Woche 08
Juristische Qualität und CI

Wir sichern unserem Auftraggeber die Freiheit von Rechtsmängeln zu. Begehen wir hier einen Fehler, müssen wir solche Mängel auf eigene Kosten beheben im Rahmen der Gewährleistung. Rechtsmängel können leicht entstehen, wenn wir die Lizenzbedingungen einzelner Bibliotheken verletzten oder gegen die DSGVO verstoßen.

1. DSGVO

Die Datenschutzgrundverordnung ist Teil des europäischen Rechts.Diese ist für sie als Team relevant, wenn sie personenbezogene Daten verarbeiten. Beispiele für personenbezogene Daten sind: Name, Emailadresse, IP-Adresse, Wohnort, Autokennzeichen (Verstoß = Ordnungswirdigkeit). Kritisch sind besonders Daten zum Gesundheitszustand, der geschlechtlichen oder politischen Orientierung von Personen (Verstoß = Staftat). Gehen sie im Team noch mal kritisch die von ihnen gespeicherten Daten durch, haben sie irgendwo personenbezogene Daten?

- Welche Daten, die sie erheben sind personenbezogen? Erheben Sie Name, Anschrift, E-Mail-Adressen, IP-Adressen, Fotos, oder andere Daten die auf einzelne Personen Rückschlüsse zulassen?
- Sind die von Ihnen erhobenen personenbezogenen Daten tatsächlich ***minimal***? Brauchen Sie diese wirklich um den Vertrag mit dem Benutzer zu erfüllen?
- Wozu genau brauchen Sie die erhobenen personenbezogenen Daten? Könnten sie diese auch durch allgemeine / anonymisierte Daten ersetzen?
- Wann wann müssen sie die Daten wieder Löschen? Welche Verwendungsarten hat ihnen der Benutzer tatsächlich erlaubt? Wie lange müssen sie die Daten aufheben, gibt es ggf. gesetzliche Fristen, die sie einhalten müssen?
- Wären sie in der Lage, einem Benutzer auskunft darüber zu erteilen, welche Daten genau über ihn / sie gespeichert sind?

Erstellen Sie im Team ein ***Verarbeitungsverzeichnis*** für die von ihnen gespeicherten personenbezogenen Daten. Die nachfolgenden Beispiele sind in Zusammenarbeit mit Prof. Dr. Hüttl entstanden. Überlegen sie welche Daten sie speichern, wann diese gelöscht werden können / dürfen / müssen. In derselben Tabelle dokumentieren sie, warum sie den Datensatz speichern müssen und wer darauf zugreifen kann, also zwei weitere Spalten.

Datenlöschung

Nr	Datensatz	Kriterien für Löschung	Gesetzl. Löschfrist	Gesetzl. Aufbewahrungsfrist
1-10	Bewerberdaten	Keine Einwilligung	Sobald Stelle besetzt	??
11-13	HR Mitarbeiter	Ausscheiden	??	10 Jahre (?)

2. TOM: Threat Modeling

Die DSGVO schreibt vor, dass sie aktuelle „Technische und Organisatorische“ Maßnahmen (TOM) ergreifen um den Schutz der Daten abzusichern. Gehen sie noch mal kritisch ihre Verteilungsarchitektur / Container-View (nach C4) durch. Haben sie alle wichtigen „Technischen und Organisatorischen Maßnahmen“ ergriffen, damit die personenbezogenen Daten bei Ihnen geschützt sind? Z.B. Übertragungskanal verschlüsseln, 2-Faktor Authentisireung, …

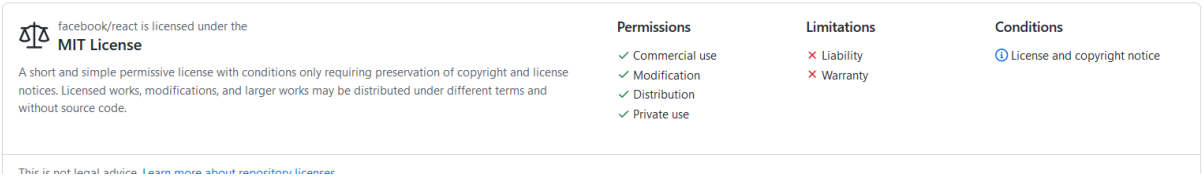
Für den Moderator: Zeichen sie die Verteilungsarchitektur auf (collaboard/miro oder Whiteboard) und lassen sie das Team mögliche Angriffe auf diese Architektur brainstormen (wenn sie Spaß an dem Thema haben, verwenden sie die Karten von Adam Shoestack). Jedes Teammitglied stellt seine Angriffsmöglichkeiten kurz vor. Versuchen sie die Angriffe nach möglicher Schadenhöhe zu sortieren und starten sie mit dem teuersten Angriff. Sie diskutieren im Team, ob ihre Architektur dagegen abgesichert ist.

3. Lizenzen (-> Lizenzauflagen, ggf. Lizenzscanner verwenden)

Wenn Sie sich für ein Frameworks verbaut haben sollten, ist es für Ihren Kunden eventuell relevant, welche Lizenzmodelle dort verwendet werden. Schauen sie z.B. noch mal bei https://openhub.net/ oder auf github dort sehen sie unter welcher Lizenz ihre Framework laufen. Erstellen sie eine Tabelle mit den genutzten Lizenzmodellen (hier kann ihnen die Gitlab-CI-Pipeline helfen, siehe unten):

Framework	Beschreibung	Version	Lizenz
React	Für die Entwicklung der Frontend-Anwendungen wird das React Framework verwendet. Dies entspricht der Vorgabe der Auftraggeber.	17.0.2	MIT
Spring	Für die Entwicklung des Java Backends wird das Spring Framework verwendet. Spring bietet sich an, da es weit verbreitet ist und	5.3.15	Apache 2.0

Prüfen sie für jedes Lizenzmodell, welche Lizenzauflagen es gibt, diese erhalten sie sehr schön übersichtlich bei github bzw. openhub. Das nachfolgende Beispiel stammt von github und zeigt die Auflagen der MIT Lizenz. Gehen sie die Lizenzen in der Gruppe durch und überlegen, was für sie zu tun ist.



In vielen Lizenzmodellen müssen sie den Lizenztext in der Lieferung beilegen.

4. CI-Pipeline (nach dem Workshop)

Eine CI-Pipeline sollten sie eigentlich schon seit der ersten Zwischenpräsentation haben. Beauftragen Sie ein Teammitglied eine „fortgeschrittene“ Version der Pipeline zu erstellen. Dieser Teil sollte nur selten ausgeführt werden z.B. bei einem Merge vom „develop“ auf den „main“ Branch. Folgende Prüfungen könnten für ihre Pipeline noch nützlich sein, Gitlab bietet hierzu passende Templates an:

1. Lizenzscanner, liefert eine Liste mit den verwendeten (gefundenen) Open Source Biblioteken
2. Secret Scanner: haben sie irgendwo im Repo einen Access-Key, Secret, Credential vergessen?
3. SAST Analyse: Statische Code Analyse nach Sicherheitslücken
4. Dependency Analyse: hat eine ihrer Bibliotheken bekannte Sicherheitslücken (CVE-Analyse)

SonarQube sollten sie nach Möglichkeit ebenfalls integrieren, um die Code Qualität abzusichern (vgl. letztes Aufgabenblatt). Auch die mögliche Testautomatisierung sollte von der Pipeline aufgerufen werden.