Akamai

WHITEPAPER

Grundlagen der API-Sicherheit: Entwickeln Sie Ihr Wissen weiter und sichern Sie das Unternehmen

Einführung

APIs haben sich schnell von einem Detailaspekt der Implementierung zu einem strategischen Treiber für digitale Innovation entwickelt. Jedes Mal, wenn ein Kunde, ein Partner oder ein Lieferant digital mit einem Unternehmen in Kontakt tritt, arbeitet eine API hinter den Kulissen, um einen reibungslosen Datenaustausch zu ermöglichen.

Mit der zunehmenden Verbreitung von APIs nehmen auch die mit ihnen verbundenen Risiken zu. Im Wettlauf um die schnelle Erstellung und Veröffentlichung neuer Anwendungen und KI-unterstützter Services werden die zugrunde liegenden APIs oft falsch konfiguriert, verfügen nicht über ausreichende Sicherheitskontrollen und sind anfällig für leicht durchführbare Angriffe.

Die Folge: APIs haben sich zu einem bedeutenden Angriffsvektor entwickelt, und viele Sicherheitsteams haben Nachholbedarf, was ihre API-Sicherheitsstrategien betrifft. Aus diesem Grund wird die API-Sicherheit für IT- und Sicherheitsverantwortliche zusehends zu einer der wichtigsten strategischen Prioritäten.

Wenn Sie sich mit den Grundlagen der API-Sicherheit vertraut machen möchten oder eine Liste mit den wichtigsten zu beantwortenden Fragen zusammenstellen wollen, finden Sie in diesem Leitfaden die dafür erforderlichen Informationen. Im Einzelnen gehen wir auf die folgenden Themen ein:

- **Die verschiedenen Arten von APIs**
- **Welche Bedeutung hat API-Sicherheit heute für Unternehmen?**
- **Best Practices zur Abwehr von API-Sicherheitsrisiken**
- **Gängige API-Angriffs- und Missbrauchsmethoden**

Um direkt zu den Best Practices für die API-Sicherheit zu gelangen, können Sie auf Seite 10 fortfahren.

Inhaltsverzeichnis

API-Grundlagen	4–9
API-Sicherheit erklärt	10–12
API-Sicherheitsrisiken und API-Missbrauch	13–18
API-Sicherheitslösungen und -trends	19–22

API-Grundlagen

Was ist eine Web-API?

Eine Web-API (Web Application Programming Interface) besteht aus einem oder mehreren Endpunkten zu einem definierten, in der Regel in JSON oder XML ausgedrückten Anfrage-Antwort-Nachrichtensystem. Diese Endpunkte sind öffentlich über das Web zugänglich, meist über einen HTTP-basierten Webserver.

Mit anderen Worten: Eine Web-API ist das, woran die meisten Leute bei der Abkürzung „API“ denken. Es handelt sich um eine Sammlung von Endpunkten. Endpunkte bestehen aus Ressourcenpfaden, den Vorgängen, die mit diesen Ressourcen ausgeführt werden können, und der Definition der Ressourcendaten (in JSON, XML, Protobuf oder einem anderen Format).

Web-APIs unterscheiden sich von anderen APIs, beispielsweise solchen, die vom Betriebssystem oder von Bibliotheken für Anwendungen auf demselben Computer zugänglich gemacht werden. Der allgemeine Begriff „API“ bezieht sich jedoch in der Regel auf eine HTTP-basierte (Web-)API, insbesondere im Kontext der digitalen Transformation und der API-Sicherheit in Unternehmen.

Welches sind die häufigsten Arten von APIs?

Die folgende Tabelle enthält Begriffe, die sich auf verschiedene Nutzungsmodelle und technische Ansätze für API-Implementierungen beziehen. Web-APIs werden als HTTP-basierte APIs definiert. Die vier wichtigsten Arten von Web-APIs sind derzeit RESTful, SOAP, GraphQL und gRPC. In der Tabelle werden diese verbreiteten Arten und einige weitere definiert.

API-Nutzungsmodell	Beschreibung
Öffentliche API	Eine API, die allen Entwicklern kostenlos über das Internet zur Verfügung gestellt wird.
Externe API	Diese Arten von APIs werden häufig auch als öffentliche APIs bezeichnet. Sie sind über das Internet zugänglich.
Private API	Eine API, die in einem geschützten Rechenzentrum oder einer geschützten Cloud-Umgebung für die Nutzung durch vertrauenswürdige Entwickler implementiert wird.
Interne API	Wird häufig in derselben Bedeutung wie private APIs verwendet.
Drittanbieter-API	Bietet programmatischen Zugriff auf spezielle Funktionen und/oder Daten aus einer Drittanbieterquelle zur Verwendung in einer Anwendung.
Partner-API	Eine Art von Drittanbieter-API, die nur ausgewählten, autorisierten Geschäftspartnern zur Verfügung gestellt wird.
Authentifizierte API	Eine API, auf die nur Entwickler mit entsprechender Autorisierung zugreifen können (oder Angreifer, die sich unbefugt Zugang zu den Anmeldedaten verschafft haben).
Nicht authentifizierte API	Eine API, auf die programmgesteuert zugegriffen werden kann, ohne dass spezielle Anmeldeinformationen erforderlich sind.
HTTP-API	Eine API, die das Hypertext-Übertragungsprotokoll als Kommunikationsprotokoll für API-Aufrufe verwendet.

RESTful API	RESTful (Representational State Transfer) ist die häufigste Art von Web-APIs. Sie verwendet Nur-Text, HTML, XML, YAML oder JSON, um Daten bereitzustellen. RESTful APIs sind leicht von modernen Frontend-Frameworks (z. B. React und React Native) zu nutzen und erleichtern die Entwicklung von Webanwendungen und Apps. Sie sind mittlerweile der De-facto-Standard für alle Web-APIs, einschließlich jener, die für B2B verwendet werden.
GraphQL	GraphQL-APIs sind der neuere, von Facebook entwickelte Standard. Er ermöglicht Datenbankzugriff über einen einzelnen POST-Endpunkt (in der Regel /graphql). Er löst ein häufiges RESTful-API-Problem, das darin besteht, dass mehrere Aufrufe erforderlich sind, um eine einzelne UI-Seite auszufüllen.
SOAP	SOAP verwendet ausführliche XML (Extensible Markup Language) für RPCs (Remote Procedure Calls). Es ist nach wie vor in älteren APIs zu finden.
XML-RPC	XML-RPC ist eine Methode zur Durchführung von Prozeduraufrufen über das Internet, bei der eine Kombination aus XML zur Codierung und HTTP als Kommunikationsprotokoll verwendet wird.
gRPC	gRPC-APIs sind ein von Google entwickeltes, leistungsstarkes Binärprotokoll über HTTP/2.0, das hauptsächlich für die Ost-West-Kommunikation (im internen Netzwerk) verwendet wird.
OpenAPI	OpenAPI ist eine Beschreibungs- und Dokumentationsspezifikation für APIs. Es kann hilfreich sein zu wissen, dass sich der Begriff Swagger auf die ursprüngliche Spezifikation und OpenAPI auf den von der OpenAPI-Initiative entwickelten offenen Standard bezieht.

