DIE API-SICHERHEITS-LANDSCHAFT IM WANDEL

KIN LANE, API-EVANGELIST

WHITEPAPER

INHALTSÜBERSICHT

EINLEITUNG

In den letzten Jahren hat sich die API-Sicherheitslandschaft grundlegend gewandelt. Schuld daran sind nicht zuletzt die zunehmende Anzahl an Bedrohungen und der sich ändernde Lebenszyklus für die Implementierung, Verwaltung und Prüfung sowie den Betrieb von APIs. Während dieser Zeit lag der Fokus vor allem auf der Authentifizierung, Autorisierung, Ratenbegrenzung und anderen wesentlichen Features des API-Managements. API-Management-Lösungen bieten zwar wichtige Sicherheitstools, doch das Thema geht noch viel weiter. Dieses Whitepaper befasst sich mit der bisherigen und künftigen Situation sowie der schnellen Veränderung und Weiterentwicklung der API-Landschaft.

DIE AKTUELLE API-SICHERHEIT

Vieles, was wir über den Schutz von Webanwendungen wissen, gilt auch für APIs. Dennoch müssen wir unsere Sicherheitspraktiken weiterentwickeln, um die individuellen Anforderungen der API-Nutzung über mobile, voicebasierte und andere neue Anwendungen hinweg zu berücksichtigen. Anstatt uns darauf zu fokussieren, welchen Nutzen Websicherheitspraktiken für die API-Sicherheit haben könnten, sollten wir analysieren, wie sich diese Praktiken von den Anforderungen der API-Sicherheit unterscheiden. Beispielsweise handelt es sich bei API-Clients häufig um mobile Geräte und Anwendungen, während Website-Clients in erster Linie Browser und Suchmaschinen-Bots sind. Bei der API-Sicherheit geht es darum, autorisierten Benutzern einen Zugriff auf nützliche Daten, Inhalte, Medien, Algorithmen und andere digitale Ressourcen zu ermöglichen und gleichzeitig unerlaubte Benutzer fernzuhalten. APIs gehen über einfaches Web-Publishing hinaus und erlauben eine tiefergehende Interaktion mit Daten und Anwendungen, was ein neues Sicherheitskonzept erfordert.

In den letzten zehn Jahren ist das API-Management zu einem Synonym für API-Sicherheit geworden. Authentifizierung und Ratenbegrenzung gelten als grundlegende Sicherheitsfeatures beim API-Management und sorgen dafür, dass interne Gruppen, Partner und externe Entwickler sicher auf Ressourcen zugreifen können. In dieser dynamischen Bedrohungslandschaft müssen wir einerseits auf das etablierte API-Management und die vorhandenen Sicherheitspraktiken setzen und gleichzeitig unsere Tools entsprechend den individuellen API-Anforderungen erweitern.

Bei der API-Sicherheit geht es nicht einfach nur um ein paar Stationen im API-Lebenszyklus. Während heutige Systeme zunehmend abgekoppelt und durch Continuous-Deployment- und -Integration-Praktiken weiter verteilt werden, nehmen die Bedrohungen konstant zu. Aus einer Handvoll APIs und Webservices werden Tausende Microservices und ereignisbasierte Lösungen, die über viele Infrastrukturanbieter und Regionen rund um den Globus hinweg betrieben werden können. Die API-Sicherheit muss über das API-Management hinausgehen und sämtliche API-Prozesse berücksichtigen.

HEUTIGE BEDROHUNGEN

APIs sind heute vielen unterschiedlichen Bedrohungen ausgesetzt. Gleichzeitig sehen immer mehr Hacker API-basierte Digital-Transformation-Initiativen als eine ideale Gelegenheit, um sich Zugriff auf Unternehmensdaten, -anwendungen und -systeme zu verschaffen. Da APIs häufig in Webentwicklungsprojekten genutzt werden, betrachten viele IT-Administratoren diese Implementierungen als sicher, weil sie als Teil der regulären Prozesse verwaltet werden. Doch die Bereitstellung von APIs bietet Hackern eine neue attraktive Möglichkeit, auf sensible Daten zuzugreifen. Lassen Sie uns einen Blick auf ein paar weit verbreitete Arten von API-Angriffen werfen und uns darüber Gedanken machen, warum bestehende Sicherheitspraktiken dagegen nicht ankommen.

