Akamai

WHITEPAPER

Anatomie eines API-Angriffs

Verstehen von BOLA und dem Ausnutzen von Sicherheitslücken im Bestandsmanagement

Einführung

Die meisten Sicherheitsteams sind sich heute darüber im Klaren, dass die proaktive Erkennung von Bedrohungen ein wesentlicher Bestandteil eines effektiven Sicherheitsprogramms für Unternehmen ist, insbesondere wenn es um Programmierschnittstellen (Application Programming Interfaces, APIs) geht. APIs bieten oft direkten Zugriff auf Daten, Funktionen und Workflows. Obwohl grundlegende Maßnahmen am Netzwerkrand zum Schutz von Anwendungen weit verbreitet sind, nehmen der API-Missbrauch und andere Arten von Angriffen zu. Einige der bekanntesten Sicherheitsvorfälle, die in den letzten Jahren für Schlagzeilen sorgten, hatten mit APIs zu tun. Zum besseren Verständnis dieser Angriffsprofile, wie z. B. Broken Object Level Authorization (BOLA) und dem Ausnutzen von Schwachstellen im unsachgemäßen Bestandsmanagement, werden in diesem Whitepaper folgende Punkte erläutert:

- Überblick über die Grundlagen von APIs
- Warum API-Sicherheit ein Thema von wachsender Bedeutung ist
- Aufzeigen der wichtigsten API-Sicherheitsbereiche anhand einiger aufsehenerregender API-Sicherheitsvorfälle
- Veranschaulichung der Funktionen, die für eine effektive Suche nach API-Bedrohungen erforderlich sind

Grundlagen zu APIs und Endpunkten

Lassen Sie uns zunächst einige grundlegende Begriffe klären. APIs werden für viele Zwecke verwendet, von B2C-Funktionen (Business-to-Consumer) über B2B-Zusammenarbeit und -Integration (Business-to-Business) bis hin zu internen Entwicklungs- und Integrationsfunktionen. Web-APIs, die über dasselbe HTTP-Protokoll kommunizieren, das von Webbrowsern verwendet wird, sind das am häufigsten verwendete Implementierungsmodell. Die spezifischen Funktionen, die diese APIs bieten, werden manchmal als Services oder API-Produkte bezeichnet.

Wenn man über API-Sicherheit nachdenkt, ist es auch wichtig, das Konzept eines Endpunktes zu verstehen. Während dieser Begriff manchmal für Computergeräte von Endnutzern verwendet wird, hat er im Zusammenhang mit APIs eine andere Bedeutung. Sie können sich einen API-Endpunkt als eine einzelne zugängliche Ressource vorstellen, die Teil der API ist, zusammen mit der Operation, die damit ausgeführt werden kann.

> **Hier ein einfaches Beispiel. Ein API-Endpunkt, der Bestellinformationen für ein bestimmtes Unternehmen zurückgibt, könnte wie folgt dargestellt werden: GET /orders/{orderID}. In diesem Fall ist GET eine spezifische HTTP-Methode, während orders und orderID die spezifischen Ressourcen darstellen, die über die API angefordert werden.**

Warum sind APIs die nächste große Herausforderung für die Sicherheit?

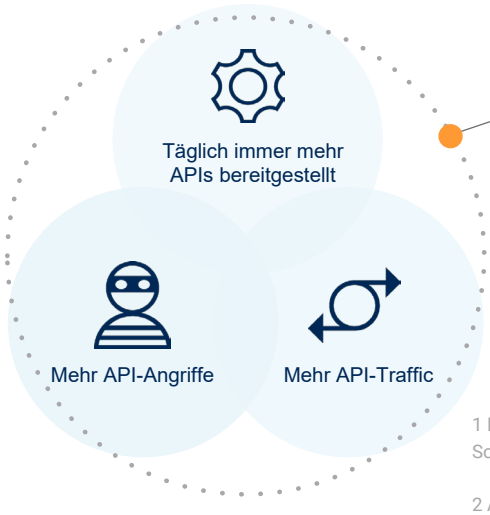
In der Vergangenheit war es vielleicht das Ziel eines Angreifers, in das Rechenzentrum eines Unternehmens einzudringen, um auf die Daten des Unternehmens zuzugreifen und sie von einem bestimmten Server zu entwenden. Oder er hat versucht, den Netzwerktraffic des Unternehmens zu untersuchen, um an sensible Daten zu gelangen. In solchen Szenarien könnte sich die proaktive Erkennung von Bedrohungen auf Aktivitäten wie Penetrationstests konzentrieren, um mögliche Einstiegspunkte für Angreifer abzuschneiden.

