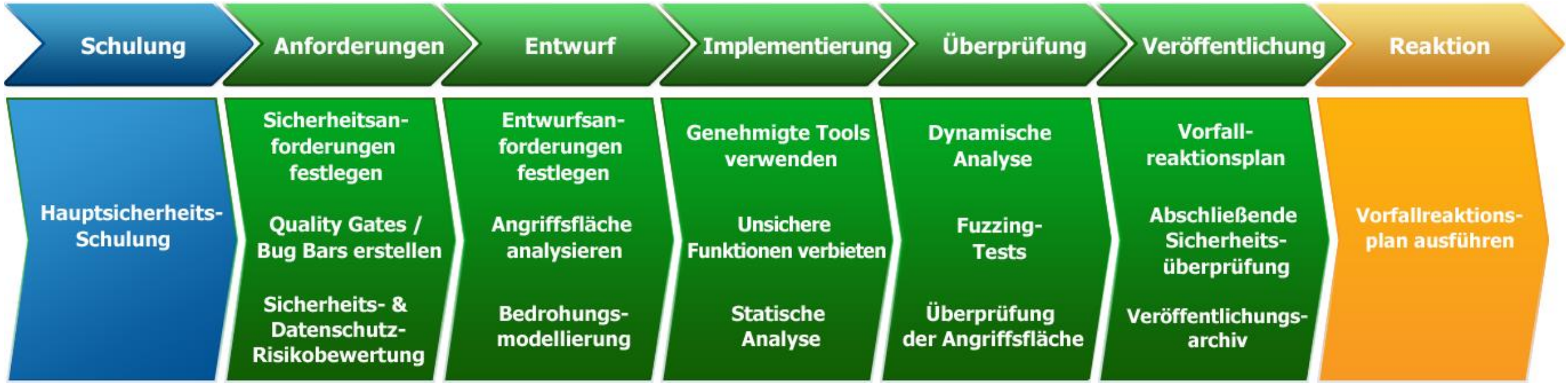
Sichere App-Entwicklung | Dr. Alexey Tschudnowsky | Bereichsleiter Software Development

WARUM?

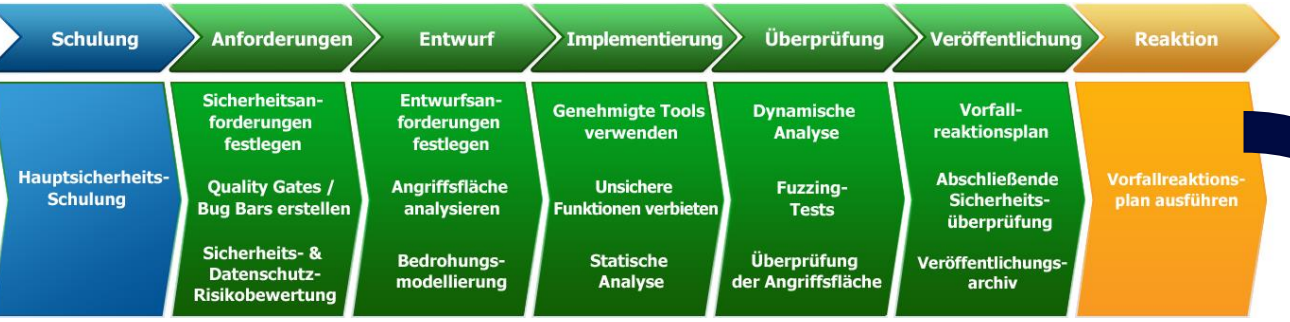
Wahrscheinlichkeit, Schweregrad, Folgen und Kosten von Sicherheitsschwachstellen verringern

Shift Left Security

Microsoft Security Development Lifecycle



Quelle: Microsoft, SDL Process Guidance Version 5.2



Microsoft SDL in agiler Form

Regelmäßige Aktivitäten
- Trainings
- Penetrationtests
- Prozessbewertung
- Aktualisierung von Werkzeugen
- Änderungsmanagement

