EU-Vorschriften zu mehr Cybersicherheit

Die Europäische Union bringt im Rahmen ihrer Digitalstrategie zwei wichtige Gesetze auf den Weg: den Cyber Resilience Act (CRA) sowie die sogenannte NIS2-Richtlinie.Da wir immer digitaler werden, muss auch immer mehr Wert auf Onlinesicherheit, oder wie es auf Neudeutsch heißt, Cybersecurity, gelegt werden.

Die EU-Kommission hat am 15. September 2022 einen Entwurf des CRA vorgeschlagen, der noch vom europäischen Parlament und vom Rat angenommen werden muss.

Recht auf Updates durch den CRA

Mit dem CRA werden erhöhte Sicherheitspflichten auf Hersteller, Vertreiber, Importeure und Händler von IT-Produkten zukommen. Dazu zählen insbesondere geplante Meldepflicht für aktiv ausgenutzte Schwachstellen und Sicherheitsvorfälle. Durch die Vorgaben des CRA sollen sicherere Hardware- und Softwareprodukte gewährleistet werden.

Zu den dabei erfassten Produkten zählt jedes Software- oder Hardwareprodukt sowie seine Datenfernverarbeitungslösungen, einschließlich Software- oder Hardwarekomponenten, die separat in Verkehr gebracht werden. Er ist anwendbar auf „Produkte mit digitalen Elementen“, also auf solche Produkte, die ohne ihre digitalen Elemente nicht sinnvoll genutzt werden können (z. B. Smartphones). Der CRA teilt die Produkte mit digitalen Elementen in drei Kategorien ein:

- Standardkategorie
- kritische Klasse I
- kritische Klasse II

In die Standardkategorie fallen voraussichtlich gut 90 Prozent aller Produkte, wie z. B. Textverarbeitung, Fotobearbeitung oder Festplatten. Zum CRA gehören verschiedene Anhänge. Darin, nämlich in Anhang III, werden die kritischen Produkte mit digitalen Elementen der Klassen I und II aufgeführt. Zur kritischen Klasse I zählen u. a. Software für Identitätsmanagementsysteme, Browser, Passwortmanager, Antivirensoftware, VPN-Lösungen, Netzwerkmanagementsysteme, Werkzeuge zur Verwaltung der Netzwerkkonfiguration, Systeme zur Überwachung des Netzwerkverkehrs, Verwaltung von Netzwerkressourcen, Systeme zur Verwaltung von Sicherheitsinformationen und -ereignissen (SIEM), Update-/Patch-Verwaltung (einschließlich Bootmanager), Systeme zur Verwaltung der Anwenderkonfiguration, Software zur Verwaltung mobiler Geräte, Firewalls, Router/Modems, Anwenderspezifische integrierte Schaltungen (ASIC) oder auch Industrielle Automatisierungs- und Steuerungssysteme (IACS). Zur kritischen Klasse II zählen hingegen insbesondere Betriebssysteme für Server, Desktops und mobile Geräte, Infrastrukturen für öffentliche Schlüssel und Aussteller digitaler Zertifikate, Hardwaresicherheitsmodule (HSM), sichere Kryptoprozessoren, Smartcards, Smartcard-Lesegeräte und Token oder auch Geräte des industriellen Internets der Dinge.

Folgende Pflichten sollen zukünftig auf die vom Anwendungsbereich des CRA erfassten Unternehmen zukommen:

- Berücksichtigung der Cybersicherheit schon in der Planungs-, Entwurfs- und Entwicklungsphase sowie auch in der Produktions-, Liefer- und Wartungsphase („Security by Design“)
- umfangreiche Dokumentationspflichten in Bezug auf Cybersicherheitsrisiken
- Meldepflicht für aktiv ausgenutzte Schwachstellen und Vorfälle
- Überwachungs- und Beseitigungspflichten von Schwachstellen während der erwarteten Produktlebensdauer

(max. fünf Jahre)
- Pflicht zur Lieferung von klaren und verständlichen Gebrauchsanweisungen
- Pflicht zur Bereitstellung von bestimmten Pflichtinformationen (u. a. Name, Anschrift und Kontaktdaten des Herstellers, Typen-, Chargen-, Versions- bzw. Seriennummer, Verwendungszweck, Art der technischen Sicherheitsunterstützung, die der Hersteller anbietet, sowie der Zeitpunkt, bis zu dem sie geleistet wird)
- Pflicht zur Bereitstellung von Sicherheitsupdates für jedenfalls fünf Jahre

Ganz konkret und unabhängig von der jeweiligen Kategorie müssen Produkte mit digitalen Elementen zudem immer einer Risikobewertung unterzogen werden.

Unter die Produkte mit digitalen Elementen im Sinne des CRA fallen jedoch weder „Produkte mit digitalen Inhalten“ noch „digitale Produkte“. Die Erstgenannten zeichnen sich dadurch aus, dass der digitale Teil des Produkts für dessen Funktionsfähigkeit nicht von zentraler Bedeutung ist (z. B. Kühlschrank mit Bestellfunktion via App). Bei den „digitalen Produkten“ handelt es sich um rein digitale Produkte (z. B. Apps oder Musikdateien). Der CRA hat folglich einen sehr weit gefassten Anwendungsbereich, von dem nur ein paar spezifische Produktkategorien ausgenommen werden, wie beispielsweise Medizinprodukte. Software, die als Dienstleistung angeboten wird, also Software-as-a-Service-bzw. Cloudleistungen, wird ebenfalls gesondert geregelt.

Sichere Infrastrukturen durch NIS2

Speziell auf den Bereich der sogenannten kritischen Infrastruktur (KRITIS) zielt die NIS2-Richtlinie ab. Es geht also um den besseren Schutz von Stromversorgung, Wasserwerken oder Telekommunikationsleitungen. Es soll ein Höchstmaß an

Ausfallsicherheit gewährleistet werden, um beispielweise Strom-Blackouts oder Störungen der Trinkwasserversorgung für die Bevölkerung möglichst zu vermeiden oder jedenfalls so schnell und gut wie möglich auf derartige Störungen reagieren zu können.

In Deutschland ist mit Blick auf den KRITIS-Sektor bereits 2015 das IT-Sicherheitsgesetz (IT-SiG) in Kraft getreten. Darin war auch eine Änderung des damaligen § 13 Telemediengesetz (TMG) enthalten, der in einem Absatz 7 die Pflicht für alle Websitebetreiber zur Absicherung ihrer Websites mit sich brachte (z. B. durch Verschlüsselung nach dem Stand der Technik per SSL-/TLS-Zertifikat). Diese Norm findet sich nach der letzten Änderung des TMG nun in § 19 Abs. 4 des Telekommunikations-Telemedien-Datenschutz-Gesetzes (TTDSG). Seit Mai 2021 ist nunmehr das zweite IT-Sicherheitsgesetz (IT-SiG2) in Kraft, welches sowohl den Adressatenkreis als auch den Pflichtenkatalog der KRITIS-Betreiber merklich erweitert hat.

Aber auch auf EU-Ebene tut sich einiges. 2016 ist die Richtlinie (EU) 2016/1148 des Europäischen Parlaments und des Rates vom 6. Juli 2016 über Maßnahmen zur Gewährleistung eines hohen gemeinsamen Sicherheitsniveaus von Netz- und Informationssystemen in der Union (kurz: NIS-Richtlinie) in Kraft getreten. Sie ist der europäische Rahmen für Cybersecurity im KRITIS-Bereich und soll ein hohes Sicherheitsniveau für Netzwerke und Informationssysteme sicherstellen.

Seit dem Jahr 2021 wird die NIS-Richtlinie überarbeitet. Ihr Nachfolger, die sogenannte NIS2-Richtlinie, soll den bestehenden Rechtsrahmen modernisieren, um die Herausforderungen des zunehmenden Grades an Digitalisierung und der stetig wachsenden Bedrohungen für die Cybersicherheit meistern zu können. Nach Inkrafttreten der NIS2-Richtlinie muss diese noch in das jeweilige nationale Recht der EU-Mitgliedsstaaten umgesetzt werden. In NIS2 wird zwischen

kritischen und wichtigen Einrichtungen unterschieden:

- *Kritische Einrichtungen:* Energie (Strom, Fernwärme und Fernkälte, Erdöl, Erdgas und Wasserstoff); Verkehr (Luft, Schiene, Wasser und Straße); Bankenwesen; Finanzmarktinfrastrukturen; Gesundheitswesen; Herstellung pharmazeutischer Erzeugnisse (einschließlich Impfstoffe und kritischer Medizinprodukte); Trinkwasserversorgung; Abwasserwirtschaft; digitale Infrastrukturen (Internetknoten, DNS-Anbieter, Anbieter von Clouddienstleistungen, Anbieter von Rechenzentrumsdiensten, Netze zur Bereitstellung von Inhalten, öffentliche elektronische Kommunikationsnetze und elektronische Kommunikationsdienste,…); öffentliche Verwaltung; Weltraum.
- *Wichtige Einrichtungen:* Post- und Kurierdienste; Abfallwirtschaft; Chemikalien; Lebensmittel; Herstellung anderer Medizinprodukte, von Computern, Elektronik und Kraftfahrzeugen sowie Maschinenbau; Anbieter digitaler Dienste (Onlinemarktplätze, Onlinesuchmaschinen und Plattformen der sozialen Netzwerke).

Sowohl die kritischen als auch die wichtigen Einrichtungen müssen u. a. folgende Cybersecurity-Maßnahmen treffen:

- Erlass und Umsetzung von Richtlinien für Risiken und Informationssicherheit
- Umsetzung von Maßnahmen zur Prävention, Detektion und Bewältigung von Cybersecurity-Vorfällen (Sicherheitspannen)
- Ergreifen von Maßnahmen zum Business Continuity Management (BCM) inkl. Backup- bzw. Krisenmanagement
- Gewährleistung der Sicherheit bei der Beschaffung von IT- und Netzwerksystemen
- Beachtung von Vorgaben für Kryptografie bzw. Verschlüsselung

- Umsetzung angemessener Maßnahmen zur Zugangskontrolle
- Einsatz sicherer Sprach-, Video- und Textkommunikation
- Einsatz gesicherter Notfallkommunikationssysteme

Durch die NIS2-Richtlinie wird der Aspekt der Cybersecurity zukünftig in der gesamten Lieferkette zu berücksichtigen sein. Außerdem sollen die Aufsicht und die Zusammenarbeit zwischen den Behörden und den von NIS2 betroffenen Betreibern innerhalb der EU vertieft werden. Die Sanktionen bei Verstößen gegen die NIS2-Vorgaben sollen durch die einzelnen EU-Mitgliedsstaaten selbst geregelt werden. Allerdings wird von Seiten des EU-Gesetzgebers bestimmt, dass die Sanktionen wirksam, verhältnismäßig und abschreckend sein müssen. Gegen kritische Einrichtungen sollen Geldbußen mit einem Höchstbetrag von mindestens 10 Mio. Euro oder von mindestens 2 Prozent des gesamten weltweit erzielten Vorjahresumsatzes verhängt werden können. Bei Sanktionen gegen wichtige Einrichtungen soll der Höchstbetrag mindestens 7 Mio. Euro oder 1,4 Prozent des Vorjahresumsatzes betragen.

Praxistipp

Neben dem CRA und NIS2 beinhaltet die Strategie der EU noch weitere Gesetze, die den Umgang mit digitalen Daten regeln sollen. Dazu zählen insbesondere der Data Governance Act (DGA) zur Förderung der Weiterverwendung von Daten des öffentlichen Sektors, der Digital Markets Act (DMA) und der Digital Services Act (DSA) zur Regulierung großer Onlineplattformen, der Artificial Intelligence Act (AIA) zur Regulierung von Künstlicher Intelligenz (KI) oder auch der Data Act (DA) zur besseren Weiterverwendung von Unternehmensdaten. Bei diesen Rechtsakten handelt es sich nicht um bloße Zukunftsmusik, denn der DMA ist bereits seit dem 1. November 2022 in Kraft. Der DGA wird ab dem 24. September 2023 anwendbar sein, der DSA bereits ab dem 2. Mai 2023.

- Michael Rohrlich hat als Rechtsanwalt und Fachautor seinen

Kanzleisitz in Würselen, Nähe Aachen. Seine beruflichen Schwerpunkte liegen auf dem Gebiet des Onlinerechts sowie des gewerblichen Rechtsschutzes. Weitere Infos zu den Themen aus den Rechtsbeiträgen sowie Gesetze und Gerichtsentscheidungen bietet er unter www.rechtssicher.info an.

Übersichten und unterschiedliche Listen

Links – Cybersecurity

- cybercrime magazine

Dienstleistungen

- Website-Pflege – was ist das?

-  analyse digitale präsenz

onlinesicherheit

-  onlinesicherheit, handbuch, österreich
-  gesetzliche verpflichtung zur Wartung von Webanwendungen, Profihost

Gute-Wichtige-Webseiten

-  JSON in CSV konvertieren

-  Effektives Security Awarness Training
-  distributor
-  distributor - österreich
-  hornetsec-preise-konkurrenz-deutschland
-  Open-Source-Monitoring mit Zabbix

-  Anbieter für Cloud-basierte IT-Sicherheitslösungen

WordPress Erweiterungen

-  Simples Link Directory
-  Introducing Table of Contents for Divi
-  Preis-Tabellen-Divi
-  Advanced Toggle

wordpress security-liste

-  Auflistung von WordPress - Sicherheitslücken
-  Diverse Auflistungen - WordPress
-  Wordpress - Hackerone
-  Wordpress Patches
-  Wordpress Sicherheits-APP

.

Hacking WordPress – Ein Blick hinter die Kulissen

Angreifen und Sichern von WordPress

Hacking WordPress – Ein Blick hinter die Kulissen

Wie Angreifer WordPress-Installationen hacken bzw. Schwachstellen in Plugins, Themes oder Konfigurationen ausnutzen.

1. Hilfe ich wurde gehackt!

Allein in Deutschland ist der **Verbreitungsgrad** von WordPress enorm – Tendenz weiter steigend. Im weltweiten Vergleich mit anderen Content Management Systemen (CMS) hält WordPress einen Anteil von bis zu 51% (Top Million Sites). Beeindruckende Zahlen also, die WordPress immer stärker in den Fokus von **professionellen Angreifern** rückt. So attackierte ein Botnet im April diesen Jahres WordPress-Installationen weltweit.

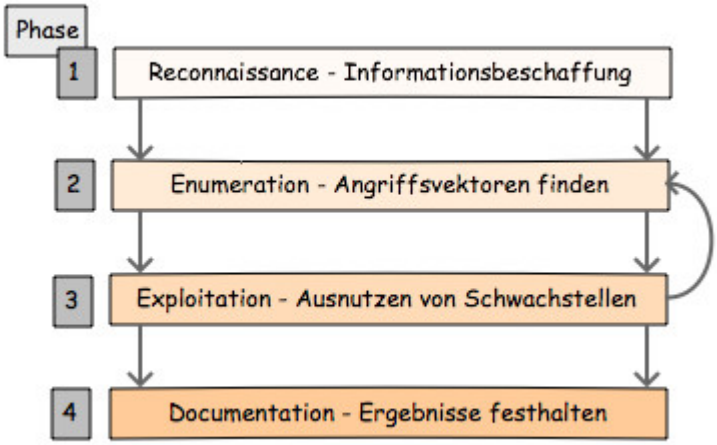
Zum Schutz und Absicherung von Installationen wurden bereits zahlreiche Anleitungen veröffentlicht. Empfehlenswert sind »Hardnening WordPress« oder auch meine Artikelserie »WordPress absichern«.

In diesem Beitrag werden allerdings keine weiteren **Schutzmaßnahmen** vorgestellt, sondern wie Angreifer vorgehen, um WordPress-Installationen zu hacken. Es soll ein kleiner Einblick hinter die **Kulissen** sein – in der Realität existieren weitaus mehr Möglichkeiten und Varianten.

2. Hinweis zu »Hacking WordPress«

Der Angriff von WordPress-Installationen oder Systemen ohne Erlaubnis bzw. Einverständniserklärung stellt eine **strafbare Handlung** dar. Wer ohne vertragliche Grundlage fremde Systeme angreift begibt sich auf sehr dünnes Eis. Die nachfolgenden Informationen dienen der Aufklärung und sollten lediglich im Rahmen eines Penetrationstests Verwendung finden. Im Gegensatz zu illegalen Hacking-Angriffen stellt ein Penetrationstest ein **auftragsgesteuerter** Einbruch in ein oder mehrere Systeme dar.

Das Vorgehen dient im Grunde der »Qualitätskontrolle« der aktuell umgesetzten IT-Sicherheit im Unternehmensumfeld.



Ein Angriff / Penetrationstest lässt sich in unterschiedlichen **Phasen** unterteilen, von denen ein Teil sequentiell wiederholt wird. Phase 1 dient zunächst der **Informationsgewinnung** über das Ziel. Während ein Penetrationstester in Phase 4 die Ergebnisse festhält, wird sich ein Angreifer diesen Schritt wohl eher sparen…

3. Informationsgewinnung – Phase 1

Im ersten Schritt wird ein Angreifer möglichst viele **Informationen** über sein Ziel sammeln, die für den weiteren Verlauf von Interesse sein können. Zu diesem Zweck werden verschiedene öffentlich verfügbare Informationsquellen durchsucht. Diese werden im Anschluss ausgewertet und sollen Aufschluss darüber geben, über welchen Weg ein Einbruch am **einfachsten** realisiert werden kann. Für diesen Zweck stehen unterschiedliche Tools zur Verfügung – die meisten davon befinden sich auf der Linux Distribution Kali. Die Distribution wird sowohl von **Hackern**, als auch von **Penetrationstestern** zur Auffindung von Schwachstellen / Sicherheitsanalysen eingesetzt.

Dabei helfen Tools die unter »Information Gathering« zusammengefasst sind. Letztendlich werden in der ersten Phase

folgende Ziele verfolgt:

- Ziel identifizieren
- System / Anwendungsversion bestimmen
- Verfügbare Netzwerk-Ports
- Laufende Services
- Verteidigungsstrategien erkennen
- […]

Du kannst den Blog aktiv unterstützen!

No Tracking. No Paywall. No Bullshit.

Die Arbeit von kuketz-blog.de finanziert sich zu 100% aus den Spenden unserer Leserinnen und Leser. Werde Teil dieser Community und unterstütze auch du unsere Arbeit mit deiner Spende.

Mitmachen →

3.1 Beispiel: WordPress Identifikation

Das Verstecken der **WordPress-Versionsnummer** oder sonstigen **Meta-Daten** wird bei Laien oftmals mit dem Schutz gegen Spambots oder Sicherheitslücken in Verbindung gebracht. In der Tat lassen sich damit die besonders »dämlichen« Bots an der Nase herumführen, aber bereits semi-professionelle Varianten lassen sich von den Security by Obscurity Maßnahmen nicht beirren. Sie benutzen ausgeklügelte Methoden zur Feststellung ob eine Seite mit WordPress betrieben wird.


```
[*] Num of checks set to: 100

------------------------------------------------------
[*] Input plugin list set to: wp_plugin_list_2013_feb.txt
[*] Num of threats set to: 10
------------------------------------------------------

==> Results for: http://[redacted] <==

[i] Wordpress version found:  3.5.2
[i] Wordpress last public version: 3.5.2


[*] Search for installed plugins


[i] Plugin found: google-sitemap-generator
    |_Latest version:  3.2.9
    |_ Installed version: 3.2.8

[i] Plugin found: jetpack
    |_Latest version:  2.1.2
    |_ Installed version: 2.3.1
    |_CVE list:
    |___CVE-2011-4673: (http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2011-4673)

[i] Plugin found: si-contact-form
    |_Latest version:  3.1.8.1
    |_ Installed version: trunk

[i] Plugin found: wp-pagenavi
    |_Latest version:  2.83
    |_ Installed version: 2.83
```

Wer selbst mal schauen möchte ob seine WordPress-Installation als solche erkannt wird kann folgende Webseite nutzen: [Is it WordPress?]

Mehr Informationen benötigt? Beispielsweise alle installierten Plugins? Auch gar kein Problem mit dem Tool [plecost]. Hier ein Fingerprint einer WordPress-Installation:

Mit Hilfe der gesammelten Informationen lässt sich WordPress bzw. eines der installierten Plugins gezielt angreifen. Details zu Schwachstellen für bestimmte Versionen stellt beispielsweise CVE-Details zur Verfügung.

3.2 Beispiel: System identifizieren

Nmap ist ein Werkzeug zum **Scannen** und **Auswerten** von Hosts in einem Netzwerk und fällt in die Kategorie der Portscanner. Der Name steht für Network Mapper. Nmap wird in erster Linie für Portscanning eingesetzt. Daneben verfügt es über weitere Techniken, wie beispielsweise die Erkennung des eingesetzten Betriebssystems (OS-Fingerprinting).

Letztendlich dienen solche Informationen wiederum als **Ausgangspunkt** für die weiteren Phasen, in denen Schwachstellen aktiv ausgenutzt werden.

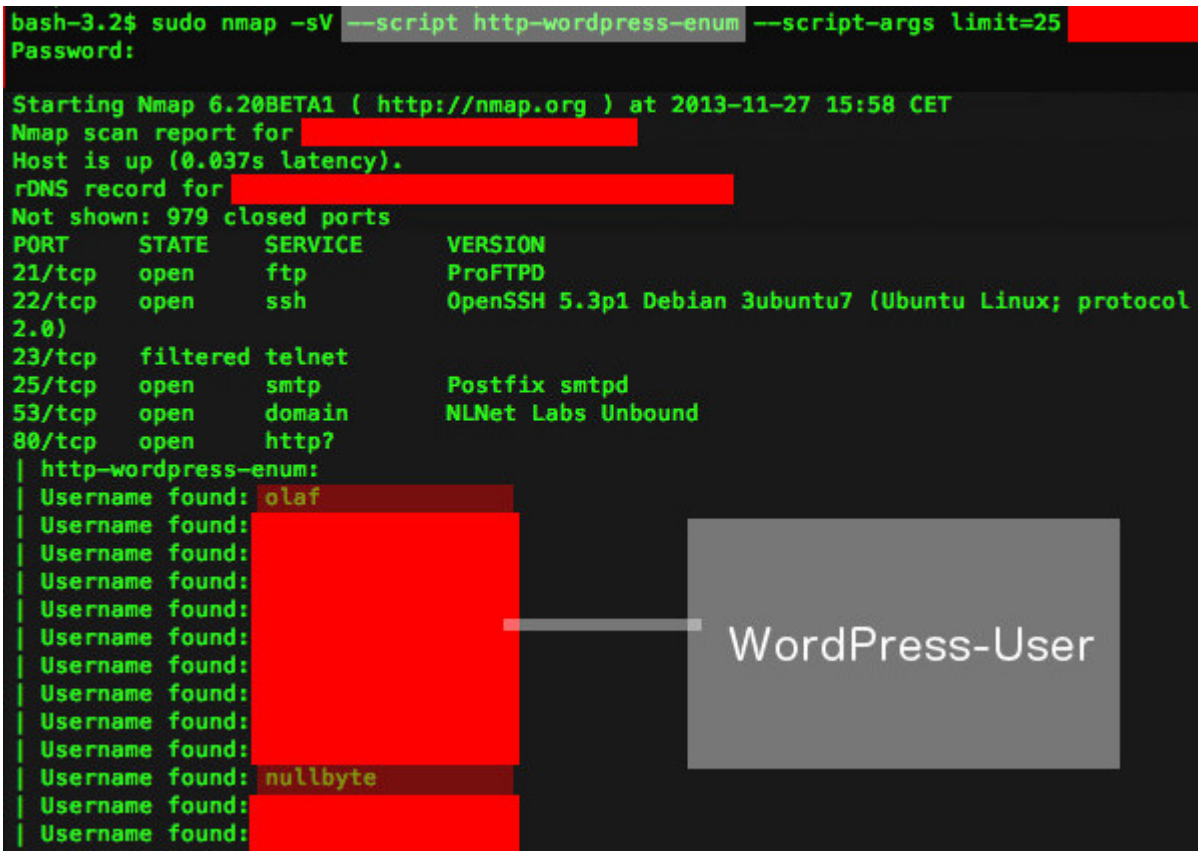
3.3 Beispiel: Erkennung von Benutzer-Accounts

Um sich in den **Administrationsbereich** von WordPress einzuloggen ist die Kombination aus einem Benutzernamen und Passwort erforderlich. Falls ein Angreifer im Vorfeld den Benutzernamen »erraten« kann, benötigt er im Anschluss lediglich das korrekte Passwort. Insgesamt erleichtert das ein erfolgreiches Eindringen in den sensiblen Administrationsbereich.

Oft genügt dazu die Eingabe von

wordpress-blog-adress.de/?author=1

in die Browser-Zeile. In der Standard-Installation bekommt ein Administrator / Nutzer eine eindeutige **Identifikationsnummer** zugewiesen. Meist endet diese auf **author=1** bzw. kann durch den Austausch der 1 am Ende leicht durchprobiert werden.



Falls der WordPress-Betreiber dies manuell geändert hat hilft ein Skript für nmap – wer probiert schon gerne alle

Kombinationen durch:

4. Angriffsvektoren finden – Phase 2

Ausgehend von den in Schritt eins gesammelten Informationen werden anschließend mögliche **Einstiegspunkte** in das System identifiziert. Mit Hilfe von Tools und manuellen Abfragen wird konkrekt nach Schwachstellen und Lücken gesucht, die einen Einbruch ermöglichen. Unter »Vulnerability Analysis« sind die benötigten Tools zusammengefasst und dienen folgenden Zielen:

- Schwachstellen identifizieren
- Identifizieren und priorisieren von System Zugangspunkten
- Risiken einschätzen
- […]

WordPress auf Schwachstellen und Konfigurationsfehler prüfen

Für Deine WordPress-Installation habe ich ein **spezielles** Leistungspaket im Angebot:

- Scan Deiner WordPress-Installation auf Schwachstellen
- Auswertung und Beurteilung der gefundenen Schwachstellen
- Auf Basis der Ergebnisse erhälst Du von mir individuelle Maßnahmenempfehlungen zur Behebung und Absicherung

Wenn du Deine WordPress-Installation **nachhaltig** absichern möchtest, kannst Du mich gerne kontaktieren.

Gut zu wissen: Sicherheit erlangst Du nicht durch die Installation unzähliger Security-Plugins, sondern durch eine saubere Konfiguration, stetige Updates und proaktive Maßnahmen

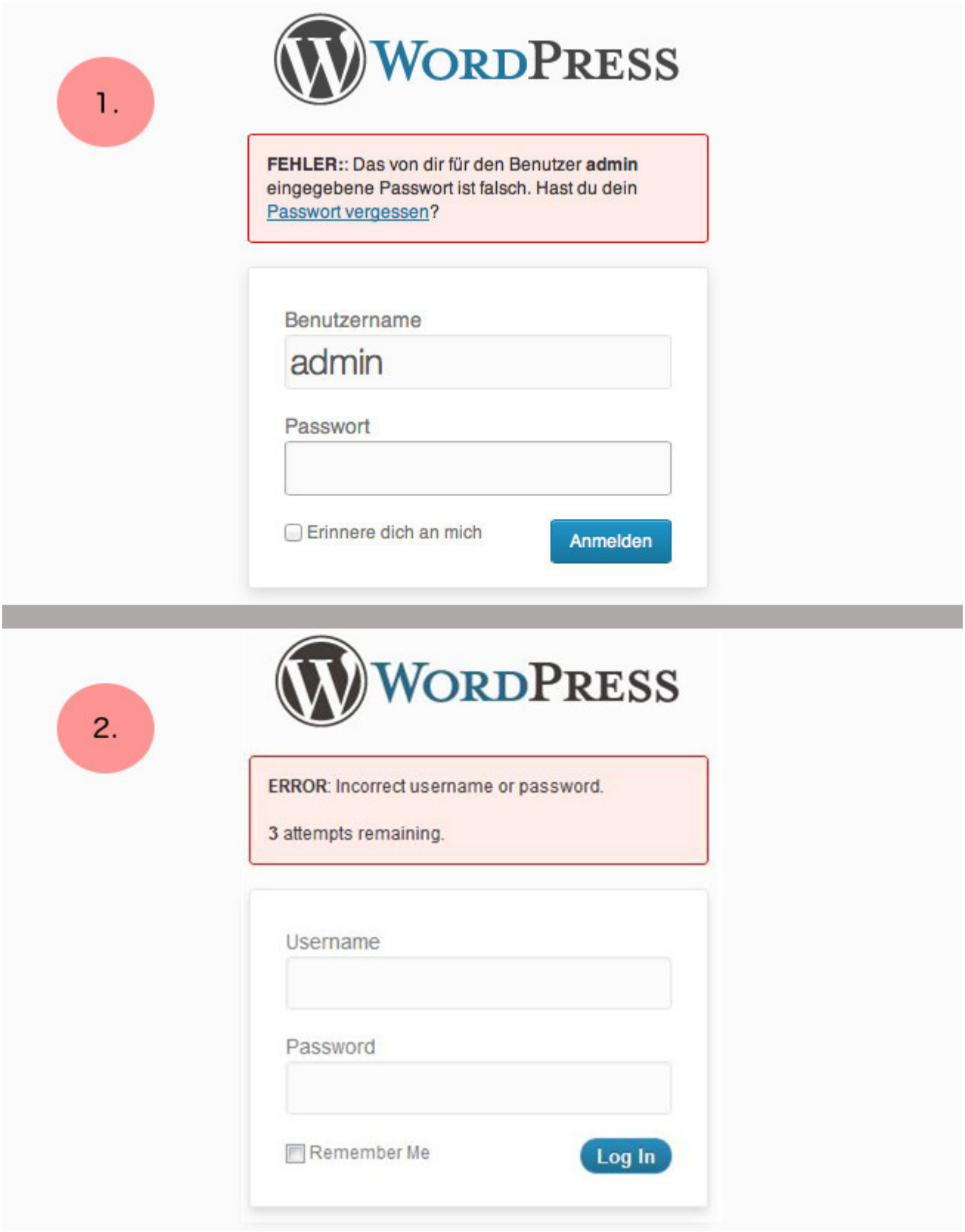
zur Absicherung. Kontakt aufnehmen

4.1 Administrationsbereich

Äußert beliebt als Einstiegspunkt ist der Login zum **Administrationsbereich** von WordPress – nicht zuletzt deswegen, weil sich ein Angriff bei vielen Installationen mit einfachen Mitteln bewerkstelligen lässt.

Über den Browser lässt sich prüfen, ob der Administrationsbereich generell für jeden erreichbar ist:

wordpress-blog-adress.de/wp-admin



Nach der Eingabe eines Benutzernamens und Passwort kann zunächst die Reaktion von WordPress erkundet werden.

Im ersten Beispiel wird der Account »**admin**« bestätigt. Dieser ist also vorhanden und dient vermutlich der Administration der

WordPress-Installation.

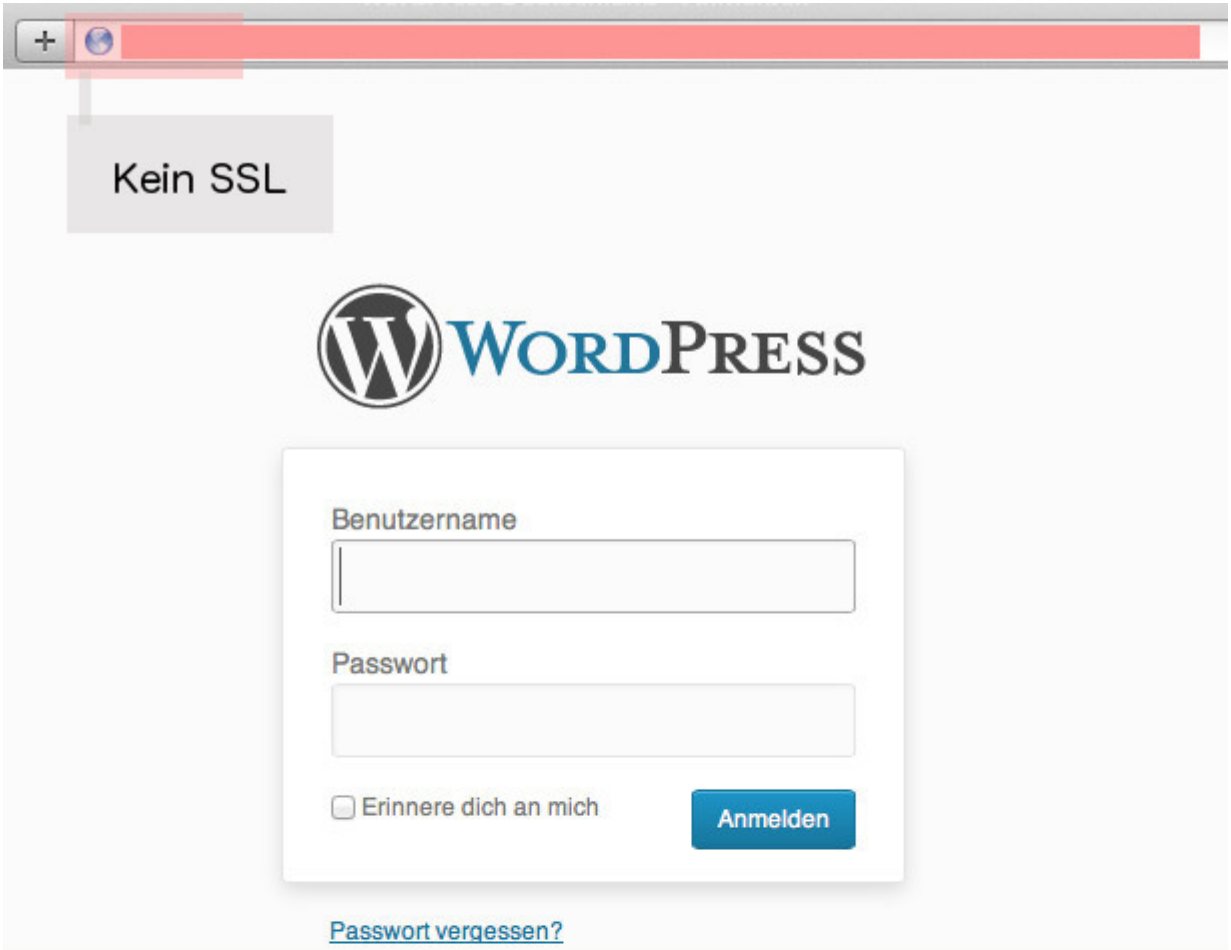
Aus Beispiel zwei lässt sich die Verwendung eines Security-Plugins ableiten. Vermutlich kommt hier Login LockDown / Limit Login Attempts oder ein ähnliches Plugin zum Einsatz. Diese protokollieren fehlgeschlagene **Login-Versuche**. Falls ein Anmeldeversuch innerhalb von 5 Minuten dreimal hintereinander fehlschlägt blockiert das Plugin die anfragende IP-Adresse beispielsweise für eine Stunde. Script-Kiddies und dämliche Bots lassen sich von solchen Maßnahmen abschrecken – professionelle Angreifer hingegen weniger.

