OTTO-FRIEDRICH-UNIVERSITÄT
BAMBERG

Qualitative Analyse von Stakeholdern in einem Unternehmen zu Sichtweisen auf Privatsphäre bei Softwareentwicklern bzw. -administratoren

VON

Samet Murat Akcabay

Lehrstuhl für Privatsphäre und Sicherheit in Informationssystemen

2. August 2020

Zusammenfassung

Diese qualitative Analyse beschäftigt sich mit den Sichtweisen auf Privatsphäre und Sicherheit von Stakeholdern in einem Softwareentwicklungs-Unternehmen. Mit der stetig wachsenden Digitalisierung spielt der Datenschutz, und somit die Privatsphäre und Sicherheit, eine immer größere Rolle bei Nutzern des Internets. Aus praktischer Sicht sind die Sichtweisen auf diese unterschiedlich, weswegen bei der Recherche darauf Wert gelegt wurde, möglichst viele Experten aus unterschiedlichen Berufsgruppen (z.B. Werkstudenten, Festangestellte, Team-Leiter o.Ä.) einzubeziehen und deren Aussagen auszuwerten. Nach erfolgreicher Auswertung der Expertenbefragungen hat sich ergeben, dass die Berufsgruppe, damit verbunden die Arbeitsweise und der -umfang, [...] keinen Einfluss auf die Einstellung gegenüber personenbezogenen Daten und die Verarbeitung dieser in DevOps-Tools haben.

Inhaltsverzeichnis

1 Die Relevanz der Privatsphäre im 21. Jahrhundert - eine Einleitung

„We believe privacy is a fundamental human right“ [**WWDC:20**] - mit dieser Aussage betritt Craig Federighi, der Vizepräsident der Softwareentwicklung von Apple, einem Multimilliarden-Unternehmen der Software- und Hardwareentwicklung aus den Vereinigten Staaten, zur Vorstellung der neuen Privatsphäre-Richtliniendes Unternehmens zur sogenannten, firmeneigenen „Worldwide Developer Conference“ die Bühne.
Und dieser Glaube ist nicht unbegründet: Mit der zunehmenden Digitalisierung in den letzten Jahrzehnten werden Menschen, und Internetnutzer im Spezifischen, häufiger vor dieses Dilemma gestellt. Entwicklungen wie dem Internet der Dinge (IoT), Industrie 4.0, „Smart Home“, Wearables (diverse Sensoren an der Kleidung selbst, Smartwatches, Fitnesstracker) uvm. ermöglichen neue (und zum Teil unerforschte) Angriffsmöglichkeiten für Cyberkriminalität [**BLB:18**]. Dies zeichnet sich vor allem durch den Anstieg der Höhe der Fallzahlen und der gleichzeitigen Abnahme der Aufklärungsquote aus: Im ersten Halbjahr 2018 wurden im Durchschnitt 13.000 Malware-Samples am Tag neu entdeckt [**GDB:18**], während diese Zahl 2017 noch bei [...] lag.
Ähnlich sieht es hier bei dem Anstieg neuer Schadprogramme aus: Der G Data Blog, ein deutsches Softwareunternehmen, welches mehrfach für seine Sicherheitslösungen ausgezeichnet wurde [**GD+1**], beschreibt dabei, dass die Zahl der neuen Schadprogrammtypen seit 2007 einen Anstieg um das knapp 63-fache vermerken konnte [**GDB:17**]. Es ist anzunehmen, dass dies sicherlich auch damit zusammenhängt, dass das Internet in den letzten beiden Jahrzehnten einen regelrechten Ausbruch in der Nutzung erlebt hat. Während die Computer- und Internetnutzung 2012, einem Jahr, in welchem im Durchschnitt bereits knapp 46% der Generation Plus50 Internetnutzer waren []
Nun könnte man davon ausgehen, dass sich das Internet etabliert

2 Theoretischer Hintergrund

Um in den folgenden Kapiteln ein grundlegendes Verständnis der Begrifflichkeiten, Zusammenhänge und Konzepte gewährleisten zu können, ist es notwendig, diese vorab zu definieren und einzugrenzen. Da sich diese Arbeit mit der Privatsphäre von Stakeholdern in einem Softwareentwicklungsunternehmen beschäftigt, ist es zunächst erforderlich, die Grenzen der Privatsphäre klar zu definieren und anhand dieser, eine Messung zu schaffen. Des weiteren müssen die Stakeholder, welche in dieser Analyse betrachtet werden, identifiziert und im weiteren Verlauf in Gruppen gegliedert werden. Im Anschluss werden DevOps definiert und anhand gewählter Beispiele, eine Analyse dieser aufgebaut.