Extern getriggerte Aktivitäten
- Sicherheitsvorfallbehandlung

Definition
- Product Backlog
- Datenschutz- und Sicherheitskonzept
- Richtlinien- und Vorgaben
- (Sicherheits-)Testkonzept
- Rollen und Verantwortlichkeiten
- Quality Gates
- Trainingsplan

Refinement
- Sprint Backlog

Release

#1 Rollen festlegen

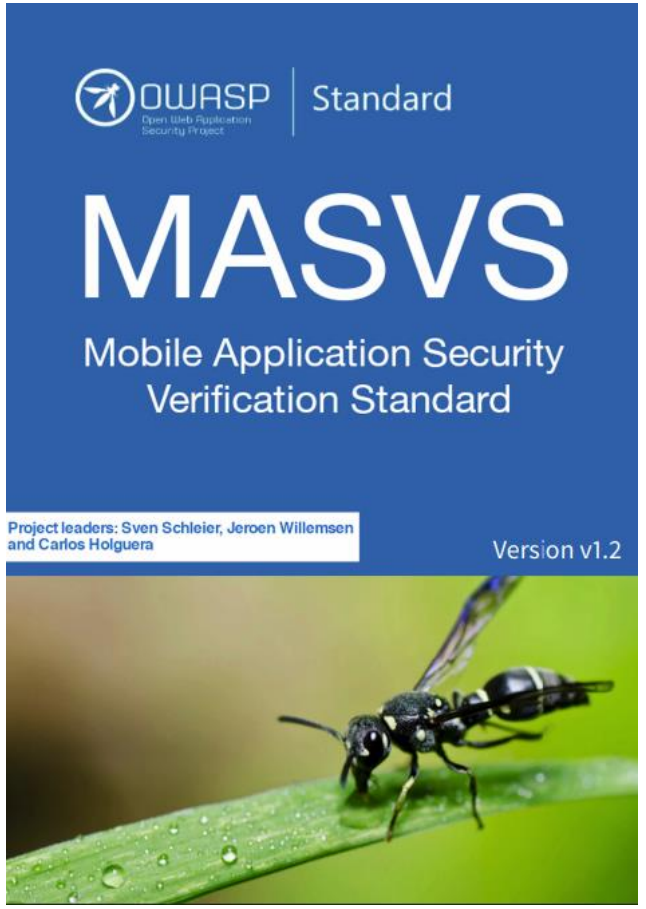
Rollen

- SDL Process Owner
- Datenschutz- und Sicherheitsbeauftragter
- Product Owner
- Softwareentwickler
- Tester
- Security Tester
- Scrum Master
- Software Architect
- Release Manager
- Risk Manager
- Security Experte

#2 Security Trainings

Regelmäßige Basis- und Spezialschulungen

- Basisschulung „Datenschutz & Sicherheit“
- Produktspezifische Schulungen:
    - Techniken der sicheren Entwicklung für mobile Geräte (OWASP MASVS)
    - OWASP Top 10
    - Sicherheit und Infrastruktur (WAF, DMZ, Firewall, IDS, IPS, Logging, Monitoring)
    - Secure Coding in Java



Quelle: https://github.com/OWASP/owasp-masvs

#3
Sicherheitsanforderungen identifizieren

Sicherheitsanforderungen identifizieren

- Gesetze (z.B. DSGVO)
- BSI-Richtlinien (z.B. TR-02102 Kryptographische Verfahren)
- In der Spezifikation enthaltene Anforderungen (zu verwendende Protokolle, Datenschutzanforderungen,…)
- Anforderungen aus OWASP MASVS
- Secure Coding Guidelines für iOS und Android

Nachweisbarkeit der Erfüllung ist wichtig!