LOGIN-ANGRIFFE

Ein Login-Angriff beginnt typischerweise damit, dass ein Hacker die Umgebung gründlich ausspäht und anschließend Angriffe ausführt, um Anmeldesysteme zu umgehen oder auszuhebeln. Die API-Surface-Area für beliebte mobile Anwendungen wird oft mithilfe handelsüblicher Proxy-Software ausgearbeitet und als maschinenlesbares Konzept in GitHub veröffentlicht, damit Hacker die Infrastruktur einer Anwendung ausnutzen können. Mit diesem Konzept können Hacker API-Pfade untersuchen und nach potenziellen Zugriffspunkten (z. B. Anmeldedienste) für die API Ausschau halten. Interessant für Hacker ist etwa das Authentifizierungssystem, das eine oder mehrere Zugriffstechnologien einschließlich Standardanmeldung, OAuth-Token, API-Schlüssel oder eine Kombination davon (z. B. API-Schlüssel + Token) unterstützen kann. Nachdem der Hacker das Authentifizierungsschema ausgespäht hat, entwickelt er Angriffe, um den Service zu kompromittieren. Nach der Kompromittierung des Anmeldedienst untersucht der Angreifer die APIs weiterhin nach Schwachstellen. Hier wird deutlich, wie wichtig es ist, API-Angriffe vor und nach der Anmeldung besser zu verstehen.

BEISPIELE FÜR LOGIN-ANGRIFFE

Der IRS – die US-Bundessteuerbehörde – gab im Februar 2016 bekannt, dass die Konten von knapp 400.000 Steuerzahlern durch einen Angriff in der API „Get Transcript" kompromittiert worden waren. Die Hacker nutzten einen Brute-Force-Angriff in Kombination mit separat gesammelten persönlichen Informationen, um auf Steuererklärungen zuzugreifen. Neben Sozialversicherungsnummern konnten Hacker auch Gehalts- und andere sensible Informationen einsehen.

In einem anderen Fall kompromittierten Hacker die Cloud-API eines Mobilgeräteherstellers. Angeblich nutzten sie einen Brute-Force-Angriff, um die Passwörter bzw. Konten von Prominenten zu kompromittieren. Die Hacker luden persönliche Informationen auf den Konten herunter und veröffentlichten Nacktfotos, die später auf den Angriff zurückgeführt wurden.

Anmeldedienste bieten eine beliebte API-Angriffsfläche. In letzter Zeit wurden bei vielen Sicherheitsvorfällen Accountdaten wie Benutzernamen und Passwörter offengelegt, die die Benutzer wahrscheinlich auch für Unternehmenskonten verwendet haben. Einem kürzlich erschienenen HBR-Artikel zufolge haben Password-Guessing-Angriffe mit gestohlenen Passwörtern eine Erfolgsrate von 2 Prozent. Das mag vielleicht harmlos klingen, aber 2 Prozent von einer Million sind 20.000 – also 20.000 erfolgreich erratene Passwörter. Hacker können Botnets mieten, die so programmiert sind, dass sie API-Aufrufe versenden, um Benutzernamen und Passwörter in API-Anmeldediensten zu testen. Obwohl API-Management-Systeme ungültige Anmeldeversuche ablehnen, besitzen diese Systeme keine geeigneten Mechanismen, um Clients daran zu hindern, kontinuierlich neue Kombinationen auszuprobieren. Viele Hacker halten die Anfragerate unter der Ratenbegrenzung und ändern regelmäßig die IP-Adresse, sodass eine Kontrolle äußerst schwer ist. Erfolgreiche Versuche verlaufen oft unbemerkt.

Darüber hinaus können Hacker auch über Man-in-the-Middle-Angriffe API-Schlüssel oder Token stehlen, die für die Client-Authentifizierung genutzt werden. Dabei bringen sie Benutzer dazu, eine Verbindung zu einem kompromittierten System herzustellen, das dann den Token oder Schlüssel des Benutzers erfasst. Der Hacker legt die gestohlenen Anmeldedaten vor, um Zugriff auf API-Services zu erlangen. Da die richtigen Anmeldedaten durch den Client vorgelegt werden, können API-Management-Systeme solche Angriffe nicht erkennen.