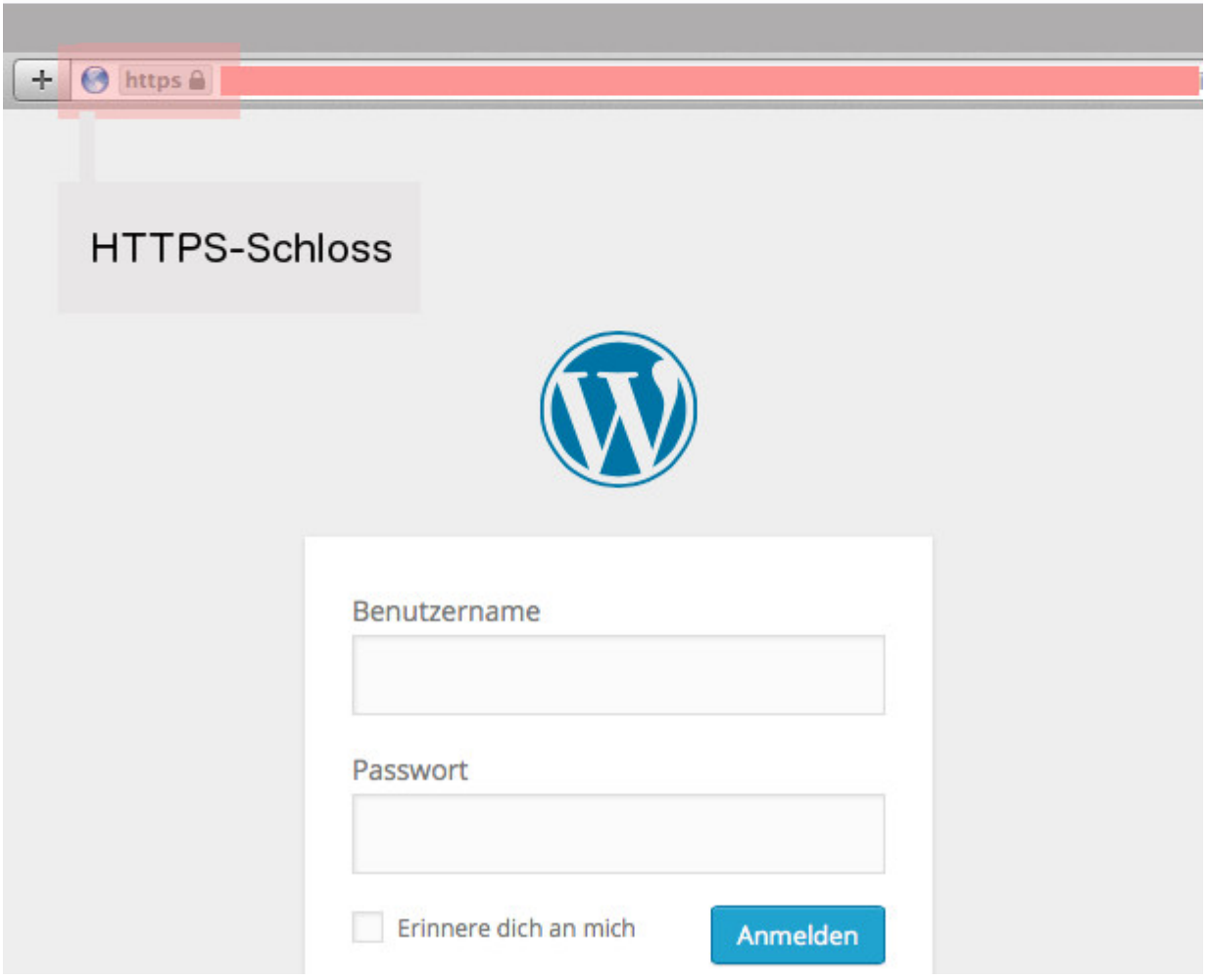
4.2 Fehlende SSL-Verschlüsselung

Hauptsächlich wird SSL für die **Absicherung** zwischen Webbrowser und Webserver eingesetzt – also immer dann, wenn sensible Informationen über das **unsichere** Internet ausgetauscht werden sollen.

Über den Browser wird abermals der Login zum Administrationsbereich aufgerufen:

wordpress-blog-adress.de/wp-admin


Kein SSL
WordPress
Benutzername
Passwort
Erinnere dich an mich
Anmelden
Passwort vergessen?


https
HTTPS-Schloss
Benutzername
Passwort
Erinnere dich an mich
Anmelden

Falls zwischen Browser und Server keine verschlüsselte SSL-Verbindung ausgehandelt wird, können die **Anmeldedaten** mitgeschnitten werden. Ganz konkret: Ein WordPress-Blogger nutzt das kostenlose **WLAN** in seinem Lieblingskaffee und loggt sich in den Administrationsbereich ein. Da die Verbindung nicht über SSL abgesichert wird, kann einer Angreifer die Anmeldedaten im Klartext bzw. unverschlüsselt mitlesen. Solch ein Angriff ist mit einfachen Mitteln bereits von Anfängern durchführbar.

5. Ausnutzen von Schwachstellen – Phase 3

Gefundene Schwachstellen gilt es in Phase 3 gezielt auszunutzen. Dafür werden vorhandene Exploits verwendet oder neue entwickelt, die es ermöglichen Systeme zu **kompromittieren**. Falls in ein System eingerungen werden kann, ergeben sich aus dem Zugriff oftmals weitere mögliche **Angriffsziele**, die vorher nicht erreichbar waren. Mit der Toolkiste aus »Exploitation Tools« oder »Privilige Escalation« stehen in Kali genügend Mittel zur Verfügung. Verfolgt wird damit:

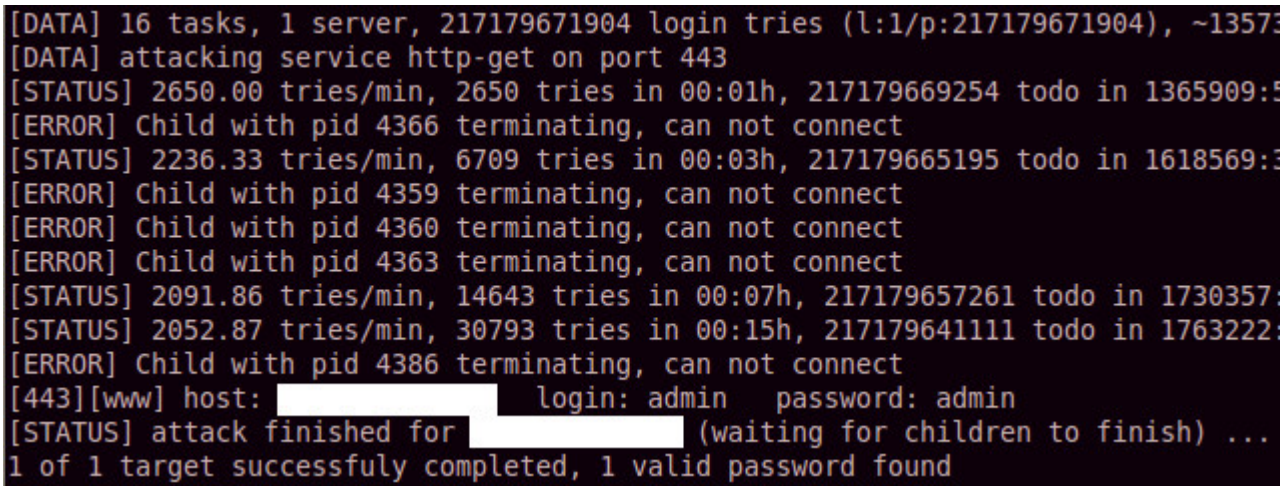
- Schwachstellen in Systemen / Anwendungen ausnutzen
- Systemzugriff erhalten
- Zugang zu geschützten Web-Bereichen
- Erfassen von sensiblen Daten
- […]

5.1 Brute-Force WP-Login

Da Administratoren über die weitreichendsten **Berechtigungen** verfügen, stellen sie ein beliebtes Ziel für Angreifer dar. Einmal eingeloggt erlauben Sie beispielsweise das Hinzufügen von schädlichen PHP- oder Javascript-Befehlen direkt über das

Dashboard. In der Informationsphase wurden bereits Anmeldeinformationen gesammelt, die gezielt für den Einbruch in das Backend genutzt werden können.

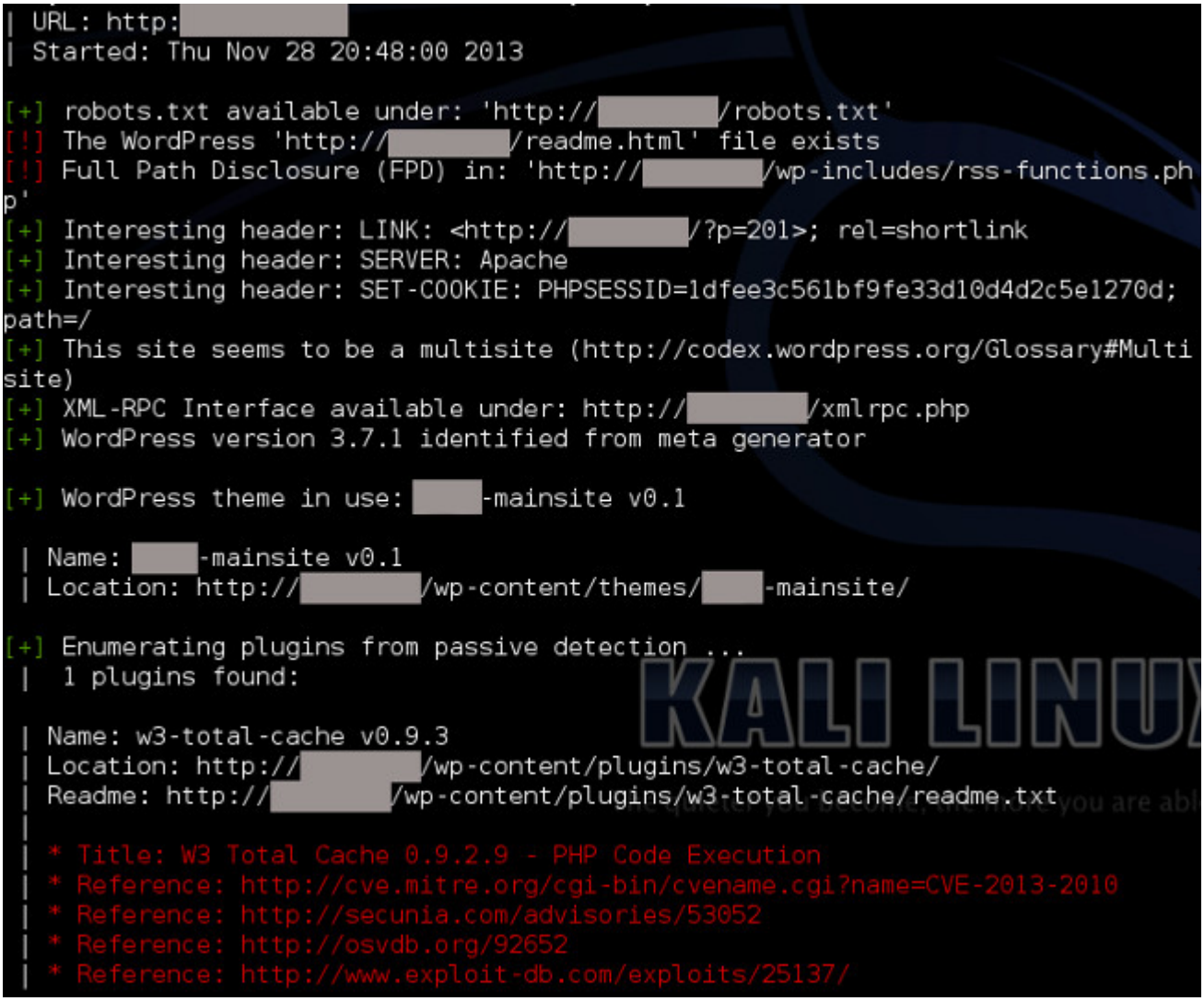
Geschützt wird der Administrationsbereich aus einer **Kombination** von Benutzername und Passwort. Falls ein Angreifer bereits über den Benutzernamen verfügt, so muss er im nächsten Schritt das Passwort »erraten«. Mittels einem Brute-Force-Angriff wird durch Ausprobieren das passende Passwort ermittelt. In freier Wildbahn führt dieser Angriff oft zum Erfolg, da viele Anwender noch immer unsichere Passwörter verwenden.



Speziell für diesen Zweck steht Hydra zur Verfügung. Neben WordPress-Installationen kann damit eine breite Palette von Systemen und Anwendungen angegriffen werden.

5.2 Das Tool WPScan

WPScan ist speziell auf WordPress zugeschnitten. Es bietet zahlreiche Funktionen, wie beispielsweise die **Erkennung** der installierten Plugins, Themes und WordPress-Versionen. Des Weiteren ist es in der Lage Benutzer-Accounts für Brute-Force-Angriffe zu »erraten« und verweist direkt auf **Schwachstellen-Datenbanken**, falls während des Scans auffällige Plugins gefunden werden. Im Beispiel wird eine Lücke im Plugin W3 Total Cache (Version 0.9.3) detektiert.



5.3 Metasploit

Metasploit ist eine Art **Allzweckwaffe** bzw. große Toolbox für Penetrationstests und Sicherheitsanalysen. Es besteht aus unterschiedlichen Teilbereichen, Teilprojekten und Modulen – der Umfang erlaubt den Einsatz in allen **Phasen** eines Penetrationstests. Auch Angreifer machen sich Metasploit zu Nutze, um in fremde Systeme einzudringen. Hier lediglich ein kurzer Einblick in das Metasploit Universum.

Das Metasploit Modul »**wordpress_login_enum**« dient zur Feststellung von gültigen Benutzer-Accounts und kann im Anschluss einen Passwort-Rate-Angriff durchführen.

```
msf > use auxiliary/scanner/http/wordpress_login_enum
msf auxiliary(wordpress_login_enum) > show options

Module options (auxiliary/scanner/http/wordpress_login_enum):

   Name               Current Setting  Required  Description
   ----               ---------------  --------  -----------
   BLANK_PASSWORDS    true             yes       Try blank passwords for all users
   BRUTEFORCE         true             yes       Perform brute force authentication
   BRUTEFORCE_SPEED   5                yes       How fast to bruteforce, from 0 to 5
   PASSWORD                            no        A specific password to authenticate
   PASS_FILE                           no        File containing passwords, one per
   Proxies                             no        Use a proxy chain
   RHOSTS                              yes       The target address range or CIDR ic
   RPORT              80               yes       The target port
   STOP_ON_SUCCESS    false            yes       Stop guessing when a credential wo
   THREADS            1                yes       The number of concurrent threads
   URI                /wp-login.php    no        Define the path to the wp-login.phr
   USERNAME                            no        A specific username to authenticate
   USERPASS_FILE                       no        File containing users and passwords
   USER_FILE                           no        File containing usernames, one per
   VALIDATE_USERS     true             yes       Enumerate usernames
   VERBOSE            true             yes       Whether to print output for all att
   VHOST                               no        HTTP server virtual host
```

```
msf auxiliary(wordpress_login_enum) > set URI /wordpress/wp-login.php
URI => /wordpress/wp-login.php
msf auxiliary(wordpress_login_enum) > set PASS_FILE /tmp/passes.txt
PASS_FILE => /tmp/passes.txt
msf auxiliary(wordpress_login_enum) > set USER_FILE /tmp/users.txt
USER_FILE => /tmp/users.txt
msf auxiliary(wordpress_login_enum) > set RHOSTS 192.168.1.201
RHOSTS => 192.168.1.201
msf auxiliary(wordpress_login_enum) > run

[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Runnir
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invali
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration- Usernan
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invali
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Checki
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Invali
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Enumeration - Found
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Runnir
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Skippi
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[-] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Failed
[*] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - Trying
[+] http://192.168.1.201:80/wordpress/wp-login.php - WordPress Brute Force - SUCCES
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(wordpress_login_enum) >
```

6. Weitere Möglichkeiten

Die oben dargestellten Tools und Möglichkeiten stellen lediglich eine Mini-Auswahl dar. In der **Praxis** existieren unzählige Tools und Varianten, Webanwendungen und deren Host-Systeme zu hacken. Allein in den Datenbanken von Metasploit und exploit-db.com sind hunderte von **Schwachstellen** erfasst und beschrieben. Immer wieder Ziel sind Plugins, Themes und der WordPress-Kern selbst.

Hinweis

Auch von deaktivierten Plugins oder Themes geht eine Gefahr aus. Selbst wenn sie nicht aktiv verwendet werden, so sind sie im Normalfall dennoch erreichbar. Beispielsweise erlaubt das Plugin Asset-Manager (Version <= 2.0) einen Datei-Upload in ein temporäres Verzeichnis – anschließend kann darüber Schadcode ausgeführt werden. Für diesen Einbruch muss das Plugin nicht aktiv sein, sondern lediglich auf dem Webspace vorhanden.**Lücke**: WordPress Asset-Manager PHP File Upload Vulnerability.

6.1 Angriffe auf Systemebene

Allein für Phase 1 (**Informationsbeschaffung**) wird ein Angreifer viel Zeit aufwenden, um an Daten / Informationen zu gelangen, die ihm später nützlich sein können. Immerhin hängt davon indirekt der Erfolg für den späteren Einbruch ab. Bereits einfache Wege wie, Google-Hacking (Dorks), DNS-Informationen und soziale Netzwerke stellen wichtige Informationsquellen dar. Daraus lassen sich oftmals Informationen ableiten, die entscheidende Hinweise für einen erfolgreichen Angriff bieten. Womöglich bietet eine WordPress-Installation selbst keinen **Angriffspunkt**, was den Fokus auf das Host-System richtet. Als Beispiel:

- MySQL-Datenbank

- FTP / SSH Service
- CPanel oder andere Tools für die web-basierte Administration
- phpMyAdmin Zugänge
- […]

Für die Sicherheit von WordPress müssen alle **Zahnräder** ineinandergreifen – letztendlich hat ein Angreifer immer das Ziel das schwächste Zahnrad auszumachen.

7. Fazit

Der Artikel WordPress-Hacking soll einen **Eindruck** über den Ablauf eines Angriffs vermitteln – auch wenn die Phasen leicht vermischt sind. Angreifer verfolgen damit meist unterschiedliche Ziele. Oftmals dienen infizierte WordPress-Installationen als **Ausgangspunkt** für weitere Angriffe oder zum Versenden von Spam-Mails. Neben Vandalismus und Rachegelüste sind praktisch unzählige Absichten denkbar.

Wenn eure WordPress-Installation selbst schon gehackt wurde oder ihr die Sicherheit im Vorfeld verbessern wollt, dann empfehle ich nochmals folgende Anleitungen: »Hardnening WordPress« und meine Artikelserie »WordPress absichern«.

Bildquellen:

Skull: „#9358035“, https://de.fotolia.com/id/9358035

Über den Autor | Kuketz

In meiner freiberuflichen Tätigkeit als Pentester / Sicherheitsforscher (Kuketz IT-Security) schlüpfe ich in die Rolle eines »Hackers« und suche Schwachstellen in IT-Systemen, Webanwendungen und Apps (Android, iOS). Des Weiteren bin ich **Lehrbeauftragter** für IT-Sicherheit an der dualen Hochschule Karlsruhe, schärfe durch Workshops und Schulungen das

Sicherheits- und **Datenschutzbewusstsein** von Personen und bin unter anderem auch als Autor für die Computerzeitschrift c’t tätig.

Der Kuketz-Blog bzw. meine Person ist regelmäßig in den Medien (heise online, Spiegel Online, Süddeutsche Zeitung etc.) vertreten.

Mehr Erfahren →

SCA-Tools (Lieferkettensicherheits-Tools) in der Übersicht

Die Software Composition Analysis soll Risiken aufdecken, die Entwickler beim Einsatz von Open-Source-Komponenten eingehen, und die Softwarelieferkette absichern. Der Markt für passende Produkte ist riesig und in ständiger Bewegung.

-tract

- Software Composition Analysis (SCA) ist eine Form der Codeanalyse, die ermittelt, welche Open-Source-Bibliotheken eine Software verwendet, welche bekannten Schwachstellen in ihnen enthalten sind und unter welcher Lizenz sie stehen.
- SCA-Werkzeuge unterstützen dabei und automatisieren diesen Prozess, indem sie sich in Code-Repositorys, CI/CD-Pipelines und oft auch IDEs integrieren.
- Der Markt ist geprägt von sehr vielen Anbietern und oft recht jungen Produkten. Eine Auswahl von Angeboten von etablierter Herstellern, die auf dem deutschen Markt

aktiv sind, stellt diese Übersicht vor.

Sicherheit rückt nach links. Nicht politisch, sondern im Sinne des Left Shift der DevOps-Bewegung. Es bedeutet, dass immer mehr Kompetenzen am Anfang, also links im gesamten Prozess angesiedelt sind: bei den Entwicklern. Mit DevSecOps wird aus der Verantwortung für den Betrieb (DevOps) nun Verantwortung für den sicheren Betrieb. Das macht aber Entwickler nicht auf magische Weise zu Securityspezialisten. Deshalb ist jede Unterstützung in Form von Werkzeugen oder Frameworks gefragt, die helfen, möglichst viele Risiken so früh es geht zu entdecken und Sicherheitslücken zu stopfen.

Da nahezu jedes größere Softwareprojekt Open-Source-Komponenten enthält, betrifft dies nicht nur die vom eigenen Entwicklerteam zu verantwortenden Schwachstellen, sondern auch die in den eingebundenen Abhängigkeiten. Die Aufgabe der Software Composition Analysis besteht darin, herauszufinden, welche Komponenten in welchen Versionen in der eigenen Software stecken, und dann zu ermitteln, welche schon bekannten Schwachstellen diese haben. Über diese absolute Mindestanforderung an eine SCA-Software gehen aber alle am Markt vorhandenen Systeme hinaus und bieten Einbindung in CI/CD-Pipelines oder Entwicklertools, automatische Lösungsvorschläge für gefundene Schwachstellen (Remediation), diverse Dashboards, Frameworks zum Festlegen von Richtlinien und so weiter.

Entdecken, dokumentieren, beheben

In aller Regel erfüllt SCA zudem eine Doppelfunktion. Zusätzlich zu Sicherheitsrisiken soll sie auch Compliance-Risiken identifizieren, indem sie die Open-Source-Lizenzen findet, unter denen verwendete Komponenten veröffentlicht sind. Das macht auch Rechtsabteilungen und Management zu SCA-Anwendern, die Software darf also unter Umständen nicht ausschließlich auf die Bedürfnisse von Developern

zugeschnitten sein. So gut wie immer kann ein SCA-Werkzeug SBOMs (Software Bills of Materials) erzeugen, also „Zutatenlisten“, die beispielsweise US-Behörden per Präsidentenerlass verlangen müssen [1].

Um die Komponenten zu ermitteln, lesen SCA-Tools die Abhängigkeiten aus den Manifestdateien verschiedener Paketmanager aus, etwa NPM, Maven oder NuGet; manche scannen darüber hinausgehend auch den Sourcecode selbst oder sogar Binärdateien. Für den Abgleich mit bekannten Sicherheitslücken nutzen kommerzielle Anbieter in der Regel eigene Datenbanken, Open-Source-Programme greifen oft auf frei verfügbare Quellen zurück, wie die National Vulnerability Database (NVD), die das US-amerikanische National Institute of Standards and Technology (NIST) pflegt, oder die ebenfalls recht umfassenden GitHub Security Advisories.

Der Markt für SCA-Software ist ausgesprochen vielfältig, neben ausgereiften und von großen Organisationen unterstützten Open-Source-Programmen tummeln sich Spezialhersteller und die etablierten Anbieter großer Sicherheitslösungen, die in den letzten zwei Jahren SCA entweder in ihre Suiten integriert oder separate Produkte lanciert haben. Zur großen Fülle an Herstellern und Produkten mag beitragen, dass es technisch eher eine Fleißarbeit ist, die Grundfunktionen zur Verfügung zu stellen: möglichst viele unterschiedliche Manifestformate der Paketmanager parsen und mit Datenquellen zu Sicherheitslücken abgleichen, Export in gängige SBOM-Formate, dazu noch etwas Integration in bereits vorhandene Frameworks zu Datenaufbereitung, Nutzermanagement, Entwicklertools und DevOps-Pipelines – fertig ist die SCA-Lösung.

Deshalb grenzen sich die führenden Anbieter auf diesem Gebiet auch alle durch spezielle Alleinstellungsmerkmale von ihren Mitbewerbern ab. Häufig haben sie weitere Analyseverfahren im Angebot und gestatten das zusätzliche Scannen von Source- oder Binärcode. Oft pflegen sie erweiterte Schwachstellendatenbanken, können interne Projekte in die

Analyse einbeziehen, oder sie positionieren sich gezielt als umfassende Enterprise-Lösung, die alle Anwendungsfälle abdeckt und sich an ein heterogenes Anwenderfeld richtet.

Viel Bewegung im Markt

Die OWASP listet auf ihrer Website zum SBOM-Format CycloneDX 170 Plattformen und Werkzeuge auf, die ganz oder teilweise SCA-Funktionen haben (siehe ix.de/zvbm). Beim Eingrenzen der Auswahl ist auf die jährlichen Analysen von Gartner, Forrester und Co. nur bedingt Verlass. Manche Hersteller, die laut dem einen Analysten seit Jahren eine stabile, besonders starke Marktposition haben, werden bei dem anderen nicht einmal erwähnt, andere rutschen von einem Jahr zum anderen zwischen Gartners magischen Quadranten hin und her.

Auch ist es fraglich, ob die Orientierung an den dort gelisteten Produkten immer sinnvoll ist, denn deren Schwerpunkt liegt auf großen Lösungen für den unternehmensweiten Einsatz. Nicht nur deren Implementierung kann aufwendig sein. Auch die Prozesse, an die sich alle Anwender gewöhnen müssen, sind nur mit großem Aufwand durchzusetzen. Mit etwas Pech ist das Produkt gekauft und eingerichtet, aber kaum einer nutzt seine elaborierten Features.

Für einzelne Projekte und kleinere Teams kann eine weniger umfangreiche, aber auch weniger komplexe Software die bessere Entscheidung sein – vorausgesetzt, sie lässt sich gut mit den vorhandenen Tools und Abläufen verheiraten. Open-Source-Werkzeuge, aber auch manche auf Cloud-native gebürsteten Spezialhersteller mit ihren SaaS-Angeboten kommen da am ehesten infrage.

Die hier vorgestellten Werkzeuge zählen zu den eher etablierten Produkten dieses Segmentes und stammen hauptsächlich von SCA-Spezialisten oder zumindest von Herstellern, deren sonstige Expertise in der Codeanalyse liegt

und die auch im deutschsprachigen Raum aktiv sind. Hinzu kommt eine kleine Auswahl Open-Source- oder anderer kostenloser Tools. Die Angaben in dieser Übersicht beruhen auf öffentlich zugänglichen Informationen und auf Nachfragen bei den Herstellern; soweit verfügbar wurden die aktuellen technischen Dokumentationen der Produkte herangezogen. Es sind sowohl Produkte dabei, die sich on Premises installieren lassen, als auch solche, die komplett als Service in der Cloud angeboten werden. Manche Anbieter lassen ihren Kunden die Wahl zwischen verschiedenen Bereitstellungsmethoden, andere haben hybride Modelle im Angebot.

Übersicht ausgewählter SCA-Anbieter									
Anbieter	Aqua		Anchore/Community	Synopsys	Checkmarx	FOSSA	Mend	OWASP/Community Dependency-Track	Snyk
Produkt	Aqua Supply Chain Security	Aqua Trivy	Syft/Grype	Black Duck	Checkmarx SCA	FOSSA	Mend SCA	OWASP Dependency-Track	Snyk Open Source
Bereitstellungsmodell	Public Cloud, Private Cloud, on Premises, AWS, GCP	lokal	lokal	on Premises	SaaS, Private Cloud, on Premises	SaaS, on Premises	SaaS, lokaler Scan möglich	on Premises	SaaS, lokaler Scan möglich
Export von SBOM-Formaten	SPDX, CycloneDX	SPDX, CycloneDX	SPDX, CycloneDX, eigenes Format	SPDX, CycloneDX, Protex	CycloneDX (über API auch SPDX)	SPDX, CycloneDX, weitere Formate	CycloneDX und SPDX mit separatem Tool	CycloneDX	SPDX und CycloneDX mit API und CLI (Beta)
Abgleich mit Schwachstellendatenbanken	eigene Datenbank	eigene Datenbank	durch Integration mit Grype	NIST NVD oder Black Duck Security Advisories (mit separater Lizenz)	eigene Schwachstellendatenbank, zusätzlich Datenbank bösartiger Pakete	eigene Datenbank	eigene Datenbank	NIST NVD und weitere	eigene Datenbank
Integration in DevOps-Tools	u. a. GitHub Actions, GitLab, CI CD, Jenkins, CircleCI, Terraform Cloud	GitHub Actions und Azure DevOps (offiziell), CircleCI und weitere (Community)	teilweise Community-Plug-ins	ca. 15 CI-Plattformen	Jenkins, Azure DevOps, TeamCity, Bamboo	ca. 15 CI-Plattformen	u. a. Azure DevOps, Jenkins, CircleCI, Travis	Jenkins, Maven, Gradle, GitHub Actions	CircelCI, GitHub Actions, Jenkins, Maven, TeamCity, Terraform
Code-Repositorys	GitHub, GitLab, Bitbucket, Azure	Git-basierte	n. a.	GitHub, GitLab, Bitbucket	GitHub, GitLab, Bitbucket, Perforce, Azure	GitHub, GitLab, Bitbucket, Azure, Custom Imports	Hosted Integration für GitHub, Bitbucket, Azure, Self-hosted GitHub Enterprise, BB Server, GitLab	n. a.	Git-basierte
Scan von Artefakt-Repositorys	JFrog Artifactory, Nexus	nein	n. a.	JFrog Artifactory, Nexus	JFrog Artifactory, Nexus	nein	JFrog Artifactory, GitHub Packages	nein	JFrog, Nexus und weitere
Scan von Container-Images	Docker	Docker	Docker, OCI, Singularity	Docker (OpenShift, Kubernetes Package Manager, Pivotal Cloud Foundry)	Docker, AWS ECR (mittels Syft)	OCI-Container (apt, RPM und apk)	Docker, GCR, ACR, ECR	nein	mit Snyk Container
Compliance, Lizenzinformationen	ja	eingeschränkt	eingeschränkt	ja	ja	ja	ja	nein	mit Enterprise-Lizenz
IDE-Integration	(ja)	JetBrains IDEs, VS Code, (Vim mit Community-Plug-in)	VS Code für macOS und Linux	möglich mittels Code Sight	JetBrains IntelliJ, Visual Studio Code	nein	Visual Studio, VS Code, IntelliJ IDEA, GitHub Codespaces	nein	Eclipse, JetBrains-IDEs, VS, VS Code, Language Server (Beta)
CLI	ja	ja	ja	ja	ja	ja	ja	ja	ja
API	ja	nein	nein	ja	ja	ja	ja	ja	mit Enterprise-Lizenz
unterstützte Sprachen	Java, C/C++, .NET, Node.js, PHP, Python, Go, Ruby, Rust	Go, Java, .NET, PHP, Python, Ruby, Node.js	ca. 20	ca. 25	Java, C++, .NET, Python, PHP, Swift, Objective-C, Go, Ruby	ca. 20	über 200	Java, .NET, experimentell: Python, PHP, Node.js, Ruby, Swift	ca. 15
unterstützte Paketmanager	□	ca. 10□	ca. 20	ca. 20	ca. 15	ca. 20	ca. 30	ca. 20, davon 2□3 experimentell	ca. 15
Scan von Binärdaten	Go	nein	□nein	Java, .NET, Go	nein□	nein	ja	nein	□

Übersicht ausgewählter SCA-Anbieter									
Anbieter	Aqua		Anchore/Community	Synopsys	Checkmarx	FOSSA	Mend	OWASP/Community Dependency-Track	Snyk
Produkt	Aqua Supply Chain Security	Aqua Trivy	Syft/Grype	Black Duck	Checkmarx SCA	FOSSA	Mend SCA	OWASP Dependency-Track	Snyk Open Source
Prüfung von Codeerreichbarkeit[1]	nein	nein	nein	für Java	ja, Exploitable Path	nein	ja, Reachability Path Analysis	nein	für Java, mit Snyk Code
automatisierte Remediation	nein	nein	nein	nein	Remediation Manifests für npm	automatisierte Pull Requests	automatisierte Pull Requests	nein	automatisierte Pull/Merge Requests
Definition von Richtlinien	ja	nein	nein	ja	ja	ja	ja	ja	mit Enterprise-LIzenz
Preis	auf Anfrage (Lizenzierung nach Repositorys)	kostenlos (Open Source)	kostenlos (Open Source)	auf Anfrage	auf Anfrage	ab 104 Dollar pro Entwickler und Monat, Enterprise	ab 16 000 Euro pro Jahr (für 20 Entwickler)	kostenlos (Open Source)	ab 23 Dollar pro Entwickler und Monat, limitierte Version kostenlos, Enterprise auf Anfr.
n. a. – nicht anwendbar, BB – Bitbucket; [1] Tests, ob die kompromittierte Funktion/Methode tatsächlich aufgerufen wird									

OWASP Dependency-Track

Eines der am längsten verfügbaren SCA-Tools kommt vom Open Worldwide Application Security Project: das Open-Source-Werkzeug OWASP Dependency-Track (ODT). Es gibt SBOMs im CycloneDX-Format aus und man kann sie in diesem Format auch importieren. Für den Abgleich mit bekannten Schwachstellen nutzt das Werkzeug die National Vulnerability Database, GitHub Advisories und den Sonatype OSS Index als Datenquellen. Weitere, zum Teil kostenpflichtige Quellen können Anwender über Plug-ins freischalten. Metadaten über Abhängigkeiten gewinnt ODT aus einer Reihe von verbreiteten Paketformaten, neben den üblichen NuGet, PyPi, Maven oder NPM sind auch Cargo für Rust-Projekte oder Hex für Elixir/Erlang darunter. ODT ist nicht auf die Analyse von Abhängigkeiten und Schwachstellen beschränkt, es ermöglicht auch, Richtlinien (Policies) festzulegen, die bestimmte Pakete, Softwarelizenzen oder Software mit Schwachstellen eines definierten Schweregrades ausschließen.