#4 Security Reviews

Wichtige Reviews

1. Architektur-Reviews
2. Code-Reviews
3. Abschließende Sicherheitsüberprüfung

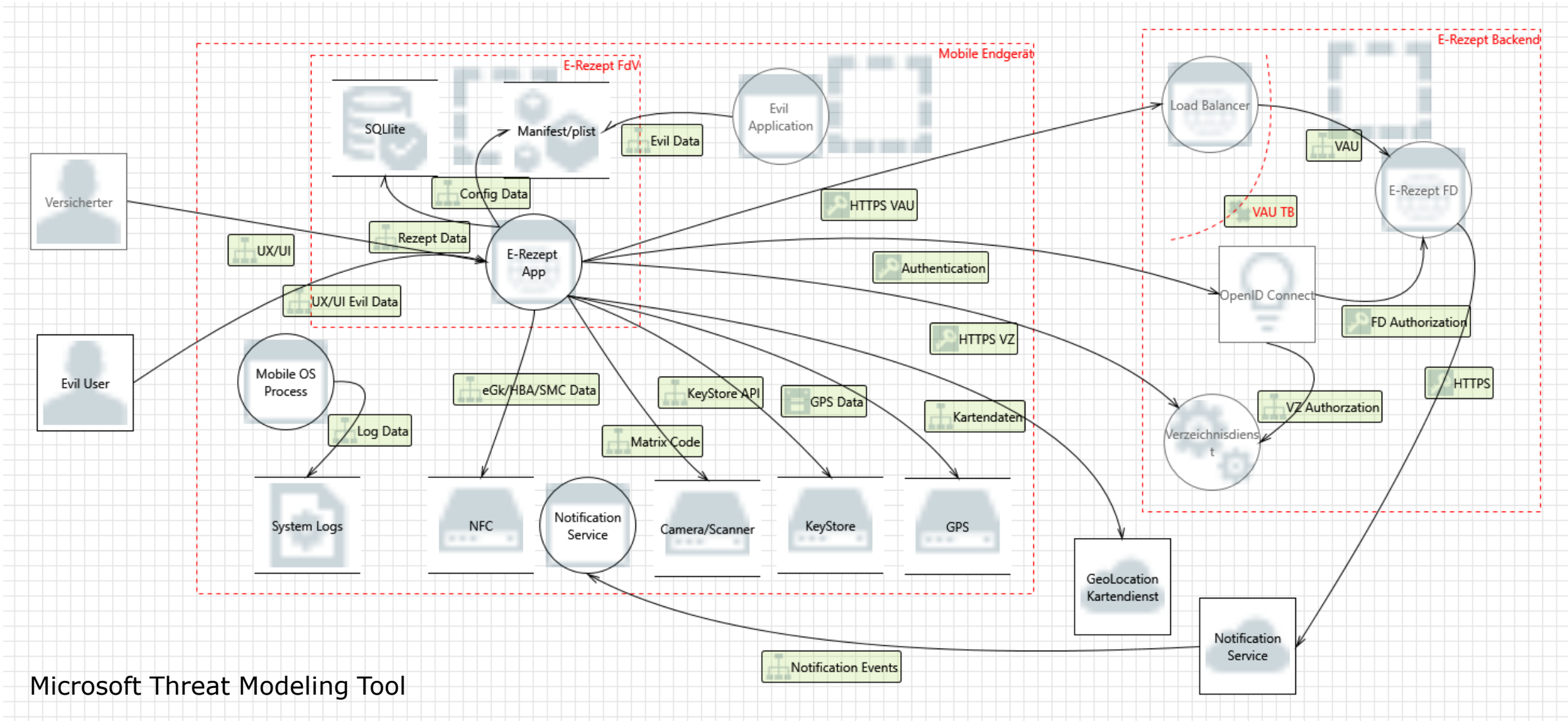
Mobile Application Security Requirements - Android