In einer Welt mit APIs sieht diese Dynamik anders aus. Viele APIs sind von Natur aus für jedermann in der Außenwelt zugänglich, wobei Anmeldedaten und Schlüssel manchmal die einzige Verteidigungslinie darstellen. Und Angreifer werden immer geschickter darin, diese Elemente zu kompromittieren. Darüber hinaus können einige der schädlichsten Arten des API-Missbrauchs von Parteien ausgehen, die Zugang zu APIs erhalten haben, diese aber auf nicht autorisierte Weise nutzen.

API-Angriffe in der realen Welt

Bei Akamai ist 31 % des gesamten Traffic, den wir schützen, API-Traffic. Dieser zunehmende API-Traffic führt zu nachgelagerten Effekten wie einer Zunahme von Angriffen und Missbrauch. Gartner prognostiziert, dass sich API-Missbrauch und Datenschutzverletzungen im Jahr 2024 verdoppeln werden. In der Zwischenzeit müssen viele Sicherheitsteams ihren Rückstand aufholen. Es gibt immer mehr APIs, während die vorhandenen Tools für Anwendungssicherheit nur einen sehr begrenzten API-Schutz bieten.

Bis 2024 werden sich API-Missbrauch und damit zusammenhängende Datenschutzverletzungen nahezu **verdoppeln.**[1]



31 % des Webtraffic stammt von APIs[2]

1 Die 10 wichtigsten Dinge, die Softwareentwickler über APIs wissen müssen

2 Akamai-Bedrohungsforscher haben festgestellt, dass 31 % des gesamten von Akamai geschützten Traffics API-Traffic ist

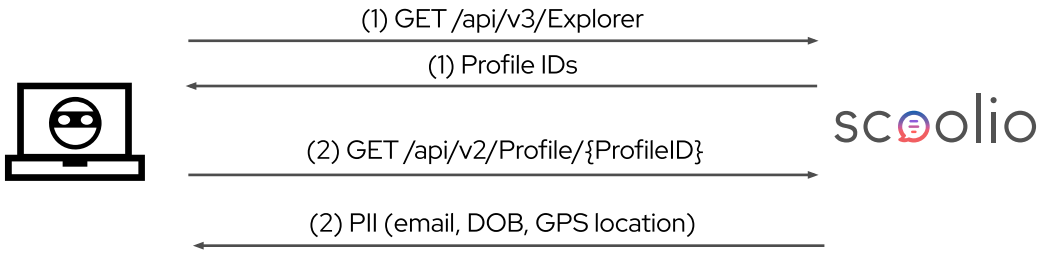
Um dieses Problem zu veranschaulichen, betrachten wir eine Fallstudie, die die realen Auswirkungen von API-Angriffen auf Unternehmen – und ihre Kunden – deutlich macht.

Fallstudie

Kontoübernahme | Scoolio

Ein aufsehenerregendes Beispiel ist ein Vorfall aus dem Jahr 2021, der die deutsche Bildungs-App Scoolio betraf. Die App sammelt umfangreiche Informationen über deren Nutzer, d. h. Schüler. Sie führt beispielsweise Persönlichkeitstests durch, bietet soziale Netzwerke und Chatfunktionen und verwaltet Aktivitäten wie Lernplanung und Nachhilfe. Diese Funktionen sammeln eine Menge personenbezogener Daten. Die Sicherheitsforscherin Lilith Wittmann entdeckte eine BOLA-Schwachstelle in den APIs der Bildungs-App, die es ermöglichte, über zwei API-Aufrufe auf personenbezogene und andere Daten beliebiger anderer Nutzer der Bildungs-App zuzugreifen.