Was ist der Unterschied zwischen APIs und Endpunkten?

Oft verwendet man den Ausdruck „API“, meint aber eigentlich einen einzelnen API-Endpunkt. APIs – manchmal spricht man auch von Services oder API-Produkten – sind Sammlungen von Endpunkten, die eine Geschäftsfunktion erfüllen. Ein einzelner Endpunkt hingegen ist eine Ressource (oder ein Ressourcenpfad, der auch als „Uniform Resource Identifier“ oder kurz: „URI“ bezeichnet wird) nebst dem damit ausgeführten Vorgang (Erstellen, Lesen, Aktualisieren oder Löschen). In RESTful APIs werden Vorgänge normalerweise den HTTP-Methoden (POST, GET, PUT und DELETE zugeordnet).

Was ist eine Nord-Süd-API?

Hierbei handelt es sich um APIs, die ein Unternehmen für die Außenwelt zugänglich hält, hauptsächlich um Geschäfte mit seinen Geschäftspartnern zu tätigen. Dies wird als API-Exposition bezeichnet. Zum Beispiel:

> **Banken, die Open Banking nutzen, können ihre Daten beispielsweise über APIs anderen Fintech- oder Finanzdienstleistern zugänglich machen.**

> **Gesundheitsorganisationen können Versicherungsgesellschaften und anderen medizinischen Unternehmen über APIs den Zugriff auf Patientenakten ermöglichen.**

> **Unternehmen im Gastgewerbe können ihre Reservierungssysteme über APIs Reisebüros oder Aggregatoren zugänglich machen.**

APIs sind das Bindeglied, das es verschiedenen Unternehmen ermöglicht, Daten auszutauschen. Nord-Süd-APIs werden oft als sicher angesehen, da der Zugriff autorisiert und authentifiziert ist. In der Regel ist dies der am schnellsten wachsende API-Typ und zu ihm gehören auch die meisten APIs. Daher bietet er bei den meisten Unternehmen auch die größte Angriffsfläche.

Was ist eine Ost-West-API?

Dies sind APIs, die ein Unternehmen intern verwendet und die für niemanden außerhalb des Unternehmens zugänglich sein sollten. Diese APIs verbinden interne Anwendungen oder Geschäftseinheiten oder Abteilungen. Es ist möglich, dass ein Entwickler einen Fehler macht, der die Ost-West-APIs versehentlich zugänglich macht. Diese APIs dürfen nicht für externe Entitäten zugänglich sein oder ihnen bekannt werden. Zu Sicherheitsverletzungen kommt es jedoch, wenn Cyberkriminelle Ost-West-APIs finden, auf die über das Internet zugegriffen werden kann.

Was sind die Unterschiede zwischen B2C-APIs und B2B-APIs?

B2C-APIs (Business to Consumer) unterstützen Web- und mobile Anwendungen. Sie werden in der Regel von modernen Frontend-Clients genutzt, um authentifizierten Endnutzern Zugriff auf die Geschäftsfunktionen des Unternehmens zu ermöglichen.

B2B-APIs (Business to Business) stellt das Unternehmen anderen Organisationen zur Geschäftsabwicklung und manchmal für Mehrwertangebote an gemeinsame Kunden bereit.

B2B-APIs ermöglichen es Unternehmen, ihre Zusammenarbeit mit Lieferanten, Wiederverkäufern und anderen Partnern zu optimieren und das Kundenerlebnis zu verbessern.

Beispiele für B2B-APIs:

Da sich die Nutzer der APIs stark voneinander unterscheiden, gibt es auch unterschiedliche Sicherheitskontrollen, die für den Schutz dieser APIs zur Verfügung stehen. Bis vor Kurzem konzentrierte sich die Branche auf B2C-Anwendungsfälle, aber selbst hier lag der Schwerpunkt nicht auf der Sicherung von B2C-APIs, sondern auf der Sicherung von Webanwendungen. Die Sicherheitstools und -kontrollen, die üblicherweise zur Sicherung von B2C-Webanwendungen eingesetzt werden, bieten bestimmte Vorteile (z. B. Web Application Firewall [WAF]/Web Application and API Protection [WAAP]). Sie bieten jedoch nicht den erforderlichen Grad an Transparenz, Echtzeitüberwachung und Schutz, um B2C-APIs vor Angriffen zu schützen.

Der Schutz von B2B-APIs wird immer schwieriger. Diese APIs sind oft leichtere Ziele für Angreifer, da für sie häufig essenzielle Schutzmechanismen fehlen. Frühere API-Sicherheitstools gestatteten nur einen begrenzten Einblick in B2B-APIs und es war schwierig, APIs zu sichern, die den Massenzugriff auf Daten für gemeinsame Nutzer ermöglichten (wie dies beispielsweise beim Open Banking der Fall ist, bei dem Fintech-Unternehmen und Finanzinstitute Kundendaten einvernehmlich teilen). Doch neuere API-Sicherheitslösungen, die Verhaltensanalysen bieten und ungewöhnliche Aktivitäten erkennen können, lösen diese Probleme.

Was ist der Unterschied zwischen privaten APIs und öffentlichen APIs?

Private APIs, die manchmal auch als interne APIs bezeichnet werden, sind für die Nutzung durch Entwickler und Auftragnehmer des Unternehmens vorgesehen. Sie sind oft Teil einer SOA-Initiative (Service-Oriented Architecture) und sollen die interne Entwicklung optimieren, indem sie verschiedenen Abteilungen oder Geschäftseinheiten ermöglichen, effizient und effektiv auf die Daten des anderen zuzugreifen.

Im Gegensatz dazu sind öffentliche APIs, die auch als externe APIs bezeichnet werden, für Nutzer außerhalb des Unternehmens zugänglich. In ihrer extremsten Variante als offene APIs können sie von jedem frei genutzt werden. In jedem Fall erfordern sie jedoch eine strikte Verwaltung und eine gute Dokumentation, damit sie von Technikern außerhalb des Unternehmens verwendet werden können.