ODT ist lokal installierbar. Ab Version 4 der Software sind Backend und Frontend voneinander getrennt. Das Backend – der API-Server – ist eine klassische Serverapplikation, die die API per Jetty zur Verfügung stellt und ihre Daten im lokalen Dateisystem und einer relationalen Datenbank speichert. Das Frontend ist eine Single-Page-Webapplikation. Sie stellt Dashboards und Reports dar und dient der Konfiguration. Am einfachsten ist die Installation als Docker-Container. Über

ein offizielles Jenkins-Plug-in oder GitHub Actions wird ODT in die CI/CD-Pipeline integriert.

Syft und Grype

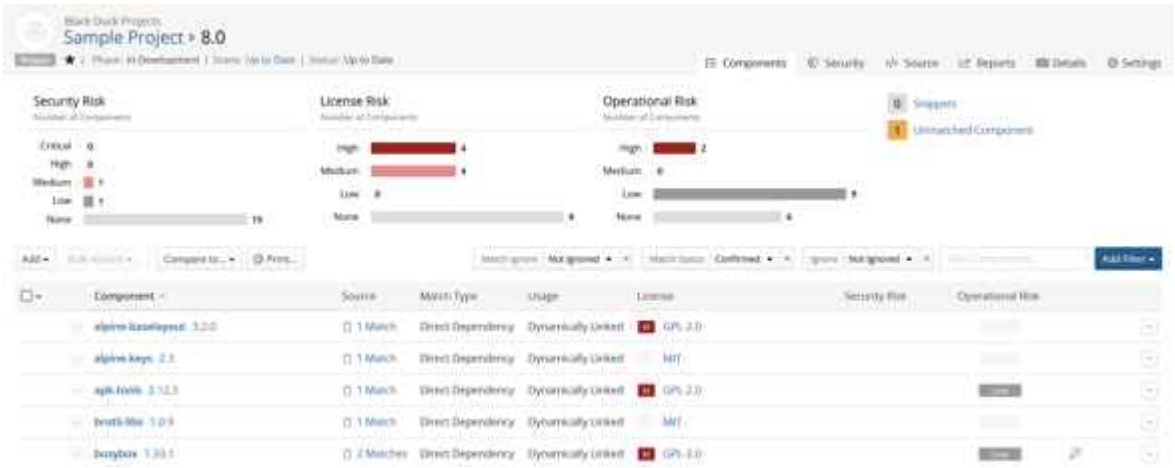
Das US-Unternehmen Anchore stellt mit Syft ein Open-Source-Tool zur Verfügung, das hauptsächlich der Erstellung von Software Bills of Materials (SBOMs) dient. Syft verfügt ausschließlich über eine Kommandozeilenschnittstelle, es stellt keine grafische Benutzeroberfläche bereit. Zusätzlich bietet Anchore das Tool Grype an, das in Verbindung mit Syft oder auch einzeln Vulnerability-Scans durchführt. Mit der Kombination von Syft und Grype lassen sich viele Anwendungsszenarien der großen kommerziellen Lösungen abdecken, wenn auch mit manuellem Konfigurationsaufwand. Für die Integration in CI/CD-Pipelines stellen Anchore und die Community Werkzeuge zur Verfügung, beispielsweise Jenkins-Plug-ins oder GitHub Actions, selbst ein IDE-Plug-in für VS Code gibt es.

Syft und Grype lassen sich lokal installieren; die Schwachstellendatenbank kommt im SQLite-Format und wird in der Standardkonfiguration automatisch über das Netz aktualisiert. Die beiden Open-Source-Werkzeuge von Anchore haben keine eigene API, als CLI-Tools mit wohldefinierten Ausgabeformaten lassen sie sich aber prinzipiell von anderen APIs benutzen. Eine der Stärken beider Anwendungen ist die Ausrichtung auf containerisierte Applikationen.

Synopsys Black Duck

Black Duck gehört zu den am längsten verfügbaren und am häufigsten eingesetzten SCA-Angeboten am Markt. Dazu kommt, dass es nach der Übernahme des Herstellers durch Synopsys mit dessen Codeanalysewerkzeugen verzahnbar ist: So lässt sich SCA mit statischen und dynamischen Codeanalysemethoden (DAST, SAST) und Fuzzing aus einer Hand kombinieren. Es ist auf den unternehmensweiten Einsatz ausgerichtet und soll dabei helfen,

in großen Projekten zentrale Sicherheits- und Compliancerichtlinien zu definieren und durchzusetzen (siehe Abbildung 1).



Die BOM-Ansicht von Black Duck listet Lizenz-, Sicherheits- und Betriebsrisiken einer Komponente gemeinsam auf. Letztere ergeben sich zum Beispiel aus Paketen, die kaum noch gepflegt werden oder eine geringe Reputation besitzen (Abb. 1). *Synopsys*

Damit einher gehen ein umfassendes Rollen- und Berechtigungsmodell, komplexe Policies und Regelsätze, die definieren, wie mit bestimmten Risiken umzugehen ist, sowie Reportgeneratoren. Entsprechend langwierig kann die Einführung des Produkts sein. Synopsys gibt die Black Duck Security Advisories heraus und verspricht, dass seine SCA-Software viele Schwachstellen schon meldet, bevor sie in der National Vulnerability Datebase auftauchen.

Black Duck integriert sich via Plug-ins in alle verbreiteten CI/CD-Frameworks, Code- und Artefakt-Repositorys. Für die IDE-Integration ist Synopsys Code Sight zuständig, ein separates Produkt, das Black-Duck-Anwender kostenlos nutzen können.

Neben Sourcecode analysiert Black Duck Java-, .NET- und Go-Binaries, binäre Repositorys im JFrog-Artifactory- und Nexus-Format und bestimmte Firmwareformate. Bei der Codeanalyse verlässt es sich nicht nur auf die Deklarationen in den Manifesten der Pakete. Der Hersteller wirbt mit einer Multi-Faktor-Open-Source-Erkennung und integriert eine proprietäre

Methode namens Codeprint, um Open-Source- und Fremdanbieter-Komponenten zu identifizieren.

Aqua Supply Chain Security

Aqua Security gilt als Spezialist für die Absicherung von containerisierten Anwendungen. Nach der Übernahme von Argon Ende 2021 – eines auf Supply Chain Security spezialisierten Start-ups aus Israel – bietet das Unternehmen mit Aqua Supply Chain Security ein Produkt an, das die wesentlichen Aspekte der SCA abdeckt und darüber hinaus weitere Sicherheitsüberprüfungen durchführt. So scannt es per statischer Codeanalyse bei Abhängigkeiten auch den Quellcode selbst, sucht nach Fehlkonfigurationen in den Build-Tools und in Infrastructure as Code. Go-Code können die Aqua-Scanner auch in Binärform untersuchen.

Eine Besonderheit stellen die erweiterten SBOMs dar, die die Plattform erzeugen kann. Die als Next Generation SBOMs bezeichneten Dokumente sind mit zusätzlichen Informationen angereichert, etwa ob Peer-Reviews stattfanden oder ob das Code-Repository eine Zwei-Faktor-Authentifizierung verlangt. Zusätzlich soll Code Signing die Integrität sicherstellen.

Compliance- und Sicherheitsfunktionen sind integriert, Aqua Supply Chain Security eignet sich also auch zur Top-Level-Beurteilung der Risiken durch Open-Source-Einsatz im gesamten Unternehmen. Für die einzelnen Open-Source-Komponenten erstellt das Produkt einen Reputation Score, aus dem Maintenance-Zustand, der Beliebtheit, der Zahl und Schwere von Sicherheitslücken und anderen Faktoren.

Aqua vermarktet Supply Chain Security innerhalb seiner Cloud-native Application Protection Platform (CNAPP), in dessen Variante Dev Security. Es wird dort von den Komponenten Risk &Vulnerability Scanning sowie Advanced Malware Protection ergänzt.

Aqua Trivy

Ein Kernbestandteil von Aqua Supply Chain Security ist der Security-Scanner Trivy, der als separates CLI-Tool vor allem in der Container-Welt häufig eingesetzt wird. Für sich genommen ist er zwar kein vollwertiges SCA-Produkt, aber er ist Open Source und deckt so viele SCA-Aspekte ab, dass er in Kombination mit ein paar Skripten und anderen Open-Source-Werkzeugen die Grundlage für eine kleine, flexible, selbst gebaute SCA-Lösung sein kann. Trivy ist kein reiner Container-Scanner, sondern kann auch Code in Git-Repositorys, auf dem lokalen Filesystem oder in Images virtueller Maschinen prüfen. Er identifiziert dort bekannte Schwachstellen, findet Abhängigkeiten, Konfigurationsfehler und sensible Informationen wie Zugangsdaten. Außerdem identifiziert er Open-Source-Lizenzen. Seit Kurzem kann Trivy auch SBOMs im SPDX- oder CycloneDX-Format erzeugen.

Trivy ist ein reines Kommandozeilenwerkzeug und somit automatisierungsfreundlich. Aqua Security stellt sogar selbst Integrationen für GitHub Actions und Azure DevOps zur Verfügung. Trivy bringt seine eigene kompakte Schwachstellendatenbank mit, bei gefundenen Lücken verlinkt er in der Ausgabe auf den entsprechenden Eintrag in der Aqua Vulnerability Database, die auch das kommerzielle Produkt Aqua Supply Chain Security nutzt.

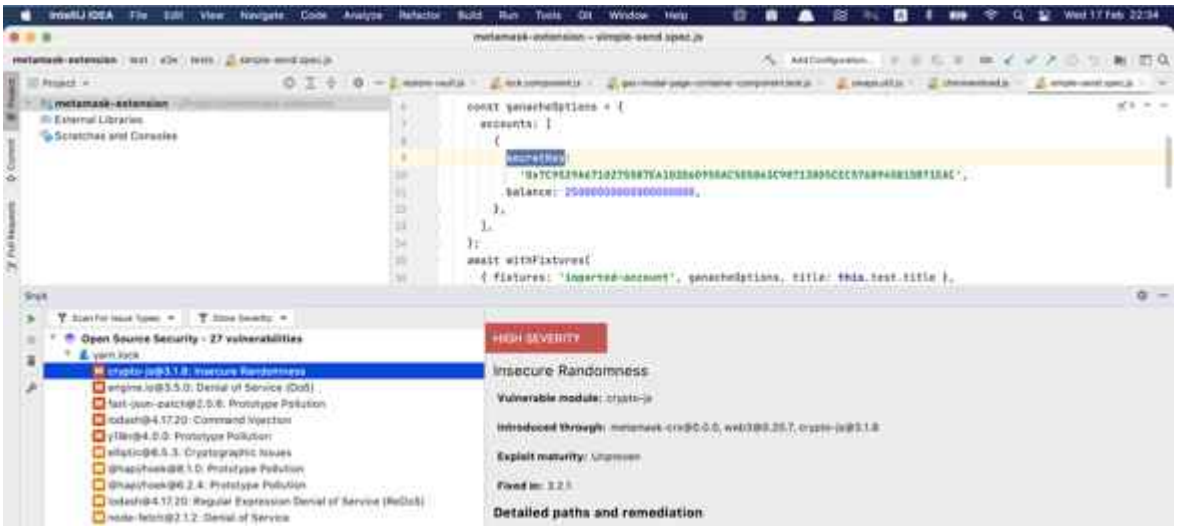
FOSSA

Das Produkt FOSSA (Free Open Source Software Analysis) des gleichnamigen Anbieters bezeichnet dieser als Open Source Risk Management Platform. Sein Schwerpunkt liegt darauf, rechtliche und Sicherheitsrisiken gemeinsam zu betrachten und die Nutzung von Open Source unternehmensweit durch Richtlinien abzudecken. Eine zentrale Policy Engine soll Rechts- und Entwicklungsabteilungen bei der gemeinsamen Ausarbeitung dieser Richtlinien unterstützen und garantieren, dass sie im

Softwarelebenszyklus durchgesetzt werden. FOSSA wirbt mit rechtssicheren, auditfähigen Berichten und automatisierten Risikobewertungen, die beispielsweise den Due-Diligence-Prozess bei Firmenübernahmen beschleunigen sollen. Für DevOps-Teams bietet FOSSA neben Integrationsmöglichkeiten in alle relevanten CI-Produkte auch eine generische CI-Schnittstelle für individuelle Pipelines an, es scannt Container nach OCI-Standard und unterstützt rund 20 verbreitete Programmiersprachen.

Snyk Open Source

Snyk vereint mehrere Produkte auf einer Plattform. Für SCA zuständig ist die Komponente mit dem Namen Snyk Open Source, daneben bietet Snyk Code eine statische Codeanalyse. Snyk Container und Snyk Infrastructure as Code sind weitere Komponenten. Snyk ist in erster Linie ein SaaS-Anbieter. In dieser Variante sind die Komponenten auch einzeln buchbar. Eine Enterprise-Lizenz umfasst immer alle Produkte; sie ist auch Voraussetzung, um Features nutzen zu können, die zu einem umfassenden SCA-Produkt gehören, wie Lizenzcompliance, Verwaltung von Richtlinien und Erstellung von Berichten sowie die Option, auf den Unternehmensservern gehostete Code-Repositorys einzubinden.



Mit einem Plug-in für JetBrains-IDEs informiert Snyk schon beim Schreiben des Codes über Sicherheitslücken in den eingebundenen Open-Source-Bibliotheken (Abb. 2). *Snyk*

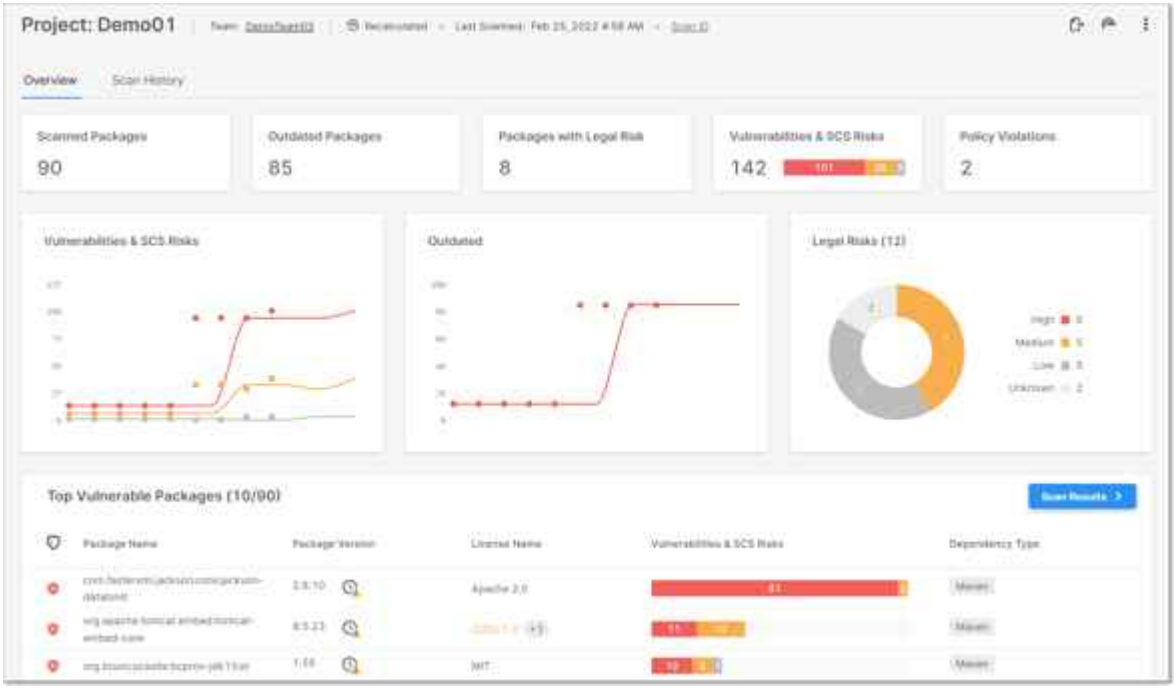
Eine vollständige On-Premises-Installation bietet Snyk nicht an – die Enterprise-Variante erlaubt es aber, ein als Snyk Broker bezeichnetes Proxysystem einzurichten, das Codescans lokal ausführt und die Kommunikation mit den Snyk-Servern über einen Tunnel absichert. Mit einem normalen SaaS-Abo erfolgt der Scan auf den Servern von Snyk.

Snyk Open Source lässt sich mit IDEs, den gängigen CI/CD-Tools und Git-basierten Repositorys verknüpfen (siehe Abbildung 2). Im Vergleich zu anderen Produkten ist vor allem die IDE-Unterstützung gut ausgebaut. So schlägt zum Beispiel das JetBrains-Plug-in mit einer als Open Source Advisor bezeichneten Funktion geeignete Open-Source-Pakete vor und bezieht dabei Popularität, Maintenance-Zustand und Bewertungen der Community ein.

Der Umgang mit SBOMs zählte lange Zeit nicht zu den Stärken von Snyk. Kürzlich hat das Unternehmen aber angekündigt, dass die API und das Kommandozeilenwerkzeug künftig SBOMs in den CycloneDX- und SPDX-Formaten exportieren sollen. In der aktuellen Betaversion der API ist das Feature bereits zu finden.

Checkmarx SCA

Checkmarx ist ein 2006 in Israel gegründetes IT-Security-Unternehmen, dessen Sicherheitsforscher wiederholt wichtige Schwachstellen aufgedeckt haben und federführend an der Erstellung der OWASP API Top Ten beteiligt sind. Erstes Produkt der Firma war CxSAST, ein Werkzeug zur statischen Codeanalyse, Checkmarx SCA (CxSCA) kam erst 2020 hinzu. Wie alle der größeren Anbieter betreibt Checkmarx seine eigene Schwachstellendatenbank, zusätzlich dazu auch eine Datenbank bösartiger Pakete, die gezielt dafür entwickelt werden, Softwareprojekte zu infiltrieren.



Dashboards wie hier bei Checkmarx gehören zur Grundausstattung aller umfangreicheren SCA-Tools (Abb. 3). *Checkmarx*

Checkmarx SCA ist Teil von Checkmarx One, dem integrierten Hauptprodukt des Herstellers, das von diesem als Application Security Testing Platform bezeichnet wird. CxSCA kann aber auch separat lizenziert werden. Am günstigsten ist die Nutzung als Managed Service, optional ist der Betrieb in einer Private-Cloud-Umgebung oder vollständig on Premises möglich. Checkmarx SCA implementiert eine Methode namens Exploitable Path, die im Sourcecode des Projekts danach sucht, welche Funktionen in den Abhängigkeiten tatsächlich aufgerufen werden. Laut Hersteller funktioniert das für jede Programmiersprache, die sich mit CxSAST untersuchen lässt. Bei Scans über die SCA-Website lädt das Tool auch den Sourcecode hoch und dort bleibt er für bis zu 24 Stunden gespeichert. Ein Resolver kann Abhängigkeiten aber auch on Premises ermitteln und schickt diese Daten dann an die Plattform zur Risikoanalyse.

Bei Verwendung von Agents oder des Resolvers gelangen nur Metadaten, Manifestdateien und Fingerprints des Sourcecodes auf die Checkmarx-Server. Zu den Metadaten zählt Checkmarx auch sämtliche Dateinamen. Daten landen in einem verschlüsselten S3-Bucket, Sourcecode wird höchstens 24

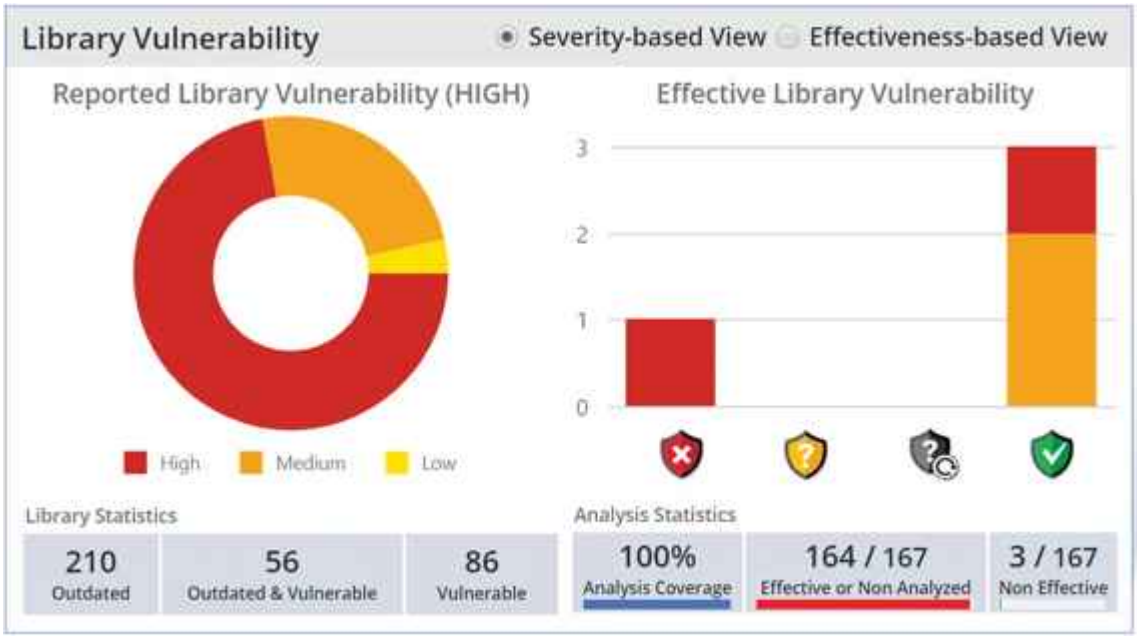
Stunden aufbewahrt.

Mend SCA

Mend, vormals Whitesource, ist ein weiterer Hersteller im Umfeld der Anwendungssicherheit, der seine Wurzeln in Israel hat. Hier war das SCA-Produkt zuerst da, SAST kam später hinzu. Mend entwickelt auch den von Entwicklern viel gelobten Renovate Bot, ein Open-Source-Werkzeug zur automatischen Aktualisierung von Dependencies. Diesen wird iX in einer der kommenden Ausgaben vorstellen.

Mend sammelt Schwachstellen und Security Advisories aus zahlreichen Quellen in einer eigenen Datenbank und scannt auch Software, die in den Manifesten der Paketmanager nicht deklariert ist. Eine der Stärken des Produkts ist das Bewertungssystem von Schwachstellen (siehe Abbildung 4). Hier berücksichtigt Mend vor allem, ob der eigene Code verwundbare Funktionen aufruft (Reachable Path Analysis). Aber auch andere, nicht direkt die Schwachstelle selbst betreffende Faktoren, die insgesamt die Auswirkungen auf die Geschäftstätigkeit widerspiegeln sollen, gehen ein.

Damit gehen entwicklerfreundliche Benachrichtigungs- und Remediation-Möglichkeiten einher. Ist Mend SCA in ein Repository integriert, kontrolliert es bei jedem Commit den Code auf vom Entwickler eingebaute Schwachstellen, Vulnerabilities in verwendetem Open-Source-Code und Lizenzverletzungen. Das Tool öffnet Pull Requests mit einem Upgrade des Pakets auf eine nicht verwundbare Version. Im einfachsten Fall ist somit die Schwachstelle mit einem Klick aus dem Abhängigkeitsbaum verschwunden.



Mend priorisiert Schwachstellen anhand verschiedener Metriken. Eine davon ist die Erreichbarkeit des Codes von der eigenen Anwendung aus (Abb. 4). *Mend*
Seine IDE-Pug-ins nennt der Anbieter Mend Advise, es gibt sie für IntelliJ Idea, WebStorm und PyCharm von JetBrains, für Visual Studio und VS Code sowie für Eclipse. Eine clevere Idee ist eine Browsererweiterung, die beim Stöbern auf Stack Overflow oder GitHub auf Sicherheitsrisiken in den gerade dargestellten oder erwähnten Komponenten hinweist.

Compliance- und Sicherheitsrichtlinien kann Mend SCA ebenfalls über entsprechende Regelwerke definieren und durchsetzen – insgesamt stehen bei diesem Produkt aber eher die Bedürfnisse der Developer als die der Rechtsabteilung im Vordergrund. In den letzten Monaten hat Mend seine API um einen SBOM-Export erweitert, vorher musste man SBOMs mit einem Tool aus dem internen Softwareinventarformat erzeugen. Jetzt lässt sich der Prozess automatisieren.

Weitere Anbieter

Contrast SCA ist Teil der vor allem im Java-Umfeld verbreiteten Secure Code Platform des Herstellers. Sie verfolgt den Ansatz, Agenten in den Code einer Anwendung zu integrieren, die im laufenden Betrieb Schwachstellen identifizieren. Diese Agenten liefern auch Informationen zu

den verwendeten Open-Source-Komponenten, aus denen die Plattform Schwachstellen identifiziert und detaillierte SBOMs generiert. Neben Java unterstützt Contrast weitere Sprachen und Plattformen etwa .NET, Python, Ruby und Go.

Die kanadische Firma **MergeBase** bewirbt ihr SCA-Produkt mit niedriger Falsch-positiv-Rate und Laufzeitüberwachung des Produktivcodes. Als SaaS ist MergeBase relativ günstig (ab 38 US-Dollar pro Entwickler), Enterprise-Varianten lassen sich auch on Premises installieren. Der Funktionsumfang ist mit dem von Snyk vergleichbar.

Revenera FlexNet Code Insights lädt entweder die gesamte Codebasis eines Projekts zum Scannen auf den Server oder verbindet den Scanserver mit einem Software-Repository, das er dann automatisch nach Vorgaben scannt. Im Unterschied zu Werkzeugen, die auf die Cloud und DevOps-Prozesse ausgerichtet sind und sich an verschiedene andere Tools andocken, hat FlexNet Code Insight eine eher konservative Herangehensweise: Das System dient als „Single Source of Truth“ für den gesamten Code des Projekts, erstellt SBOMs und identifiziert Schwachstellen.

Unternehmen, die bei ihren Artefakt-Repositorys auf JFrog setzen, können mit **JFrog XRay** die dazu passende SCA-Lösung einsetzen, die eine native Artifactory-Anbindung bietet und Zugriff zu sämtlichen Metadaten im Repository hat und auch Binaries scannt. XFrog XRay identifiziert Lizenzen und Schwachstellen, erlaubt die Definition von Policies und exportiert SBOMs, JFrog pflegt eine Schwachstellendatenbank, die sich aus der VulnDB und eigenen Einträgen speist. Mit dem FrogBot lässt sich JFrog XRay auch in GitHub-Repositorys einbinden.

Veracode kombiniert SCA mit statischer Codeanalyse. Bei Letzterer versteht es auch Cobol, PRG oder verschiedene SQL-Dialekte, ist also auch im traditionellen IT-Umfeld zu Hause. **Veracode SCA** kommt mit 13 verbreiteten moderneren Sprachen und

den entsprechenden Paketformaten zurecht. Es ist ein umfangreiches, sowohl auf Security als auch auf Compliance ausgerichtetes SCA-Produkt, das alle entscheidenden Funktionen und Integrationsmöglichkeiten mitbringt.

Die Nexus-Plattform von Sonatype ist bei Cloud-Entwicklern vor allem für ihr Artefakt-Repository bekannt, das direkt mit JFrog Artifactory konkurriert. Sicherheitsforscher kennen Sonatype eher wegen seiner Schwachstellendatenbank. Mit **Nexus Lifecycle** hat das Unternehmen ein SCA-Produkt im Angebot, das zwar auf seine übrigen Securityprodukte abgestimmt, aber nicht auf Anwender der Nexus-Repositorys beschränkt ist. Nexus Lifecycle ist ein umfassendes Produkt für den Enterprise-Einsatz.

Fazit

Für die Auswahl einer der großen kommerziellen Lösungen ist auf jeden Fall eine genaue Anforderungsanalyse sowohl seitens der Entwickler und des Sicherheitsteams als auch – wenn Complianceaspekte wichtig sind – der Rechtsabteilung notwendig. Sehr empfehlenswert zur Vorbereitung ist der 13-seitige „Open Guide to Evaluating Software Composition Tools“ der Linux Foundation, der die wichtigsten Metriken identifiziert und dabei hilft, ihre Relevanz für das eigene Projekt oder Unternehmen einzuschätzen.

Eine längere, gut geplante Testphase vor der Lizenzierung des Produktes ist unabdingbar und bei allen seriösen Anbietern möglich. Bei Herstellern, die ihre komplette Nutzer- oder Administrationsdokumentation frei verfügbar machen, lassen sich einige Anforderungen schon vorher klären, denn nicht selten zeigen die Dokumente, wie die in Fact Sheets beworbenen Features tatsächlich funktionieren, oder sie decken Einschränkungen auf.

Zu beachten ist auch, dass bei möglicherweise schnell eingekauften SaaS-Angeboten Sourcecode und Metadaten das

Unternehmen verlassen können und unter Umständen auf US-Servern landen. Im Sinne der DSGVO dürfte das meist zwar unproblematisch sein, da es sich nicht um personenbezogene Daten handelt. Aber das eine oder andere Unternehmen hat vielleicht doch gute Gründe, den Sourcecode lokal zu halten – speziell, wenn es um Auftragsentwicklung geht. Zum Glück gehen die meisten Anbieter mit Informationen, wo und wie lange Kundendaten gespeichert werden, recht transparent um.

Aus technischer Sicht essenziell ist, dass sich das SCA-Produkt an möglichst viele der im Unternehmen eingesetzten Entwicklungs- und Deployment-Werkzeuge anbinden lässt – am besten auch an solche, die für später auf der Wunschliste stehen. Kleinere Integrationen lassen sich über die API nachrüsten.

Darüber hinaus ist eine niedrige Falsch-positiv-Rate bei den gemeldeten Schwachstellen wichtig, damit das Werkzeug den Entwicklern nicht im Weg steht. Idealerweise kommt eine Überprüfung dazu, ob der Code mit der Schwachstelle überhaupt aufgerufen wird. Dieses Feature ist unter verschiedenen Namen (Reachable Path, Exploitable Path etc.) bei Anbietern verfügbar, die auch SAST-Produkte im Portfolio haben, manchmal jedoch nur für ausgewählte Sprachen.

Eine gute Integration in IDEs ist ein großes Plus, denn so verhindert man, dass Schwachstellen überhaupt den Weg in den Code finden und nicht erst beim Einchecken in das Repository oder noch später auffallen. Automatisierung und permanente Überwachung der CI-Pipelines sollte möglich sein.

Schwieriger wird es, wenn das Werkzeug dazu benutzt werden soll, unternehmensweite Policies durchzusetzen und Complianceanforderungen zu überwachen. Dann bringt ein Test der Software innerhalb eines Entwicklerteams keinen nennenswerten Erkenntnisgewinn. Hier könnte ein abteilungsübergreifendes Projektteam die Anforderungen möglichst genau spezifizieren und nach einer sinnvollen

Vorauswahl eine kleine Zahl von Anbietern genauer unter die Lupe nehmen.

Wenn es darum geht, überhaupt erstmalig werkzeuggestützte Software-Composition-Analyse zu betreiben, ließe sich alternativ in einem Developer-Team ein eher an den Bedürfnissen der Entwickler ausgerichtetes Produkt einführen. Es muss aber zumindest von seinen Spezifikationen her den Compliancebereich mit abdecken könnte und ginge erst nach positiven Erfahrungen der Developer in den unternehmensweiten Einsatz. Auch ein nicht ganz optimales Werkzeug zur Ermittlung von Risiken durch Open-Source-Software sichert die Softwarelieferkette besser ab als gar keines. (ulw@ix.de)

1. Quellen
2. Udo Schneider; SBOMs – Stücklisten für Software; iX 10/2022, S. 54
3. Weitere Infos zu Tools und Auswahlkriterien: ix.de/zvbm

---

ownCloud Infinite Scale mit Microservice-Architektur

Mit mehr als zehn Jahren Praxiserfahrung hat ownCloud seine als Dropbox-Alternative gestartete Software grundlegend überarbeitet. Für ownCloud Infinite Scale verspricht man, die Leistungsgrenzen der Plattform zu verschieben.

Von Dr. Udo Seidel

-tract

- ownCloud ist mit seiner als Dropbox-Alternative

gestarteten Software seit mehr als zehn Jahren im Unternehmens- und Behördenumfeld aktiv.
- Die zunehmend hinderlichen Leistungsgrenzen von PHP initiierten einen grundlegenden Umbau der Plattform vom LAMP-Stack zur in Go geschriebenen Microservice-Architektur.
- Dank der Zusammenarbeit unter anderem mit dem CERN ist die Software für kommende Anforderungen im Cloud-Umfeld gut gerüstet.

Vor über zehn Jahren trat ownCloud mit der gleichnamigen Datenaustauschplattform als LAMP-basierte – wobei das P hier für PHP steht – Alternative zu Dropbox an. Mittlerweile hat der Nürnberger Anbieter auch ein DSGVO-konformes SaaS-Angebot im Portfolio und konnte sich in den Jahren vor der Pandemie als zentrale Cloud-Software im Bereich Datenaustausch für die bayerischen Kommunen etablieren. Dabei zeigte sich, dass die Architektur in großen Umgebungen an die Grenzen der PHP-Leistungsfähigkeit stieß. Also zogen die ownCloud-Entwickler und -Architekten Bilanz: Was hatte sich in den letzten knapp 10 Jahren bewährt? Was passt nicht mehr so richtig? Welche Trends gilt es zu beachten? Das war die Geburtsstunde von ownCloud Infinite Scale – kurz oCIS. iX hat sich die Plattform genauer angesehen.