ID	MSTG-ID	Detailed Verification Requirement	Level 2	Status	Comment
V1		**Architecture, design and threat modelling**			
1.1	MSTG-ARCH-1	All app components are identified and known to be needed.	✓	Pass	Neue Komponenten und dokumentiert.
1.2	MSTG-ARCH-2	Security controls are never enforced only on the client side, but on the respective remote endpoints.	✓	Pass	Die App kann Rezep Token vom FD abruf Faktoren: eGK (Besit
1.3	MSTG-ARCH-3	A high-level architecture for the mobile app and all connected remote services has been defined and security has been addressed in that architecture.	✓	Pass	Die Architektur ist d
1.4	MSTG-ARCH-4	Data considered sensitive in the context of the mobile app is clearly identified.	✓	Pass	Siehe 1.3
1.5	MSTG-ARCH-5	All app components are defined in terms of the business functions and/or security functions they provide.	✓	Pass	s. MSTG-ARCH-1, v Algorithmen, App
1.6	MSTG-ARCH-6	A threat model for the mobile app and the associated remote services has been produced that identifies potential threats and countermeasures.	✓	Pass	nicht relevant bei de mobilen App wird in
1.7	MSTG-ARCH-7	All security controls have a centralized implementation.	✓	Pass	Nutzung von Featur Secure Element, Key
1.8	MSTG-ARCH-8	There is an explicit policy for how cryptographic keys (if any) are managed, and the lifecycle of cryptographic keys is enforced. Ideally, follow a key management standard such as NIST SP 800-57.	✓	Pass	Durch gemSpec_Kry
1.9	MSTG-ARCH-9	A mechanism for enforcing updates of the mobile app exists.	✓	N/A	Nicht vorgesehen in
1.10	MSTG-ARCH-10	Security is addressed within all parts of the software development lifecycle.	✓	Pass	SSDL wird eingehalt

Quelle: https://github.com/OWASP/owasp-mstg/tree/master/Checklists

#5
Bedrohungsmodellierung

Ziel: Erkennung und Prävention von Bedrohungen



Microsoft Threat Modeling Tool

#6 Statische Codeanalyse

Ziel: Kontinuierliche Codeprüfung auf Schwachstellen

Dokumentation der Bewertungsergebnisse und Maßnahmen sind wichtig für das Audit

Quelle: https://www.microfocus.com/de-de/products/application-security-testing/overview
Quelle: https://jeremylong.github.io/DependencyCheck/

Ziel: Kontinuierliche Codeprüfung auf Schwachstellen

Dokumentation der Bewertungsergebnisse und Maßnahmen sind

Release Issues

SHOW

Group By

Category

- Biometric Authentication:... 1
  - BiometricsAuthenticationC...
- Input Interception: Keybo... 1
  - AppDelegate.swift : 11
- Insecure Storage: Unspec... 1
- Link Injection: Auto Dial 4
- Privacy Violation: HTTP ... 1
- Type Mismatch: Negative... 2
- Type Mismatch: Signed t... 16

1314.7843 usr/local/share/agentWork/app/erp-app-ios/Sources/e...

1.0 | High | Input Interception: Keyboard Extensions Allowed | SMART FIX

Vulnerability | **Recommendations** | Code | Diagram | More Evidence | History

Recommendation

If the application accepts sensitive data via the keyboard, you may want to prevent the use of third party keyboards with your application. This is possible using the `UIApplicationDelegate application:shouldAllowExtensionPointIdentifier:` delegate method.

Example: In the following example the application disables the usage of keyboard extensions when the app is in the foreground:

```
func application(_ application: UIApplication, shouldAllowExtensionPointI
dentifier extensionPointIdentifier: UIApplication.ExtensionPointIdentifie
r) -> Bool {
if extensionPointIdentifier == .keyboard {
return false
}
return true
}
```

NEW Interactive Training

Through our partnership with Secure Code Warrior, try a short, hands-on challenge where sample software code