Man muss wissen, dass private APIs, auf die über das Internet zugegriffen werden kann, nicht im eigentlichen Sinne des Wortes „privat“ sind. Nehmen wir zum Beispiel die B2C-API von ACME, die nur von (intern von ACME-Ingenieuren entwickelten) ACME-Apps verwendet wird. Man könnte meinen, dies sei eine private API. Da aber der Traffic zu dieser API aus dem Internet (von außerhalb des Unternehmens) ankommt, ist diese API nicht wirklich privat. Sie wurde lediglich nicht für externe Zielgruppen veröffentlicht. Hacker greifen solche APIs regelmäßig an, indem sie Traffic abfangen und Apps rückentwickeln, um die entsprechenden APIs zu finden.

API-Sicherheit erklärt

Was ist API-Sicherheit?

API-Sicherheit ist eine Strategie, die das Ziel verfolgt, für alle APIs in einem Unternehmen Transparenz zu erreichen, strenge Tests durchzuführen und Schutz zu gewährleisten. Das schließt APIs ein, die für Anwendungen, Geschäftsprozesse und Cloud-Workloads unverzichtbar sind. Da jedoch neue interne und externe APIs schnell und in großer Zahl produziert werden, kann es schwierig sein, einen vollständiges Verständnis der gesamten API-Landschaft eines Unternehmens zu erlangen. Vielen Unternehmen fehlt der Überblick, wie viele APIs sie tatsächlich haben und welche APIs sensible Daten zurückgeben, wenn sie aufgerufen werden. Um API-Sicherheitsrisiken zu erkennen und abzuwehren, sind ausgereifte Sicherheitskontrollen erforderlich, die diese Art von Transparenz und Datenanalyse bieten. Zu den APIs, die geschützt werden müssen, gehören die folgenden:

- APIs, die Daten für Kunden oder Geschäftspartner leicht zugänglich machen
- APIs, die von Geschäftspartnern genutzt werden
- APIs, die intern implementiert und verwendet werden, um Anwendungsfunktionen und Daten auf standardisierte und skalierbare Weise für verschiedene Systeme und Nutzeroberflächen verfügbar zu machen

Eine effektive API-Sicherheitsstrategie muss systematische Verfahren zur Bewertung von Risiken und potenziellen Auswirkungen sowie geeignete Abwehrmaßnahmen umfassen. Der erste Schritt bei der Risikobewertung besteht in einer Bestandsaufnahme aller genehmigten und nicht genehmigten APIs, die vom Unternehmen veröffentlicht und verwendet werden. Diese Bestandsaufnahme sollte Attribute wie die folgenden enthalten:

- Datenklassifizierungen, bei denen zumindest zwischen „nicht sensiblen“, „sensiblen“ und „sehr sensiblen“ Daten unterschieden wird
- Risikoindikatoren wie API-Schwachstellen und Fehlkonfigurationen

Darüber hinaus müssen die Maßnahmen zur Herstellung von API-Transparenz und zur Minimierung von Risiken eine Vielzahl möglicher Bedrohungen berücksichtigen. Das schließt beispielsweise folgende Aufgaben ein:

- Erkennung und Verhinderung der Verwendung nicht genehmigter „Shadow-APIs“ (siehe Spalte rechts)
- Identifizierung und Behebung von API-Schwachstellen und Fehlkonfigurationen, die Cyberkriminelle ausnutzen könnten
- Vermeidung von illegitimer API-Nutzung wie Missbrauch von Geschäftslogik und Daten-Scraping

Inwiefern unterscheidet sich die API-Sicherheit von der Anwendungssicherheit?

API-Sicherheit und herkömmliche Anwendungssicherheit hängen zwar miteinander zusammen, doch die API-Sicherheit stellt eine besondere Herausforderung dar. Dafür gibt es zwei zentrale Gründe: die Dimension und die Komplexität des Problems.

Größerer Umfang

Drei Faktoren tragen zum rasanten Anstieg der API-Nutzung bei:

1. Die Nutzung von Microservices, einer Architektur, die die Verwendung von APIs für die Service-to-Service-Kommunikation verlangt, nimmt zu.
2. Im Direktnutzerkanal verwenden moderne Frontend-Anwendungsframeworks wie React, Angular und Vue APIs und verdrängen ältere Webanwendungen.
3. Man fügt APIs hinzu, um auch ganz neuen Kanälen gerecht zu werden (z. B. Partner, IoT und Geschäftsautomatisierung).

Flexibilität führt zu Komplexität

Im Gegensatz zu Webanwendungen sind APIs so konzipiert, dass sie auf viele verschiedene Arten programmgesteuert verwendet werden können. Das macht die Unterscheidung zwischen legitimer Nutzung einerseits und Angriffen und Missbrauch andererseits äußerst schwierig.

Gibt es API-Begriffe, die Sicherheitsteams verstehen müssen?

Im Folgenden finden Sie allgemeine Kategorisierungen und Beschreibungen von APIs, die in einem Sicherheitskontext vorkommen können.

Genehmigte APIs

Veröffentlichte API
(mit Swagger-Dokumentation o. Ä.)

Nicht genehmigte APIs

- Shadow-API
- Nicht autorisierte API
- Zombie-API
- Verborgene API

Veraltete APIs

- Außer Betrieb genommene API
- Ältere API
- Zombie-API
- Verwaiste API

Was sind die Best Practices zum Schutz von APIs?

Die Verbesserung der API-Sicherheit beginnt mit den folgenden Best Practices:

- Integrieren Sie API-Sicherheitsstandards und -verfahren in den Softwareentwicklungszyklus Ihres Unternehmens.
- Integrieren Sie API-Dokumentation und automatisierte Sicherheitstests in Ihre CI/CD-Pipelines (kontinuierliche Integration/kontinuierliche Bereitstellung).
- Stellen Sie sicher, dass geeignete und effektive Authentifizierungs- und Autorisierungskontrollen für Ihre APIs angewendet werden.
- Implementieren Sie Maßnahmen zur Ratenbeschränkung, um zu verhindern, dass APIs missbraucht oder überlastet werden.
- Verstärken Sie die Ratenbeschränkung und andere Maßnahmen auf Anwendungsebene mit speziellen Gateways und/oder Netzwerken zur Inhaltsbereitstellung, um das Risiko von DDoS-Angriffen (Distributed Denial of Service) zu minimieren.
- Machen Sie API-Sicherheitstests zu einem integralen Bestandteil Ihrer allgemeinen Testprozesse für Anwendungen.
- Führen Sie eine kontinuierliche Erkennung von APIs durch.
- Implementieren Sie einen systematischen Ansatz zur Erkennung und Behebung häufiger API-Schwachstellen, einschließlich der Top 10 API-Sicherheitsrisiken von OWASP.
- Verwenden Sie signaturbasierte Bedrohungserkennung und -prävention als Basisschutz gegen bekannte API-Angriffe.
- Erweitern Sie die signaturbasierte Erkennung durch KI und Verhaltensanalysen, um die Erkennung von API-Bedrohungen besser skalierbar, genauer, geschäftsrelevanter und widerstandsfähiger gegen neue Bedrohungen zu machen.
- Stellen Sie sicher, dass der Prozess der Überwachung und Analyse der API-Sicherheit sich über mehrere Wochen und mehrere API-Sitzungen erstreckt.
- Ergänzen Sie die API-Sicherheitsüberwachung und -Warnmeldungen durch bedarfsgesteuerten Zugriff auf API-Bestands- und Aktivitätsdaten, die von Threat Huntern, Entwicklern, DevOps und Supportmitarbeitern verwendet werden können.