Ein neues Entwicklungsmodell

Für die Anwendungsarchitektur der oCIS-Plattform hat ownCloud quasi auf der grünen Wiese angefangen – allerdings ohne die Erfahrungen oder die Neuentwicklungen des letzten Jahrzehnts zu ignorieren. Der erste oCIS-Git-Commit stammt aus dem August 2019. Anstelle des recht „gemütlichen“ PHP wechselte man auf Go als Programmiersprache. Diese erfreut sich nicht nur bei Cloud-Enthusiasten großer Beliebtheit, sondern verspricht auch einen deutlichen Leistungszuwachs. Eine weitere technische Neuerung ist der Wegfall einer zentralen (relationalen) Datenbank. Im Cloud-Umfeld ist das fast ein natürlicher

Schritt.

Die dritte wesentliche Veränderung ist das Entwicklungsmodell hinter oCIS. Sie hat sogar zwei verschiedene Dimensionen: eine organisatorische und eine technische. ownCloud setzt jetzt mehr auf Zusammenarbeit und Kooperation. So basiert die Datenspeicheranbindung auf EOS Open Storage, einem am CERN für den Einsatz beim LHC entstandenen Projekt (siehe ix.de/zsub). Eine weitere Kooperation betrifft eine freie Alternative zur Microsoft-Graph-Schnittstelle. Sie ist REST-basiert und erlaubt einen einfachen Zugriff auf die Daten verschiedener Azure-Cloud-Dienste auf Grundlage der Identität des Benutzers. Zusammen mit der Firma Kopano hat ownCloud das Projekt Libregraph ins Leben gerufen und nutzt dies in oCIS (siehe ix.de/zsub).

Dies ist aber nur die Spitze des Eisbergs. Ausgehend vom LAMP-basierten Produkt gibt es 20 dokumentierte Architekturentscheidungen, die wegweisend für die Entwicklung von oCIS waren und sind. Das Resultat ist ein Produkt mit einer Drei-Schichten-Architektur, basierend auf Microservices. Wer die Entwicklung vielleicht schon ein oder zwei Jahre beobachtet hat: Da gab es alle paar Wochen eine neue technische Vorschau. Etwas untypisch im traditionellen Open-Source-Umfeld zierte die Hauptversion schon damals eine 1 und keine 0. Im November 2022 erschien dann (konsequenterweise) oCIS 2.0.0 – reif für den produktiven Einsatz.

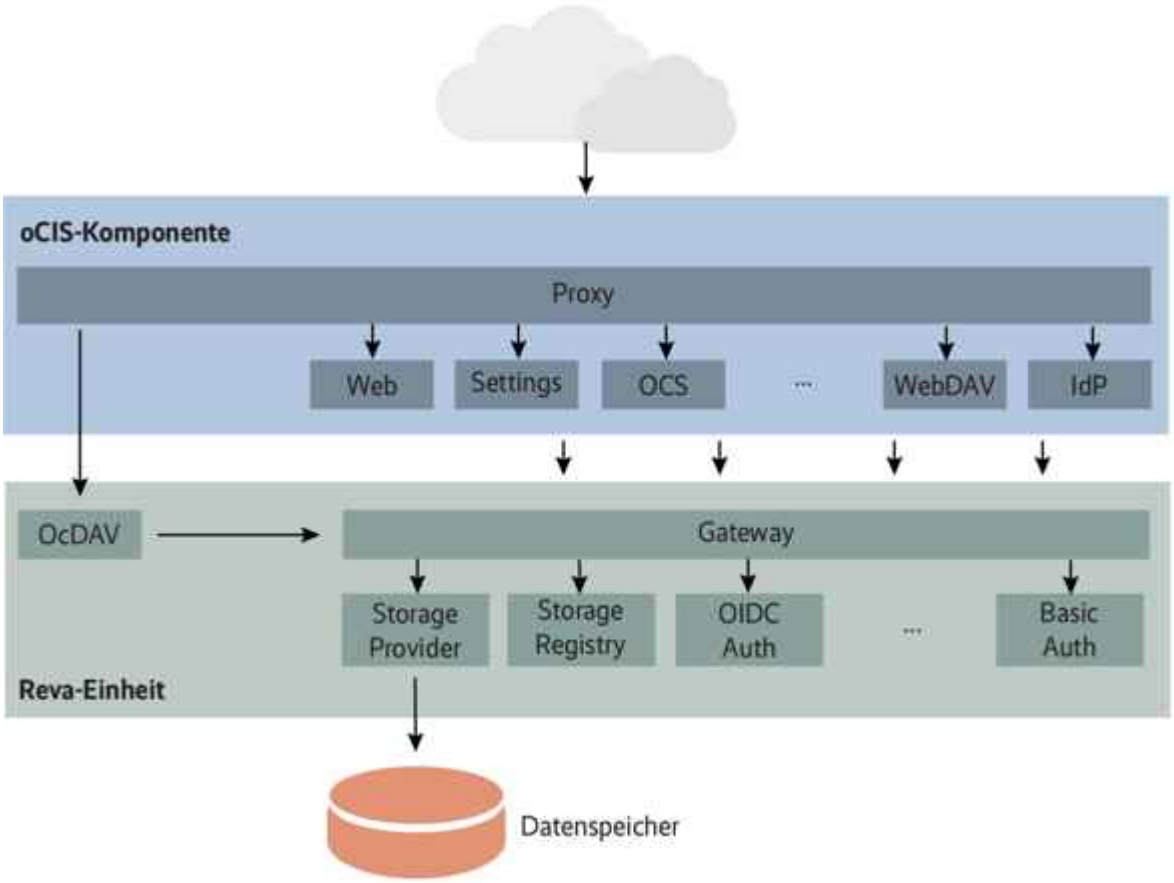
Die Software steht unter der Apache-2.0-Lizenz und läuft auf Linux und macOS. Ersteres in 32 und 64 Bit sowohl für x86 als auch für ARM. Es gibt Clients für iOS, Android, Windows, macOS und Linux sowie eine Weboberfläche für den Zugriff über einen Webbrowser. oCIS verpackt die Software in eine nicht einmal 90 MByte große Binärdatei. Für erste Schritte sind keine großen Hardwareanschaffungen nötig: Beim Hauptspeicher reichen 256 MByte, für produktive Umgebungen sollte man aber mindestens 4 GByte einplanen. Bezüglich CPU und Netzwerkbandbreite macht ownCloud keine konkreten Vorgaben. Die tatsächlichen

Anforderungen variieren je nach Anwendungsszenario zu sehr, bei der Netzwerkbandbreite spielen die Anzahl der Clients, der Veränderungen und deren Größe eine wesentliche Rolle.

Ohne weitere Unterstützung durch ownCloud sind Installation und Betrieb von oCIS quasi kostenfrei. Nur Support vom Hersteller, Zugriff auf das Kundenportal und auch Funktionen wie Clients mit dem eigenen Firmenlogo sind mit Kosten verbunden. Die Konditionen orientieren sich an der Benutzeranzahl mit skalierendem Discount bei großen Umgebungen.

Von ganz oben betrachtet

Grob betrachtet besteht oCIS aus drei Komponenten. Da ist zunächst der in unterschiedlichen Inkarnationen vorliegende Client. Der oCIS-Server-Teil besteht aus zwei Hauptkomponenten: Benutzer- und Datenverwaltung. Zum Authentifizieren der Benutzer greift oCIS auf OpenID Connect (OIDC, siehe ix.de/zsub) zurück. Damit kann die Software alle mit diesem Protokoll kompatiblen Identity Provider (IDP) verwenden. Im Cloud-Umfeld ist dieser Ansatz inzwischen der De-facto-Standard. Im einfachsten Fall kann hier ein LDAP-Server einspringen. Genau genommen ist dieser ein Identity Management System (IDM). Der LDAP-Server muss dabei nicht dediziert für oCIS installiert sein. Für die allerersten Schritte bringt oCIS einen kleinen IDP mit, der auf dem bereits erwähnten Libregraph fußt. Für das Kennenlernen der Software mithilfe lokaler Benutzer reicht das vielleicht schon. Ein realistischer Einsatz, bei dem auch Benutzer mehrerer Firmen zusammenarbeiten, sollte unbedingt OIDC verwenden.



Für oCIS setzt ownCloud auf eine dreischichtige Microservice-Architektur (Abb. 1).
Bei der Datenverwaltung gibt es zunächst die Datenträger selbst. Das können lokale Platten in einem Rechner sein, aber auch netzwerkbasierter Storage wie Amazon S3 oder NFSv4 ist möglich. Der Zugriff darauf erfolgt über Treiber. Intern verwaltet oCIS die Speicher in sogenannten Spaces, die es in einer Registry vermerkt. Die Space Registry ist der Einstieg in die unteren Schichten des oCIS-Storage-Stacks. Sie verwaltet den Namensraum der Benutzer, sprich: Welche Daten dürfen sie lesen und/oder schreiben? Über die registrierten Spaces und Treiber erfolgt dann der eigentliche Zugriff. Im Detail ist die Datenverwaltung etwas komplexer – dazu später mehr.

Für die internen Netzwerkverbindungen verwendet oCIS fast ausschließlich das von Google initiierte RPC-Framework gRPC. Durch den Microservice-Ansatz ist das nicht weiter verwunderlich. Damit geht einher, dass oCIS für sich eine ganze Reihe von Ports beansprucht. Wer für die Zukunft

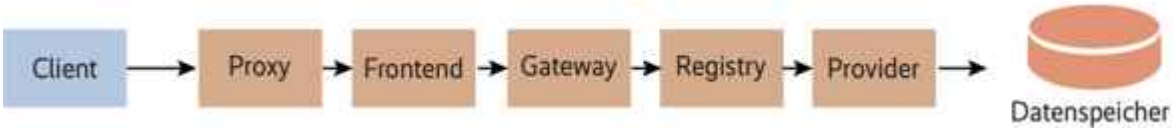
gewappnet sein will, sollte alle zwischen 9000 bis 10 000 schon mal reservieren. Im Moment scheint sich aber alles im Bereich bis 9300 abzuspielen. Neben gRPC kommt natürlich auch LDAP für die Benutzerauthentifizierung zum Einsatz. Zur einfacheren Netzwerkkommunikation verwendet oCIS intern ein Gateway. Es dient als Pförtner zu den für die Datenverwaltung zuständigen Microservices.

Die Kommunikation nach außen teilt oCIS in zwei Kategorien. Für externe Datenspeicher kommen Protokolle wie S3 oder NFSv4 zum Einsatz, die oCIS-Clients wiederum benutzen das ownCloud-eigene Synchronisierungsprotokoll auf WebDAV-Basis. Dabei ist hier eine Art API-Gateway vorgeschaltet, das als Eingang zu den Microservices für Benutzerauthentifizierung, den Einstellungen oder den ownCloud-APIs dient.

Vom Ästlein zum Zweiglein

Mit oCIS haben die ownCloud-Entwickler eine Microservice-Struktur eingeführt, die auf den ersten Blick etwas überwältigt. Die Software landet als eine Binärdatei auf dem System und startet im Vollausbau weit über 20 Dienste. Die gehören zu drei Gruppen: Benutzerverwaltung, oCIS-Komponenten und die vom CERN stammende Reva-Einheit. Noch mal zur Erinnerung: ein Gateway dient als Schnittstelle zwischen den beiden Letzteren. Zur Reva-Einheit gehören Dienste für die Kommunikation mit den Datenträgern, etwa storage provider. Dazu später mehr. Es gibt auch eine Verbindung von der Reva-Einheit zur Benutzerverwaltung. Dort sind Identity Provider und Management beheimatet und füttern die Attribute user provider und group provider mit den erforderlichen Daten. Die oCIS-Komponenten kann man als Schaltstelle des Gesamtsystems verstehen. Hier erfolgt die Verarbeitung und Weiterleitung der Clientanfragen. Es gibt jeweils einen Microservice für die verschiedenen Protokolle. Hier findet sich auch die Konfiguration des Systems wie der Detailgrad der Protokolldateien oder der Pfad zu den x509-Zertifikaten für

die verschlüsselte Kommunikation.



Dieser exemplarische Arbeitsablauf zeigt das Zusammenspiel der oCIS-Microservices (Abb. 2).
Intern benutzt oCIS das Projekt suture zur dynamischen Verwaltung der Microservices. Auch an anderer Stelle haben die ownCloud-Entwickler das Rad nicht neu erfunden. Als Gerüst für die Implementierung des Microservice-Ansatzes nutzen sie die Ergebnisse des Projektes go-micro (siehe ix.de/zsub). Um unnötige Quellcodekopien zu vermeiden und die vielen Dienste zu vereinheitlichen, kommt die Eigenentwicklung ocis-pkg zum Einsatz. Sie benutzt das go-micro-Gerüst zur Implementierung der Schnittstellen für Serverkomponenten und Clients.

Der Wechsel von LAMP zur Microservice-Architektur ist den ownCloud-Entwicklern gelungen. Das gilt ebenso für die Verwendung bereits etablierter Software und Komponenten. Die Tatsache, dass oCIS dennoch als eine einzelne Binärdatei daherkommt, erleichtert die Installation, Aktualisierung und Wartung. Die Anzahl der reservierten Ports wirkt allerdings etwas zu großzügig – zumindest in der geplanten Stufe. Das kann insbesondere dann problematisch werden, wenn die einzelnen Microservices auf verschiedenen Rechnern laufen sollen. Momentan ist das nur ein hypothetischer Fall, jedenfalls nach den Installationsrezepten von ownCloud. Generell ist die Verteilung von Microservices auf verschiedenen Rechner im Cloud-Umfeld nicht unüblich. Dann müssen die Entwickler auch über die Verschlüsselung der gRPC-Verbindungen nachdenken. Und eigentlich ist die Antwort offensichtlich: ein Service Mesh. Hier gibt es verschiedene Implementierungen und Anbieter.

Hinter den Datenkulissen

Für das Herzstück, die Datenverwaltung zum Verteilen von

Dateien und Verzeichnissen, haben sich die ownCloud-Entwickler für das ebenfalls am CERN beheimatete Projekt Reva entschieden (siehe ix.de/zsub). Das ist eine Referenzimplementierung des CS3-Schnittstellenstandards (CS3 – Cloud Storage Services for Synchronization and Sharing). Die Nutzung von Reva ist eine der oCIS-Architekturentscheidungen. Ein wichtiger Punkt war hier auch die Implementierung eines verteilten und hochverfügbaren Systems zum dauerhaften Speichern von Benutzerinformationen. In traditionellen ownCloud-Installationen war dies nicht möglich oder hätte signifikanten Entwicklungsaufwand erfordert.

Wie erwähnt kann oCIS sowohl auf lokale Datenträger als auch auf Storage im Netz schreiben. Dabei abstrahiert die Software noch mal über eine weitere Schicht: Decomposed FS. Die Kernidee dieses Dateisystems ist die Dateiverwaltung anhand von UUIDs und eine für Nutzer nicht sichtbare sowie recht flache Struktur der Datenablage. Nach außen zeigt oCIS die gewohnte Verzeichnisstruktur mit Dateien. Im Hintergrund findet oCIS die entsprechenden Daten aber nicht per Pfad, sondern per UUID. Über Treiber greift Decomposed FS auf die eigentlichen Datenträger zu.

Lokale Platten müssen ein POSIX-Dateisystem haben, die Entwickler empfehlen XFS oder ZFS – mit Einschränkungen funktioniert aber auch ext4. Ein wichtiges Merkmal der POSIX-Dateisysteme ist die verfügbare Speichergröße für erweiterte Attribute. Wie Listing 1 zeigt, speichert Decomposed FS dort viele Verwaltungsinformationen. In großen Installationen mit ext4 würde oCIS dann künstlich limitiert. Bei den Netzwerkspeichern ist die Situation komplizierter. Prinzipiell funktioniert oCIS mit NFS, S3, CephFS oder sogar EOS. Für den produktiven Einsatz ist bislang aber nur Amazon S3 freigegeben. Es ist zu erwarten, dass die anderen nachfolgen. Wer jetzt schon mal reinschnuppern will, findet bei ownCloud gute Dokumentation dafür.

Listing 1: Ausgabe von getfattr -d zeigt Extended-Attribute

```
# file: storage/users/spaces/4c/510ada-c86b-4815-8820-42cdf82c3d51/nodes/4c/[…]
user.ocis.blobid=""
user.ocis.blobsize="0"
user.ocis.name="Albert Einstein"
user.ocis.owner.id="4c510ada-c86b-4815-8820-42cdf82c3d51"
user.ocis.owner.idp="https://localhost:9200"
user.ocis.owner.type="primary"
user.ocis.parentid=""
user.ocis.propagation="1"
user.ocis.space.alias="personal/einstein"
user.ocis.space.name="Albert Einstein"
user.ocis.space.type="personal"
```

Abstraktion ist ein wiederkehrendes Motiv bei der oCIS-Datenverwaltung. Das kann beim Lesen der Dokumentation verwirrend sein. Allerdings bleibt Anwendern diese Komplexität verborgen. Nur wer es genau wissen will, muss sich mit Begriffen wie Storage Space oder Space Registry herumschlagen. Da eine ausführliche Darstellung den Artikelumfang sprengen würde, folgt hier nur eine kurze Darstellung. Storage Space ist eine logische Zusammenfassung von Dateien und Verzeichnissen. Das könnten etwa die Daten eines Projektes oder einer Abteilung sein. Wichtig ist, dass es ausschließlich einen logischen Besitzer der Daten gibt. Dieser Storage Space ist mit einer eindeutigen Nummer versehen.

Die nächste Ebene nennt sich Storage Provider. Sie verwaltet einen oder mehrere Storage Spaces. Dabei ist diese Zuordnung zwar immer eindeutig, aber veränderbar. Ein Storage Space kann von einem Provider zu einem anderen wechseln. Der Storage Provider ist außerdem für den eigentlichen Zugriff auf die physischen Datenspeicher zuständig. Dabei kommen die oben beschriebenen Treiber zum Einsatz. Der Storage Provider teilt sich die Verwaltungsaufgaben mit der Space Registry. Letztere kümmert sich um Dinge wie freien Platz, Benutzerzugehörigkeit

oder auch Quotas. Hier ist der Namensraum angesiedelt, der entscheidet, ob der Anwender Zugriff auf die Daten hat oder nicht. Der Storage Provider ist auf der unteren Schicht des Stacks unterwegs, dem Datenzugriff auf den eigentlichen Storage.

Zum Abschluss noch zwei Hinweise. Bevor eine Benutzeranfrage beim Storage Provider ankommt, durchläuft sie einige Stationen beziehungsweise Microservices. Das fängt beim oCIS-Proxy an und führt am Ende zum Gateway, das oben als Pförtner in die Reva-Welt beschrieben wurde. Wen es genauer interessiert, der kann den genauen Datenfluss in der ownCloud-Dokumentation nachlesen (siehe ix.de/zsub). Hinweis Nummer zwei ist eher praktischer Natur: Wer seine Daten von einer anderen Plattform in oCIS migrieren möchte, kann auf das Werkzeug Rclone zurückgreifen.

Potenzielle Installationsszenarien

Der Start mit oCIS unter Laborbedingungen ist denkbar einfach (siehe Kasten „Los gehts“). Auch die Integration in das Hoch- beziehungsweise Herunterfahren von Linux lässt sich ganz einfach bewerkstelligen. Die Dokumentation enthält eine detaillierte Anleitung inklusive systemd-Konfiguration. Das Gleiche gilt für die Benutzer, die auf dem Containerpfad unterwegs sind, sowohl mit als auch ohne Orchestrierung durch Kubernetes. Läuft schon ein entsprechender Cluster, lässt sich oCIS recht schnell mit den Helm Charts des Projektes aufsetzen. Container und Kubernetes sind laut Aussage des Projektes die bevorzugte Plattform. Da verwundert es nicht, dass hier quasi alles im Detail schon bereitliegt (siehe ix.de/zsub).

Sehr positiv ist zu vermerken, dass sich für quasi jede Plattform ein Installationsrezept für oCIS findet. Egal, ob echte Hardware, VM, Container mit oder ohne Orchestrierung: Jeder kommt auf seine Kosten. Die Entscheidung, die Software als einzelne ausführbare Datei auszuliefern, hat sich hier

ausgezahlt. Die Unterstützung von 32-Bit-Plattformen – sowohl auf x86 als auch auf ARM – ist inzwischen auch selten und verdient das Prädikat Luxus. Damit sind die ersten und vielleicht auch zweiten Schritte sehr einfach. Die Bewährungsprobe im harten Produktionsalltag steht aber noch aus.

Los gehts

Die ersten Schritte im Labor mit oCIS sind einfach. Die Software kommt als eine einzige Binärdatei daher. Einfach herunterladen, eventuell umbenennen und sie als ausführbar markieren – und schon kann es losgehen. Der Start erfolgt in zwei Stufen. Zunächst generiert man über das Kommando ocis init eine Konfigurationsdatei. Dabei erzeugt das Programm eine Grundkonfiguration und legt den Admin-Benutzer inklusive Passwort an. Danach lässt sich die oCIS-Instanz mit ocis server starten (siehe Listing 2). In diesem Fall laufen alle mitgelieferten Microservices; der Aufruf ocis list zeigt sie an.

Ein alternativer Weg führt über ein vorgefertigtes Docker-Image von ownCloud (siehe ix.de/zsub). Auch hier ist der Start zweiphasig. Zunächst startet man die Containerinstanz und gibt das Kommando ocis init mit auf den Weg. Danach kommt dann das Hochfahren der oCIS-Instanz. Dabei gilt es, den Port 9200 für den Zugriff auf das Webfrontend entsprechend umzuleiten:

```
docker run --rm -p 9200:9200 -v ocis-config:/etc/ocis -v ocis-data:/var/lib/ocis owncloud/ocis
```

Und es geht sogar noch einfacher. Wer nur mal in oCIS reinschauen möchte, kann die Onlinedemo benutzen (siehe ix.de/zsub). Die notwendigen Log-in-Daten stehen auf der Startseite.

Listing 2: Fünf Schritte zur ersten oCIS-Instanz

```
$ wget -nd https://download.owncloud.com/ocis/ocis/stable/2.0.0/ocis-2.0.
```

```
0-linux-amd64
...
$ mv ocis-2.0.0-linux-amd64 ocis
$ chmod 755 ocis
$ ./ocis init
...
========================================
generated OCIS Config
========================================
configpath : /home/ocis/.ocis/config/ocis.yaml
user : admin
password : 5p2ZjjGuiI#rVRF*P0SHpU+KKzu$&Dnv
$    OCIS_INSECURE=true    IDM_CREATE_DEMO_USERS=true
PROXY_HTTP_ADDR=0.0.0.0:9200 OCIS_URL=https://localhost:9200
./ocis server
```

Im Produktionsbetrieb stellt sich immer die Frage nach Skalierbarkeit, also einer hohen Verfügbarkeit und ausreichender Leistungsfähigkeit des Dienstes. Bei oCIS gilt es, dafür drei Fragen zu beantworten. Welche Microservices dürfen parallel laufen? Wie halte ich die Informationen der verschiedenen Instanzen synchron? Wie verteile ich die Last, ohne Duplikate oder Lücken zu erzeugen? Zur ersten dieser Fragen finden sich ausreichend Informationen in der Dokumentation. Sie beschreibt unter anderem, welche Dienste nur einmal laufen dürfen. Außerdem zeigt ein genauerer Blick in die Kubernetes Helm Charts, wo horizontale Skalierung einfach möglich ist. Das oCIS-Binary enthält alle Microservices und man entscheidet per Kommandozeilenoption, welche tatsächlich starten sollen.

oCIS kann sowohl vertikal als auch horizontal skalieren – typischerweise, wenn es um die Erweiterung des verfügbaren Datenspeichers geht. Sprich: Weitere Rechner mit lokalen Datenträgern treten dem oCIS-Verbund bei. Analoges gilt, wenn die Netzwerkbandbreite für die Clients an ihre Grenzen stößt.

Unbeantwortet ist dabei die Frage der Datenverteilung und -synchronisation auf den Speichermedien. Haben alle oCIS-

Instanzen Zugriff auf alle Daten? Gibt es eine Partitionierung/Unterteilung? Wenn ja, wie wissen die oCIS-Microservices, wer wofür zuständig ist? Hier gibt es keine befriedigende Antwort in der Dokumentation. Natürlich löst ein verteiltes Dateisystem wie CephFS, NFSv4 oder auch ein Cloud-Speicher S3 dieses Problem. Für den produktiven Einsatz ist aber im Moment nur S3 erlaubt.

Auch vertikale Skalierung kann vorkommen. Die oCIS-Instanzen benutzen den RAM als Zwischenspeicher für den Zugriff auf die eigentlichen Daten. Die einfachste Variante ist also die Vergrößerung des Hauptspeichers. Alternativ wäre auch das Verteilen über neue Instanzen möglich, was dann horizontales Skalieren wäre. Weniger Daten, also weniger Belastung des Hauptspeichers.

In der Dokumentation von oCIS enttäuscht jedoch der Abschnitt über Hochverfügbarkeit und Skalierung. Es fehlen klare Rezepte und Handlungsanweisungen. Hier muss ownCloud unbedingt nachbessern. Einmal, weil diese Aspekte essenziell für den seriösen Einsatz im produktiven Umfeld sind. Außerdem wirkt der Bruch im Informationsgehalt der sonst sehr guten Dokumentation etwas befremdlich.

Fazit

Mit oCIS hat ownCloud einen gewaltigen Wechsel seiner Datenverteilungsplattform vollzogen. Da ist der Technologiewechsel von LAMP zu einer Microservice-Architektur mit Go als Programmiersprache. Der geoverteilte und föderative Ansatz ist modern und den aktuellen Anforderungen entsprechend. Eine wichtige Aufgabe für die nähere Zukunft ist die Unterstützung weiterer Cloud-Speicherdienste und verteilter Dateisysteme für die Datenablage. Die Dokumentation ist insgesamt sehr umfangreich. Die Kooperationen – insbesondere mit dem CERN – geben dem Produkt eine zusätzliche Qualität in Bezug auf Planungssicherheit und Interoperabilität. Die ersten Schritte sind einfach und gut

dokumentiert. Die Bewährung im harten Produktionsalltag steht aber noch aus. (avr@ix.de)

1. Quellen
2. Links zu weiteren Hintergrundinformationen und zu Download- und Demo-Seiten: ix.de/zsub

Dr. Udo Seidel

war seit 1996 als Linux-/Unix-Trainer, Administrator, Senior Solution Engineer und Chefarchitekt tätig. Er arbeitet heute als Senior Customer Experience Architect im europäischen Gebiet für Kong Inc.

Softwarequalität mit Teamscale steuern

Die Software-Intelligence-Plattform Teamscale hilft Entwicklungs- und Testteams beim Messen der Code- und Produktqualität.

Von Dr. Carsten Weise und Christoph Singer

-tract

- Teamscale ist eine Software-Intelligence-Plattform, die

Software analysiert, überwacht und verbessert.

- Das Tool führt die Ergebnisse statischer und dynamischer Tests mit dem Quelltextrepository zusammen.
- Dashboards geben einen schnellen Überblick über den Stand der Softwarequalität. Treemaps visualisieren unter anderem die Testüberdeckung.
- Teamscale verfolgt Änderungen im Code und ermittelt Testlücken.
- Zahlreiche Programmiersprachen, externe Werkzeuge zur statischen Codeanalyse und Testüberdeckung sowie Versionskontrollsysteme lassen sich einbinden.

Um in einer immer komplexeren und anspruchsvolleren Softwarewelt qualitativ hochwertige Produkte zu liefern, braucht es eine toolunterstützte Qualitätssteuerung. Techniken wie Continuous Integration/Deployment, DevOps, Internet of Things, Internet of Production, Design Thinking und Lean Development sind auf schnellere und kürzere Iterationen sowie kleinteilige Entwicklung angewiesen. Die kleinteiligen Zyklen in der Softwareentwicklung sind Grundlage hoher Softwarequalität: Eine Applikation wird nicht in einem Produktionsschritt zum fertigen, fehlerfreien Produkt. Stattdessen messen Entwicklerinnen und Entwickler deren Qualität kontinuierlich während der Entwicklung und bessern gefundene Abweichungen vom Soll (Fehler) nach.

Dabei sind sie auf flexible, verlässliche Werkzeuge angewiesen, um schnelles Feedback zur Qualität zu bekommen und dabei nichts an Änderungen und Korrekturen zu übersehen. Solche Werkzeuge sind etwa CodeCity und Seerene sowie Teamscale, das die Firma CQSE (Continuous Quality in Software Engineering) entwickelt hat, ein 2009 gegründetes Spin-off des Lehrstuhls für Software und Systems Engineering der TU München.

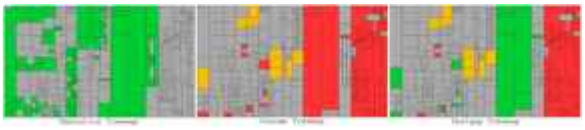
Teamscale sammelt Informationen zur Softwareerstellung und -prüfung und bereitet sie auf. Die Applikation ist ein Server,

der über ein Webinterface erreichbar ist. Die Installation des Servers ist nativ, als Docker-Image oder auch als Cloud-Service möglich. Er lässt sich nicht nur über das Webinterface, sondern auch über eine REST-API ansteuern.

Teamscale berücksichtigt zwei wesentliche Prüfaktivitäten während der Softwareerstellung: die statische Analyse und den dynamischen Test (Test des ausführbaren Codes). Das Werkzeug hilft, verschiedene Testarten durchzuführen, die Testdaten zu sammeln und zu überwachen, ob notwendige Änderungen berücksichtigt und anstehende Korrekturen ausgeführt wurden. Der Artikel beschäftigt sich zunächst mit dem dynamischen Test, geht dann auf die statische Analyse ein und betrachtet als weiteres Feature die Architekturanalyse.

Codeänderung + Testüberdeckung = Testlücke

Wie andere Testüberwachungstools kann Teamscale Testüberdeckung messen und darstellen. Es verwendet für die Visualisierung Treemaps (Abbildung 1). Die Execution Treemap zeigt die Testüberdeckung, also die Überdeckung durch die Testausführung (Test Execution). Grüne Flächen sind bereits getestete Anteile, graue Flächen die noch ungetesteten Teile. Die farbigen Flächen sind hierbei die Prozeduren (Methoden) des Codes. Die Größe der Fläche entspricht der Methodengröße (Komplexität), gemessen als Anzahl Codezeilen (LOC: Lines of Code).



Teamscale zeigt durch Treemaps Testüberdeckung, Codeänderungen und Testlücken an (Abb. 1).
Alle Qualitätsmessungen stellt Teamscale übersichtlich auf einem konfigurierbaren Dashboard dar. Dort findet man alle gängigen Darstellungsformen (beispielsweise Torten- und Balkendiagramme). Die Treemaps stechen dabei optisch hervor

und liefern eine schnelle Übersicht für Überdeckungsmaße. Sie sind interaktiv und erlauben, auf die Details zu zoomen: So kann man zum Schluss auch auf detailliertere Information als die Funktionsüberdeckung zugreifen.

Teamscale kennt zusätzlich zur Testüberdeckung auch die Codeänderung (Change Treemap). Kombiniert man diese Informationen, erhält man die Testlücke (Testgap Treemap). Jede Methode ist entweder grau (unverändert – getestet oder ungetestet), rot (neu, aber nicht getestet), gelb (geändert, aber nicht getestet) oder grün (neu oder geändert und getestet).

Diese automatische Lückenanalyse zeigt bei kontinuierlichen Änderungen im System immer den aktuellen Stand und ob alle Änderungen getestet sind. In einem Prozess mit kurzen Iterationen und vielen Änderungen erhält man damit schnelles Feedback: Entscheider sehen, ob das nächste Release reif für den Kunden ist, und Tester, wo noch nachzutesten ist. Die Lückenanalyse sagt allerdings nur aus, ob eine Änderung bereits getestet wurde. Zum Testergebnis, also ob der Test erfolgreich war, gibt sie keine Auskunft. Teamscale kennt aber auch die Testergebnisse, weshalb sich damit auch zum Testerfolg Übersichten als Treemaps erstellen lassen.

Automatisierte Auswirkungsanalyse spart Zeit

Die Lückenanalyse ist eines der wesentlichen Features von Teamscale und auch Alleinstellungsmerkmal unter den Qualitätswerkzeugen. Dahinter verbirgt sich eine inkrementelle Analysemaschine, die kontinuierlich Änderungen im Code und Testvorgänge in ihrer zeitlichen Abfolge registriert und daraus die notwendigen Informationen aufbereitet.

In der Testtheorie spricht man von Auswirkungsanalyse: Bei Änderungen müssen Tester die direkten und indirekten Auswirkungen auf das Gesamtsystem verstehen, um den Umfang der

notwendigen Tests zu bestimmen. Teamscale liefert diese Auswirkungsanalyse sofort und automatisch. Das spart Zeit und erhöht die Zuverlässigkeit, was in kontinuierlichen Entwicklungsprozessen förderlich ist.

Allerdings haben solche Methoden Grenzen: Weder eine manuelle noch eine automatisierte Auswirkungsanalyse kann zuverlässig alle Fehlerquellen identifizieren, die durch Änderungen entstehen. In diesem Sinne soll und darf man von der Lückenanalyse keine Vollständigkeit erwarten. Im Normalfall wird sie aber die wahrscheinlichsten Fehlerquellen aufdecken und damit das Risiko unentdeckter Fehler senken.

Bei der Auswirkungsanalyse bietet Teamscale eine Paretooptimierung: Es sind diejenigen Testfälle auswählbar, die bei möglichst kurzer Ausführungszeit die größte Erhöhung der Testüberdeckung erzielen. Testaktivitäten lassen sich daher effizient planen und durchführen. Insbesondere können Entwickler umfangreiche Regressionstests auf ein handhabbares Maß beschränken.

Zugriff auf Coderepositorys mit Adaptern

Für die unterschiedlichen Analysen braucht Teamscale Informationen zu Codeänderungen und zu den von Tests ausgeführten Codeteilen. Damit es Zugriff auf solche Informationen erhält, muss man es mit entsprechenden Quellen in der Entwicklungs- und Testumgebung verbinden. Für die Codeänderung ist ein Zugriff auf die Coderepositorys erforderlich. Teamscale stellt dafür viele Adapter für gängige Versionskontrollsysteme bereit. Nutzer können sich beispielsweise mit GitHub, Azure und SAP-Repository-Servern sowie mit dem eigenen Dateisystem verbinden. Teamscale erlaubt das gleichzeitige Arbeiten mit mehreren verschiedenen Repositorys.