Audit

Status
Fix Validated

Introduced Date
2021/02/01

Last Found Date
2021/02/01

Assigned User
Fiebig, Martin (martin.fiebig

Developer Status
In Remediation

Auditor Status
Pending Review

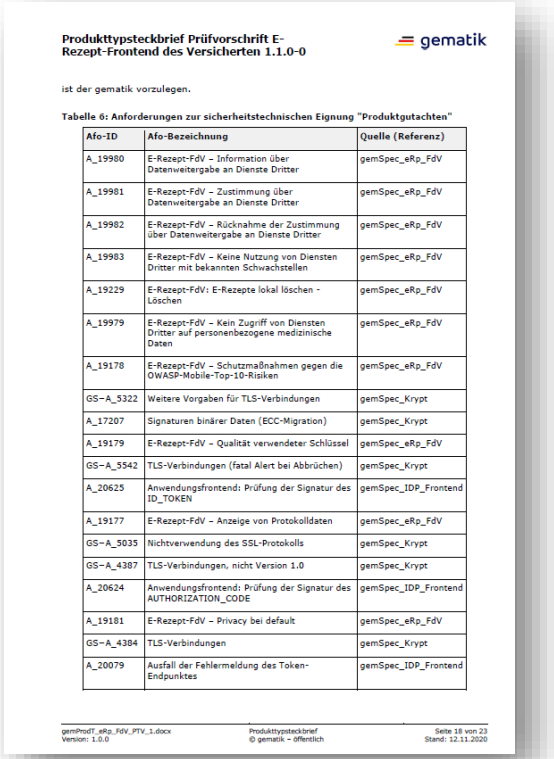
Severity
High

Comment

#7 Pentesting

Ziel: Dedizierte fachmännische Prüfung auf Schwachstellen

- WhiteBox / BlackBox Testing mit (automatisierten) Werkzeugen
- Code Reviews mit Fokus auf Sicherheitsaspekte
- Ergänzend: Prüfung der expliziten Sicherheitsanforderungen an das Produkt



Produkttypsteckbrief Prüfvorschrift E-Rezept-Frontend des Versicherten 1.1.0-0

gematik

ist der gematik vorzulegen.

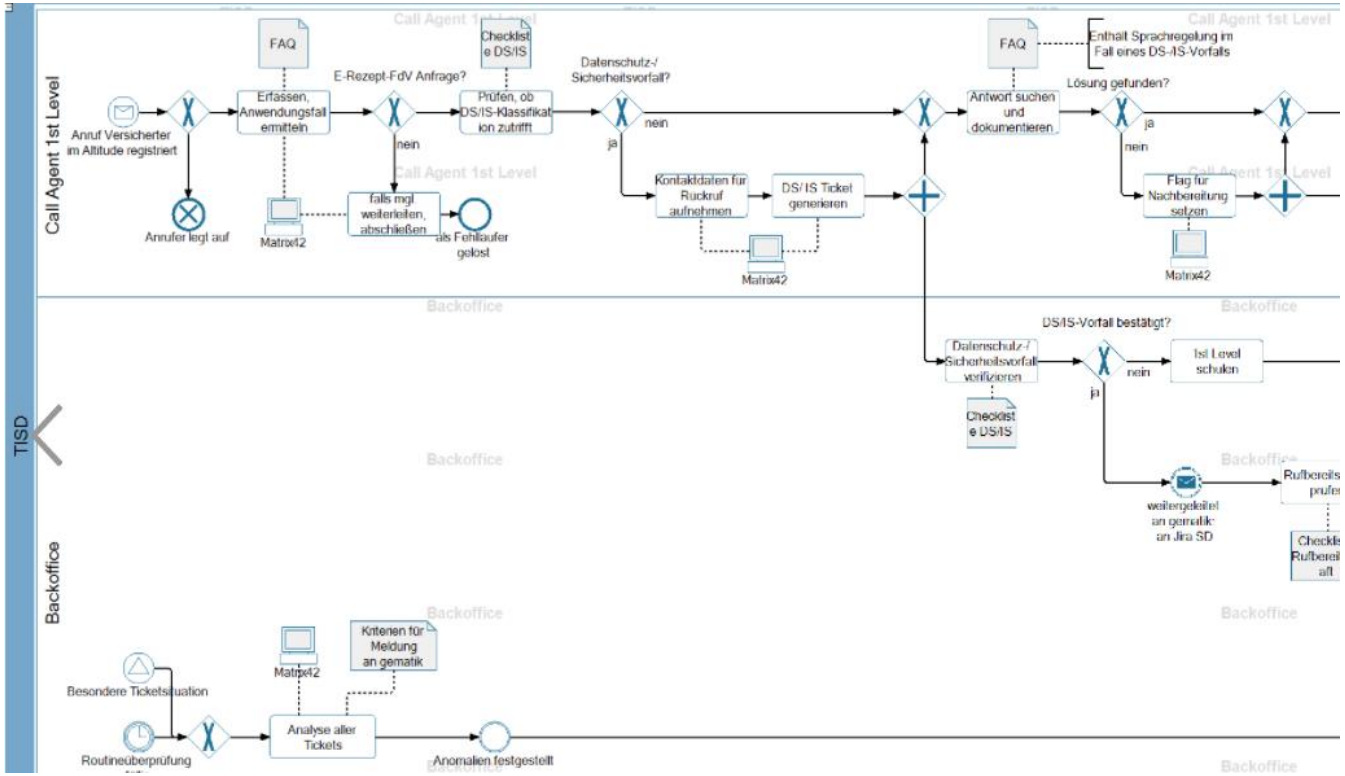
Tabelle 6: Anforderungen zur sicherheitstechnischen Eignung "Produktgutachten"