Ihre Fähigkeit, diese Best Practices für die API-Sicherheit zu implementieren, hängt davon ab, an welchem Punkt der Entwicklung einer ausgereiften API-Sicherheitsstrategie Sie sich befinden (siehe Spalte rechts).

Stufen des API-Sicherheitsreifegrads

Phase 1: Transparenz und Erkennung

Sie sind dabei, alle Ihre APIs und die von ihnen unterstützten Microservices mithilfe eines automatisierten Ansatzes zu ermitteln. Die Breite der Abdeckung ist von entscheidender Bedeutung, da übersehene APIs (z. B. nicht mehr verwendete APIs) ein vorrangiges Ziel für Cyberkriminelle sind.

Phase 2: Tests

Sie testen alle Ihre APIs, um sicherzustellen, dass ihr Code korrekt ist und sie ihre vorgesehene Funktion erfüllen. Tests, die vor der Bereitstellung einer API durchgeführt werden, sind das obere Ende dieser Reifegrad-Stufe. Risiken werden eliminiert, bevor die API in die Produktion geht, und gegebenenfalls erforderliche Fehlerbehebungen sind um ein Vielfaches kostengünstiger.

Phase 3: Risiko-Audit

Sie prüfen Ihre gesamte API-Umgebung kontinuierlich, um falsch konfigurierte APIs oder andere Fehler zu identifizieren. Ihr Audit gewährleistet außerdem eine angemessene Dokumentation jeder API und ermittelt, ob die APIs sensible Daten enthalten und ob sie über geeignete Sicherheitskontrollen verfügen.

Phase 4: Laufzeitschutz

Sie verwenden eine Lösung mit automatisiertem Laufzeitschutz, die zwischen normaler und ungewöhnlicher API-Aktivität unterscheiden kann. Durch diese Überwachung von API-Interaktionen können Sie in Echtzeit Verhaltensweisen erkennen, die auf eine Bedrohung hinweisen.

Phase 5: Reaktion

Sie verfügen über Lösungen, die auf verdächtiges API-Verhalten reagieren, wie z. B. eine WAF oder ein API-Gateway, das verdächtigen Traffic blockiert, bevor er auf wichtige Ressourcen zugreifen kann. Ihre Lösungen verwenden nutzerdefinierte, automatisierte Regeln.

Phase 6: Suche nach Bedrohungen

Sie führen regelmäßig forensische Analysen vergangener Bedrohungsdaten durch, um zu erfahren, ob Warnungen Bedrohungen korrekt erkannt haben und ob Muster aufgetreten sind, die eine proaktive Bedrohungssuche mithilfe einer Kombination aus ausgeklügelten Tools und menschlicher Intelligenz ermöglichen.

API-Sicherheitsrisiken und API-Missbrauch

Was ist eine API-Schwachstelle?

Eine API-Schwachstelle ist ein Software- oder Systemkonfigurationsfehler, den ein Angreifer ausnutzen kann, um auf sensible Anwendungsfunktionen oder Daten zuzugreifen oder eine API anderweitig zu missbrauchen. Die OWASP Top 10 der API-Sicherheitsrisiken bieten einen nützlichen Überblick über einige der am häufigsten missbrauchten API-Schwachstellen, die Unternehmen identifizieren und beheben sollten.

Werden alle API-Schwachstellen der OWASP Top 10 API-Sicherheitsrisiken berücksichtigt?

Die OWASP API Security Top 10 eignen sich hervorragend als Ausgangspunkt für Unternehmen, die ihre API-Sicherheit verbessern möchten. Ihre Kategorien decken ein breites Spektrum möglicher API-Risiken ab. Die in den OWASP API Security Top 10 enthaltenen Kategorien sind recht weit gefasst. Daher ist es wichtig, sich auch die jeweiligen Unterbereiche genau anzusehen. API-Angreifer versuchen häufig, Autorisierungsprobleme auszunutzen (die von OWASP umfassend behandelt werden). Doch es gibt auch API-Risiken, die vollständig außerhalb der OWASP API Security Top 10 liegen. Das gilt zum Beispiel für die Ausnutzung von Logikfehlern.

Wie können APIs missbraucht werden?

APIs können auf verschiedene Arten angegriffen und missbraucht werden. Einige der häufigsten Varianten sind:

- **Ausnutzung von Schwachstellen:** Technische Sicherheitslücken in der zugrunde liegenden Infrastruktur können zu einer Gefährdung der Server führen. Beispiele reichen von den Sicherheitslücken in Apache Struts (CVE-2017-9791, CVE-2018-11776) bis hin zu den Sicherheitslücken in Log4j (CVE-2021-44228).
- **Missbrauch von Geschäftslogik:** Logikmissbrauch liegt vor, wenn ein Angreifer Fehler im Anwendungsdesign oder in der Implementierung ausnutzt, um unerwartetes und nicht genehmigtes Verhalten zu provozieren. Diese Szenarien verursachen Stress für CISOs und ihre Teams, da veraltete Sicherheitskontrollen hier nutzlos sind.
- **Nicht autorisierter Datenzugriff:** Eine weitere verbreitete Form des API-Missbrauchs ist die Ausnutzung defekter Autorisierungsmechanismen für den Zugriff auf Daten, die eigentlich unzugänglich sein sollten. Für diese Schwachstellen gibt es viele Bezeichnungen wie etwa „fehlerhafte Autorisierung auf Objektebene“ (Broken Object-Level Authorization: BOLA), „unsicherer direkter Objektverweis“ (Insecure Direct Object Reference: IDOR) und „fehlerhafte Autorisierung auf Funktionsebene“ (Broken Function-Level Authorization: BFLA).