2.1 Die Privatsphäre - eine Eingrenzung

Der Begriff Privatsphäre ist im Wortschatz der deutschen Sprache tief verankert und findet im Alltag in vielerlei Hinsicht Gebrauch: Menschen decken Notebook- und Smartphonekameras ab, protestieren gegen Überwachungskameras im öffentlichen Raum [**Stallwood:2013aa**] oder möchten nicht, dass ihre persönlichen und privaten Daten auf sozialen Netzwerken ohne ihre Zustimmung bzw. ohne ihre Kenntnis verbreitet werden [**Picchi:2018aa**] - doch was genau bedeutet Privatsphäre?
Die Privatsphäre lässt sich definieren als einen eingegrenzten, nichtöffentlichen Raum, in welchem ein Individuum bzw. Individuen nach eigenem Belieben, ohne äußere Einflüsse oder die Beobachtung durch Unbeteiligte, zur freien Entfaltung der eigenen Person, handeln kann [**Pettinger:2020aa**]. Möchte man diese Definition nun auf die Softwareentwicklung widerspiegeln, so müssen die möglichen, vorherrschenden Komponenten von persönlichen Daten näher betrachtet werden: In Unternehmen, welche sich mit der Softwareentwicklung befassen, müssen Daten vorhanden sein, welche Individuen individuell und bestimmen können (Grund liefern!) - dies kann in Form von IDs und individuell gewählten Nutzernamen vorzufinden sein, aber auf der Kehrseite auch mit Klarnamen, E-Mail Adressen sowie eindeutig identifizierbaren IDs.

2.2 Stakeholder

Als Stakeholder werden jene „Personen, Gruppen oder Institutionen bezeichnet, die von den Aktivitäten eines Unternehmens direkt oder indirekt betroffen sind oder [...] ein Interesse an diesen [...] haben“ [**Fleig:2016aa**] - dies kann von Kunden und Lieferanten bis zu den eigenen Mitarbeitern und Eigentümern reichen [**Fleig:2016aa**].

2.3 DevOps

„Die Zufriedenheit unserer Kunden liegt uns am Herzen“ - durch dieses Motto strahlen Unternehmen ihre Kundenorientierung aus und möchten in der Regel zusätzlich durch Feedback, Kulanz, einem überzeugenden Produkt oder einem allgemein positiv zurückgebliebenem Eindruck beim Kunden überzeugen. In Softwareentwicklungsunternehmen vereint DevOps die benötigten Technologien, die Prozesse und die Menschen miteinander, um für diese Kunden andauernd hochwertige Produkte anbieten zu können [**MSAzure:2020aa**]. Der Begriff setzt sich zusammen aus „Development“ (dt. *Entwicklung*) von Software sowie „Operations“ (dt. *Vorgänge*), welche damit zusammenhängen [**MSAzure:2020aa**] und repräsentiert dabei die „Zusammenarbeit von Entwicklung und Betrieb“ [**Hasselbring:2015aa**] dar. Dies erfolgt durch die Einführung von DevOps-Methoden und -Tools, sodass den zuvor getrennten Bereichen, und zusätzlichen wie beispielsweise der Qualitätssicherung, der Sicherheit etc., die Möglichkeit zur Koordination und Zusammenarbeit geboten wird [**MSAzure:2020aa**].

Im Folgenden werden beispielhaft repräsentative DevOps-Methoden und -Tools definiert und aufgelistet.

2.3.1 Continuous Integration und Continuous Delivery

In der Softwareprogrammierung ist es mit zunehmender Komplexität und Umfang des Programms üblich, mehrere einzelne Module in dieses einzubinden, weswegen sich die gängige Vorgehensweise in der Softwareentwicklung insofern gestaltet, dass das Gesamtprojekt, welches aus diesen einzelnen Modulen besteht, durch mehrere Teams parallel vorangetrieben [**HJL:2018aa**]. Statt auf eine täglich erfolgende Kompilierung auf einem extra angelegten „Build“-Server warten zu müssen, kommt heutzutage in modernen Unternehmen die sogenannte „Continuous Integration“ (dt. *Kontinuierliche Integration*) zum Einsatz [**HJL:2018aa**]: Diese bezeichnet „eine Technik der agilen Softwareentwicklung [...], [welche die] in kleinen Schritten vorgenommenen Änderungen [...] regelmäßig zusammenführt“ [**Dirk:2018aa**], also eine kontinuierliche Integration von Softwarekomponenten wie z.B. eines Guided User Interfaces (dt. *grafische Benutzeroberfläche*), eines Application Programming Interfaces (dt. *Programmierschnittstelle*) etc., in das gemeinsame Projekt bzw. den gemeinsamen Code [**HJL:2018aa**].
In der Arbeit als Team kann der Code dann für dasselbe Projekt also nicht