API-DDoS-ANGRIFFE

Im Gegensatz zu massierten DDoS-Angriffen, die Schutzmechanismen außer Gefecht setzen, werden API-DDoS-Angriffe häufig durch mehrere Clients ausgeführt, die Datenverkehr versenden, um einen API-Service zu überlasten. Da jeder Hacker normale Datenmengen versendet, lassen sich diese Angriffe nur schwer identifizieren, ohne die aggregierte Traffic-Rate auf jedem einzelnen API-Service zu analysieren. Versierte Hacker können sogar Kontrollen zur Ratenbegrenzung erkennen und Traffic-Raten anpassen, um unter der Drosselungsgrenze zu bleiben und eine Erkennung zu vermeiden. Obwohl API-Management-Systeme eine Ratenbegrenzung nutzen, um die Aktivitäten einzelner Clients zu kontrollieren, sind diese Systeme in der Regel nicht in der Lage, die aggregierten Traffic-Raten mehrerer Clients einzusehen, um DDoS-Angriffe zu stoppen.

BEISPIELE FÜR API-DDoS-ANGRIFFE

Da DDoS-Angriffe den Zugriff stören, ohne Daten zu kompromittieren, melden Organisationen solche Angriffe in der Regel nicht. Beispielsweise hielt der IRS den Angriff, bei dem die Konten von Steuerzahlern kompromittiert worden waren, zunächst für einen DDoS-Angriff. Erst später wurde festgestellt, dass man es mit einem wesentlich schwerwiegenderen Datenleck zu tun hatte.

Ein Video-Streaming-Service veröffentlichte vor kurzem ein Dokument über einen internen Angriff, bei dem seine Microservices-Architektur genutzt wurde, um die Backend-Systeme zu kompromittieren. Die eingehende Anfrage löste eine Kettenreaktion mit zahlreichen weiteren Anfragen aus, die die Server in die Knie zwangen.

DDoS-Angriffe zielen unter anderem auf Anmeldeservices sowie Sitzungsverwaltungs- und viele andere Dienste ab, die wichtig für die Zuverlässigkeit einer Anwendung sind. Beispielsweise könnte eine Gruppe von Hackern gleichzeitig Login-Anfragen versenden und berechtigten Nutzern so eine Anmeldung in der API erschweren. Ähnliche Angriffe lassen sich auch auf Cookie-Management- oder Token-Servern ausführen. Zudem könnten Hacker einen Warenkorb oder einen anderen wichtigen API-Service mit betrügerischen Anfragen bombardieren, um die Serviceverfügbarkeit zu beeinträchtigen.

Neben der Beeinträchtigung von Anwendungen können DDoS-Angriffe auch hohe Computing-Kosten verursachen, insbesondere in Cloud-Umgebungen mit nutzungsbasierten Zahlungspraktiken. Viele Organisationen haben flexible Computing-Services eingerichtet, die in Spitzenzeiten anspringen. Ohne geeignete Kontrollen können DDoS-Angriffe den Kundenzugriff auf wichtige Services behindern und gleichzeitig die Computing-Kosten in die Höhe treiben.

ANGRIFFE AUF ANWENDUNGEN UND DATEN

Mit den richtigen Anmeldedaten können Insider und Hacker auf beliebige Systeme oder Daten zugreifen – einen zusätzlichen Aufwand, um die Sicherheit zu kompromittieren, müssen sie nicht betreiben. Offene Geschäftsmodelle und Digital-Transformation-Initiativen haben heute oft zur Folge, dass Partner über eigene Unternehmensanmeldedaten mithilfe von APIs auf Anwendungen und Daten zugreifen können. Allerdings werden Unternehmen in den seltensten Fällen informiert, wenn Partnermitarbeiter ihre Organisation verlassen. Das macht es für sie schwierig, zwischen berechtigtem und unberechtigtem API-Verkehr zu unterscheiden. Mit der zunehmenden Verbreitung von Phishing-Kits und Keyloggern sollten Organisationen außerdem stets im Hinterkopf behalten, dass Anmeldedaten nicht ausreichen, um ihre wichtigsten Ressourcen zu schützen. Da Angreifer mit kompromittierten Anmeldedaten wie berechtigte Clients aussehen, haben API-Management-Systeme keine Chance, kompromittierte Benutzer zu erkennen, die über die APIs auf Anwendungen zugreifen. Noch dazu herrscht in Unternehmen die Annahme, dass APIs, die für den internen Gebrauch entwickelt wurden, weniger anfällig für Angriffe von außen sind. Die zunehmende Verbreitung kompromittierter Anmeldedaten und der Mangel an zentralisierter API-Governance sorgen oft dafür, dass APIs, die nur für den internen Gebrauch bestimmt waren, auch außerhalb des Unternehmens eingesetzt werden.