- **Kontoübernahme:** Nach einem Diebstahl von Anmeldedaten oder gar einem XSS-Angriff (Cross-Site Scripting) kann ein Konto übernommen werden. Sobald dies geschieht, lassen sich selbst hervorragend programmierte und perfekt gesicherte APIs missbrauchen. Mit einer API-Sicherheitslösung, die Verhaltensanalysen bietet, können Sie authentifizierte Aktivitäten von unrechtmäßiger Nutzung unterscheiden.
- **Daten-Scraping:** Wenn Unternehmen Datensätze über öffentliche APIs zur Verfügung stellen, können Cyberkriminelle diese Ressourcen aggressiv abfragen, um große, wertvolle Datensätze komplett zu erfassen.
- **Business Denial of Service (DoS):** Wenn API-Angreifer oder -Nutzer das Backend auffordern, umfangreiche Aufgaben auszuführen, können sie auf Anwendungsebene eine Service-Erosion oder einen kompletten Denial of Service verursachen (eine häufige Schwachstelle in GraphQL, die allerdings bei jeder ressourcenintensiven API-Endpunktimplementierung auftreten kann).

Was ist eine Zombie-API?

Aufgrund der sich verändernden Markt- und Geschäftsanforderungen unterliegen APIs einem ständigen Wandel. Da neue Endpoint-Implementierungen herausgebracht werden, um neue Geschäftsanforderungen zu erfüllen, Bugs zu beheben und technische Verbesserungen einzuführen, verschwinden ältere Versionen dieser Endpunkte. Die Steuerung des Stilllegungsprozesses alter Endpunkte ist keine triviale Aufgabe. Häufig bleiben veraltete Endpoint-Implementierungen aktiv und zugänglich. Man spricht hier auch von Zombie-Endpunkten.

Wie finde ich die verschiedenen Arten von Shadow-APIs?

Eine Möglichkeit zur unternehmensweiten Erkennung von Shadow-APIs besteht darin, den API-Traffic in Ihrem Netzwerk zu erfassen und zu analysieren. Beispiele für Quellen von API-Traffic:

Sobald Rohdaten aus allen verfügbaren Quellen erfasst sind, können KI-Techniken verwendet werden, um sie in einen umfassenden Bestandsüberblick über alle APIs, Endpunkte und Parameter umzuwandeln. Anschließend können zusätzliche Analysen durchgeführt werden, um diese Elemente zu klassifizieren und Shadow-APIs zu identifizieren, die entfernt oder in formelle Governance-Prozesse integriert werden müssen.

Wie schützt man interne APIs und B2B-APIs?

Die Antwort hängt von der Definition des Wortes „intern“ ab. Manche Teams bezeichnen APIs, die über das Internet für die Webanwendungen oder Apps des eigenen Unternehmens zugänglich sind, als „interne APIs“. Es mag sein, dass die Dokumentation für diese APIs in der Tat nur für Mitarbeiter und Auftragnehmer des Unternehmens zugänglich ist. Doch mittlerweile sind Hacker versiert darin, Apps zu analysieren und die APIs über Toolkits für die Demontage von Apps und Proxys wie Burp Suite zu rekonstruieren.

Wenn „interne APIs“ jedoch als Ost-West-APIs definiert sind, auf die von außerhalb des Unternehmens nicht zugegriffen werden kann, bleibt als Hauptgefahr nur noch die Bedrohung durch Insider. Schützen Sie Ost-West-APIs und Ihre B2B-APIs wie die meisten anderen APIs: Beginnen Sie mit der Sicherung des Softwareentwicklungszyklus (Software Development Life Cycle, SDLC) und stellen Sie dann sicher, dass der Zugriff nur mit Authentifizierung und Autorisierung erfolgt. Sie können auch die Verwaltung von Kontingenten, Ratenbeschränkungen und Maßnahmen bei plötzlichen Traffic-Spitzen (Spike Arrests) implementieren. Darüber hinaus können Sie Ihre APIs mit WAFs/WAAPs vor bekannten Bedrohungen schützen. Bei B2B-APIs sollten Sie strenge Authentifizierungsmechanismen wie mTLS hinzufügen, da die Transaktionen ihrem Wesen nach anfällig und häufig sperrig sind.

Sowohl für Ost-West- als auch für B2B-APIs empfehlen wir, Verhaltensanalysen zu verwenden. Das gilt besonders dann, wenn viele Entitäten beteiligt sind, was die Unterscheidung zwischen legitimem und illegitimem Verhalten erschweren kann. Zum Beispiel:

> **Woher können Sie wissen, ob die API-Anmeldedaten eines bestimmten Nutzers kompromittiert wurden?**

> **Woher können Sie wissen, ob Ihre Rechnungs-API von einem Nutzer missbraucht wird, der Rechnungsnummern auflistet, um Kontodaten zu stehlen?**

Der Schutz von B2B-APIs und Ost-West-APIs erfordert Geschäftskontext, der nicht allein durch die Analyse technischer Elemente wie IP-Adressen und API-Token gewonnen werden kann. Die Verwendung von maschinellem Lernen und Verhaltensanalysen zur Gewinnung von Einblicken in geschäftsrelevante Entitäten ist die einzige Möglichkeit, Risiken effektiv zu verstehen und zu verwalten. Geschäftskontext- und Verlaufsdaten-Benchmarks für die normale Verwendung von APIs durch bestimmte Entitäten wie Ihre Nutzer, Partner oder auch Geschäftsprozesseinheiten (Rechnung, Zahlung, Bestellung usw.) ermöglichen es, Anomalien zu erkennen, die andernfalls nicht entdeckt würden.

Bieten API-Gateways ausreichenden Schutz vor Risiken?

Viele Unternehmen, die einen strategischen Ansatz für APIs verfolgen, verwenden API-Gateways. Die meisten API-Gateways verfügen über umfassende integrierte Sicherheitsfunktionen, von denen Unternehmen Gebrauch machen sollten. An erster Stelle ist hier die Authentifizierung zu nennen (und auch die Autorisierung, wenn Sie OpenID Connect nutzen können). Authentifizierung, Autorisierung und Kontingentverwaltung am API-Gateway reichen jedoch aus mehreren Gründen nicht aus:

Die Erkennungslücke bei API-Gateways: API-Gateways bieten nur Transparenz und Kontrolle der APIs, für deren Verwaltung sie konfiguriert sind, sodass sie bei der Erkennung von Shadow-APIs und -Endpunkten nicht effektiv sind.

Die Sicherheitslücke von API-Gateways: API-Gateways können Authentifizierungs- und bis zu einem gewissen Grad die Autorisierungsschemas durchsetzen. Sie prüfen jedoch keine Payloads (wie WAFs und WAAPs es tun), noch erstellen sie Verhaltensprofile, um Missbrauch zu erkennen.

Was sind die häufigsten Fehler bei der API-Konfiguration?