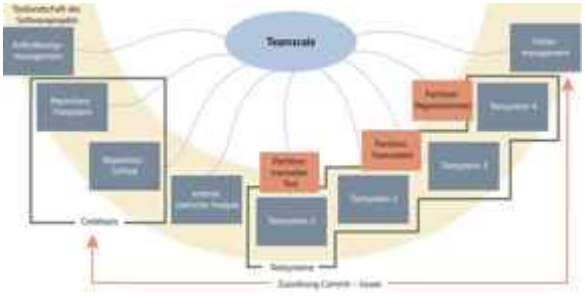
Teamscale benötigt für die Testüberdeckung Überdeckungsinformation aus der Testausführung. Das geschieht

entweder durch manuelles oder automatisiertes Hochladen der Information in einem gängigen Dateiformat (etwa JaCoCo, LLVM Coverage oder LCOV) oder automatisch über Coverage-Agenten. Teamscale beherrscht alle geläufigen Programmiersprachen und Dateiformate. Ordnet man die Information verschiedenen Partitionen zu, ist eine kleinteilige Analyse nach eigenen Vorstellungen möglich.

Die Testüberdeckung ist im automatisierten und im manuellen Test messbar. Wichtig ist nur, dass während der jeweiligen Testausführung ein Werkzeug die Testüberdeckung aufzeichnet und die Daten an Teamscale überträgt. Mittels Partitionen lassen sich manuelle und automatisierte Tests unterscheiden – auch weitere Differenzierungen bei den Testaktivitäten sind möglich. Ebenso können Anwender über die Partitionen die Tests auf verschiedenen Testservern zu einer gemeinsamen Messgruppe zusammenfassen. Die Systeme müssen dabei dedizierte Testsysteme sein, da alle Programmausführungen in die Messungen eingehen. Aber selbst hier kann man bei genügendem Verständnis der Teamscale-Schnittstelle individuelle Lösungen aufbauen.

Überwachung mit dem Datenkraken

Teamscale gestaltet sich als Datenkraken im positiven Sinn: Es sammelt die wichtigen Messdaten aus der jeweiligen Komponente der Toollandschaft (Abbildung 2). Verknüpft man die Daten für komplexe Analysen, erhält man aussagekräftige Statistiken.



Teamscale als Datenkraken sammelt Messdaten aus allen Komponenten und überwacht sie (Abb. 2).
Zusätzlich lässt sich Teamscale mit dem Fehlermanagement

verbinden. Instrumentiert man das Fehlermanagement, sodass Teamscale den Zusammenhang zwischen Commits im Repository und Issues im Fehlermanagement versteht (zum Beispiel durch Eintrag der Issue ID in der Commit Message), sind Änderungen und Testüberdeckung auch auf Issue-Ebene nachverfolgbar. Anwenderinnen und Anwender erhalten damit eine wertvolle Unterstützung, detailliert den verschiedenen Aktivitäten der Weiterentwicklung und Fehlerkorrektur zu folgen und Lücken aufzuspüren. Zudem verbindet sich Teamscale mit dem Anforderungsmanagement.

Teamscale greift auf Informationen aus Entwicklung und Test zu und kann deshalb viele Messungen, Statistiken und Übersichten erstellen. Die für ein Projekt wichtigsten Messungen und Statistiken stellt es in einem Dashboard dar (Abbildung 3).

Das Dashboard fasst alle Ergebnisse übersichtlich mit Diagrammen und Treemaps zusammen (Abb. 3).

Qualität des Codes durch Analysen steuern

Bei DevOps und Continuous Integration wird nicht nur dynamisch getestet, sondern der Code schon vor dem Build der Software ausführlich statischen Analysen unterzogen. Teamscale beherrscht die statische Analyse, indem es Werkzeuge wie beispielsweise CLang-Tidy (C, C++, Objective-C), ESLint (JavaScript), FindBugs (Java), StyleCop (C#), Pylint (Python) oder SAP Code Inspector (ABAP) einbindet. Auch kennt es eine Reihe eingebauter statischer Analysen wie unbenutzten Code, Verschachtelungstiefe und Clone Detection.

Insbesondere Clone Detection ist eine wertvolle Hilfe, ähnliche Codeteile aufzuspüren, die meist auf Copy-Paste-Verstöße gegen Wiederverwendbarkeit (DRY – Don’t repeat yourself) hinweisen. Ebenso lassen sich eigene Regeln für die statische Analyse hinzufügen. Sowohl beim ersten Hinzufügen eines Repositorys zu den Teamscale-Schnittstellen als auch bei jeder Änderung im Repository durchläuft das Tool automatisch die statischen Analysen.

Teamscale liefert eine umfassende Übersicht über die Befunde der statischen Analyse, die man durch geeignete Diagramme im Dashboard darstellen kann. Entsprechend konfiguriert, sammelt Teamscale weitestgehend automatisch Daten über sämtliche Prüfaktivitäten im Softwarelebenszyklus und zeigt sie übersichtlich auf Dashboards an.

Damit offeriert es umfassendes Fast Feedback bei der iterativen Entwicklung. Die hohe Flexibilität von Teamscale scheint dabei dem Umfang und der Art der erfassten Messungen kaum Grenzen zu setzen. Insofern ist es als Single Point of Truth bei kontinuierlicher Entwicklung einsetzbar und liefert alle für das Bewerten der aktuellen Softwarequalität nötigen Berichte.

Architekturanalyse deckt Abhängigkeiten auf

Zusätzlich zu den Funktionen, die Codequalität und Testdurchführung steuern, bietet Teamscale noch ein weiteres ungewöhnliches Feature: Anwender können die Systemarchitektur spezifizieren, das heißt, das Gesamtsystem in seine Teilsysteme zerlegen und deren Beziehungen zueinander modellieren. Über einen mit dem Coderepository verbundenen UML-Editor sind die Teilsysteme in der Architekturmodellierung direkt aus dem existierenden Code auswählbar.

Softwarearchitekten können beim Architekturentwurf schrittweise vorgehen. Das ist besonders dann vorteilhaft,

wenn noch keine vollständige Architektur des Systems besteht, sondern die Architektur aus historischen Gründen erst im Nachhinein durch Reverse Engineering des bestehenden Systems spezifiziert oder im Rahmen einer iterativen Entwicklung angepasst wird.

Die Zerlegung in Teilsysteme erlaubt, die Messungen auf die jeweiligen Teile zu beschränken und dadurch eine feingranulare Qualitätsanalyse zu erhalten, die nur die Teile betrachtet, die von Belang sind. Teamscale stellt durch eine statische Architekturanalyse mit dem Abgleich gegen das Coderepository auch Verstöße gegen die Architektur bei Systemänderungen fest – etwa Abhängigkeiten zwischen Teilsystemen, die im Architekturentwurf nicht vorgesehen sind.

Bewertungsgrundlage: Cloud-Version, Linux und Docker

Für den Artikel kam die Cloud-Testversion von Teamscale zum Einsatz – den Zugriff erhält man durch eine einfache Anfrage über die Webseite. Teamscale ist als native Installation für Windows, Linux, macOS und auch als Docker-Image verfügbar. Da es eine webbasierte Client-Server-Lösung ist, kann man Teamscale ebenfalls als Cloud-Lösung erhalten.

Die Tests sind mit der Cloud-Version durchgeführt. Hierzu wurden zwei Standardbeispiele von CQSE verwendet, aber auch eigene eingebunden. Um die Installation auszuprobieren, haben wir Teamscale nativ unter Linux und mit Docker unter Linux und Windows installiert.

Teamscale bietet viele Features für die Qualitätssteuerung. Um sie zu nutzen, muss man es in viele Bereiche der Softwareentwicklung einbinden. Durch die vielen möglichen Diagnosen, notwendigen Anbindungen an die Entwicklungs- und Testlandschaft und eingebundenen externen Werkzeuge (statische Analyse, Versionskontrolle, Test) ist die Konfiguration von Teamscale eine komplexe Aufgabe, die Anwender nicht

unterschätzen sollten.

Auch wenn sich die Testversion in Eigenregie installieren lässt, sollten Anwender auf den angebotenen CQSE-Support zurückgreifen – nur dann können sie das volle Potenzial von Teamscale in einem Pilotversuch nutzen.

Bedienung überwiegend intuitiv

Die User Experience ist gut: Hat man sich erst einmal an die Teamscale-Logik gewöhnt, findet man wertvolle Informationen für Entwicklung und Test – aber es dauert mitunter, die Logik hinter den Funktionen zu erfassen. Im Vergleich mit ähnlich komplexen Werkzeugen ist die Bedienung dennoch einfach und (meistens) intuitiv.

Der Wert von Teamscale für den einzelnen Kunden hängt davon ab, wie gut es in die eigene Entwicklungs- und Testlandschaft eingebunden ist. Je besser die eigenen Prozesse und die eigene Werkzeuglandschaft strukturiert sind, desto einfacher ist die Integration.

Ein wichtiger Punkt ist, ob Teamscale die vom Nutzer verwendeten Techniken berücksichtigt. Es kommt schon jetzt mit einer umfangreichen Menge daher: 29 Programmiersprachen, 18 externe Werkzeuge für die statische Analyse, alle gängigen Versionskontrollsysteme; zudem noch die Integration in IDEs und die Implementierung von Fehlermanagementsystemen (Abbildung 4).

Teamscale beherrscht viele Programmiersprachen und bindet achtzehn Werkzeuge für die statische Analyse ein (Abb. 4). Auch auf den Webseiten und im Gespräch mit dem Support weist CQSE darauf hin, dass es neue Sprachen und Werkzeuge gerne integriert. Nicht die Integration zusätzlicher Technologie ist das Problem, sondern dass Anwender den Überblick darüber bekommen.

Welche Voraussetzungen erforderlich sind

Um Teamscale einzusetzen, braucht es ein Softwareprojekt mit Zugriff auf den Quellcode. Zudem muss Teamscale die verwendete Programmiersprache und die eingesetzten Tools berücksichtigen. Auf Kundenwunsch passt CQSE es an weitere Techniken an. Auch eine IDE-Integration zum Anzeigen der erhobenen Informationen ist möglich. Arbeiten im Projekt mehrere Teammitglieder zusammen, ist es wichtig, ein gemeinsames Verständnis von Qualität zu schaffen. Dazu zählt es, Coding Guidelines für die Codestruktur (Codezeilen pro Datei, Methodenlänge) festzulegen, Redundanzen zu vermeiden, Code zu formatieren, zu dokumentieren und Tests zu schreiben. Außerdem betrifft es den Prozess der Softwareentwicklung: ein Versionsverwaltungssystem mit einem Branching-Konzept verwenden oder eine CI-Pipeline einsetzen, um Schritte im Entwicklungsprozess und die Ausführung von Tests zu automatisieren.

Teamscale ist auch ohne ein Versionsverwaltungssystem

verwendbar. Dazu laden Anwender in bestimmten Abständen den aktuellen Stand des Codes hoch, Teamscale bildet dann das Delta zur vorherigen Version.

Die Installation von Teamscale allein verbessert allerdings noch nicht die Qualität des Quellcodes oder verringert die Anzahl der Fehler. Vielmehr gilt es, die gesammelten Informationen auch zu nutzen. Diese Aufgabe sollten Quality Engineers übernehmen. Sie überwachen, dass die Entwicklungsteams die definierten Guidelines und Prozesse einhalten, kümmern sich um die Konfiguration und Lauffähigkeit von Teamscale und schauen, dass das Architekturdiagramm auf dem neuesten Stand ist. Außerdem liefern sie Informationen zum Qualitätstrend und besonderen Findings und stellen sicher, dass das Team offene Punkte im Backlog anlegt.

Für den Start empfiehlt es sich, kleine und erreichbare Ziele zu definieren. Vermutlich führt Teamscale anfangs viele Findings auf. Hier darf man nicht den Überblick verlieren, sondern muss sich im ersten Schritt auf ein, zwei Ziele konzentrieren. Das kann beispielsweise sein, neu auftretende Probleme im gerade angepassten Code zu beseitigen.

Setzen Anwender ein Versionsverwaltungssystem ein, können Pull Requests sinnvoll sein. Bevor ein Merge mit dem Hauptzweig stattfinden kann, sind bestimmte Quality Gates zu erreichen: zum einen, dass Teamscale keine neuen Findings listet (der Code muss den geltenden Qualitätsstandards entsprechen), und zum anderen, dass mindestens ein anderer Entwickler den Review durchführt.

Wann ist es sinnvoll, Teamscale einzusetzen?

Mehrwert kann Teamscale in jedem Softwareprojekt bieten, auch wenn es von nur einem Entwickler betreut wird, da die Analysen eine große Arbeitserleichterung sind und helfen, die Softwarequalität zu steigern. Je größer das Entwicklerteam

ist, desto wichtiger ist es sicherzustellen, dass jeder die gleichen Richtlinien einhält und somit eine einheitliche Basis gleicher Qualität geschaffen wird. Teamscale fungiert hier als zentrale Anlaufstelle, um einen schnellen Überblick über die derzeitige Codequalität zu bekommen. Entwickler erhalten dadurch direkt Feedback zu ihrem neu eingecheckten Code und darüber, an welcher Stelle sie gegebenenfalls nachbessern müssen.

Darüber hinaus kann es sinnvoll sein, Teamscale einzusetzen, wenn man Software mit einer großen Codebasis neu entwickeln oder die Struktur von bestehendem Code verbessern will. Vor allem, wenn Dritte den Code geschrieben haben, hilft es schnell dabei, Probleme und Schwachstellen im Code aufzudecken: Duplikate, Verstöße gegen die Coding Conventions, zu lange Methoden und Verletzungen der Softwarearchitektur fallen hierunter. Auch herauszufinden, welche Teile der Code beim Nutzen einer Funktion durchläuft, ist beim Refactoring behilflich und beschleunigt den Prozess.

Nützlich ist dabei die Code-Coverage-Analyse, mit der sich die Codeausführung im produktiven Betrieb auswerten lässt. Entwickler können ermitteln, welche Komponenten die Anwender des Programms wie oft benutzen, um ihnen eventuell eine höhere Bearbeitungspriorität zuzuweisen. Ein Fehlverhalten einer dieser Komponenten würde schnell mehrere Nutzer der Software betreffen, was ein hohes Risiko ist. Zudem kann eine solche Analyse aufzeigen, welche Teile des Codes die Anwender nicht mehr benötigen. Bei einem Refactoring können Entwickler diese Teile des Codes löschen, was in Folge die Codequalität steigert.

Fazit

Teamscale senkt durch sein Fast Feedback die Entwicklungs- und Testlaufzeiten. Es fördert kleinteiliges Arbeiten, was das Aufspüren und Lokalisieren von Fehlern erleichtert. Außerdem erlaubt es Anwendern schnell zu arbeiten, weil es umfangreiche

Prüfungen der erreichten Überdeckung automatisiert liefert. Die Paretooptimierung bringt weitere Informationen, um Testlaufzeiten zu verbessern: Wichtige Testfälle sind damit schnell und effizient auswählbar.

Als komplexes Werkzeug besitzt Teamscale Features, die ähnlichen Werkzeugen fehlen. Das hat seinen Preis: Derzeit kostet es 550 Euro im Monat für bis zu fünf Benutzer. Dafür erhält man auch Support. Wie bei solchen Werkzeugen üblich, kann man es mit einer Testversion ausprobieren, für die es ebenfalls Support gibt. Für Lehreinrichtungen und Open-Source-Projekte ist der Einsatz von Teamscale kostenlos.

Teamscale ist ein Tool, das Testspezialisten kennen sollten. Es empfiehlt sich gerade für größere Teams, die mit kurzen Releasezyklen arbeiten. Prinzipiell sind die erstellten Qualitätsmessungen und Berichte für jede Art von Entwicklungs- und Testteam von großem Nutzen. (nb@ix.de)

Wertung

schnelles Feedback bei iterativer Entwicklung und häufigen Änderungen Dashboard an eigene Bedürfnisse anpassbar, bietet schnelle Übersicht über den aktuellen Qualitätsstand sehr großer Funktionsumfang, der stetig erweitert wird Unterstützung vieler Programmiersprachen und Technologien sehr hohe Konfigurierbarkeit, lässt sich an die eigene Systemlandschaft anpassen leicht erweiterbar durch eine gut dokumentierte REST-API

- sehr guter Support durch den Anbieter

UI nicht immer selbsterklärend, könnte an manchen Stellen benutzerfreundlicher sein anfangs hohe Einstiegshürde, bis alle Systeme korrekt angebunden sind noch kein Visual-Studio-Code-Plug-in (ist geplant), Funktion ist aber über das CLI nutzbar

- preislich im Mittelfeld, für nicht umsatzstarke Softwareprojekte vermutlich zu teuer

Daten und Preise

Hersteller: CQSE

Produktname: Teamscale

Preise: 550 Euro/Monat (fünf Benutzer); kostenlos für Open-Source-Projekte und Bildungseinrichtungen

Download: cqse.eu/de/teamscale/download/

1. Quellen
2. Dokumentation, Download, Videos: ix.de/zm29

Dr. Carsten Weise

arbeitet als Trainer für ISTQB-Zertifizierungskurse und Testautomatisierung bei der imbus AG.

Christoph Singer

ist Berater mit Schwerpunkt Testautomatisierung bei der imbus AG.

KI-Recht: ChatGPT, Bard und Co.

Über Risiken beim Einsatz von KI wird nicht erst seit ChatGPT diskutiert. Der geplante EU AI Act wird jedoch nicht alle Aspekte regeln.

Von Tobias Haar

-tract

- Die KI-Regulierung der EU soll noch 2023 kommen. Sie soll insbesondere Grundrechtsverletzungen bei Betroffenen verhindern, deren Daten unter KI-Einsatz verarbeitet werden.
- Der zugrunde liegende Ansatz ist risikobasiert und sieht für die verschiedenen Risikostufen daran angepasste Maßnahmen und Pflichten vor.
- Auch eine EU-Richtlinie für KI-Haftung ist in Vorbereitung. Sie soll Geschädigten die Beweislast erleichtern und sieht schon dann einen Anspruch auf Schadenersatz vor, wenn ein ursächlicher Zusammenhang des Schadens mit der KI nach vernünftigem Ermessen wahrscheinlich ist.

„Wir können also nicht wissen, ob uns die künstliche Intelligenz unendlich helfen wird, ob sie uns ignoriert und beiseiteschiebt oder ob sie uns möglicherweise zerstört.“ So äußerte sich bereits 2017 der berühmte Physiker Stephen Hawking. Der jüngste Hype um den Einsatz künstlicher Intelligenz als Chatbots in Suchmaschinen, namentlich ChatGPT in Microsofts Bing oder Bard bei Google, wirft ein Schlaglicht

auf die aktuellen Entwicklungen im Bereich der KI. Zwar befindet sich die juristische Betrachtung dieser Phänomene noch am Anfang, erste Diskussionen werden aber bereits vehement geführt. Chatbots sind ein anschauliches Beispiel für den KI-Einsatz, um sich einigen wichtigen KI-Rechtsfragen zu nähern.

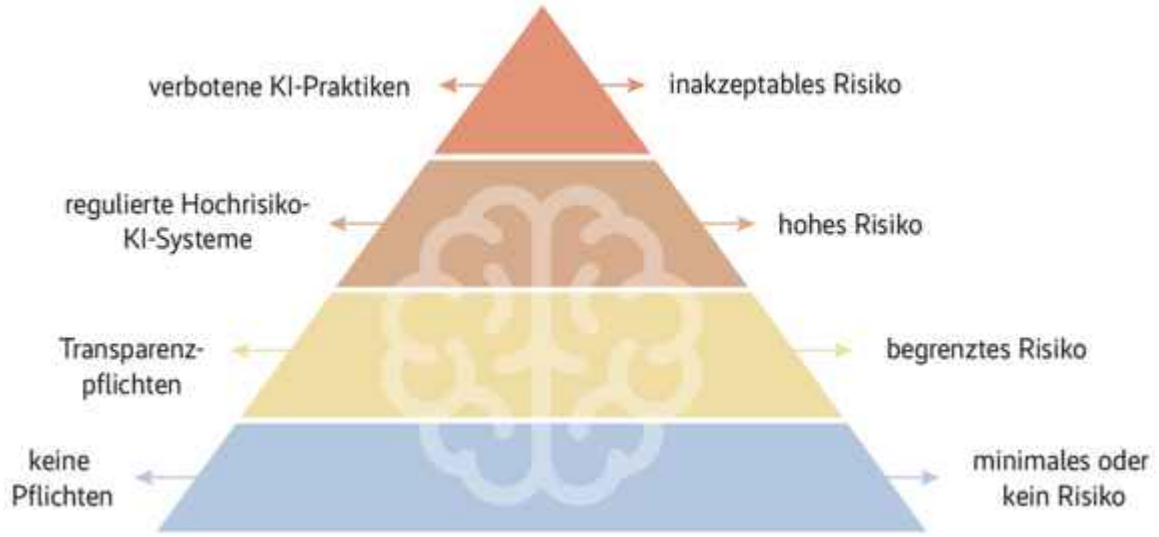
EU-Regulierung in den Startlöchern

Auf EU-Ebene wird derzeit am „Artificial Intelligence Act“, kurz AI Act, gearbeitet. Die Verordnung steht bereits kurz vor der abschließenden Abstimmung zwischen EU-Kommission, EU-Rat und EU-Parlament. Wie auch die Datenschutz-Grundverordnung wird sie nach Wirksamwerden in allen EU-Staaten unmittelbar gelten, ohne dass es nationaler Umsetzungen bedarf. Mit einem Inkrafttreten ist noch 2023 zu rechnen, da den EU-Institutionen aufgrund der für 2024 anstehenden Europawahlen nicht mehr viel Zeit bleibt, ihre Agenda umzusetzen. Gelingt dies nicht, droht eine erhebliche Verzögerung der einheitlichen Regulierung von KI in der EU.

Die EU beschreibt das Spannungsfeld, in dem der AI Act regulierend eingreifen soll, wie folgt: „Der Ansatz der EU für künstliche Intelligenz konzentriert sich auf Exzellenz und Vertrauen, um die Forschungs- und Industriekapazitäten zu stärken und die Grundrechte zu gewährleisten.“ Seit zwei Jahren wird am Verordnungstext gefeilt. Über 3000 Änderungsanträge wurden eingereicht. Bereits die Definition von KI ist umstritten. Ein aktueller Vorschlag lautet, den Anwendungsbereich auf Systeme zu beschränken, „die durch maschinelle Lerntechniken und wissensbasierte Ansätze entwickelt wurden“.

Der AI Act wird einen risikobasierten Ansatz verfolgen. Das soll gewährleisten, „dass die auf dem EU-Markt in Verkehr gebrachten und in der Union verwendeten Systeme künstlicher Intelligenz (KI) sicher sind und die bestehenden Grundrechte und die Werte der Union wahren“. Verboten werden soll der KI-

Einsatz bei der sozialen Bewertung von Personen. Für Hochrisikosysteme sind unter anderem Vorgaben im Bereich Datenqualität und Dokumentation vorgesehen. Daneben sind Transparenzvorgaben und weitere Schutzmaßnahmen zugunsten der Betroffenen geplant. Wie üblich sind schmerzhafte Bußgelder und behördliche Maßnahmen bei Nichteinhaltung vorgesehen.



Der risikobasierte Ansatz der KI-Regulierung soll verhindern, dass Grundrechte Betroffener und Werte der Europäischen Union verletzt werden. *Europäisches Parlament*

Mitten in den Diskussionen um den AI Act hat das Bundesverfassungsgericht Stellung zu einem der umstrittenen Themen genommen. Es geht darum, wie die Rechtsgrundlagen für die Datenauswertung mit KI durch die Polizei zur Gefahrenabwehr ausgestaltet sein müssen. Grundsätzlich ist eine automatisierte Datenauswertung nach Auffassung der Karlsruher Richter zulässig. Sie störten sich aber daran, dass die Polizeigesetze in Hamburg und Hessen nicht regeln, wann früher erhobene Daten für einen neuen Zweck ausgewertet werden dürfen.

Missbrauch von Datenverknüpfungen vermeiden

Insbesondere bei Daten, die aus einer Onlinedurchsuchung oder einer Wohnraumüberwachung stammen, müssen die Grundrechte der Betroffenen ausreichend gewahrt werden. Schafft KI durch die

erstmalige Verknüpfung von Informationen gänzlich neues Wissen, gilt es, das allgemeine Persönlichkeitsrecht besonders zu berücksichtigen. Unter Fachleuten wird für die neu zu fassenden Regelungen in den Polizeigesetzen ähnlich dem AI-Act-Ansatz ein abgestufter Ansatz diskutiert. Je nach abzuwehrender Gefahr dürften danach bestimmte Analyseinstrumente zum Einsatz kommen. Zu berücksichtigen wäre auch die zuvor angewandte Methode zur Erhebung der Daten.

Immer wenn von Daten die Rede ist, ist auch die Datenschutz-Grundverordnung (DSGVO) nicht weit. Sie gilt dann, wenn ein Chatbot personenbezogene Daten verarbeitet. Sie greift nur dann nicht, wenn es sich um nicht personenbezogene Daten handelt, also bei anonymen oder anonymisierten Daten. Wenn beim KI-Einsatz nur solche Daten verwendet werden, ist die juristische Compliance deutlich einfacher (siehe [1]).

Jede Form der Datenverarbeitung muss zulässig sein. Das ist sie, wenn die DSGVO oder eine informierte Einwilligung des Betroffenen dies erlauben. Wie in allen anderen Fällen kann der Betroffene bei KI-Einsatz Auskunfts-, Berichtigungs- und Löschungsrechte geltend machen. Wenn Unternehmen wie das hinter ChatGPT stehende OpenAI in Europa aber keine Niederlassung haben, wird die Rechtsdurchsetzung schwierig.

Von Bedeutung beim KI-Einsatz ist Artikel 22 der DSGVO. Er verbietet es, rechtlich erhebliche oder beeinträchtigende Entscheidungen über Menschen zu treffen, die ausschließlich einer automatisierten Datenverarbeitung entstammen. Ausdrücklich genannt wird in diesem Zusammenhang das Profiling. Ganz allgemein bedarf es einer DSGVO-konformen Datenschutzfolgenabschätzung, wenn risikobehaftete Datenverarbeitungen erfolgen. Dies ist insbesondere bei Gesundheitsdaten der Fall.

Bei Regelverletzungen droht Verbot

Jüngst hat die italienische Datenschutzaufsicht den Chatbot

Replika verboten. Zur Begründung verwies sie auf die Verletzung von Transparenzvorschriften nach DSGVO und die unzulässige Verarbeitung von Daten Minderjähriger. Brisant an dem Chatbot ist, dass dieser auf Empathie aufbaut und beworben wird als „der KI-Begleiter, der sich kümmert“. Angeblich soll die App zudem Daten im Auftrag russischer Behörden sammeln. Künftig müssen sich solche KI-Anwendungen auch am AI Act messen lassen.

Wer aber muss letztlich das Datenschutzrecht beim Einsatz von KI einhalten? Verantwortliche Stelle aus DSGVO-Sicht für Datenverarbeitungen ist, wer „allein oder gemeinsam mit anderen über die Zwecke und Mittel der Verarbeitung von personenbezogenen Daten entscheidet“. Im Fall von ChatGPT wäre dies OpenAI, im Falle der Einbindung von Bard in Microsoft-Office-Applikationen wäre es Microsoft und so weiter. Wenn allerdings der Nutzer selbst KI-Systeme mit personenbezogenen Daten füttert, kommt auch er als Verantwortlicher in Betracht. Womöglich sind KI-Anbieter und -Nutzer auch als gemeinsam Verantwortliche in der Pflicht. Es kommt auf den Einzelfall an.

Und wer zahlt, wenn durch KI-Einsatz ein Schaden entstanden ist? Wenn einer Person ein Schaden entstanden ist und ein Vertrag zwischen Nutzer und Anbieter besteht, ergibt sich ein etwaiger Schadenersatzanspruch aus dem Vertrag. Nach hiesiger Rechtsordnung greifen ergänzend zu einem Vertrag die schuldrechtlichen Vorschriften im Bürgerlichen Gesetzbuch.

Erst 2022 hat der Gesetzgeber im BGB den Mängelbegriff im Kaufrecht und zahlreiche weitere Vorschriften ergänzt – unter anderem, um die zunehmende digitale Bereitstellung von Waren, aber auch rein digitale Dienstleistungen juristisch besser zu berücksichtigen. Auch auf mangelhafte KI-Leistungen sind diese Vorschriften anwendbar.

KI-Chatbots und das Urheberrecht

KI-gestützte Chatbots wie ChatGPT geben Texte aus. Aus juristischer Sicht ist das relevante Rechtsgebiet in diesem Zusammenhang das Urheberrecht. In dessen Zentrum steht das Grundverständnis, dass nur kreativ-schöpferische Leistungen eines Menschen urheberrechtlichen Schutz genießen können. Damit scheidet die KI selbst als urheberrechtsschöpfend von vornherein aus. Sie ist nur ein handwerkliches oder technisches Werkzeug eines dahinterstehenden Menschen. Ihre rein technische Leistung spielt aus urheberrechtlicher Sicht keine Rolle.

Auch der KI-Entwickler und -Betreiber hat an einem konkreten Ergebnis eines durch Nutzer eingesetzten KI-Chatbots keinen eigenen schöpferisch-kreativen Anteil. Er kann sich ebenfalls nicht auf das Urheberrecht stützen. Denkbar wäre allenfalls, dass der Gesetzgeber ein Leistungsschutzrecht für KI-generierte Inhalte schafft. Durch den Hype rund um ChatGPT und Co. könnte die bisher eher schwache Diskussion hierüber an Fahrt gewinnen. Ein Ergebnis ist derzeit nicht absehbar.

Ein KI-Entwickler kann sich allerdings auf den Urheberrechtsschutz für Computerprogramme oder das gesetzliche Leistungsschutzrecht für Datenbankhersteller berufen. Sie schützen ihn vor einer unerlaubten Übernahme der konkreten KI-Umsetzung. Daher kann er Geld für die Nutzung „seiner KI“ verlangen und ein entsprechendes Geschäftsmodell aufsetzen. OpenAI probiert dies derzeit mit ChatGPT aus. Eine Variante von ChatGPT wurde auch in die Microsoft-Suchmaschine Bing eingebunden, wofür Microsoft mehrere Milliarden US-Dollar gezahlt hat.

Wem gehören KI-generierte Inhalte?

Vorsicht ist aus urheberrechtlicher Sicht bei der Nutzung KI-generierter Inhalte geboten. Soweit diese auf urheberrechtlich geschützten Vorlagen beruhen und es sich nicht um gemeinfreie

oder etwa auf Basis von Open-Source-Lizenzen für solche Zwecke nutzbare Werke handelt, kann man bei deren Nutzung durch etwaige Urheber in Anspruch genommen werden. OpenAI verweist darauf, dass ChatGPT ausschließlich Werke verwendet, bei denen keine Rechte Dritter entgegenstehen. Überprüfbar ist das im Einzelfall nicht. Den Nutzer eines urheberrechtsverletzenden Chatbot-Ergebnisses schützt seine Unwissenheit im Ernstfall nicht. Im Urheberrecht existiert kein Schutz des guten Glaubens.

Sind urheberrechtliche Inhalte nicht gemeinfrei oder frei nutzbar, stellt sich die Frage nach anderen Rechtfertigungen zur Verwendung von Werken im Rahmen des Text und Data Mining für KI-Zwecke. § 44 b des Urhebervertragsgesetzes (UrhG) versteht hierunter „die automatisierte Analyse von einzelnen oder mehreren digitalen oder digitalisierten Werken, um daraus Informationen über Muster, Trends und Korrelationen zu gewinnen“.

Eine Schranke des Urheberrechts enthält § 60 d des Urhebervertragsgesetzes für den Bereich des Text und Data Mining, sie erlaubt die automatisierte Auswertung – allerdings nur für die nicht kommerzielle wissenschaftliche Forschung. Dabei darf es auch keine beeinflussende Zusammenarbeit zwischen dem Forschungsinstitut und einem privaten Unternehmen geben.

§ 44 b UrhG besagt: „Zulässig sind Vervielfältigungen von rechtmäßig zugänglichen Werken für das Text und Data Mining.“ Dieser vermeintliche Freibrief gilt nicht unbeschränkt. Werke dürfen danach nur dann für Data Mining genutzt werden, „wenn der Rechtsinhaber sich diese nicht vorbehalten hat“. Bei online zugänglichen Werken muss ein solcher Nutzungsvorbehalt in maschinenlesbarer Form vorliegen.

Festlegen, was ein Webcrawler darf

Hier kommt der Robot Exclusion Standard ins Spiel. Ist im Stammverzeichnis einer Domain eine robot.txt-Datei hinterlegt, sollten Webcrawler diese zunächst auslesen. Der Domain-Inhaber kann darin festlegen, welche Suchmaschinen welche Inhalte unter einer Domain auslesen dürfen. Dieser Standard ist rechtlich möglicherweise als „Stand der Technik“ bindend. Er wird von den meisten Suchmaschinenbetreibern beachtet. Derzeit wird eine ähnliche Lösung mit dem Attribut „No AI“ diskutiert, um Webseiteninhalte vor der Nutzung durch KI-Systeme zu schützen.

Der Vollständigkeit halber stellt sich außerdem die Frage, ob dem KI-Nutzer nicht ein Urheberrechtsschutz an seinen Anweisungen an die KI zustehen kann. Es kommt beim sogenannten Prompt Engineering auf den Einzelfall an. Einfache Aufgabenstellungen fallen sicher nicht darunter, komplexe und individuelle Anweisungen möglicherweise schon. Urheberrechtlicher Schutz kann schließlich entstehen, wenn ein KI-Inhalt durch einen Menschen in ausreichender Schöpfungshöhe bearbeitet wird. Dann entsteht an den Bearbeitungen Urheberrechtsschutz.

Zwischen dem KI-Anbieter und dem KI-Nutzer besteht in der Regel ein Vertrag. Dieser enthält Regelungen über das Rechtsverhältnis zwischen beiden und auch im Hinblick auf die von der KI erzeugten Ergebnisse. An diesen erhält der KI-Nutzer meist sämtliche Rechte zur weiteren Nutzung. Ob daran ein Urheberrecht entsteht oder nicht, kann der Vertrag aber

nicht regeln. Ein Urheberrecht entsteht kraft Gesetzes, nicht kraft eines Vertrages.