erst wie zuvor nach dem Fertigstellen der anderen Teilbereiche mit in den Gesamtcode des Projekts aufgenommen werden, sondern nach eigenem Bedarf, sogar mehrmals am Tag [**Dirk:2018aa**]. Dadurch kann der Code für die beteiligten Programmier schon viel früher offengelegt werden, sodass Tests und damit verbunden, Fehlerbehebungen früher und effizienter durch Einsparung von Zeit und Ressourcen, durchgeführt können - trotzdem müssen diese sich zunächst Umorientieren, da ein bereits gängiger Prozess neu aufgelegt und zusätzliche Prozesse sowie Server neu eingebunden werden müssen [**HJL:2018aa**; **Dirk:2018aa**]. Dieser gesamte Prozess funktioniert durch das Einspeisen der zusammengefügten Änderungen des Codes in ein Repository, womit im Anschluss eine Versionskontrolle durch die Übernahme von einer aktuellen Version des Codes durch einen automatisierten Build-Prozess [**HJL:2018aa**].

Der Begriff „Continuous Delivery" (dt. *Kontinuierliche Auslieferung*) geht einen Schritt weiter: Eine Sammlung, welche „kurze Entwicklungszyklen und [eine] schnelle Auslieferung von Software-Updates [durch Techniken, Prozesse und Werkzeuge]" [**Ilanrr:2017aa**] ermöglicht, beschreibt man als Continuous Delivery. Anstatt konventionell darauf zu setzen, Software erst zu programmieren, dann einem Test und einer Evaluation zu unterziehen, bevor diese an Kunden ausgeliefert wird, kann durch Continuous Delivery eine vollfunktionsfähige Version der Software in jedem Entwicklungsstand nach Bedarf vom Kunden bezogen werden - in der Regel ohne Fehler zu enthalten [**Ilanrr:2017aa**].

2.3.2 Continuous Delivery vs. Continuous Deployment

2.3.3 Versionsverwaltung

Durch die Zusammenarbeit im Team an einem gemeinsamen Projekt werden Softwareentwickler häufig vor das Problem der Versionsverwaltung gestellt: Wie können mehrere Personen an einem gemeinsamen Projekt und z.T. sogar am selben Programm gleichzeitig arbeiten? Wie können verschiedene Versionen eines Programms einheitlich und übersichtlich verwaltet werden? Und wie verfährt man möglichst effizient mit Code, welcher von unterschiedlichen Autoren geschrieben wurde und fügt diese zusammen? Zur Lösung dieser Konflikte ist die Versionsverwaltung heutzutage in Softwareentwicklungsunternehmen essentiell. Diese „ist ein System, welches die Änderungen an einer oder einer Reihe von Dateien über die Zeit hinweg protokolliert, sodass man später auf eine bestimmte Version zurückgreifen kann." [**Scott-Chacon:2020aa**]. Laut der Webseite Open Hub, welche zur Katalogisierung von Open-Source-Software verwendet wird und wie hier, den Anteil von Versionverwaltungstools anhand der registrierten Daten indexiert, stellen Git (71%) und Subversion (24%) die Spitze dar [**Inc.:2020aa**].
Git unterscheidet hierbei zwischen der lokalen, zentralen und verteilten Versionsverwaltung: Die lokale Versionsverwaltung befasst sich dabei mit der „[Verwaltung] aller Änderungen relevanter Dateien in einer Datenbank" [**Scott-Chacon:2020aa**], wohingegen die zentrale mit dem Problem der Kooperation mit anderen Entwicklern an gemeinsamer

Software [**Scott-Chacon:2020aa**]. Der letzte Punkt der verteilten Verwaltung behandelt das vollständige Kopieren eines sogenannten „Repositories“ (dt. *Quelle bzw. Lager von Software*) von einem Server anstatt lediglich den letzten Stand zu beziehen [**Scott-Chacon:2020aa**].

3 Forschungsmethode

Dabei wurde die Vorgehensweise zur Durchführung von Experteninterviews gewählt: Hierzu wurden einzelne Softwareentwickler von diversen Unternehmen sorgfältig anhand ihrer Qualifikation, Berufserfahrung und der Position im Unternehmen gewählt - um ein möglichst weites Spektrum an unterschiedlichen Personengruppen abzudecken, wurden diese in Berufseinsteiger, welche kaum bis relativ wenig Berufserfahrung vorweisen können, mehrjährig Festangestellte Softwareentwickler und Team-Leads bzw. Projektleiter, welche zusätzlich noch verantwortlich für Entwicklergruppen sind, untergliedert.