Die Zahl möglicher API-Fehlkonfigurationen ist nahezu endlos, da APIs auf so vielfältige Weise verwendet werden. Es gibt jedoch einige besonders verbreitete Szenarien, was Fehlkonfigurationen betrifft:

Fehlerhafte oder keine Authentifizierung

Die Authentifizierung ist grundlegend für den Schutz sensibler Daten, die über APIs zur Verfügung gestellt werden. Im ersten Schritt muss sichergestellt werden, dass für alle APIs, die sensible Daten enthalten, zunächst eine Authentifizierungslösung eingerichtet ist. Es ist aber auch wichtig, Authentifizierungsmechanismen durch Ratenbeschränkung vor Brute-Force-Angriffen, Credential Stuffing und der Verwendung gestohlener Authentifizierungstoken zu schützen. Manchmal können Fehlkonfigurationen auftreten, die es API-Nutzern ermöglichen, Authentifizierungsmechanismen zu umgehen. Häufig geschieht das im Zusammenhang mit der Tokenverwaltung (exemplarisch zu nennen sind hier einige berüchtigte Probleme bei der JWT-Validierung oder die Nichtüberprüfung des Tokenbereichs).

Fehlerhafte Autorisierung

Eine der häufigsten Anwendungen von APIs ist der Zugriff auf Daten oder Inhalte, einschließlich vertraulicher Informationen. Autorisierung ist der Prozess, bei dem vor einem Zugriff auf bestimmte Daten überprüft wird, ob der betreffende API-Nutzer berechtigt ist, auf diese Daten zuzugreifen. Dies kann auf Objekt- oder Ressourcenebene erfolgen (z. B. kann ich auf meine Bestellungen zugreifen, aber nicht auf die von anderen Personen) oder auf Funktionsebene (wie dies häufig bei administrativen Funktionen der Fall ist). Aufgrund der hohen Anzahl von Edge-Fällen und -Bedingungen und der verschiedenen Flows, die API-Aufrufe zwischen Microservices durchlaufen können, ist es nicht leicht, bei der Autorisierung die richtige Lösung zu finden. Wenn Sie keine zentrale Autorisierungs-Engine haben, weist Ihre API-Implementierung wahrscheinlich einige der möglichen Schwachstellen auf, zum Beispiel BOLA und BFLA.

Fehlerhafte Sicherheitskonfiguration

Zusätzlich zu den oben erwähnten Authentifizierungs- und Autorisierungsproblemen gibt es viele mögliche Arten fehlerhafter Sicherheitskonfigurationen. Dazu gehören unsichere Kommunikation (z. B. Nichtverwendung von SSL/TLS oder Verwendung anfälliger Verschlüsselungssammlungen), ungeschützte Cloud-Speicher und allzu großzügige CORS-Richtlinien (Cross-Origin Resource Sharing).

Mangel an Ressourcen und Ratenbeschränkung

Wenn APIs implementiert werden, ohne dass die Anzahl der Aufrufe, die API-Nutzer tätigen können, begrenzt wird, können Cyberkriminelle die Systemressourcen überlasten, was zu einer Verschlechterung des Service oder einem kompletten Denial of Service (DoS) führen kann. Zumindest müssen Ratenbeschränkungen für den Zugriff auf nicht authentifizierte Endpunkte durchgesetzt werden, wobei Authentifizierungs-Endpunkte sehr wichtig sind. Andernfalls wird es unweigerlich zu Brute-Force-Attacken sowie Credential-Stuffing- und Credential-Validation-Angriffen kommen.

Was sind API-Angriffe?

API-Angriffe sind Versuche, APIs für schädliche oder anderweitig unzulässige Zwecke zu verwenden. API-Angriffe können in verschiedenen Formen vorkommen, darunter:

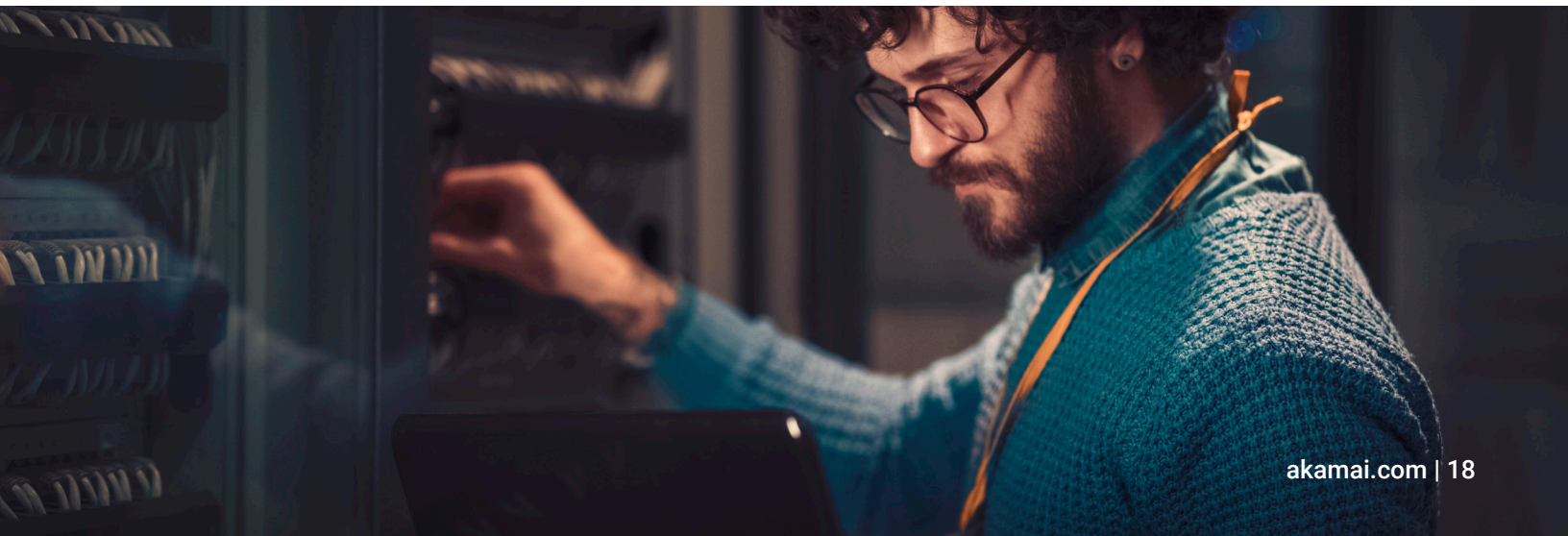
- Ausnutzen technischer Schwachstellen in API-Implementierungen
- Verwenden gestohlener Anmeldedaten und anderer Methoden zur Kontoübernahme, um sich als legitimer Nutzer zu tarnen
- Missbrauch der Geschäftslogik, der das Ausnutzen von APIs auf unerwartete Weise ermöglicht

Was ist Credential Stuffing für APIs?