BEISPIEL FÜR ANGRIFFE AUF ANWENDUNGEN UND DATEN

Eine beliebte Social-Media-Site gab im September 2017 bekannt, dass Informationen von über 6 Millionen Benutzerkonten durch eine Schwachstelle in ihrer API offengelegt worden waren. Obwohl Passwortdaten nicht kompromittiert wurden, konnten die Hacker auf die Kontaktinformationen einiger prominenter Kunden zugreifen.

ARTEN VON ANWENDUNGS- UND DATENANGRIFFEN

Anwendungs- und Datenschwachstellen hängen von der offengelegten API-Funktionalität ab. So ist eine API mit einer Read-only-Funktion wahrscheinlich nicht anfällig für einen Injection-Angriff. Allerdings verfügen APIs häufig über mehrere Funktionen. Hier einige Beispiele von Angriffen, die diese Funktionen nutzen:

- **Datenextraktion oder -diebstahl:** Hacker können einen Angriff so programmieren, dass Informationen nicht nur aus einem, sondern aus vielen Accounts gesammelt werden.
- **Datenlöschung oder -manipulation:** Ein verärgerter Mitarbeiter könnte Informationen löschen, um Systeme zu sabotieren. Ebenso könnte ein Hacker Daten ändern, um Informationen zu kompromittieren.
- **Einschleusung von Daten in einen Anwendungsservice:** Hacker könnten große Datendateien laden, um einen Überlauf des Systemspeichers zu verursachen, oder übermäßig viele Daten einschleusen, um einen API-Service zu überlasten.
- **Einschleusung von bösartigem Code:** Hacker könnten bösartigen Code wie einen Keylogger einschleusen, um andere Benutzer, die auf den Service zugreifen, zu kompromittieren.
- **Massive Anwendungsaktivität:** Hacker können Aufrufe generieren, die ungewöhnlich hohe Systemressourcen erfordern und die Antwortzeit des Servers beeinflussen.

Dies sind nur einige Beispiele von API-Schwachstellen, die durch kompromittierte Anmeldedaten ausgenutzt werden. Organisationen sollten ihre API-Management-Systemimplementierungen überprüfen und die API-Services identifizieren, bei denen sensible Informationen offengelegt werden könnten. Mindestens diese API-Services sollten über die grundlegende Zugriffskontrolle hinaus geschützt werden, die von API-Management-Systemen bereitgestellt wird. Befassen wir uns als Nächstes damit, was man beim Aufbau dieser Sicherheitsschicht berücksichtigen sollte.

API-SICHERHEIT DER ZUKUNFT

Künftig sollte der Fokus darauf liegen, intelligente API-Sicherheitsfunktionen fest in die API-Prozesse zu integrieren. Die IT- und Sicherheitsteams müssen agil sein, schnell reagieren und über die neuesten Informationen – sowohl aus internen als auch externen Quellen – verfügen. Werfen wir einen Blick darauf, wie das am besten gelingt.

VERHALTENSANALYSEN

Bevor es API-Management-Systeme gab, konnten Organisationen mit dem API-Verbraucherverhalten und der Ressourcennutzung kaum etwas anfangen. Jetzt haben Unternehmen ein umfassendes Verständnis darüber, welche Verbraucher auf ihre Ressourcen zugreifen und welche Verhaltensweisen sie aus technischer und betriebswirtschaftlicher Sicht fördern sollten. Durch API-Authentifizierung und die Zusammensetzung von Services lassen sich Verhaltensmodelle für API-Prozesse umsetzen. Diese Regeln legen fest, welche Gruppen zu welchen Kosten auf welche Ressourcen zugreifen können. Durch API-Management kann positives Verhalten behindert und gefördert werden, doch wie lässt sich das auf die Sicherheit übertragen? Durch API-Monitoring erhalten wir einen Einblick in die API-Nutzung, aber wir bekommen nur minimale Informationen über das Verhalten von Übeltätern. Wir haben keine Tools, um unbekannte Verhaltensmodelle zu identifizieren oder gar unsere Reaktion zu automatisieren und zu skalieren.

Das API-Management und -Monitoring sollte sicherheitsbezogene Verhaltensmodelle umfassen, da sich regelbasierte Zugriffskontrollmodelle für APIs nur schwer skalieren lassen. Organisationen sollten bestehende Informationsquellen wie Plattformprotokolle und externe Bedrohungsinformationen von Third-Party-Netzwerken nutzen, die Bedrohungsdaten austauschen. Nur so können sie einen umfassenden Einblick in das Echtzeitverhalten gewinnen und APIs vor Bedrohungen schützen, die auf Anbieter von API-Plattformen abzielen.