3.1 Experteninterviews als Mittel zur Analyse

3.2 Identifikation der Stakeholder in einem Softwareentwicklungsunternehmen

Ziel dieser qualitativen Analyse ist es, die Denk- bzw. Sichtweisen von Stakeholdern in Softwareentwicklungsunternehmen zu erfassen und anhand dieser Ergebnisse, Schlüsse in Bezug auf die eigene Privatsphäre dieser zu ziehen. Um ein grundlegendes Verständnis in den folgenden Kapiteln gewährleisten zu können ist es zunächst notwendig, diese Stakeholder zu definieren und anhand davon, potenzielle Stakeholder im Rahmen dieser Forschung zu identifizieren.
Im Falle dieser Analyse wird die Sichtweise der einzelnen Stakeholder auf die eigene Privatsphäre im angestellten Unternehmen betrachtet. Gemessen wird dieser Aspekt durch die Relevanz, wie viel Wert die befragten Softwareentwickler beispielsweise darauf legen, personenbezogene Daten in einem möglichst weiten Spektrum für einen möglichst langen Zeitraum einsehen zu können und auf der Kehrseite, wie weit sie bereit sind, eigene personenbezogene Daten preiszugeben bzw. mit Kollegen und anderen Stakeholdern zu teilen, um ein gut funktionierendes Glied eines Teams oder eines Unternehmens sein zu können.

In den folgenden Unterkapiteln werden die untergliederten Interessensgruppen und ihre entsprechende Notwendigkeit näher erläutert. Die Begründung für die spezifische Wahl dieser besagten Gruppen wird im nächsten Kapitel klassifiziert.

3.2.1 Berufseinsteiger, Praktikanten und Werkstudenten

Die primäre Gruppe zeichnet sich durch Softwareentwickler aus, welche kaum bis wenig Berufserfahrung besitzen und eventuell geringere Qualifikationen, als die anderen Interessensgruppen aufweisen könnten. Diese Merkmale finden sich in Berufseinsteigern wieder, welche vor Kurzem aus einer abgeschlossenen Ausbildung oder eines abgeschlossenen Studiums stammen und nun im Beruf Softwareentwickler sind. Selbe Punkte lassen sich allerdings auch auf (längerfristige) Praktikanten und Werkstudenten, welche neben ihrem Studium in einem Unternehmen bis zu 20 Stunden in der Woche tätig sind, übertragen - es ist also wenig Berufserfahrung vorhanden, aber grundlegende Kenntnisse in der Informatik, vor allem in Bezug auf DevOps, welche im nächsten Unterkapitel näher erläutert werden, lassen sich in diesen Interessensgruppen wiederfinden.

3.2.2 Softwareentwickler

In der sekundären Gruppe befinden sich Softwareentwickler, welche als Festangestellte mehrjährige Erfahrungen in der Entwicklung in einem Unternehmen gesammelt haben. Diese sind in den meisten Fällen Teil eines Teams, welche gemeinsam an einem Projekt arbeiten - im weiteren Verlauf der Analyse wird diese Personengruppe als „Softwareentwickler“ genannt. Aufgrund der gesammelten Erfahrung hat diese Gruppe in der Regel auch häufig mit DevOps Kontakt und kann so begründetere und für die Analyse signifikantere Aussagen als die primäre Gruppe treffen. Die grundlegenden Kenntnisse in der Informatik reichen weiter durch regelmäßige Anwendung in der Praxis, weswegen auch fachspezifischere Fragen mit Hinblick auf die Erfahrungen gestellt werden können.

3.2.3 Team-Leads, Projektleiter (und Geschäftsführer?)

3.2.4 Externe Ansprechpartner

4 Ergebnisse

Dies ist das Kapitel, in welchem ich meine Ergebnisse präsentiere und evaluiere.

5 Diskussion der Ergebnisse

5.1 Implikationen in der Theorie

5.2 Implikationen in der Praxis

5.3 Limitation der Arbeit

6 Fazit und Ausblick

Dies ist das Kapitel, in welchem ich meine Bachelorarbeit abschließe und ein Fazit angebe.

Literaturverzeichnis

Eidesstattliche Erklärung

Ich erkläre hiermit gemäß § 17 Abs. 2 APO, dass ich die vorstehende Bachelorarbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Bamberg, den _____ _____

Samet Murat Akcabay