Die unbeabsichtigte Offenlegung von Nutzer-ID- und Passwortinformationen von Websites und SaaS-Plattformen (Software-as-a-Service) ist mittlerweile keine Seltenheit mehr. Häufig führen solche Vorfälle dazu, dass große Mengen von Anmeldedaten online verbreitet werden. Credential Stuffing bezeichnet die Verwendung von Authentifizierungsdaten, die von zuvor angegriffenen Leak-Websites stammen und für automatisierte Anmeldeversuche auf anderen Websites genutzt werden. Diese Technik basiert auf der Prämisse, dass ein bestimmter Prozentsatz der Nutzer dieselben Anmeldedaten für mehrere Websites verwendet. Angreifer nehmen zunehmend direkt die APIs und deren Authentifizierungsmechanismen ins Visier. So können sie den Angriff leichter automatisieren, da APIs für eine einfachere Nutzung erstellt werden.

Was ist Datenextraktion über APIs?

Im Erfolgsfall ist Datenextraktion ein häufiges Resultat von API-Angriffen und -Missbrauch. In manchen Fällen bezieht sie sich auf hochsensible, nicht öffentliche Informationen, die von Cyberkriminellen durch einen API-Angriff gestohlen wurden. Sie kann jedoch auch für weniger schwere Arten von API-Missbrauch angewendet werden. Ein Beispiel dafür wären aggressive Daten-Scrapings von öffentlich verfügbaren Daten, um große Datensätze zu erstellen, die in aggregierter Form wertvoll sind.



API-Sicherheitslösungen und -trends

Was sind die neuesten Trends bei der API-Sicherheit?

Im Folgenden beschreiben wir die wichtigsten Trends, die Sicherheitsexperten bei der Entwicklung einer API-Sicherheitsstrategie berücksichtigen sollten:

Verhaltensanalysen und Erkennung von Anomalien: Anstatt mögliche Angriffe vorherzusagen und sich zur Risikominderung nur auf signaturbasierte Erkennung und vordefinierte Richtlinien (z. B. WAFs) zu verlassen, setzen Unternehmen zunehmend auch auf maschinelles Lernen und Verhaltensanalysen, um API-Aktivitäten in einem Geschäftskontext zu beobachten und Anomalien zu erkennen.

Übergang von On-Premise zu SaaS: Während viele API-Sicherheitsprodukte der ersten Generation lokal bereitgestellt wurden, erfreuen sich SaaS-basierte Ansätze aufgrund ihrer Geschwindigkeit, ihrer einfachen Bereitstellung und ihrer Fähigkeit, die Leistungsfähigkeit von maschinellem Lernen in großem Umfang zu nutzen, wachsender Beliebtheit.

Analyse größerer Zeitfenster: API-Sicherheitsansätze, die nur einzelne API-Aufrufe oder kurzfristige Sitzungsaktivitäten analysieren, werden durch Plattformen ersetzt, die API-Aktivitäten über Tage und manchmal Wochen hinweg analysieren können. Das Spektrum reicht hier von der grundlegenden automatischen Optimierung der WAF-Richtlinien bis hin zur Durchführung von Verhaltensanalysen und der Erkennung von Anomalien.

DevSecOps – Einbeziehung nicht sicherheitsrelevanter Stakeholder: Eine der besten Möglichkeiten, API-Risiken zu minimieren, besteht darin, stärkere Verknüpfungen zwischen API-Sicherheitsstrategien und -tools und den Entwicklern und Systemen zu schaffen, die an der Erstellung, Implementierung und Konfiguration von APIs beteiligt sind.

API-aktivierte API-Sicherheit: Erkennung und Abwehr aktiver API-Angriffe und -Missbrauchsversuche sind sehr wichtig. Daneben suchen vorausschauend denkende Unternehmen nach Wegen, wie sie den bedarfsgesteuerten Zugriff auf API-Sicherheitsdaten und -erkenntnisse nutzen können, um die Bedrohungssuche, die Reaktion auf Vorfälle und die API-Entwicklung zu verbessern.

Was ist signaturbasierte API-Sicherheit?

Signaturbasierte API-Sicherheitstechniken suchen nach bekannten Angriffsmerkmalen und -mustern. Sie generieren dann Sicherheitswarnungen und andere automatisierte Reaktionen, wenn Übereinstimmungen beobachtet werden. Der Vorteil: Wenn also ein Unternehmen über eingehenden API-Traffic informiert wird, der kompromittiert ist oder ungewöhnliches Verhalten zeigt, kann es signaturbasierte API-Sicherheit verwenden, um diesen Traffic sofort zu blockieren.

Sie sollten eine WAF finden, die Teil einer größeren WAAP-Lösung ist und erweiterte Erkennungen durch maschinelles Lernen bietet, das aus Signaturmustern von Angriffen lernt und auf unterschiedlichen Skalierungsstufen flexibel bleiben kann. Der WAAP-Schutz sollte in eine API-Sicherheitslösung integriert sein, die Verhaltensanalysen und individuell angepasste Antworten bietet. So lässt sich aus beiden Welten das Beste herausholen. Zusammen bieten diese Lösungen intern und extern vollständige Transparenz, Erkennung und Reaktion.

Was ist API-Erkennung und -Reaktion?

API-Erkennung und -Fehlerbehebung ist eine wichtiger werdende Kategorie der API-Sicherheit, bei der es um die tiefgreifende Analyse historischer Daten geht. Damit soll Folgendes erreicht werden:

- Festlegen einer Ausgangsbasis für das Verhalten aller API-Nutzer
- Erkennen von Angriffen und Anomalien, die auf möglichen API-Missbrauch und API-Ausnutzung hinweisen

Eine effektive API-Erkennung und -Reaktion in größerem Maßstab kann nur im Rahmen eines SaaS-Modells bereitgestellt werden, da für ressourcenintensive ML-Techniken umfangreiche Datensätze benötigt werden.

Was ist Advanced API Threat Protection?

Advanced API Threat Protection ist ein SaaS-basierter Ansatz für API-Sicherheit, der Verhaltensanalysen mit Threat Hunting kombiniert. Damit soll Folgendes erreicht werden:

- Ermitteln aller APIs, die von einem Unternehmen verwendet werden, einschließlich Shadow- oder Zombie-APIs
- Einsatz von maschinellem Lernen zum Einbringen von Geschäftskontext, um zu zeigen, wie APIs verwendet und missbraucht werden
- Führen Sie Verhaltensanalysen und Threat Hunting für APIs und API-Aktivitätsdaten durch

Was ist eine API-Sicherheitsplattform?