Unternehmen sollten einen API-Plattform-Ansatz intern entwickeln und schnell implementieren, um auf Bedrohungen reagieren zu können. Unsere Modelle müssen automatisiert werden, ohne dabei auf Mitarbeiter an den richtigen Stellen zu verzichten. Bei der API-Sicherheit geht es nicht darum, alles akribisch zu definieren und einzurichten. Vielmehr geht es darum, sämtliche Ressourcen im Rahmen eines gut durchdachten Konzepts mit geeigneten Identity- und Access-Management-Kontrollen ins Web zu stellen. Wichtig ist, einen klaren Überblick darüber zu haben, wer auf welche Ressourcen zugreifen kann, positive Verhaltensweisen zu fördern und negatives Verhalten schnell zu unterbinden.

API-ANALYSEN

Analysefunktionen wurden als Teil der API-Management-Suite fest in die API-Prozesse integriert. Künftig sollten Analysefunktionen an allen Punkten entlang des API-Lebenszyklus präsent sein, angefangen beim Definitions- und Designprozess über die Implementierung und Verwaltung bis hin zu Überwachung und Prüfung sowie allen anderen API-Prozessen während des Betriebs. Wir müssen die API-Analyse um robustere Funktionen erweitern, sodass sie mit anderen Initiativen rund um die API-Sicherheit eng verzahnt ist.

Die Analysefunktionen müssen mit den API-Sicherheitsprozessen abgestimmt sein und umfassende Einblicke in diese Prozesse ermöglichen. Wichtig ist auch eine Analyse der API-Sicherheitspraktiken und -modelle sowie der Umsetzung. Der Schwerpunkt bei der API-Analyse sollte in den kommenden Jahren darauf liegen, einen Einblick in Bedrohungen zu bieten und ein besseres Verständnis über die Sicherheit sowie den geschäftlichen Teil der API-Prozesse zu ermöglichen. Uns muss bewusst sein, dass das Thema API-Sicherheit niemals abgeschlossen ist, selbst nicht mit entsprechender Verschlüsselung, Authentifizierung und Ratenbegrenzung und einem Dashboard, das uns sagt, welche User und Anwendungen aktiv unsere APIs nutzen.

TESTEN

API-Serviceprovider bieten uns eine externe Perspektive, wenn es um die Überwachung von APIs geht. Beim Monitoring und Testen der einzelnen APIs wird ein einheitlicher Satz an Aussagen verwendet, die bestimmen, was eine API tun und nicht tun sollte. Durch das Testen der Verfügbarkeit, Integrität und Performance jeder API lassen sich Sicherheitslücken sowie Fehler im API-Code erkennen. Über Angriffsvektoren werden zusätzliche oder fehlerhafte Informationen eingeschleust, die außerhalb des geplanten Verhaltens eines API-Prozesses fallen. Da Kriminelle Fehler häufig ausnutzen, spielt das Testen von APIs eine wichtige Rolle für unsere API-Sicherheit.

Moderne Ansätze für das Testen von APIs erlauben strukturierte, wiederholbare, regelmäßige Tests, die anhand ereignisbasierter Muster erfolgen. Diese Tests lassen sich als Teil von Sicherheitsvereinbarungen nutzen, um sicherzustellen, dass APIs erwartungsgemäß im Rahmen der von den IT- und Sicherheitsteams definierten Sicherheitsrichtlinien funktionieren. API-Überwachung und Sicherheitstests gehen Hand in Hand. Mithilfe von API-Sicherheitstests können Sie die Performance und Integrität Ihrer API-Aufrufe analysieren. Eine geringe API-Performance ist oft das erste Warnsignal für einen Sicherheitsvorfall. Durch das Testen der Integrität von API-Anfragen und -Antworten lassen sich Sicherheitslücken identifizieren, bevor sie das gesamte System erfassen.

MODELLIERUNG

Durch maschinelles Lernen können Sie Modelle entwickeln, um positive Verhaltensweisen zu fördern und negatives Verhalten zu unterbinden. Zur Umsetzung dieser Modelle sind allerdings geeignete Tools und Services nötig. Entwickeln lassen sich diese Modelle auf Basis bestehender Protokollierungspraktiken, die Teil der API-Prozesse sind, oder durch die Integration externer Informationsquellen, um zusätzliche Bedrohungsmuster einzubeziehen, die nicht intern erfasst werden. Ein Vertrag zwischen API-Providern und Verbrauchern, der die vorgesehenen Anwendungsbereiche für eine bestimmte API definiert, ist sehr nützlich, da man so durch künstliche Intelligenz und maschinelles Lernen normale Verhaltensweisen modellieren und Anomalien ganz einfach identifizieren kann.