Das funktionierte so:



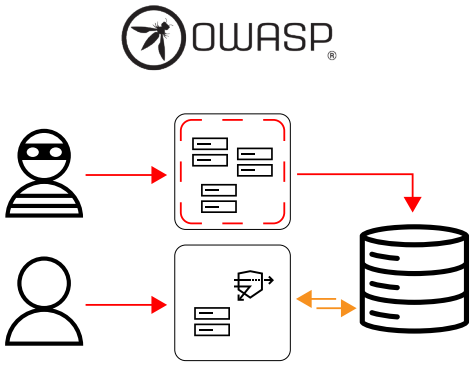
Schritt 1
Einen GET /api/v3/Explorer-API-Aufruf senden.
Dieser Aufruf gab UUIDs aus, die in dieser Implementierung als ProfileID bezeichnet werden.

Schritt 2
Einen GET /api/v2/Profile/{ProfileID}-API-Aufruf senden.
Diese Anfrage gab umfassende personenbezogene Daten des betreffenden Nutzers aus, einschließlich E-Mail, Geburtsdatum, GPS-Standort und mehr.

Der Wert der Verwendung von UUIDs

Während beide Szenarien auf die Verwendung von UUIDs abzielen, ist die Verwendung von UUIDs tatsächlich eine sehr gute Praxis. Die Verwendung zufällig generierter Zahlen anstelle einer vorhersehbaren Folge von Nutzer-IDs erschwert es einem Angreifer, massenhaft Zugriff auf Nutzerinformationen zu erlangen. Das Problem entsteht, wenn UUID-Informationen zu freizügig weitergegeben und mit BOLA-Schwachstellen hinzukommen.

Unsachgemäßes Bestandsmanagement



Eine weitere Facette dieses Exploits einer API-Schwachstellen ist, dass er unsachgemäße Bestandsverwaltung ausnutzt, was gleichzeitig Platz 9 der OWASP API Top 10 ist. Wenn man sich die Abfolge des Angriffs genau ansieht, stellt man fest, dass der erste Schritt auf Version 3 der API angewendet wird, während der zweite Schritt auf Version 2 abzielt. In Version 3 wurden Verbesserungen vorgenommen, um den Zugriff auf personenbezogene Daten stärker zu kontrollieren. Diese Verbesserungen wurden jedoch dadurch untergraben, dass die anfälligere Version 2 weiterhin für jedermann zugänglich war. Letztendlich waren sowohl Version 2 als auch Version 3 von der BOLA-Schwachstelle betroffen. Dadurch dass Version 2 unnötigerweise noch vorhanden war, waren die Auswirkungen der Schwachstelle jedoch noch gravierender.

Welche Schritte unternehmen Unternehmen heute, um ihre APIs zu schützen?

Viele Unternehmen konzentrieren sich bei der API-Sicherheit auf diese drei Säulen:

1. Zentralisierte Autorisierung – Die Implementierung einer zentralisierten Autorisierungs-Engine für alle API-Zugangsports reduziert das Risiko von API-Schwachstellen, indem Entwicklungsfehler vermieden werden, die zu unzureichenden Autorisierungsmechanismen führen.

2. API-Tests – Eine zweite wichtige Praxis ist das Testen von APIs. Durch das Testen auf alle Schwachstellen, insbesondere auf fehlerhafte Autorisierung, mittels statischer Codeanalyse und dynamischer Tests werden Probleme frühzeitig im Entwicklungsprozess erkannt.

3. Laufzeitschutz – Die dritte Säule besteht aus einer Reihe von Laufzeitschutzmaßnahmen für die Produktionsumgebung. Selbst die proaktivsten Teams können nicht alle Schwachstellen vor der Bereitstellung entdecken. Daher ist es wichtig, den Nutzerzugriff auf Produktionsdaten zu kontrollieren und die Ausnutzung bekannter Kategorien von Schwachstellen so weit wie möglich zu verhindern.