Eine API-Sicherheitsplattform ist ein SaaS-basiertes Angebot, das speziell für folgende Zwecke konzipiert wurde:

- Erstellen eines fortlaufend aktualisierten Bestandsüberblicks über alle APIs, die im Unternehmen verwendet werden (unabhängig davon, ob sie genehmigt sind oder nicht)
- Analyse der APIs und ihrer Nutzung, um den Geschäftskontext zu ermitteln und eine Ausgangsbasis für das erwartete Verhalten zu ermitteln
- Erkennen von Anomalien bei der API-Nutzung und gegebenenfalls Bereitstellung von Warnungs- und Unterstützungsdaten für das SIEM (Security Information and Event Management) und für SOAR-Workflows (Orchestrierung, Automatisierung und Reaktion im Bereich Sicherheit)
- Bereitstellen eines bedarfsgesteuerten Zugriffs auf API-Bestands-, Aktivitäts- und Bedrohungsinformationen sowohl für sicherheitsrelevante als auch für andere Stakeholder

Was ist ein Unternehmen für API-Sicherheit?

Da IT- und Sicherheitsverantwortliche APIs mittlerweile strategischer einsetzen, müssen sie unter Umständen spezialisierte API-Partner hinzuziehen. Die drei häufigsten Arten von API-Unternehmen sind:

- API-Gateway-Unternehmen, die Technologie bereitstellen, um API-Aufrufe zentral anzunehmen und an die entsprechenden Backend-Ressourcen und Microservices weiterzuleiten
- Anbieter von API-Sicherheitsplattformen, die sicherstellen, dass Unternehmen alle aktiven APIs und ihre potenziellen Risiken kennen, Angriffe und Missbrauch entdecken können, umfassende Tests ermöglichen und umfangreiche Daten über die Verwendung von APIs bereitstellen
- Anbieter von WAAP- und API-Sicherheitsplattformen, die für eine nahtlose Übertragung von API-Trafficdaten sorgen und gleichzeitig die Möglichkeit bieten, APIs auf und außerhalb der Plattform zu erkennen; ideal für die Lieferantenkonsolidierung und das Schließen digitaler Lücken

Was ist Threat Hunting in APIs?

Beim Threat Hunting wird aktiv nach unbekannten oder bislang unentdeckten Bedrohungen gesucht. Dieser proaktive Ansatz ist entscheidend, um neue und bislang nicht beobachtete Bedrohungen zu erkennen und abzuwehren, bevor sie größeren Schaden anrichten. Die Verhaltensanalyse zählt zu den Haupttechniken des Threat Hunting. Dabei wird das Verhalten von APIs analysiert, um verdächtige oder ungewöhnliche Aktivitäten zu identifizieren. Wenn beispielsweise eine API plötzlich innerhalb kurzer Zeit Tausende von Datensätzen anfordert, kann dies ein Hinweis darauf sein, dass die API-Geschäftslogik kompromittiert ist. Moderne API-Sicherheitslösungen bieten spezifische Threat-Hunting-Funktionen, sodass Sicherheitsteams mögliche Bedrohungen frühzeitig erkennen und mit Gegenmaßnahmen beantworten können.

Was ist WAAP?

Schutz für Webanwendungen und APIs (Web Application and API Protection, WAAP) ist eine Kategorisierung, die das Marktforschungsunternehmen Gartner für seine Brancheninformationen zu neuen Lösungen für Web- und API-Schutz verwendet. Es handelt sich um eine Weiterentwicklung der früheren Branchenberichterstattung zum WAF-Markt als Reaktion auf die wachsende strategische Bedeutung der API-Sicherheit und die Umstellung von WAF-Plattformen auf die Cloud als verwaltete SaaS.

Wie sieht ein Beispiel für eine API-Dokumentation aus?

Die gängigste Form der API-Dokumentation für RESTful APIs (die gängigste Art von Web-API) ist eine Sammlung von Swagger-Dateien, die auf der OpenAPI-Spezifikation basieren. Im Idealfall wird die API-Dokumentation von Entwicklern erstellt, wenn eine API entwickelt oder implementiert wird. In der Realität ist die API-Dokumentation jedoch häufig veraltet, was zu einer Diskrepanz zwischen der realen API-Nutzung und ihrer Dokumentation führt. Um dieses Problem zu beheben, können einige API-Sicherheitsplattformen Swagger-Dateien aus der tatsächlichen API-Aktivität generieren und die Lücken zwischen Dokumentation und tatsächlicher Bereitstellung hervorheben. Dies ist ein fester Bestandteil einer jeden API-Risikobewertung.

Gibt es eine API-Sicherheitscheckliste, die Unternehmen befolgen sollten?

Eine effektive API-Sicherheit erfordert viele detaillierte Schritte und fortlaufende Verfahren, die speziell auf das Unternehmen zugeschnitten sind. Im Folgenden finden Sie jedoch eine API-Checkliste, die Sicherheitsteams als Ausgangspunkt für den Ausbau ihrer API-Sicherheit verwenden können:

- Enthält Ihr API-Sicherheitsansatz einen Mechanismus für die kontinuierliche unternehmensweite API-Erkennung?
- Ist das API-Sicherheitsmanagement in die umfassenderen Sicherheits- und Risikomanagementpraktiken des Unternehmens integriert?
- Implementieren Sie einen allgemeinen API-Sicherheitsansatz, der Sie nicht an bestimmte Rechenzentrums- oder Cloud-Infrastrukturmodelle bindet?
- Vermittelt Ihr Ansatz Ihren Teams den Geschäftskontext, den sie benötigen, um die beobachteten API-Aktivitäten und mögliche Risiken wirklich zu verstehen?
- Verfügen Sie über eine Strategie für die Zwei-Wege-Automatisierung Ihrer API-Sicherheitsplattform und anderer damit verbundener Geschäftsprozesse wie SIEM/SOAR, Threat Hunting, Dokumentation, DevOps-Tools usw.?
- Ergreifen Sie Maßnahmen, um nicht sicherheitsrelevante Stakeholder wie zum Beispiel Entwickler in Ihre API-Sicherheitstools und -prozesse einzubinden?

Akamai schützt die Anwendungen, die Ihr Unternehmen vorantreiben, an jedem Interaktionspunkt – ohne die Performance oder das Kundenerlebnis zu beeinträchtigen. Unsere globale Plattform liefert Skalierbarkeit sowie transparente Einblicke in Bedrohungen. Gemeinsam mit Ihnen können wir auf diese Weise Bedrohungen erkennen und abwehren, damit Sie Markenvertrauen aufbauen und Ihre Vision umsetzen können. Möchten Sie mehr über die Cloud-Computing-, Sicherheits- und Bereitstellungslösungen von Akamai erfahren? Dann besuchen Sie uns unter akamai.com und akamai.com/blog oder folgen Sie Akamai Technologies auf X (ehemals Twitter) und LinkedIn. Veröffentlicht: 09/24.