Wenn eine KI beispielsweise einen Artikel für die iX schreibt, hat niemand an dem Chatbot-generierten Text das Urheberrecht. Im Text können aber Werke Dritter verarbeitet worden sein, wodurch das Urheberrecht dieser Dritten tangiert wurde. Ist das nicht der Fall, ist der Text urheberrechtsfrei und kann von jedermann genutzt werden. Einem Abdruck in der iX steht nichts im Wege.

Auch KI-Haftung soll reguliert werden

Unterdessen arbeitet die EU an einer Richtlinie zur KI-Haftung. Als Richtlinie ist sie nach ihrer Verabschiedung in nationales Recht umzusetzen. Sie soll Geschädigten die Beweislast erleichtern. So soll eine Kausalitätsvermutung greifen und ein Anspruch auf Schadenersatz schon dann bestehen, wenn eine Person für eine bestimmte für den Schaden relevante Verpflichtung verantwortlich war und „ein ursächlicher Zusammenhang mit der KI-Leistung nach vernünftigem Ermessen wahrscheinlich ist“.

Bei Hochrisiko-KI-Systemen sollen Richter zudem die Offenlegung von Informationen über solche Systeme anordnen können. Damit will man die verantwortlichen Personen ermitteln. Andererseits unterliegen offengelegte Informationen dem Geheimnisschutz. Auswirkungen auf den Bereich KI könnte auch die gleichfalls auf EU-Ebene diskutierte neue Produkthaftungsrichtlinie haben. Sie bietet Verbrauchern bei fehlerhaften Produkten eine verschuldensunabhängige Schadenshaftung. Mit ihr soll die seit 1985 EU-weit geregelte Produkthaftung an neue Technologien, etwa KI, angepasst werden.

Auf die KI-Anbieter kommen mehr und mehr regulatorische Pflichten und Complianceanforderungen zu. Deshalb müssen sie sich um ihre Haftungsrisiken Gedanken machen und entsprechende

Vorkehrungen treffen. Das Trainieren von KI-Systemen auf Rechtskonformität ist eine der großen Herausforderungen beim Einsatz von KI. Dazu müssen aber zunächst einmal die gesetzlichen Vorgaben verfeinert werden. Sie werden aber auch nur den Rahmen festlegen. Die konkreten Ausgestaltungen hängen dann vom jeweiligen Einsatz von KI in der Praxis ab. Letztlich werden Gerichte darüber befinden, ob die Vorgaben in ausreichendem Maße eingehalten wurden oder nicht.

Auch KI-Nutzer müssen vor der Verwendung von KI-generierten Informationen juristische Aspekte beachten. Sie können sich im Bereich Datenschutz oder Urheberrecht in der Regel nicht auf ihre Unwissenheit berufen. Juristische Fallen entstehen, wenn Nutzer die Herkunft der durch KI verwendeten Datensätze nicht kennen und nicht kontrollieren können. Gleiches gilt für die eingesetzten Algorithmen. Zumindest bei KI-Anwendungen von Drittanbietern ist eine Kontrolle in den meisten Fällen nicht möglich.

Fazit

Künstliche Intelligenz fordert das Recht heraus. Je tiefer die KI in die verschiedenen Lebensbereiche eindringt, umso mehr müssen sich Gesetzgeber und Richter mit den daraus resultierenden Risiken befassen. Letztlich können davon zunächst fernliegende Rechtsbereiche betroffen sein, etwa das Schulrecht bei der Frage der Zulässigkeit KI-generierter Hausaufgaben. Es geht aber eben auch um komplexe Fragen im Haftungs-, Datenschutz- oder Urheberrecht (siehe Kasten „KI-Chatbots und das Urheberrecht“). Die EU plant mit dem AI Act einen risikobasierten Ansatz bei der KI-Regulierung bis hin zum Verbot einzelner KI-Ansätze. Auf Anbieter wie Nutzer kommen umfassende juristische Risiken zu. Das KI-Recht ist gekommen, um zu bleiben. (ur@ix.de)

ChatGPT im juristischen Einsatz

Künstliche Intelligenz ist nicht nur eine Herausforderung für

die Juristerei. Sie kann ihr auch bei der Arbeitsbewältigung helfen. Richter, Staatsanwälte und Rechtsanwälte können von sprachmodellbasierten KI-Lösungen profitieren, die beispielsweise Urteile analysieren und Entscheidungsmuster aufzeigen. Hilfreich kann es auch sein zu wissen, wie ein bestimmtes Gericht in einem ähnlich gelagerten Verfahren entschieden hat. Das wäre ein großer Schritt hin zu einem „Predictive Decision-Making“, einem planbaren Ausgang eines Rechtsstreits.

Macht Chat-GPT Anwälte obsolet? Chatbots können dabei helfen, rechtssuchenden Bürgern den Gang zum Anwalt zu ersparen. Auch bisher war es möglich, zu Rechtsfragen zu „googeln“. Zukünftig könnte eine individuelle Rechtsberatung durch KI-Systeme hinzukommen. Sie wäre womöglich billiger als das Hinzuziehen eines Rechtsanwalts. Allerdings ist die Juristerei extrem komplex und jeder Einzelfall ist individuell zu betrachten. Dabei spielt der verfassungsrechtlich bei nahezu allen Rechtsfragen zu berücksichtigende Verhältnismäßigkeitsgrundsatz eine bedeutende Rolle. Er könnte eine KI – noch – an ihre Grenzen führen.

KI im juristischen Einsatz

Bereits heute werden KI-Systeme bei der Vertragsprüfung eingesetzt. Sie können Muster erkennen und beispielsweise auf Regelungslücken hinweisen. Derzeit werden solche Systeme etwa zur Prüfung von Vertraulichkeitsvereinbarungen eingesetzt. Auch an komplexere Verträge wagen sich die ersten Lösungen heran.

Weitere Anwendungsfelder für KI im Rechtsbereich gibt es bei Themen wie Contract Lifecycle Management. Im Bereich der DSGVO-Compliance können Webseiten auf die Einhaltung datenschutzrechtlicher Vorgaben etwa hinsichtlich der Datenschutzerklärung überprüft werden.

Welche weiteren Anwendungsfälle in Zukunft erlaubt sein

werden, hängt von den regulatorischen Rahmenbedingungen ab. Denkbar sind KI-Systeme, die bei Gesetzesverstößen automatisch Sanktionen verhängen. Bei Parkzeitüberschreitungen ist das sicher eher akzeptabel als bei schwerwiegenden Straftaten. Hier dürfte Artikel 101 des Grundgesetzes eine Rolle spielen: „Niemand darf seinem gesetzlichen Richter entzogen werden.“ Dabei muss es sich (noch) um einen Menschen handeln.

1. Quellen
2. Tobias Haar; Nutzer unbekannt; Rechtsfragen der Pseudonymisierung und Anonymisierung; iX 5/2021, S. 86

Tobias Haar

ist Rechtsanwalt mit Schwerpunkt IT-Recht bei Vogel & Partner in Karlsruhe. Er hat zudem Rechtsinformatik studiert und hält einen MBA.

So programmiert es sich mit Sprach-KI

Sprach-KI soll Programmierern Routineaufgaben abnehmen und Codezeilen beim Schreiben sinnvoll ergänzen. Am Beispiel von GitHub Copilot lässt sich zeigen, wie sich diese Systeme einsetzen lassen und was dabei zu beachten ist. Denn noch darf man sich nicht blind auf die Hilfe verlassen.

Von Philipp Braunhart

-tract

- Der KI-Assistent Copilot von GitHub soll beim Entwickeln helfen und repetitive Standardaufgaben automatisieren.
- Dafür lässt sich der Assistent per Plug-in in gängige IDEs einbinden.
- Copilot berücksichtigt die aktuelle Eingabe und die umliegenden Zeilen, um Entwicklern passende Codevorschläge zu bieten.
- Der Einsatz klappt am besten, wenn man sich das Programm als einen zusätzlichen Entwickler vorstellt, der helfen soll und dementsprechend Informationen benötigt.

„Die neue, heißeste Programmiersprache ist Englisch“, das meinte Andrej Karpathy, der ehemalige Head of AI von Tesla, Anfang des Jahres auf Twitter. Dass er damit nicht ganz falschliegt, zeigt eine Vielzahl von aktuellen Programmierassistenten, die auf generativer künstlicher Intelligenz basieren. Einer davon ist GitHub Copilot, der im Kontext des aktuellen Inhalts meist sehr genau versteht, was gerade gewünscht ist, und sehr selten komplett danebenliegt. Hinter Copilot steckt das KI-Modell Codex von OpenAI, das natürliche Sprache und Code verstehen und generieren kann.

Technischer Hintergrund

GitHub Copilot basiert auf Codex, einer weitertrainierten Variante des Large Language Models GPT-3 von OpenAI. Das Modell verwendet die Transformerarchitektur. Durch mehrere Terabyte an Textdaten unter anderem aus den Archiven von Common Crawl und WebText2, Büchern und der englischen Wikipedia hat OpenAI das Modell auf das Verstehen von Sprache trainiert. Das Feintuning von Codex basiert auf Milliarden von Codezeilen von öffentlichen GitHub-Repositorien. GitHub Copilot filtert und bewertet die Vorschläge von Codex, bevor

es diese an die Nutzer sendet.

Copilot erzeugt sowohl Code als auch Kommentare, wobei das Programm den Kontext des zu erzeugenden Codes zur Vorhersage verwendet. Dies ist der Inhalt der Datei mit Fokus auf die aktuelle Zeile und die darüberliegenden. Die Vorschläge können innerhalb einer Zeile sein, eine komplette oder sogar mehrere Zeilen umfassen. Damit lässt sich das Programmieren stark beschleunigen.

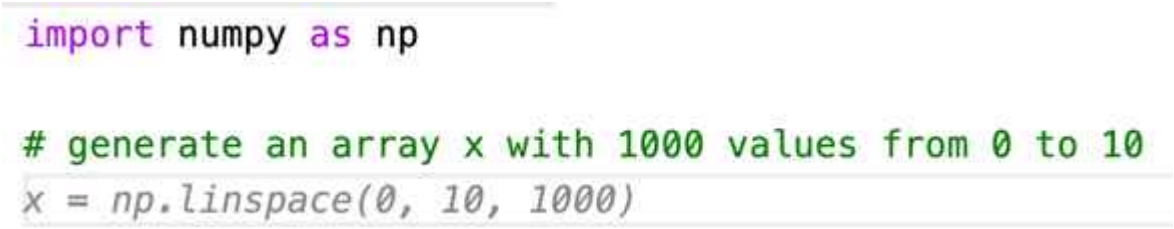
Die Integration in verschiedene IDEs erfolgt durch Plug-ins. Copilot unterstützt nahezu sämtliche Programmiersprachen, am besten sind die Ergebnisse für Python, JavaScript, TypeScript, Ruby, Go, C#, C++ und Java. Auch Sprachen wie SQL, HTML und YAML beherrscht Copilot. Dank dieser Vielfalt ist er nahezu für alle Entwickler eine Hilfe.

Benutzer können sich auf ihre Aufgabe konzentrieren, ohne zwischen der IDE und Webseiten zur Informationssuche wie Stack Overflow zu wechseln. Das kommt gut an: Während der Assistent nach dem Release 27 Prozent der Inhalte von mit ihm editierten Dateien generierte, kommen laut Plattformbetreiber nun 46 Prozent der Files von Copilot, im Fall von Java sogar 61 Prozent, Stand Februar 2023.

Fliegen mit Copilot

Copilot generiert Codevorschläge direkt während des Schreibens. Die Angebote der KI lassen sich per Klick oder Tastendruck annehmen oder ablehnen. Die Schaltflächen über dem Vorschlagsfeld bieten die Möglichkeit, mehrere Vorschläge zu durchsuchen oder sogar ein neues Fenster mit bis zu 10 Entwürfen zu öffnen (siehe Abbildung 1). Eine Alternative besteht darin, Kommentare zu schreiben, um gewünschten Code generieren zu lassen. Dann zeigt Copilot die Codevorschläge in der Zeile unter dem Kommentar an (siehe Abbildung 2).

Einmal in die IDE eingebunden, macht Copilot Vorschläge zum Vervollständigen von angefangenen Codezeilen, hier am Beispiel eines Arrays (Abb. 1).



Copilot lässt sich auch per Kommentar steuern. Hier folgt der KI-Assistent dem Befehl zum Erstellen eines Arrays mit 1000 Werten (Abb. 2).

Das System erstellt nicht nur einzelne Codezeilen: Aus detaillierten Kommentaren oder begonnenen Funktionsdefinitionen mit Input und Output generiert Copilot Codeblöcke. Auch Docstrings erstellt der KI-Assistent. Abbildung 3 zeigt, wie Copilot bereits bei jedem Schritt vom Schreiben der Kommentare bis hin zur Input- und Output-Definition hilft.

```
import pandas as pd
import numpy as np
import plotly.express as px

# generate a dataframe with x from 0 to 10
# and y1 sin, y2 cos
x = np.arange(0, 10, 0.01)
df = pd.DataFrame({
    'x': x,
    'y1': np.sin(x),
    'y2': np.cos(x),
})

# create a function to plot the above dataframe
# with ploty express

def plot(df):
    fig = px.line(df, x='x', y=['y1', 'y2'])
    fig.show()
```

Nichts mehr selbst schreiben: Nur über Kommentare hat Copilot hier den gesamten Code generiert. Dabei lassen sich auch eingebundene Bibliotheken ansteuern (Abb. 3).
Es ist jedoch wichtig zu beachten, dass dabei durchaus Fehler passieren oder das System suboptimalen Code produzieren kann. Daher ist es notwendig, die Vorschläge vor dem Annehmen immer sorgfältig zu überprüfen.

Auch die Qualität der Dokumentation im Code lässt sich so leicht verbessern: Einfach zusätzliche Kommentare und Dokumentation auf Basis des Codekontextes von Copilot generieren lassen. Dabei erkennt und imitiert der Assistent den persönlichen Stil im Code und in den Kommentaren.

Tipps für eine sichere Landung

Ein entscheidender Faktor für die optimale Nutzung von Copilot ist das Bereitstellen von ausreichenden Informationen, um relevante und präzise Vorschläge zu erhalten. Das ist vergleichbar mit einem anderen Programmierer, der auf entsprechende Informationen und Anleitung angewiesen ist. Die wichtigste Frage beim Schreiben von Kommentaren oder Docstrings ist also: Welche Informationen benötigt ein Programmierer, um die Anforderungen zu erfüllen? Je detaillierter die Angaben sind, desto besser kann Copilot weiterhelfen.

Ein weiterer wesentlicher Aspekt ist sauberer und konsistenter Code. Hierbei sollte man gute, klare und präzise Kommentare, Docstrings und aussagekräftige Namen für Funktionen, Klassen, Variablen und Argumente verwenden. Logik und die Intention des Codes sollten leicht zugänglich sein. Copilot kann dabei sogar unterstützen und generiert dadurch anschließend bessere Codevorschläge.

Soll Copilot Funktionen aus Bibliotheken vorschlagen, reicht es meist, Letztere am Anfang der Datei zu importieren. Zusätzlich hilft es, die Bibliothek in einem Kommentar dort zu erwähnen, wo man sie einsetzen will. Bei der Arbeit mit Datenbanken oder Dataframes ist es sinnvoll, das Schema, wie Spaltennamen, Typen und einige Beispiele, als Kommentare bereitzustellen. Dann versteht Copilot die verwendeten Objekte genauer und macht Vorschläge mit den passenden Namen (siehe Abbildung 4).

```
import pandas as pd
import matplotlib.pyplot as plt

# load data
# DataFrame has columns: ['id', 'name', 'age', 'gender', 'height', 'weight']
pd.read_csv(path)

# histogram
plt.hist(data['height'], bins=10)
```

Informiert man die Sprach-KI per Kommentar über die Namen von Spalten eines Dataframes, dann schlägt das Programm selbstständig mögliche Felder beim Plotten der Daten vor (Abb. 4).
Um Copilot effizient zu nutzen, empfiehlt es sich, Tastenkombinationen in der IDE einzurichten. Geeignete Funktionen für Hotkeys sind: Vorschläge annehmen, ablehnen, nur das nächste Wort des Vorschlags akzeptieren, zwischen verschiedenen Optionen navigieren oder ein separates Fenster mit Vorschlägen öffnen. Dadurch reduziert Copilot die Zeit für repetitive oder mühsame Aufgaben, wie das Schreiben von Unit-Tests, das Behandeln von Exceptions, das Loggen oder das Ausgeben von Informationen.

Tricks für Überflieger

Es gibt auch noch weitere spannende Aspekte an GitHub Copilot jenseits des Vervollständigens von Code. Einer dieser Aspekte ist das Konvertieren von Code in andere Programmiersprachen. Hierdurch lässt sich etwa Legacy-Code leichter portieren (siehe Abbildung 5). Darüber hinaus hilft Copilot auch beim Umwandeln von Code von einem Framework in ein anderes. Das beschleunigt das Konvertieren zwischen verschiedenen Frameworks (siehe Abbildung 6).

```
# Convert this from Python to R
# Python version

import numpy as np
import matplotlib.pyplot as plt

x = np.random.randn(1000)
plt.hist(x, bins=20)
plt.show()

# End

# R version

x <- rnorm(1000)      # <- generated by Copilot
hist(x, breaks=20)    # <- generated by Copilot
```

Will man seinen Code von einer Programmiersprache in eine andere übersetzen, kann eine Sprach-KI helfen. Das Beispiel zeigt das Erstellen und Plotten von 1000 Beispieldaten in Python und in R (Abb. 5).

```
# Rewrite this to use plotly express instead of matplotlib

import numpy as np
import matplotlib.pyplot as plt

x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)

plt.plot(x, y)
plt.show()

# plotly express version:

import numpy as np                      # <- generated by Copilot
import plotly.express as px             # <- generated by Copilot

x = np.linspace(0, 2 * np.pi, 100)      # <- generated by Copilot
y = np.sin(x)                           # <- generated by Copilot

fig = px.line(x=x, y=y)                 # <- generated by Copilot
fig.show()                              # <- generated by Copilot
```

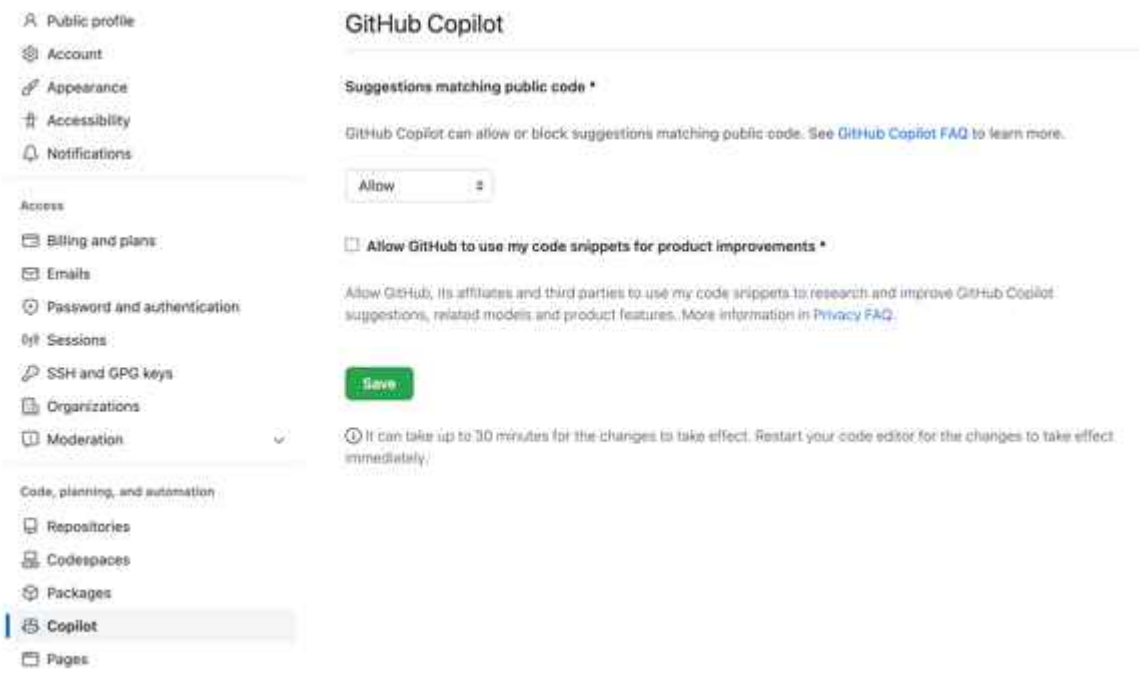
Ähnlich wie beim Übersetzen von Code in unterschiedliche Programmiersprachen dolmetscht Copilot auch zwischen verschiedenen Frameworks und wie hier verschiedenen Bibliotheken einer Sprache (Abb. 6).
Wenn es darum geht, neue Frameworks oder Bibliotheken zu lernen, ist Copilot eine große Hilfe. Vorschläge als Kommentar, basierend auf dem Wunsch des Users, sind ein guter Ausgangspunkt. Damit können Entwickler ihr Wissen schnell erweitern und erfolgreich anwenden.

Zuletzt kann man Copilot auch für das schnelle Generieren von Daten einsetzen. Etwa wenn Daten für Tests nötig sind, reicht die Angabe von ein paar Beispielen und Copilot generiert automatisch zahlreiche weitere Daten.

Turbulenzen

Trotz seines Nutzens weist das Tool jedoch auch Grenzen und potenzielle Risiken auf, die Anwender berücksichtigen sollten. So kann es geschehen, dass Copilot exakte Kopien von Code aus

öffentlichen Repositorys oder Quellen vorschlägt. Hierfür lässt sich eine Filteroption auf der Einstellungsseite aktivieren (siehe Abbildung 7). Dort sieht man auch, dass GitHub den Code von Anwendern standardmäßig für das Retrainieren der KI-Modelle hinter Copilot nutzt und somit potenziell mit anderen Nutzern teilt. Auch dies lässt sich in den Einstellungen deaktivieren.



Zwei besonders wichtige Optionen finden sich im Menü von Copilot. Einerseits lässt sich verhindern, dass der Assistent Code aus öffentlichen Repositorien exakt repliziert. Andererseits kann man per Checkbox verhindern, dass GitHub den gescannten und erzeugten Code zum Trainieren des Modells verwendet und man so möglicherweise Betriebsgeheimnisse verrät (Abb. 7).
Durch das Teilen von eigenem Code können Risiken für sensible Daten entstehen. GitHub versichert, dass die hohen Standards für Datenschutz und Datensicherheit auf ihrer Plattform eingehalten werden und der Code nur zum Erstellen von Vorschlägen genutzt wird. Noch mehr Kontrolle geben Self-Hosting-Alternativen für KI-Codeassistenten (siehe unten).

Generell ist der rechtliche Status der Trainingsdaten von Copilots KI-Modellen noch nicht komplett geklärt und Inhalt einiger aktuell laufender Klagen. OpenAI hat zahlreiche

GitHub-Repositorys, die unter Copyleft-Lizenzen wie der GPL stehen, für das Training verwendet. Copilots Modell weist nicht auf die Lizenzen des eingebauten Codes hin und ist selbst nicht Open Source. Hier bleibt die Rechtsprechung abzuwarten.

Das momentan limitierte Kontextfenster ist eine andere Einschränkung. Der Assistent sollte eigentlich lediglich den Inhalt der aktuell bearbeiteten Datei berücksichtigen. Teilweise entsteht bei der Arbeit allerdings der Eindruck, dass Copilot auch andere offene Dateien heranzieht. Allerdings verwendet das System nicht die gesamte Codebasis eines Projekts.

Unsicherer Code und Schwachstellen stellen ebenfalls eine Herausforderung dar. Eine aktuelle Forschungsarbeit deutet darauf hin, dass Nutzer von Tools wie Copilot dazu tendieren, Code mit Schwachstellen leichter zu akzeptieren, sich aber gleichzeitig in Sicherheit wiegen (siehe ix.de/zxwq). Mit einem Copilot-Update im Februar 2023 führte GitHub ein Feature ein, das potenzielle Schwachstellen in den Vorschlägen scannt und automatisch unterdrückt. Das System filtert dabei typische Schwachstellen wie hartcodierte Credentials, SQL- oder Pfadinjektionen.

Einige Probleme sind damit sicherlich behoben, aber natürlich findet das Programm dadurch nicht alle Schwachstellen. Es bedarf also weiterhin großer Aufmerksamkeit, gerade bei sicherheitsrelevanten Anwendungen. Zuletzt sei erwähnt, dass Copilot am besten auf Englisch arbeitet. Arbeiten Anwender mit anderen Sprachen, kann das zu Einschränkungen führen.

Alternative Reisemöglichkeiten

Zu Copilot gibt es diverse Alternativen, darunter sowohl kommerzielle als auch Open Source. AWS bietet beispielsweise Code Whisperer an, Salesforce mit CodeGen ein quelloffenes Projekt, das sich mit Fauxpilot selbst hosten lässt.

Weitere Optionen sind Tabnine und Replit Ghostwriter. Für JupyterLab, Colab-Notebooks oder BigQuery ist zudem CodeSquire als Browsererweiterung verfügbar. Der Markt entwickelt sich stetig und so kommen und gehen die Alternativen. Unter allen Optionen scheint GitHub Copilot derzeit am ausgereiftesten und zuverlässigsten zu sein.

Was klar ist: Copilot ist mehr als nur ein einfaches Autocomplete-Werkzeug. Es ist der Anfang eines größeren Wandels im Programmieren, der darauf abzielt, mehr Aspekte des Entwicklungsprozesses zu automatisieren.

Die Fähigkeit, Code aus Kommentaren und Beispielen innerhalb der eigenen IDE zu generieren, reduziert Ablenkungen, beschleunigt das Programmieren und verbessert die Qualität der Dokumentation. Für den richtigen Einsatz und gute Vorschläge sollten Entwickler im Kopf behalten, dass das System die gleichen Informationen benötigt wie ein Kollege, den man um Rat bittet. Forscher bei Google unterstreichen die Vorteile von Programmierassistenten. In einer Untersuchung mit der Inhouse-KI will man herausgefunden haben, dass sich die Produktivität mithilfe des Tools messbar erhöhen lässt. (pst@ix.de)

1. Quellen
2. Weitere Informationen zu Copilot und dem Programmieren mit KI-Assistenten finden sich unter ix.de/zxwq.

Philipp Braunhart

ist Entwickler und Berater im Bereich Machine Learning und Mitgründer der ingenio ai GmbH. Er programmiert, entwirft und

baut KI-Lösungen für den direkten Einsatz in den Bereichen Bio-, Agrar- und Medizintechnologie.

ChatGPT per API einbinden

Ein Programmbeispiel zeigt, wie man ohne Overhead und Umwege die Modelle hinter ChatGPT per OpenAI-API in eigenen C#-Anwendungen nutzt.

Von Daniel Basler

-tract

- Mit der OpenAI-API lässt sich das ChatGPT-Modell gpt-3.5-turbo in eigene Programme einbinden.
- Anstelle von Token aus unstrukturiertem Text verwendet ChatGPT eine Folge von Nachrichten zusammen mit Metadaten. OpenAI nennt das Format Chat Markup Language (ChatML).
- Mit einer in C# implementierten WPF-App können Interessierte erste Erfahrungen mit dem Einbinden der Sprach-KI in eigene Anwendungen sammeln.
- Durch Zugriff auf verschiedene Modellparameter bietet die API die Möglichkeit, den Umgang mit Sprachmodellen zu erlernen und zu verbessern.

Mit ChatGPT (OpenAI) und Bard beziehungsweise LaMDA von Google ist die künstliche Intelligenz (KI) massenkompatibel geworden. ChatGPT hat sich auf Grundlage von Transformertechnologie und dem Attention-Mechanismus mit Millionen von Textseiten vollgesogen und daraus ein großes Modell der menschlichen Sprache gebaut: ein Large Language Model (LLM).

Dadurch kann ChatGPT eine Vielzahl von Aufgaben der natürlichen Sprachverarbeitung ausführen, darunter das Generieren, Zusammenfassen und Klassifizieren von Text und Frage-Antwort-Systeme. Des Weiteren erstellt ChatGPT Programme und hilft beim Debuggen von Code. All diese Aufgaben beherrschen auch andere Modelle aus der GPT-Reihe von OpenAI in unterschiedlichem Ausmaß. ChatGPT selbst ist für den Dialog mit Nutzern optimiert, basiert aber ansonsten auf der Generation GTP 3.5 und damit den leistungsfähigsten und neuesten Modellen, die OpenAI derzeit im Angebot hat. OpenAI macht seine GPT-Modelle über eine einheitliche API (Application Programming Interface) verfügbar, über die sich mit diesen Modellen kommunizieren lässt. Damit ist es also möglich, eigene Programme mit den Fähigkeiten von ChatGPT auszustatten, seit Anfang März ist auch der Zugriff auf das ChatGPT-Modell über die API möglich. Unser Beispiel zeigt, wie sich die API und die Modelle von OpenAI mit C# ansprechen lassen. Über die API lassen sich die Modelle auch über unterschiedliche Parameter für verschiedene Zwecke optimieren.

Softwareautomatisierung mit generativer KI

Bei ChatGPT und den anderen GPT-Modellen von OpenAI handelt es sich um generative KI-Stacks. Hierbei kommen Deep- und Machine-Learning-Algorithmen sowie Generative Adversarial Networks (GAN) zum Einsatz, die aus vorhandenen Texten, Audiodateien oder Bildern Inhalte erstellen. Die Modelle Erlernen Muster in der Eingabe und verwenden sie, um ähnliche Inhalte zu erzeugen.

Transformermodelle wie GPT oder LaMDA simulieren kognitive Prozesse und versuchen die Bedeutung der Eingabedaten auf unterschiedliche Weise zu messen. Man trainiert die Modelle darauf, die Sprache oder das Bild im Kontext zu verstehen, Klassifizierungen zu erlernen und Texte oder Bilder aus großen Datensätzen zu generieren.

Das verwendete Sprachmodell Generative Pre-trained Transformer, kurz GPT, kann Befehle in natürlicher Sprache auch in Programmcode übersetzen. GPT ist also in der Lage, Websites oder Codebeispiele über automatisches Codegenerieren zu erzeugen. Das aus GPT entstandene Modell Codex ist auf Codegenerierung optimiert. GitHubs Tool Copilot liegt das Codex-Modell zugrunde. Copilot analysiert Kommentare aus dem Code und schlägt einzelne Codezeilen oder Funktionen während des Entwickelns vor. Das am 1. März für die API veröffentlichte gpt-3.5-turbo-Modell, auf dem ChatGPT basiert, ist für die Ein- und Ausgabe von Konversationschats optimiert. Des Weiteren ist die Turbo-Modellfamilie auch die erste, die regelmäßige Modellupdates erhält.

Das Projekt

Das Programmierbeispiel zeigt, wie man mit der OpenAI-API auf GPT-3-Modelle zugreift. Dort hat man Zugriff auf gpt-3.5-turbo. Das Projekt Example-ChatGPTApplication ist überschaubar und greift per HTTP und JSON auf den API-Endpunkt zu. Zwar ist ein Zugriff auch über die Bibliothek .NET SDK for OpenAI GPT-3 möglich, die Library kommt jedoch mit einem viel größeren Funktionsumfang, den man für das Beispiel hier nicht benötigt. Das Programm kommt als Windows-Presentation-Foundation-App (WPF) mit einer einfachen Benutzeroberfläche für Anfragen, Modellauswahl und Einstellungen für Parameter daher. Man erstellt die Anwendung mit der Visual Studio Community Edition 2022 und dem .NET Framework 7.0. Um schlank zu bleiben, verwendet das Beispiel Code aus einer Code-Behind-Datei der Klasse MainWindow.xaml.cs.

Zum Verwenden der API benötigt man einen aktiven Account bei OpenAI, über den sich ein Authentifikationsschlüssel für die Verwendung der API erstellen lässt. OpenAI berechnet zurzeit 0,02 US-Dollar für 1000 Eingabetoken bei den bisherigen GPT-Modellen. Bei ChatGPT belaufen sich die Kosten auf 0,002 US-Dollar pro 1000 Token. Token stellen Teile von Wörtern dar,

wobei 1000 Token etwa 750 Wörtern entsprechen. Beim Anmelden erhält man in der Regel ein Startguthaben von etwa 18 US-Dollar, das sich in den ersten drei Monaten frei verwenden lässt. Ist es aufgebraucht, benötigt man einen neuen API-Key, den OpenAI in Rechnung stellt.

Die Oberfläche

Das Programm verwendet die Projektvorlage WPF Application mit der Projektbezeichnung ExampleChatGPTApplication und dem .NET Framework 7.0. Visual Studio erstellt aus der Vorlage automatisch eine bereits lauffähige WPF-Applikation. Listing 1 zeigt die Erweiterung der Oberfläche in der XAML-Datei MainWindows.xaml. Das GUI lässt sich nach eigenem Geschmack erstellen oder später anpassen.