Afo-ID	Afo-Bezeichnung	Quelle (Referenz)
A_19980	E-Rezept-FdV – Information über Datenweitergabe an Dienste Dritter	gemSpec_eRp_FdV
A_19981	E-Rezept-FdV – Zustimmung über Datenweitergabe an Dienste Dritter	gemSpec_eRp_FdV
A_19982	E-Rezept-FdV – Rücknahme der Zustimmung über Datenweitergabe an Dienste Dritter	gemSpec_eRp_FdV
A_19983	E-Rezept-FdV – Keine Nutzung von Diensten Dritter mit bekannten Schwachstellen	gemSpec_eRp_FdV
A_19229	E-Rezept-FdV: E-Rezepte lokal löschen - Löschen	gemSpec_eRp_FdV
A_19979	E-Rezept-FdV – Kein Zugriff von Diensten Dritter auf personenbezogene medizinische Daten	gemSpec_eRp_FdV
A_19178	E-Rezept-FdV – Schutzmaßnahmen gegen die OWASP-Mobile-Top-10-Risiken	gemSpec_eRp_FdV
GS-A_5322	Weitere Vorgaben für TLS-Verbindungen	gemSpec_Krypt
A_17207	Signaturen binärer Daten (ECC-Migration)	gemSpec_Krypt
A_19179	E-Rezept-FdV – Qualität verwendeter Schlüssel	gemSpec_eRp_FdV
GS-A_5542	TLS-Verbindungen (fatal Alert bei Abbrüchen)	gemSpec_Krypt
A_20625	Anwendungsfrontend: Prüfung der Signatur des ID_TOKEN	gemSpec_IDP_Frontend
A_19177	E-Rezept-FdV – Anzeige von Protokolldaten	gemSpec_eRp_FdV
GS-A_5035	Nichtverwendung des SSL-Protokolls	gemSpec_Krypt
GS-A_4387	TLS-Verbindungen, nicht Version 1.0	gemSpec_Krypt
A_20624	Anwendungsfrontend: Prüfung der Signatur des AUTHORIZATION_CODE	gemSpec_IDP_Frontend
A_19181	E-Rezept-FdV – Privacy bei default	gemSpec_eRp_FdV
GS-A_4384	TLS-Verbindungen	gemSpec_Krypt
A_20079	Ausfall der Fehlermeldung des Token-Endpunktes	gemSpec_IDP_Frontend

#8 Vorfallreaktionsplan

Ziel: Reduzierung des Schadens und schnelle Schwachstellenbehebung

- Kontaktpersonen und Maßnahmen müssen im Vorfeld festgelegt sein
- Überwachung und Bewertung der bekanntgewordenen Schwachstellen jeden Sprint
- Vorbereiteter Reaktionsplan



Herausforderungen

Herausforderungen

1. Alle Reviews konsequent durchführen
2. Geeignete SAST-Werkzeuge finden
3. Dokumentation und Nachweise pflegen

Geht‘s besser?

Weitere Optimierungen

1. DAST / Fuzzying
2. Effizientere Reviews
3. Bedrohungsanalysen in den Workflow integrieren

Danke!

Dr. Alexey Tschudnowsky
Twitter: @calexey
Mail: alexey.tschudnowsky@gematik.de

Bildquellen

- https://pixabay.com