Neue API-Sicherheitsanbieter trainieren ihre Modelle über mehrere API-Provider und Datenbanken mit Bedrohungsinformationen hinweg. Nun liegt es an den API-Sicherheitsserviceprovidern, ausgereifte, weiterentwickelte Machine-Learning-Modelle zu erstellen und sich auf dem Markt durchzusetzen. Für API-Provider geht es darum herauszufinden, welche Sicherheitsanbieter die besten Modelle zur Ergänzung ihrer internen Sicherheitsplattformen bieten.

REPORTING

Viele Organisationen haben Schwierigkeiten damit, die enormen Mengen an Protokolldaten zu den API-Aktivitäten manuell zu überprüfen. Oft hinterlassen Angreifer Hinweise bzw. Spuren in den Protokolldaten, doch operative Teams haben häufig weder die Personalstärke noch die Kenntnisse, um diese Bedrohungen rechtzeitig zu erkennen und Angriffe abzuwehren. Machine-Learning-Systeme für die Analyse von API-Protokollen sind in der Lage, ungewöhnliche Aktivitäten, die auf einen Angriff hindeuten könnten, zu identifizieren. Diese Systeme können verdächtige Aktivitäten in forensischen Berichten zusammenfassen, die Sicherheitsanalysten anschließend nach Hinweisen auf Angriffe überprüfen können. Außerdem bieten diese Berichte umfassende Informationen zu API-Aktivitäten, die Organisationen in stark regulierten Branchen als Input für das Compliance-Reporting nutzen können.

COMPLIANCE

Die Corporate-Security-Compliance muss nicht Teil der gesetzlichen Compliance sein. Sie könnte organisationsweite, flexibel umsetzbare Sicherheitsrichtlinien umfassen. Ihr API-Sicherheitsteam sollte Branchenempfehlungen und Richtlinien des National Institute of Standards and Technology (NIST) für Best Practices beachten. Beim Thema API-Sicherheit ist die Einhaltung von unternehmensinternen und gesetzlichen Vorschriften manchmal sehr anspruchsvoll. Die Befolgung von Best Practices kann die Auswirkungen einer API-Sicherheitslücke entscheidend beeinflussen.

FAZIT

API-Sicherheitsprovider der nächsten Generation bieten Services und Tools, um API-Management und -Überwachung und andere operative Sicherheitspraktiken zu automatisieren, zu skalieren und zu ergänzen. Die neuesten API-Sicherheitsprovider auf dem Markt stellen neue Prozesse und Technologien bereit, um bestehende API-Implementierungen zu erweitern. Ein bedeutender Trend ist die Nutzung des maschinellen Lernens, um die riesigen Mengen an Protokolldaten zu analysieren und eine Verbindung zu API-Management-Lösungen herzustellen. So können Sicherheitsanalysten Bedrohungen aufdecken und stoppen, bevor es zu einem Vorfall kommt. Organisationen müssen auch in ihre internen Kapazitäten investieren, um mit API-Serviceprovidern zusammenzuarbeiten, deren Know-how effizient zu nutzen und ihnen Best Practices rund um die API-Sicherheit zur Verfügung zu stellen, mit denen sich Vorfälle in der Zukunft verhindern lassen.

Viele API-Serviceprovider setzen den Fokus für die API-Bereitstellung auf den API-Lebenszyklus Diese Provider nutzen API-Definitionen, um die API-Landschaft abzubilden, die API-Implementierung in beliebigen Infrastrukturen zu unterstützen und Continuous-Deployment-Workflows bereitzustellen. Viele Elemente werden containerisiert und abgekoppelt, wobei Sicherheitsaspekte über zentrale Tools und Services hinweg gemanagt werden.

Wer in 2020 zu den Gewinnern zählen will, muss das Thema API-Sicherheit proaktiv angehen. Jetzt ist ein denkbar ungünstiger Zeitpunkt, um Bedrohungen zu ignorieren. Der Markt für Daten ist derzeit lukrativ, leider auch für Hacker – Sie sollten also nicht den Fehler machen, dieses Thema auf die leichte Schulter zu nehmen.

#3352 (DE) | 11.18 | v04