Listing 1: Aufbau der Oberfläche in der MailWindow.xaml-Datei

```
<Window x:Class="ExampleChatGPTApplication.MainWindow"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:ExampleChatGPTApplication"
        mc:Ignorable="d"
        WindowStartupLocation="CenterScreen"
           Title="MainWindow" Height="500" Width="720"
Loaded="Window_Loaded">
    <Grid Margin="0,0,10,0">
          <TextBox Height="396" HorizontalAlignment="Left"
Margin="11,10,0,0"  Name="chatBox"  VerticalAlignment="Top"
Width="535"
                                        IsReadOnly="True"
TextWrapping="WrapWithOverflow"
VerticalScrollBarVisibility="Visible"/>
              <Label  Content="Message:"  Height="28"
HorizontalAlignment="Left" Margin="11,411,0,0" Name="label5"
```

```
VerticalAlignment="Top" />
          <TextBox Height="22" HorizontalAlignment="Left"
Margin="86,414,0,0" Name="messageText" VerticalAlignment="Top"
Width="460" />
                 <Button  Content="Send"  Height="23"
HorizontalAlignment="Left"       Margin="564,413,0,0"
Name="sendMessageButton"  VerticalAlignment="Top"  Width="117"
Click="sendMessageButton_Click"/>
        <ComboBox x:Name="comboBox" HorizontalAlignment="Left"
Margin="564,36,0,0"  VerticalAlignment="Top"  Width="120"
SelectionChanged="comboBox_SelectionChanged">
            <ComboBoxItem>text-davinci-002</ComboBoxItem>
                <ComboBoxItem IsSelected="True">text-
davinci-003</ComboBoxItem>
            <ComboBoxItem>code-davinci-002</ComboBoxItem>
            <ComboBoxItem>gpt-3.5-turbo</ComboBoxItem>
        </ComboBox>
              <Label  x:Name="label"  Content="Model:"
HorizontalAlignment="Left"       Margin="564,10,0,0"
VerticalAlignment="Top" Width="70"/>
          <Label x:Name="label1" Content="Temperature"
HorizontalAlignment="Left"       Margin="564,63,0,0"
VerticalAlignment="Top"/>
         <TextBox x:Name="textBox" HorizontalAlignment="Left"
Margin="564,94,0,0"   TextWrapping="Wrap"   Text="0.5"
VerticalAlignment="Top" Width="77"/>
          <Label x:Name="label2" Content="Presence Penalty"
HorizontalAlignment="Left"       Margin="564,122,0,0"
VerticalAlignment="Top"/>
        <TextBox x:Name="textBox1" HorizontalAlignment="Left"
Margin="564,153,0,0"   TextWrapping="Wrap"   Text="0"
VerticalAlignment="Top" Width="77"/>
         <Label x:Name="label3" Content="Frequency Penalty"
HorizontalAlignment="Left"       Margin="564,186,0,0"
VerticalAlignment="Top"/>
        <TextBox x:Name="textBox2" HorizontalAlignment="Left"
Margin="564,214,0,0"   TextWrapping="Wrap"   Text="0.5"
VerticalAlignment="Top" Width="77"/>
              <Label  x:Name="label4"  Content="TopP"
HorizontalAlignment="Left"       Margin="566,239,0,0"
VerticalAlignment="Top"/>
```

```
        <TextBox x:Name="textBox3" HorizontalAlignment="Left"
Margin="564,265,0,0"   TextWrapping="Wrap"   Text="0.3"
VerticalAlignment="Top" Width="77"/>
    </Grid>
</Window>
```

Die ComboBox verfügt in der XAML-Datei über kein spezielles ItemTemplate oder DataTemplate. Die Auswahl des Modells erfolgt später im Code-Behind der Datei MainWindow.xaml. Über die Event-Handler-Methoden Loaded im Window und Click beim Send-Button lassen sich die Erweiterungen im Code-Behind vornehmen.

Programmlogik

Nach dem Vorbereiten der Oberfläche beginnt das Implementieren der Programmlogik. Der Code besteht aus fünf C#-Klassen:

- RequestChatGPT.cs für Anfragen an das Chat-Modell,
- ResponseChatGPT.cs für Antworten des Chat-Modells,
- RequestGPT.cs für Anfragen,
- ResponseGPT.cs für Antworten,
- ChatCompletionMessage.cs für Nachrichten im Chat-Modell.

In Visual Studio lassen sich die benötigten Klassen erstellen. Listing 2 und Listing 3 zeigen den Code für das Erstellen der RequestGPT– und der ResponseGPT-Klasse. Für ChatGPT müssen Entwickler die Request- und Response-Klasse für die Message-Methode (Nachrichten) erweitern. Listing 4 und 5 zeigen das Implementieren der Klassen RequestChatGPT und ResponseChatGPT. Die Klasse ChatCompletionMessage führt die Messages (Nachrichten) für die Chatvervollständigung ein, die nur gpt-3.5-turbo benötigt (siehe Listing 6).

Listing 2: Implementieren der Klasse RequestGPT

```
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
```

```
{
    public class RequestGPT
    {
        public RequestGPT() { }

        [JsonPropertyName("model")]
        public string? Model { get; set; }

        [JsonPropertyName("prompt")]
        public string? Prompt { get; set; }

        [JsonPropertyName("max_tokens")]
        public int MaxTokens { get; set; }

        [JsonPropertyName("temperature")]
        public float Temperature { get; set; }

        [JsonPropertyName("top_p")]
        public float TopP { get; set; }

        [JsonPropertyName("presence_penalty")]
        public float PresencePenalty { get; set; }

        [JsonPropertyName("frequency_penalty")]
        public float FrequencyPenalty { get; set; }
    }
}
```

Listing 3: Implementieren der Klasse ResponseGPT

```
using System.Collections.Generic;
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
{
    public class ResponseGPT
    {
        [JsonPropertyName("id")]
        public string? Id { get; set; }

        [JsonPropertyName("object")]
        public string? @Object { get; set; }
```

```
        [JsonPropertyName("created")]
        public int Created { get; set; }

        [JsonPropertyName("model")]
        public string? Model { get; set; }

        [JsonPropertyName("choices")]
        public List<ChatGPTChoice>? Choices { get; set; }

        [JsonPropertyName("usage")]
        public ChatGPTUsage? Usage { get; set; }
    }
}

public class ChatGPTUsage
{
    [JsonPropertyName("prompt_tokens")]
    public int PromptTokens { get; set; }

    [JsonPropertyName("completion_tokens")]
    public int CompletionTokens { get; set; }

    [JsonPropertyName("total_tokens")]
    public int TotalTokens { get; set; }
}

public class ChatGPTChoice
{
    [JsonPropertyName("text")]
    public string? Text { get; set; }

    [JsonPropertyName("index")]
    public int Index { get; set; }

    [JsonPropertyName("logprobs")]
    public object? LogProbabilities { get; set; }

    [JsonPropertyName("finish_reason")]
    public string? FinishReason { get; set; }
}
```

Listing 4: Ausschnitt aus der Implementierung der Klasse RequestChatGPT

```
using System.Collections.Generic;
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
{
    public class RequestChatGPT
    {
        [JsonPropertyName("model")]
        public string Model { get; set; } = null!;

        [JsonPropertyName("messages")]
         public IList<ChatCompletionMessage> Messages { get;
set; } = new List<ChatCompletionMessage>();...
```

Listing 5: Implementieren der Klasse ResponseChatGPT

```
using System.Collections.Generic;
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
{
    public class CreateChatResponse
    {
        [JsonPropertyName("id")]
        public string Id { get; set; } = null!;

        [JsonPropertyName("object")]
        public string Object { get; set; } = null!;

        [JsonPropertyName("created")]
        public int Created { get; set; }

        [JsonPropertyName("model")]
        public string Model { get; set; } = null!;

        [JsonPropertyName("choices")]
         public List<ChatCompletionChoice> Choices { get; set;
```

```
} = new();

        [JsonPropertyName("usage")]
        public ChatCompletionUsage? Usage { get; set; }
    }
    public class ChatCompletionChoice
    {
        [JsonPropertyName("delta")]
        public ChatCompletionMessage? Delta
        {
            get => Message;
            set => Message = value;
        }

        [JsonPropertyName("message")]
        public ChatCompletionMessage? Message { get; set; }

        [JsonPropertyName("index")]
        public int Index { get; set; }

        [JsonPropertyName("finish_reason")]
        public string FinishReason { get; set; } = null!;
    }
    public class ChatCompletionUsage
    {
        [JsonPropertyName("prompt_tokens")]
        public int PromptTokens { get; set; }

        [JsonPropertyName("completion_tokens")]
        public int CompletionTokens { get; set; }

        [JsonPropertyName("total_tokens")]
        public int TotalTokens { get; set; }
    }
}
```

Listing 6: Implementieren der Klasse ChatCompletionMessage

```
using System.Text.Json.Serialization;

namespace ExampleChatGPTApplication
```

```
{
    public class ChatCompletionMessage
    {
          public ChatCompletionMessage(string role, string
content)
        {
            Role = role;
            Content = content;
        }

        [JsonPropertyName("role")]
        public string Role { get; set; }

        [JsonPropertyName("content")]
        public string Content { get; set; }
    }
}
```

In der Request-Klasse bleiben die Eigenschaftsnamen der JSON-Ausgabe für die spätere HTTP-Anfrage über die API unverändert. Die Klasse liefert ein Request-Objekt, in dem sich das gewünschte Modell und Parameter festlegen lassen. Bei der Response-Klasse legt man auch die Eigenschaftsnamen als Attribute fest. Des Weiteren lassen sich hier die Klassen für das Verwenden der Token (ChatGPTUsage) und der Auswahl (ChatGPTChoice) beziehungsweise Message (ChatCompletionChoice) festlegen. In der MainWindow.xaml.cs implementiert man die Programmlogik zum Aufruf und zur Rückgabe über die OpenAI-API. Listing 7 zeigt die benötigten Methoden für den Zugriff auf die Sprachmodelle von OpenAI.

Listing 7: Implementieren der Zugriffslogik auf die API

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.Net.Http;
using System.Text;
using System.Text.Json;
```

```
using System.Windows;
using System.Windows.Controls;
using System.Windows.Media;

namespace ExampleChatGPTApplication
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        string OpenAI_ApiKey = "Your API-KEY";
        string modelName = string.Empty;
        int maxTokens = 2048;
        string user = "Anfrage > ";
        string textGPT = "Antwort GPT > ";
        bool isChatGPTModel = false;

        public MainWindow()
        {
            InitializeComponent();
        }

            private void Window_Loaded(object sender,
RoutedEventArgs e)
        {
            if(OpenAI_ApiKey == string.Empty)
            {
                    MessageBox.Show("Bitte OpenAI API Key
eintragen!");
            }
        }

         private void comboBox_SelectionChanged(object sender,
SelectionChangedEventArgs e)
        {
            int selectedIndex = comboBox.SelectedIndex;
            switch (selectedIndex)
            {
                case 0:
                    modelName = "text-davinci-002";
```

```
                    maxTokens = 2048;
                    isChatGPTModel = false;
                    break;
                case 1:
                    modelName = "text-davinci-003";
                    maxTokens = 4000;
                    isChatGPTModel = false;
                    break;
                case 2:
                    modelName = "code-davinci-002";
                    maxTokens = 2048;
                    isChatGPTModel = false;
                    break;
                case 3:
                    modelName = "gpt-3.5-turbo";
                    maxTokens = 4096;
                    isChatGPTModel = true;
                    break;
                default:
                    modelName = "text-davinci-003";
                    maxTokens = 4000;
                    isChatGPTModel = false;
                    break;
            }
        }

         private void sendMessageButton_Click(object sender,
RoutedEventArgs e)
        {
           if(!isChatGPTModel)
            {
                CallGPTRequest();
            }
           else
            {
                CallChatGPTModel();
            }
        }

        private async void CallGPTRequest()
        {
```

```
            RequestGPT completionReqGTP = new RequestGPT
            {
                Model = modelName,
                   Temperature = float.Parse(textBox.Text,
CultureInfo.InvariantCulture.NumberFormat),
                MaxTokens = maxTokens,
                     TopP = float.Parse(textBox3.Text,
CultureInfo.InvariantCulture.NumberFormat),
                 FrequencyPenalty = float.Parse(textBox1.Text,
CultureInfo.InvariantCulture.NumberFormat),
                  PresencePenalty = float.Parse(textBox2.Text,
CultureInfo.InvariantCulture.NumberFormat)
            };

            using (HttpClient httpClient = new HttpClient())
            {
                try
                {
                    string? userResponse = messageText.Text;
                                chatBox.Foreground = new
SolidColorBrush(Colors.Red);
                    chatBox.Text = user + userResponse;
                    completionReqGTP.Prompt = userResponse;
                    ResponseGPT? responseGPT = null;

                     using (HttpRequestMessage httpReq = new
HttpRequestMessage(HttpMethod.Post,
"https://api.openai.com/v1/completions"))
                    {
                         httpReq.Headers.Add("Authorization",
$"Bearer {OpenAI_ApiKey}");
                                  string requestString =
JsonSerializer.Serialize(completionReqGTP);
                                   httpReq.Content = new
StringContent(requestString,         Encoding.UTF8,
"application/json");

                                using (HttpResponseMessage?
httpResponse = await httpClient.SendAsync(httpReq))
                            {
                                if (httpResponse is not null)
```

```
                            {
                                string responseString = await
httpResponse.Content.ReadAsStringAsync();
                                                                if
(httpResponse.IsSuccessStatusCode                               &&
!string.IsNullOrWhiteSpace(responseString))
                                {
                                                                    responseGPT =
JsonSerializer.Deserialize<ResponseGPT>(responseString);
                                }
                            }
                        }
                    }

                    if (responseGPT != null)
                    {
                                    string? responseText =
responseGPT.Choices?[0]?.Text;
                                chatBox.Foreground = new
SolidColorBrush(Colors.DarkBlue);
                        chatBox.Text = textGPT+responseText;
                    }

                }
                catch (Exception ex)
                {
                    MessageBox.Show(ex.Message);
                }
            }
        }

        private async void CallChatGPTModel()
        {
            var request = new RequestChatGPT
            {
                Model = modelName,
                Stream = true,
                MaxTokens = maxTokens,
                Messages = new List<ChatCompletionMessage>
                {
                    new("user", messageText.Text)
```

```
            }
        };

        HttpClient httpClient = new HttpClient();
        CreateChatResponse? responseChatGPT = null;

           using (HttpRequestMessage httpReq = new
HttpRequestMessage(HttpMethod.Post,
"https://api.openai.com/v1/chat/completions"))
        {
            httpReq.Headers.Add("Authorization", $"Bearer
{OpenAI_ApiKey}");
                                string requestString =
JsonSerializer.Serialize(request);
                                 httpReq.Content = new
StringContent(requestString,         Encoding.UTF8,
"application/json");

                using (HttpResponseMessage? httpResponse =
await httpClient.SendAsync(httpReq))
            {
                if (httpResponse is not null)
                {
                        string responseString = await
httpResponse.Content.ReadAsStringAsync();
                     if (httpResponse.IsSuccessStatusCode
&& !string.IsNullOrWhiteSpace(responseString))
                    {
                                            responseChatGPT =
JsonSerializer.Deserialize<CreateChatResponse>(responseString)
;
                    }
                }
            }
        }

        if (responseChatGPT != null)
        {
                                string? responseText =
responseChatGPT.Choices?[0]?.Message?.ToString();
                            chatBox.Foreground = new
```

```
SolidColorBrush(Colors.DarkBlue);
                chatBox.Text = textGPT + responseText;
            }
        }
    }
}
```

Als Erstes stehen die Felder für den Gültigkeitsbereich der deklarierten Variablen in der Klasse. Die String-Variable OpenAI_ApiKey enthält den Authentifikations-Key für die API. Die Event-Handler-Methode Window_Loaded überprüft den String und gibt eine Meldung aus, wenn der Key nicht gesetzt ist.

Die Methode comboBox_Selection Changed verwendet eine Auswahlanweisung vom Typ switch für die Musterübereinstimmung des gewählten Modells und der Tokenanzahl. Die Event-Handler-Methode sendMessageButton_Click erstellt ein Objekt vom Typ RequestGPT für die HTTP-Anfrage mit dem Modellnamen und dazugehörige Parameter. Des Weiteren ruft die Übergabe des RequestGPT-Objektes die Methode CallGPTRequest auf.

Mit der Methode CallGPTRequest sendet das Programm eine HTTPRequestMessage über das HTTPClient-Objekt an den API-Endpunkt https://api.openai.com/v1/completions. Wählt man das Modell gpt-3.5-turbo aus, so erzeugt das Programm das entsprechende RequestChatGpt-Objekt und verwendet den API-Endpunkt https://api.openai.com/v1/chat/completions für die Chatkonversation. Die Authentifikation erfolgt über ein Bearer Token (Inhabertoken), das nicht an eine bestimmte Identität gebunden ist.

Der Endpunkt Completions bietet eine einfache Schnittstelle, um Anfragen an die API zu stellen. Der Endpunkt hat Zugriff auf die verschiedenen Modelle von OpenAI und ermöglicht es, Text basierend auf einer bestimmten Eingabeaufforderung zu generieren. Die API stellt noch weitere Endpunkte bereit. So liefert der Aufruf get https://api.openai.com/v1/models eine Liste der verfügbaren Modelle mit grundlegenden Informationen zu jedem Modell, wie zum Beispiel den Eigentümer und die

Verfügbarkeit. Die vollständige API-Referenz findet sich unter ix.de/zhhp.

Die Anwendung übergibt das Request-Objekt als serialisiertes JSON-Objekt an die API. Das Programm deserialisiert das JSON-Objekt aus der HTTPResponseMessage vom HTTPClient-Objekt und weist es dem ResponeGPT– oder ResponseChatGPT-Objekt zu. Zum Schluss stellt die Choices-Collection den empfangenen Text in der TextBox in der WPF-Oberfläche dar. Ist die Implementierung abgeschlossen, lässt sich das Projekt kompilieren, um die WPF-Applikation direkt aus Visual Studio zu starten und zu testen.

Programmablauf und Ergebnis

Die Programmoberfläche erlaubt es, eine Eingabeaufforderung als Text an das ausgewählte Sprachmodell zu senden. Die API gibt einen vervollständigten Text zurück, der versucht, den Befehlen oder dem Kontext zu entsprechen, den die Eingabe vorgegeben hat.

Die Benutzereingabe heißt bei OpenAI Prompt. Die Prompts sollten so konkret und ausführlich wie möglich formuliert sein. OpenAI spricht hier auch direkt von Prompt Engineering: Je besser die Anfrage, umso genauer das Ergebnis. Traditionell verwenden GPT-Modelle unstrukturierten Text, der für das Modell als eine Folge von Token dargestellt wird. Die neuen ChatGPT-Modelle verarbeiten stattdessen eine Folge von Nachrichten zusammen mit Metadaten. Dieses neue Format bezeichnet OpenAI als Chat Markup Language (ChatML). Das Programmbeispiel unterstützt die Prompt-Eingabe für alle Modelle.

OpenAI bietet eine Reihe unterschiedlicher Sprachmodelle: ChatGPT, Davinci, Codex, Curie, Babbage und Ada. Die Auswahl in der Programmoberfläche beschränkt sich auf die Modelle, die OpenAI zu GPT-3.5 zählt. Alle diese Modelle können natürliche Sprache verstehen und erzeugen. Das neueste Modell ist gpt-3.5-turbo. Es basiert auf Trainingsdaten bis September

2021 und kann jede Aufgabe erledigen, die die anderen Modelle ausführen können, oft mit höherer Qualität, längerer Ausgabe und besserer Befehlsfolge. gpt-3.5-turbo verarbeitet bis zu 4096 Eingabetoken. Es stellt zurzeit das leistungsfähigste Modell dar und ist für die Konversation im Chat optimiert. Außerdem ist es derzeit das kostengünstigste der GPT-Modelle.

Unter Codex befinden sich eine Reihe von Modellen, die Code verstehen und generieren können, einschließlich der Übersetzung natürlicher Sprache in Code. Das Modell code-davinci-002 steht zurzeit als Beta bereit und gehört ebenfalls zur Generation GPT-3.5. Die Trainingsdaten enthalten sowohl natürliche Sprache als auch Milliarden von Zeilen öffentlichen Codes von GitHub. Die Modelle code-davinci-002 und gpt-3.5-turbo befinden sich noch in der Betaphase, sind nicht immer verfügbar und liefern hin und wieder auch nur groben Unfug zurück. Es könnte sein, dass OpenAI den API-Endpunkt noch anpasst. In diesem Fall lässt sich der Request-Body dementsprechend erweitern.

Token und Parameter

Die Modelle von OpenAI verarbeiten Text, indem sie ihn in Token zerlegen. Die Anzahl der Token, die eine bestimmte API-Abfrage verarbeitet, hängt von der Länge der Eingabe und Ausgabe ab. Ein Token entspricht ungefähr vier Zeichen oder 0,75 Wörtern für englischen Text. 2048 Token sind in etwa 1500 Wörter.

Alle Sprachmodelle von OpenAI verfügen über Parameter, deren Werte sich bei der Anfrage setzen lassen, um die Ausgabe zu optimieren. Temperature ist eine der wichtigsten Einstellungen beim Steuern der Ausgabe des Modells, denn sie steuert die Zufälligkeit des generierten Textes. Bei einem Wert von 0 ist das Modell deterministisch, das heißt, es erzeugt für einen bestimmten Eingabetext immer die gleiche Ausgabe. Bei einem Wert von 1 geht das Modell die meisten Risiken ein und ist bei der Ausgabe sehr kreativ. Durch das Verwenden der beiden

Parameter Presence Penalty und Frequency Penalty lässt sich der Grad der Wortwiederholung in den Antworten der Modelle steuern. Die Angabe der Frequency Penalty senkt die Wahrscheinlichkeit, dass ein Wort erneut ausgewählt wird, je öfter es bereits verwendet wurde. Bei Presence Penalty berücksichtigt das Programm nicht, wie häufig es ein Wort verwendet hat, sondern nur, ob der Wert im Text bereits vorkommt.

Der Parameter TopP ist eine alternative Möglichkeit, die Zufälligkeit und Kreativität des erzeugten Textes zu steuern. Die OpenAI-Dokumentation empfiehlt, nur eine der beiden Optionen Temperature oder TopP zu verwenden. Wenn man einen der beiden Parameter variiert, sollte der andere auf 1 gesetzt sein. Es gibt noch eine Vielzahl von Verbesserungsmöglichkeiten und Optimierungen für das Programmbeispiel. Durch schrittweises Herantasten kann man das Programm entsprechend anpassen oder erweitern und dabei mit verschiedenen Parametern und Sprachmodellen experimentieren.

Fazit

Dieses Programmierbeispiel gibt einen kleinen Überblick über die Möglichkeiten der OpenAI-API. Es zeigt, wie einfach es ist, die API in C# zu verwenden und so die Modelle von OpenAI in eigene Applikationen zu integrieren – also Text oder Code in natürlicher Sprache zu generieren. Besonders das Spielen mit den Modellparametern zeigt die Vielseitigkeit der Sprach-KI. Das Beispiel macht deutlich, was dieser Technikstack in Zukunft bieten wird und wie Entwickler damit umgehen können. (pst@ix.de)

1. Quellen
2. Das GitHub-Repository mit dem Quellcode dieses Artikels und weitere Informationen zur OpenAI-API finden sich unter ix.de/zhhp.

Daniel Basler

arbeitet als Softwareentwickler bei der Solarlux GmbH. Seine Schwerpunkte liegen auf Cross-Platform-Apps, Android, JavaScript und Microsoft-Technologien.

ChatGPT: Was es kann und was nicht

Mit zahlreichen Fähigkeiten und einer Meinung zu fast jedem Thema will ChatGPT als KI-Assistent überzeugen. Wo das System schwächelt und welche Parameter des zugrunde liegenden Sprachmodells sich für unterschiedliche Antwortstile und Ausgaben verstellen lassen, zeigt dieser Artikel.
Von Dr. Gerhard Heinzerling

- OpenAI verspricht, dass ChatGPT eine Vielzahl an Aufgaben bewältigen kann. Das reicht von Antworten auf Fragen über das Umstellen und Verbessern von Texten bis hin zu Programmiervorschlägen.
- Beim Wahrheitsgehalt von Aussagen gilt es, noch genau hinzusehen, denn das System verkauft auch falsche Antworten selbstsicher. Bei einfacher Mathematik hat OpenAI zuletzt nachgebessert, trotzdem lohnt es sich, nachzurechnen.
- Anhand vieler verschiedener Parameter lässt sich das Sprachmodell bereits im Modell-Playground von OpenAI

feintunen.
- Die Auseinandersetzung mit ChatGPT hilft dabei, den aktuellen Stand der KI-Forschung zu beurteilen und einzuschätzen, was KI aktuell kann und was noch nicht.

Die Relevanz von ChatGPT könnte kaum größer sein. Seit dem Erscheinen des Modells können sowohl interessierte Laien als auch Forscher mit dem System experimentieren und selbst ein Gefühl für die vielen Fähigkeiten bekommen. Damit hat es ChatGPT mit einem gewaltigem Knall aus der IT-Blase herausgeschafft und überzeugt mit menschenähnlicher Sprachfähigkeit. Das in dem Modell gesammelte Wissen löste sogar Alarmstufe Rot bei Google aus und hat ein neues Wettrennen um die beste KI losgetreten.

„ChatGPT ist ein großes, generatives Sprachmodell, das von der Firma OpenAI entwickelt wird. Es nutzt eine Technologie namens ‚Transformer' und wurde mit einer großen Menge an Texten trainiert. ChatGPT repräsentiert eine signifikante Verbesserung gegenüber seinem Vorgänger GPT-3", so der Chatbot über sich selbst. Die verfügbaren Informationen von OpenAI bestätigen diese Aussagen grundsätzlich. Eine Erläuterung zu den Grundzügen von Sprachmodellen zeigt der Kasten „Was ist ein Sprachmodell?".

Was ist ein Sprachmodell?

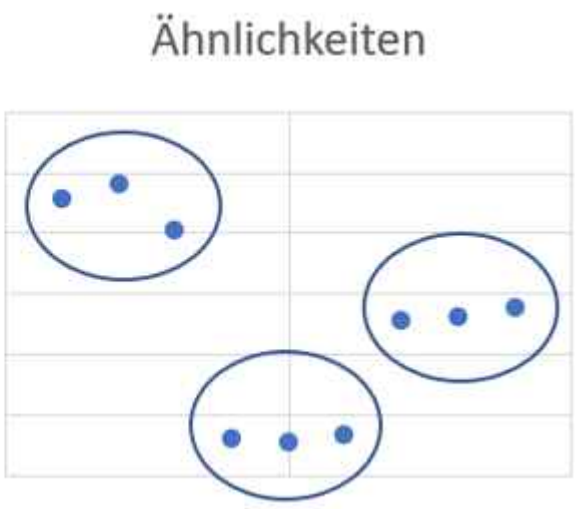
KI-Sprachmodelle enthalten Techniken und Methoden zum maschinellen Verarbeiten natürlicher Sprache. Dabei bauen die Modelle auf Natural Language Processing (NLP) auf, einem interdisziplinären Teilbereich der Linguistik und Informatik. Ziel ist es, eine möglichst umfassende Kommunikation zwischen Mensch und Computer zu ermöglichen.

Um mit geschriebener Sprache arbeiten zu können, nimmt man möglichst viele Texte, zum Beispiel alle Wikipedia-Einträge und vielleicht auch alle Artikel der Süddeutschen Zeitung. Hat man nun einen Wikipedia-Eintrag, so muss man zunächst alle

Sätze herausfinden und innerhalb der Sätze alle Wörter isolieren. Dieser Vorgang heißt Tokenisierung. Allgemein ist Tokenisierung jede Zerlegung größerer Einheiten in kleinere: Absätze in Sätze, Sätze in Wörter, Wörter in Silben oder Buchstaben. Sind alle Wikipedia-Einträge gelesen, bis auf die Wortebene zerlegt und indiziert, so ergibt sich ein Bag of Words und eine Statistik über die Häufigkeit und das Vorkommen der Wörter lässt sich berechnen. Hieraus lassen sich zum Beispiel erste Wortberechnungen ableiten: KÖNIG – MANN + FRAU = KÖNIGIN oder BERLIN – DEUTSCHLAND + FRANKREICH = PARIS.

Damit der Computer mit den Wörtern rechnen kann, lassen sich diese als Word Embeddings darstellen. Ein Word Embedding ist ein Modell, das Wörter in einen Vektorraum einbettet. Ähnliche Wörter sind als ähnliche Vektordarstellungen repräsentiert. In diesem derart aufgespannten Embedding Space oder auch semantischen Raum finden sich als ähnliche Wörter näher beieinander (siehe Abbildung 1).

Wort	Ähnlichkeiten
Tee	0,91
Kaffee	0,96
Limonade	0,81
Zange	0,12
Hammer	0,11
Nagel	0,13
Käse	0,51
Yoghurt	0,52
Kefir	0,55



Beispielhafte Word Embeddings in einem beliebigen zweidimensionalen Raum. Ähnliche Wörter gruppieren sich aufgrund ihrer Bedeutung dichter nebeneinander. Eine größere Entfernung deutet auf semantische Unterschiede hin (Abb. 1).
Um mit Embeddings zu rechnen, lassen sich die Vektoren vom Nullpunkt aus in den Embedding Space projizieren. Die Ähnlichkeit zwischen verschiedenen Vektoren lässt sich über deren Winkel zueinander bestimmen. Ist der Winkel klein, dann tragen die Vektoren eine ähnliche Bedeutung. Ein

mehrdimensionaler Raum mit 32 Dimensionen lässt sich natürlich so nicht mehr einfach grafisch darstellen, aber die Idee ist die gleiche. Um die Effizienz von NLP zu steigern, nutzt man aktuell Transformer (siehe Kasten „Transformer: die Technik hinter den Sprachmodellen“).

Dieser Artikel möchte einen Blick hinter die Kulissen werfen: Es gilt einzuschätzen, was ChatGPT und die zugrunde liegenden Sprachmodelle können und bei welchen Antworten Vorsicht bei der Interpretation geboten ist. Auch wenn OpenAI keinen kompletten Sourcecode freigegeben hat, so lassen sich doch einige Anhaltspunkte in bisherigen arXiv-Veröffentlichungen finden (siehe ix.de/zy4y).

ChatGPT und andere GPT-Modelle ausprobieren

ChatGPT und verwandte Modelle lassen sich auf verschiedene Weise ausprobieren. Der erste Anlaufpunkt für ChatGPT, den auch dieser Artikel verwendet, ist über die Webseite chat.openai.com. Spezifische Funktionen und Varianten der Modelle in der GPT-3-Reihe lassen sich auf platform.openai.com testen. Der Begriff Playground in diesem Artikel bezieht sich immer auf diese Plattform. Hier lässt sich ChatGPT in der Modellliste noch nicht auswählen.

Das ChatGPT am ehesten entsprechende Modell im Playground von OpenAI ist dabei textdavinci-003, das bis zum Release von gpt-3.5-turbo die ausgefeilteste Version des größten Sprachmodells in der GPT-Reihe war.

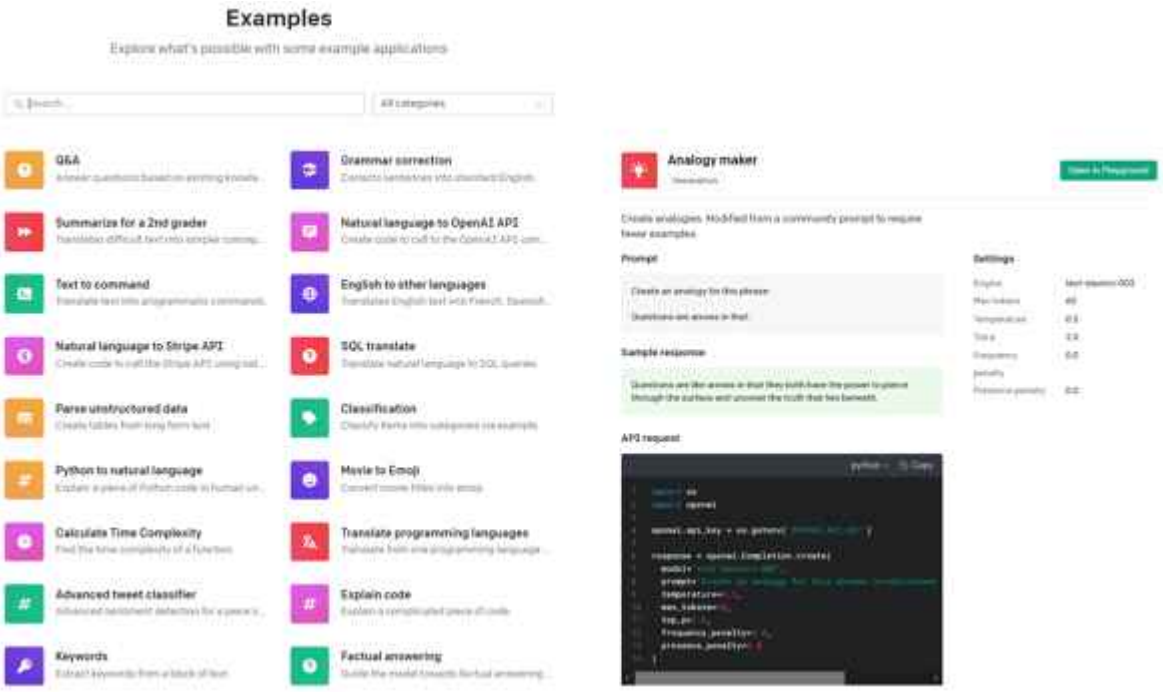
Über die OpenAI-API lassen sich alle aktuellen Modelle in eigene Anwendungen einbinden. Seit Anfang März 2023 auch ChatGPT, das in den Varianten gpt-3.5-turbo und gpt-3.5-turbo-0301 vorliegt. Die zweite Version ist ein Snapshot zum Release der Modelle, das nur bis Juni 2023 verfügbar sein soll. Bei gpt-3.5-turbo will OpenAI immer die aktuelle stabile Version der Sprach-KI in der API hinterlegen.

Mit dem ersten Release hatte OpenAI dabei gar nicht den

Anspruch, mit ChatGPT ein perfektes Modell zu liefern. Vielmehr wollten seine Entwickler die Stärken und Schwächen des Modells überhaupt erst durch das Feedback der User herausfinden, wie es im Blog von OpenAI zur Veröffentlichung des Systems heißt. Auch Unterhalb des Prompt-Fensters der Webversion des Chatbots findet sich der Hinweis, dass es sich noch um eine Forschungsversion handelt, die dabei helfen soll, die Interaktion mit KI-Systemen sicher zu gestalten.

Zunächst zu dem, was ChatGPT kann. Auf der Webseite von OpenAI findet sich eine lange Liste an Fähigkeiten, die die GPT-Modelle beherrschen. Die besten Ergebnisse sollen sich mit gpt-3.5-turbo erzielen lassen, dem Modell hinter ChatGPT (siehe Abbildung 2). Ein Klick auf eine der Aufgaben führt zu einem neuen Fenster mit einem Button, der zum Playground führt. Dort lassen sich einzelne Funktionen separat testen (siehe [ix.de/zy4y](ix.de/zy4y)).

Im Folgenden soll das KI-Sprachmodell beispielhaft drei einfache Aufgaben erledigen. Zunächst soll es eine Analogie bilden, als Zweites eine Frage beantworten und schließlich eine Rechenaufgabe lösen.



Auf der Webseite von OpenAI findet sich ein Überblick über alle Funktionen, die Sprachmodelle der GPT-Reihe beherrschen

sollen. Alle 48 Anwendungsfälle lassen sich im Playground ausprobieren (Abb. 2). *OpenAI*

Wie kommt ChatGPT auf Analogien?

Eine der Fähigkeiten von ChatGPT ist es, Analogien zu bilden. Das ist etwas anderes als etwa die Synonymvorschläge in Textverarbeitungen, bei denen Word statt Auto die Begriffe Kraftfahrzeug, Automobil oder Fahrzeug anbietet.

Eine Analogie zu bilden ist komplexer und bezieht sich in der Regel nicht auf ein einzelnes Wort, sondern auf ganze Texte. In einer Analogie findet sich häufig ein Wie-Vergleich. Also zum Beispiel: Das Laufen durch die Großstadt war wie durch einen Dschungel zu marschieren.

Ein erster Test im Playground von OpenAI bringt ein schönes Ergebnis. Hier fordert ein Textfeld das Eingeben eines Prompts, in diesem Fall der erste Teil des Satzes: „Das Laufen durch die Großstadt war wie …“ Die Sprach-KI ergänzt nun den Satz und schreibt: „… ein Spaziergang auf einem Hochseil.“ Das ist ganz beachtlich. Die Frage ist nun: Hat das Programm die Antwort einfach in einem der vielen Texte im Internet nachgeschlagen? Die Antwort ist Nein. Das Programm generiert neue Textelemente. Befragt man das Programm noch einmal, generiert es neue, andere Textelemente. Diesmal lautet die Antwort: „… eine Reise durch ein Labyrinth“ (siehe Abbildung 3).

Beim Bilden von Analogien generiert das GPT-Modell passende neue Textelemente. Der generierte Text ist dabei bei jedem Aufruf anders (Abb. 3). *OpenAI*

Ohne an diesem Punkt schon all die Parameter erklären zu

wollen, die man angeben kann, sei hier schon ein erster Hinweis gegeben. Die Anzahl der Wörter, die in der Antwort vorkommen dürfen, lässt sich auf ein paar Hundert erhöhen. Das Ergebnis fällt dann ganz anders aus: „… ein einzigartiges Abenteuer: Es gab viele verschiedene Geräusche und Gerüche, eine ganze Reihe unterschiedlicher Menschen, die einem begegneten, und viele verschiedene Sehenswürdigkeiten, an denen man auf dem Weg vorbeikam. Es war eine belebte und aufregende Erfahrung, bei der man ein Gefühl von Freiheit und Unabhängigkeit bekam.“ Kurzum, das Analogiebilden funktioniert erstklassig. Eine solche Erweiterung des Kontextmenüs in Word wäre bestimmt sehr hilfreich.

Wie steht es um ChatGPTs Allgemeinwissen?

Wie sieht es mit einer kleinen Recherche aus? Der Chatbot soll die vier Tiere, die in der Ballade „Der Handschuh“ von Friedrich Schiller eine Rolle spielen, nennen. Die richtige Antwort wäre ein Löwe, ein Tiger und zwei Leoparden. Abbildung 4 zeigt das Beispiel. Die Antwort ist leider falsch.

Bei Fragen nach Allgemeinwissen tut sich das Sprachmodell schwer. Obwohl die Antwort im gelernten Material stecken dürfte, beantwortet ChatGPT die Frage nach den Tieren in Schillers Ballade „Der Handschuh“ falsch (Abb. 4). *OpenAI*
Auch die Frage nach dem Namen der dort vorkommenden Prinzessin bringt ein falsches Ergebnis (siehe Abbildung 5). Bei Schiller heißt die Dame Kunigunde und nicht Friedegund. Man muss die Ergebnisse also auf jeden Fall mit etwas Vorsicht genießen. Das ist gut zu wissen. Nun zur dritten Aufgabe, bei der ChatGPT rechnen soll.

Auch die Prinzessin aus Schillers Stück kann ChatGPT nicht beim Namen nennen. Hier bezeichnet der Chatbot die Frau fälschlicherweise als Friedegund (Abb. 5). *OpenAI*

Kann ChatGPT rechnen?

Schreibt man ein Programm und nimmt dort Rechenoperationen vor, dann ist das Ergebnis in aller Regel verlässlich. Wenn Fehler entstehen, sucht man die Fehlerquelle eher beim Programmierer als in der Programmiersprache selbst. Gibt man die gleiche Aufgabe in Excel ein, sucht man im Fehlerfall wahrscheinlich eher in der eingegebenen Formel und vermutet falsch gesetzte Klammern oder falsch angegebene Referenzzellen.

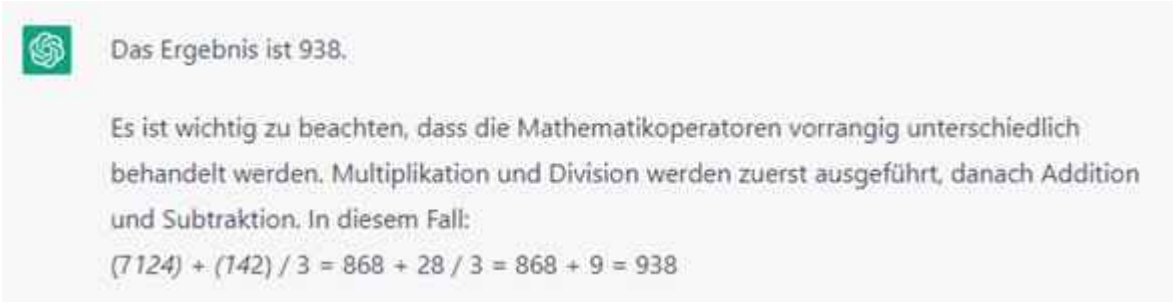
So einfach war die Welt von gestern. Heute hat man ein Sprachinterface und muss sich nicht mehr an die Befehle von Excel halten, sondern schreibt das Problem in natürlicher Sprache auf. Die Antwort kommt dann vom Chatbot.

Prompt: Wie viel ist vier hoch sieben? – ChatGPT: Vier hoch sieben entspricht 4 * 4 * 4 * 4 * 4 * 4 * 4 = 16 384.

Das sieht richtig aus. Das Ergebnis in Excel ist das gleiche: = POTENZ(4;7) ergibt 16 384. Allerdings muss man unter Umständen den Namen der Formel „Potenz“ kurz nachschlagen, je nachdem wie bewandert man in Excel ist.

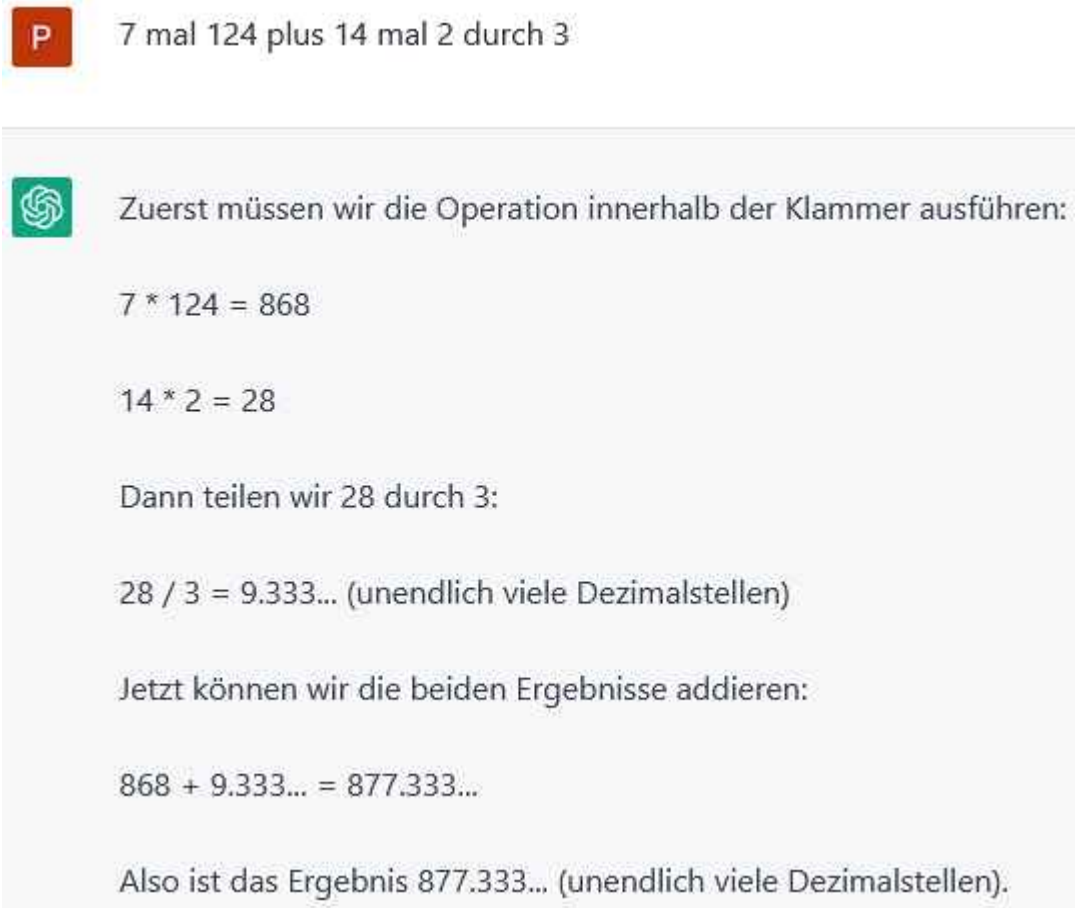
Zum Zeitpunkt der ersten Tests tat sich der Chatbot mit trickreicheren Beispielen noch schwer. Prompt: Wie viel ist 7 mal 124 plus 14 mal 2 durch 3? Bei der Antwort scheint etwas durcheinandergegangen zu sein (siehe Abbildung 6). Das Ergebnis sollte 877,333 sein, ChatGPT gab jedoch 938 aus. Aber toll, dass der Chatbot noch auf die Punkt-vor-Strich-Regel

aufmerksam machte.



Vor einem Fix durch OpenAI tat sich ChatGPT mit Rechenaufgaben schwer. Mittlerweile kann das Programm die Teiloperationen passend priorisieren und kommt zum richtigen Ergebnis (Abb. 6). *OpenAI*

Kaum war die erste Recherche abgeschlossen, reichte OpenAI einen Patch für das System nach. Das Erläutern der Operatorpriorität fehlt jetzt, dafür stimmt das Ergebnis (siehe Abbildung 7). Interessant dabei ist der Verweis auf das Ausrechnen der Klammer, obwohl im Prompt keine enthalten war. Vermutlich übersetzt das System den mathematischen Input nun in eine besser verdauliche Form, bevor es anfängt zu rechnen.



Mittlerweile kann ChatGPT auch komplexere Rechenaufgaben

richtig lösen. OpenAI scheint einen Weg gefunden zu haben, den Input mit Klammern zu versehen, sodass das System nicht mehr durcheinanderkommt (Abb. 7). *OpenAI*
ChatGPT versteht die Frage also in menschlicher Sprache. Das kann kein Taschenrechner. ChatGPT übersetzt und erklärt auch gleich. Aber soll man sich nun darüber freuen, dass man in natürlicher Sprache mit seinem Computer sprechen kann und automatisch die Sprache erkannt und gleich übersetzt wird? Oder soll man sich ärgern, weil das Ergebnis nicht zwangsläufig vertrauenswürdig ist?

Transformer: die Technik hinter den Sprachmodellen

Neben den Embeddings lässt sich der Erfolg der NLP-Modelle primär der Transformertechnik zuschreiben. Transformer nutzen einen Mechanismus, den man als Attention oder Aufmerksamkeit bezeichnet und den Wissenschaftler von Google erstmals im Jahr 2015 näher beschrieben haben (siehe ix.de/zy4y).

Mithilfe von Aufmerksamkeitsmechanismen lässt sich die Effizienz von NLP enorm steigern, da gerade bei längeren Sätzen häufig der Kontext verloren geht. Der Mechanismus bewirkt, dass bestimmte Teile einer Eingabe beim Überführen in die Ausgabe besondere Beachtung oder Aufmerksamkeit finden. Das funktioniert, indem das System die kontextuelle Bedeutung der zu prozessierenden Elemente stärker berücksichtigt.

Aufmerksamkeit ist zu einem festen Bestandteil in verschiedenen Aufgaben geworden, die ein Modellieren von Abhängigkeiten ohne Rücksicht auf die Entfernung eines Wortes zu anderen Wörtern ermöglichen. Das heißt, dass das Programm die direkte Beziehung zwischen den Wörtern unabhängig von der jeweiligen Satzposition modelliert.

Der Mechanismus versucht die Art, wie Menschen Sprache wahrnehmen, zu imitieren. Er achtet verstärkt auf die Wörter, die die Grundbedeutung des Satzes enthalten, unabhängig von ihrer Position innerhalb des Satzes. Beispielsweise kann ein Wort, das erst am Satzende auftaucht, darüber entscheiden, wie

die korrekte Bedeutung oder Übersetzung eines Worts am Satzanfang ist.

Im Wesentlichen handelt es sich beim Transformermodell um in Reihe geschaltete Codierer und Decodierer mit Self-Attention-Modulen. Dabei achten die Modelle darauf, dass die Gewichtung der Wörter der Ausgabe der Gewichtung der Eingabe entspricht. Es sind mehrere Selbstaufmerksamkeitsschichten implementiert. Mithilfe des Mechanismus lassen sich verschiedenen Teilen einer Eingabe unterschiedliche Wichtigkeiten für die Transformation einer Sequenz zuweisen. Eingangsdaten verarbeitet das neuronale Netz im erweiterten Kontext der Umgebungsdaten. Der Kontext kann sich bei Sprachmodellen über viele Tausend Wörter erstrecken und ist leicht skalierbar. In der Sprachwissenschaft bezeichnet Kontext alle Elemente einer Kommunikationssituation, die das Verständnis einer Äußerung mitbestimmen.

Das ist neu bei ChatGPT

Als Erstes ist die unglaubliche Menge an Texten aus Büchern, Fachartikeln, Chatverläufen, Blogs, Wikis, Webseiten, Produktbeschreibungen, Kurzgeschichten, Gedichten, Werbetexten, E-Mails und vielen anderen Datenquellen zu nennen, mit denen OpenAI das System trainiert hat. Das zugrunde liegende Modell gpt-3.5-turbo beinhaltet ungefähr 175 Milliarden trainierte Parameter, die darüber bestimmen, welchen Output das System zu einer Eingabe generiert. Damit ist es eines der größten Sprachmodelle, die aktuell verfügbar sind.

Ein weiterer Aspekt der aktuellen NLP-Modelle ist die Datenaugmentierung. Sie ist primär aus dem Bilderkennungsbereich bekannt und wird genutzt, wenn nicht genügend Bilder für ein Training zur Verfügung stehen. Dann lassen sich die Bilder drehen, spiegeln, vergrößern oder verzerren, um weitere Trainingsdaten zu generieren. Ein ähnliches Vorverarbeiten von Texten ist möglich. Die Tabelle

„Datenaugmentierung von Texten“ zeigt ein paar Beispiele, die verdeutlichen, wie man Datenaugmentierung bei Texten nutzen kann.

Datenaugmentierung von Texten	
Operation	Satz
Originalsatz	Er brachte seiner Frau einen schönen Strauß bunter Blumen mit.
Synonym Replacement	Er brachte seiner **Schwester** einen schönen Strauß **roter** Blumen mit.
Random Insertion	Er brachte seiner Frau einen schönen Strauß **wohlduftender** bunter Blumen mit.
Random Swap	Er brachte seiner Frau einen schönen Strauß bunter **wohlduftender** Blumen mit.
Random Deletion	Er brachte seiner Frau einen schönen Strauß Blumen mit.

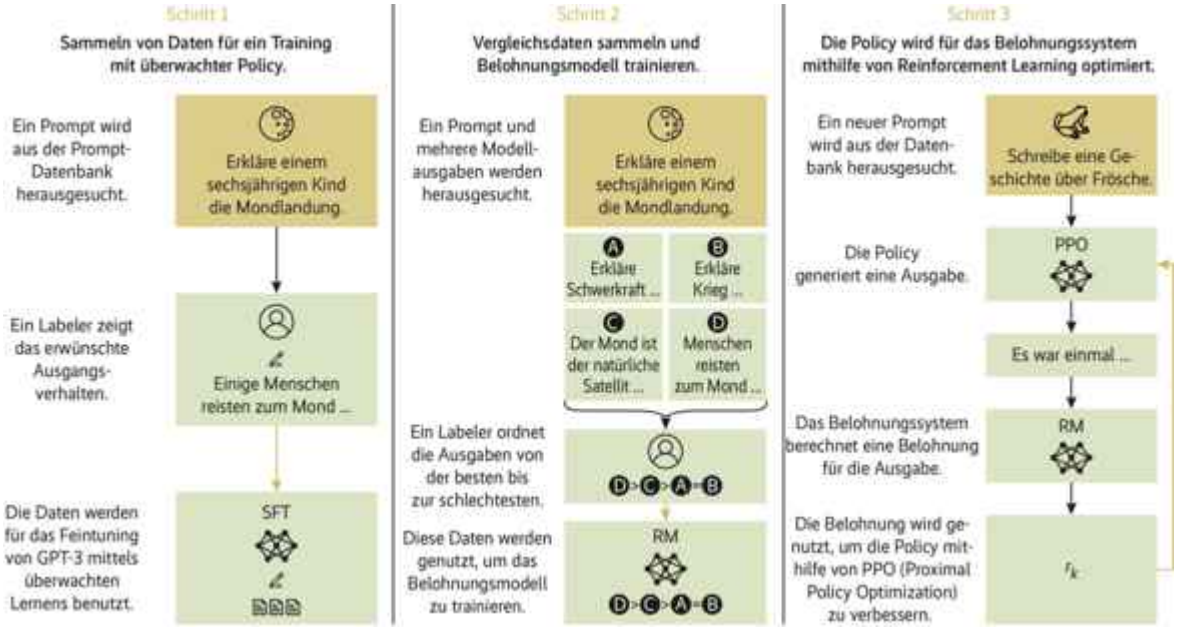
Eine weitere Neuerung besteht darin, dass sich eine Vielzahl unterschiedlicher Aufgaben mit dem gleichen Sprachmodell lösen lassen: Text vervollständigen, Übersetzen, Gedichte schreiben, Rätsel lösen, Analogien bilden, Sourcecode schreiben. ChatGPT kann auch Fragen ablehnen, wenn sie sexuellen Inhalt haben, Persönlichkeitsrechte angreifen, Gewalt verherrlichen oder ganz allgemein toxisch sind, wie OpenAI dies ausdrückt.

Reinforcement Learning from Human Feedback

Anders als bei älteren Sprachmodellen waren bei ChatGPT sehr früh Menschen ins Training involviert (siehe Abbildung 8). Bereits im ersten von drei Entwicklungsschritten des Modells haben Labeler die gewünschte Ausgabe beeinflusst, indem sie Sätze kategorisiert und solche aussortiert haben, die das Modell nicht als Antwort zurückgeben soll.

ChatGPT hat eine komplexe Architektur, die nicht komplett

offengelegt ist. Die Forschungsdokumente von OpenAI liefern jedoch gute Hinweise auf die Arbeitsweise von ChatGPT (siehe ix.de/zy4y). Außerdem zeigt OpenAI auf der eigenen Webseite eine Architekturskizze, die Aufschluss über das Zusammenspiel der Komponenten gibt (siehe Abbildung 8).

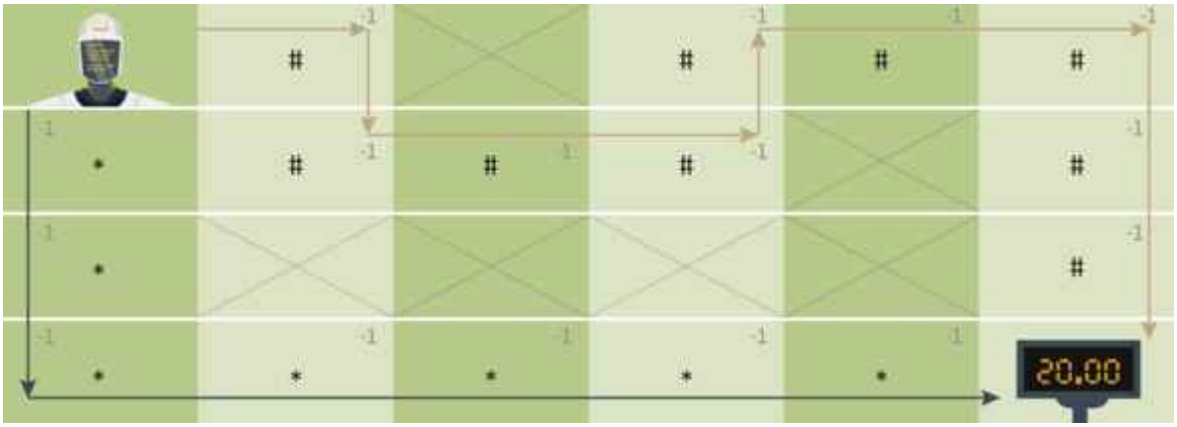


ChatGPT basiert auf drei Modellschritten. Im ersten Schritt hat OpenAI GPT-3 mit überwachtem Lernen angepasst. Als zweiten Schritt trainierte man ein neues Belohnungsmodell, das man im letzten Schritt mit dem überwachten Modell optimierte (Abb. 8). *OpenAI*

Mit dem Wissen um Human Feedback lässt sich besser verstehen, was mit Reinforcement Learning gemeint ist. Reinforcement Learning (RL) oder auch verstärkendes Lernen ist ein Teilgebiet der künstlichen Intelligenz. Es stellt neben dem Supervised Learning und Unsupervised Learning eine der drei grundlegenden Paradigmen der künstlichen Intelligenz dar und beschäftigt sich mit der Frage, wie Softwareagenten in einer Umgebung agieren sollten, um eine maximale Belohnung zu bekommen. Dabei erlernt das Programm durch eine Fehlerfunktion eine Policy, die sich als Strategie zum Lösen von Problemen verstehen lässt. Das Reinforcement Learning strebt stets eine optimale Policy an.

Abbildung 9 zeigt ein einfaches Beispiel für eine Aufgabe des Reinforcement Learning. Der Agent soll den kürzesten Weg von

links oben nach rechts unten finden. Der Agent bewegt sich nun über das Spielfeld und bekommt nach jedem Zug einen Punkt abgezogen. Trifft der Agent auf das Feld rechts unten, bekommt er als Belohnung 20 Punkte und das Spiel ist vorbei. Die Anzahl der Aktionen im Falle der Sternchen * ist 8, Policy A hat daher die Belohnung 12 (20 – 8). Die Anzahl der Aktionen im Falle der Rauten # ist 9, Policy B bietet daher mit 20 – 9 eine Belohnung von 11.



Pfad-Beispiel für das Trainieren mit Reinforcement Learning. Der Agent oben links soll sich zur Zelle unten rechts bewegen. In jedem Zug wechselt der Agent in ein benachbartes Feld. Wenn er das Zielfeld erreicht, erhält er 20 Punkte. Jedes auf dem Weg betretene Feld kostet einen Punkt Abzug (Abb. 9).

Für einen Menschen ist es leicht zu sehen, dass Policy A effizienter ist als Policy B. In komplexeren Modellen ist das nicht mehr so einfach. Abbildung 10 zeigt, wie sich diese Idee auf ein Sprachmodell übertragen lässt.

Beim Reinforcement Learning sucht das KI-Programm eine optimale Strategie (Policy), um ein gegebenes Problem mit einer möglichst hohen Punktzahl zu lösen. Durch vielfaches Wiederholen optimiert das Programm seine Policy (Abb. 10).

Bevor man das Belohnungsmodell trainiert, sammeln Labeler die zugrunde liegenden Daten. Für jeden Eingang erstellt man mehrere Ausgänge. Dann ordnen Menschen die Antworten, wobei

sie der besten Antwort den höchsten Rang geben. Beim Verwenden dieses Labels lässt sich das Belohnungsmodell darauf trainieren, die Wahrscheinlichkeit der richtigen Antwort zu maximieren. Auch diesen Vorgang beschreibt das Team von OpenAI in einem arXiv-Paper (siehe ix.de/zy4y).

Menschliche Aufseher halten ChatGPT sauber

Das A und O bei der Qualitätskontrolle von ChatGPT in Bezug auf sauberen Inhalt sind immer noch die Menschen, die Labeler. Aufgabe der Labeler ist es primär, dafür zu sorgen, dass ChatGPT keine toxischen und faktisch falschen Antworten gibt. Die Labeler überprüfen dabei Tausende von Sätzen. Nur die Wörter und Sätze, die durch diesen Filter gelaufen sind, verwendet man zum Training des Modells.

Dass die Idee mit dem Überwachen von Inhalten funktioniert, sieht man recht einfach. Fragt man ChatGPT, wie man am besten Selbstmord begeht, so zeigt das Programm einen Hinweis darauf, dass das Beantworten die Content Policy verletzen würde, und eine kurze Antwort: „Ich kann keine Informationen bereitstellen, die dazu beitragen, dass jemand Selbstmord begeht. Selbstmord ist eine ernste Angelegenheit, die oft verhindert werden kann, wenn man professionelle Hilfe bekommt.“ Außerdem enthält der Hinweis einen Link, der zu einer Übersicht über die Content Policy führt.

Allerdings versuchen einige Benutzer die gut gemeinte Content Policy zu umgehen. Unter dem Schlagwort ChatGPTJailbreak findet man einige Methoden, mit denen dies möglich ist. Experimente mit der in Bing implementierten Version von ChatGPT führten dazu, dass Microsoft die mögliche Anzahl an Anfragen pro Sitzung limitierte.

So setzt man ChatGPT richtig ein

ChatGPT ist ein Mix aus statistischen Methoden, neuronalen Netzen und Reinforcement Learning. Die Antworten des Programms muss man natürlich mit der gleichen Sorgfalt prüfen, wie es allgemein für Informationen aus Internetquellen gilt.

Als vor wenigen Jahrzehnten das Lexikon noch als Quelle der Wahrheit diente, waren Fragen wie zum Beispiel die nach dem längsten Fluss der Welt für den Normalverbraucher nur damit zu beantworten. Spätestens als Wikipedia auf den Plan getreten war, ging die Diskussion um die Wahrheit in die nächste Runde. Seit einiger Zeit geht noch der Ausdruck Fake News durch die Medien.

Auch bei ChatGPT ist nicht jede Antwort faktisch richtig. Es ist eben erst mal einfach nur ein Sprachmodell. Und zwar ein Sprachmodell, das sich beeinflussen lässt. Die folgenden Parameter finden sich allesamt im OpenAI Playground (siehe ix.de/zy4y) auf der rechten Seite neben dem Eingabefenster. Um diese Werte bei ChatGPT zu ändern, muss man sie im Prompt ansprechen. So lässt sich die Wortzahl der Antwort gut steuern, manche der Optionen funktionieren jedoch nicht. Beim Verwenden der aktuellen OpenAI-API lassen sich die Parameter ebenfalls beeinflussen.

Im Feld „Modus“ lassen sich drei Varianten unterscheiden. „Complete“ ist die Standardeinstellung, hiermit lässt sich Text ergänzen. Mit „Insert“ lassen sich inmitten eines bestehenden Textes weitere Wörter einfügen. Durch „Edit“ lässt sich ein bestehender Text ändern. Zum Beispiel können Nutzer aus einem Dictionary, wie man es von Python her kennt, einen ausformulierten Text generieren.

Bei Modell steht standardmäßig „text-davinci-003“. Das ist das aktuelle Sprachmodell. Es lässt sich jedoch auch auf die Codex-Serie umschalten, zum Beispiel auf code-davinci-002. Die Antwort aus dem reinen Sprachmodell ist etwas ausführlicher.

Durch den Prompt „Bitte zeige mir, wie man in JavaScript eine Funktion schreibt, die die Wörter in einem Text rückwärts schreibt“ generiert das GPT-Modell auf Basis von text-davinci-003 das Ergebnis im Listing.

Listing: GPT-generierte JavaScript-Funktion zum Rückwärtsschreiben

```
function reverseWords(text) {
  // Der Text wird in ein Array aufgeteilt, indem die Wörter durch Leerzeichen getrennt werden.
  let wordsArray = text.split(" ");
// Für jedes Wort im Array wird die Reihenfolge der Buchstaben umgekehrt.
let reversedWordsArray = wordsArray.map(word => {
    return word.split("").reverse().join("");
});

  // Der Array mit den umgekehrten Wörtern wird erneut in einen String umgewandelt.
  return reversedWordsArray.join(" ");
}
```

Die Antwort von code-davinci-002 sieht etwas anders, etwas nüchterner aus:

```
function reverseWords(str) {
  return str.split(' ').map(function(word) {
    return word.split('').reverse().join('');
  }).join(' ');
}
```

GPT-Parameter: Temperatur, Wortzahl und Stoppsequenzen

Die Temperatur ist der Parameter, der über die Variabilität der Antworten entscheidet. Manche sagen auch, dass die Temperatur die Kreativität des Modells steuert. Vielleicht ist das ein wenig zu anthropomorph. Auf jeden Fall gilt, je höher die Temperatur ist, umso wahrscheinlicher ist es, dass das

Sprachmodell ein unerwartetes Ergebnis liefert. In einigen Fällen führt eine hohe Temperatur jedoch auch zu unbrauchbaren Ergebnissen. Umgekehrt erhöht eine niedrigere Präzision die Wahrscheinlichkeit, dass etwas Neues in der Antwort auftaucht.

Das lässt sich mit einem einfachen Beispiel verdeutlichen. Auf den Prompt „Was gibt es morgen zum Frühstück?“ antwortet GPT mit Temperatur = 0,0: „Das hängt davon ab, wo Sie frühstücken. In einem Restaurant oder Café können Sie die Speisekarte einsehen, um zu sehen, was es zum Frühstück gibt.“ Mit Temperatur = 0,7: „Das kommt ganz darauf an, wo Sie frühstücken. In einem Restaurant oder Café würden Sie eine Speisekarte mit Frühstücksoptionen vorfinden oder Sie könnten auch zu Hause frühstücken.“ Mit Temperatur = 1,0 antwortet die KI ausführlicher: „Das hängt ganz von Ihnen ab! Möglicherweise werden Sie ein klassisches Frühstück wie Eier, Speck, Toast, Würstchen, Getreide und Obst haben, oder Sie können eine ausgefallenere Mahlzeit wie ein süßes Omelett, einen Grünkohleintopf oder einen Smoothie zubereiten.“

Mit dem Parameter Maximum Length lässt sich die Länge einer Antwort sehr genau festlegen. Gibt man 200 Wörter ein, so ist die Ausgabe etwa 195 bis 205 Wörter lang. Damit lässt sich die Qualität der Antwort beeinflussen. Gemeinsam mit der Temperatur steuern diese Parameter, wie sich das Modell einsetzen lässt.

Über eine Stoppsequenz können Nutzer einen Abbruch des Antwortgenerierens veranlassen. Hier kann man bis zu vier Sequenzen eingeben, die dazu führen, dass das Programm die Antwort beendet. Ein Beispiel für eine Stoppsequenz wäre „Politikerin“. Fragt man etwa: „Wer ist Angela Merkel?“, so gibt das Programm eine typische erklärende Antwort. Mit der Stoppsequenz endet GPT direkt nach: „Angela Merkel ist eine deutsche“, da es den definierten Stopp erreicht.

GPT-Parameter: TopP, Strafen und Wahrscheinlichkeit

Um den Parameter TopP zu verstehen, muss man etwas tiefer in das Thema Suche einsteigen. Die Kontexthilfe allein gibt da etwas holprig Auskunft: „TopP steuert die Diversität über Nukleus-Sampling. Das bedeutet, dass die Hälfte aller wahrscheinlichkeitsgewichteten Optionen berücksichtigt werden.“ Übersetzt bedeutet das, dass das System bei einem TopP von 1,0 alle Token im Vokabular verwendet, während GPT bei einem Wert von 0,5 nur die 50 Prozent häufigsten Token in Betracht zieht.

Mit der „Frequency Penalty“ und der „Presence Penalty“ bestraft man das Modell bei Wiederholungen. Beide Parameter lassen sich zwischen den Werten null und zwei einstellen. Die Häufigkeitsstrafe tritt ein, wenn das nächste zu generierende Wort bereits mehrfach im Text vorgekommen ist. Die Präsenzstrafe verhindert, dass das Sprachmodell ein Wort, das bereits im Text vorkommt, erneut generiert.

Mit der Einstellung „Best of“ lassen sich serverseitig mehrere Antworten generieren, von denen das Programm dann die beste anbietet. Erhöht man den Parameter bis maximal 20, spielt das Programm entsprechend viele Antworten aus. Dabei ist zu beachten, dass bei einem Bezahlmodell jede Antwort abgerechnet wird.

Mit der Option „Show probabilities“ lässt sich per Mausklick die Wahrscheinlichkeit eines generierten Wortes anzeigen. Abbildung 11 verdeutlicht das am Prompt „Edward Munch war“.

Im OpenAI Playground lässt sich die Wahrscheinlichkeit der

generierten Wörter anzeigen. Interessierte können so nachvollziehen, wie es zu einer Ausgabe gekommen ist (Abb. 11). *OpenAI*

Was man von ChatGPT erwarten kann

Der Fortschritt der KI verläuft schleichend und in kleinen Wellen. Die Reaktion der Öffentlichkeit schwankt dabei zwischen Überraschung, Faszination, Ernüchterung und Enttäuschung. Das zeigen auch das jüngste Beispiel ChatGPT, das Konkurrenzsystem Bard und die Bing-Implementation. Die KI-Neuerungen können Angst vor einem allwissenden Monster auslösen, das über ein eigenes Bewusstsein verfügt und die Menschheit bedroht. Dazu kommt die Angst vor dem Verlust des eigenen Jobs, den die KI womöglich besser ausführen kann als ein Mensch.

Der Hype um ChatGPT ist dabei ein erfrischender KI-Moment, der den aktuellen Forschungsstand mal wieder in den Fokus rückt und zu einer guten Einschätzung führen kann, was aktuell geht und was nicht. Es zeigt sich auch, dass bei manchen Antworten noch ein wenig Vorsicht geboten ist. Nutzer müssen die Bots mit Sinn und Verstand verwenden – man nimmt ja auch keinen Mixer zum Staubsaugen. (pst@ix.de)

1. Quellen
2. Eine Sammlung von Quellen zu Sprachmodellen und deren Training sowie Links zu ChatGPT und dem OpenAI Playground finden sich unter ix.de/zy4y.

Dr. Gerhard Heinzerling

hat 1999 über die Frage, wie Wörter im Gehirn gespeichert sind, promoviert. Danach arbeitete er als SAP-Berater und ist heute im Bereich der Bilderkennung mittels KI bei der Firma Arineo angestellt.