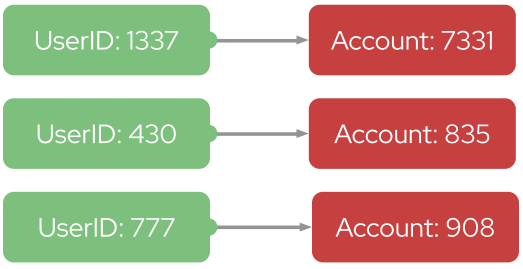
Diese drei Maßnahmen bilden eine ausgezeichnete Grundlage für Ihre API-Sicherheitsstrategie. Es ist jedoch wichtig zu bedenken, dass sie weder perfekt noch umfassend sind. Beispielsweise können selbst Unternehmen mit zentralisierter Autorisierung nicht garantieren, dass die Entwickler immer den Best Practices folgen. Und schließlich sind die vorhandenen Tools zum Schutz von Anwendungen oft gut darin, bekannte Angriffsmuster zu erkennen, aber weniger gut darin, differenziertere Bedrohungen wie BOLA zu erkennen.

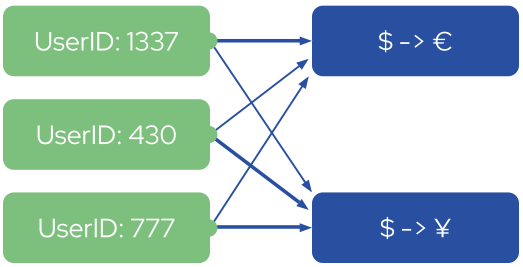
Wie lässt sich mit fortschrittlicheren BOLA-Erkennungstechniken auf dieser Basis aufbauen?

Einer der Schlüssel zur Erkennung und Abwehr von BOLA und anderen differenzierten API-Schwachstellen ist die Modellierung der Beziehungen zwischen den an der API-Aktivität beteiligten Entitäten. Dazu gehören nicht nur die Ressourcen selbst, sondern auch die Akteure, z. B. die Nutzer, die versuchen, auf die Ressourcen zuzugreifen. Durch die Zuordnung dieser Beziehungen zwischen den Akteuren und Geschäftsprozessen, die mit einer API interagieren, kann bei der Analyse ansonsten identischer API-Ereignisse zwischen legitimen und illegalen Aktivitäten unterschieden werden.

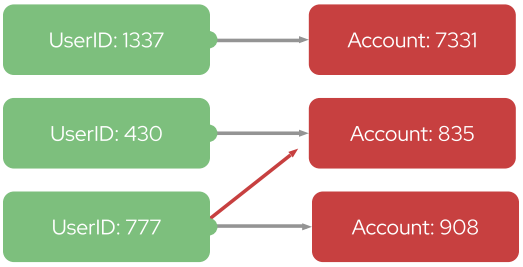
Beispiel für die Beziehungszuordnung

Um die Beziehungszuordnung besser zu verstehen, betrachten wir dieses grundlegende Beispiel. Eine Banking-App unterstützt zwei Aktionen. Die erste Aktion ist das Lesen Ihrer Kontodaten, einschließlich Informationen wie Kontostand, letzte Transaktionen usw. Die zweite Aktion ist die Anzeige von Wechselkursen. Die Beziehung zwischen Nutzern und Ressourcen ist in diesen Beispielen sehr unterschiedlich. Der Zugriff auf Kontoinformationen sollte auf einen einzelnen Nutzer beschränkt sein. Im Gegensatz dazu sollte die Wechselkursfunktion allen Nutzern zur Verfügung stehen.





Obwohl dies ein sehr einfaches Beispiel ist, ist es viel praktischer, BOLA zu verhindern oder aufzudecken, wenn man ein ausgefeilteres Modell der Beziehungen zwischen Entitäten erstellt.



Hier sehen wir einen Nutzer, der versucht, auf ein Konto zuzugreifen, das ihm nicht gehört. Der spezifische API-Aufruf mag identisch sein, aber der zusätzliche Kontext, der durch die Entitätszuordnung bereitgestellt wird, macht deutlich, dass dies nicht erlaubt sein sollte.

Erweiterte BOLA-Angriffserkennung in der Praxis

Als Nächstes wollen wir dieses Konzept auf komplexere Beispiele anwenden, z. B. auf die Schwachstellen der Fallstudie. Nachfolgend sehen Sie Ausschnitte der am Szenario beteiligten Entitäten:

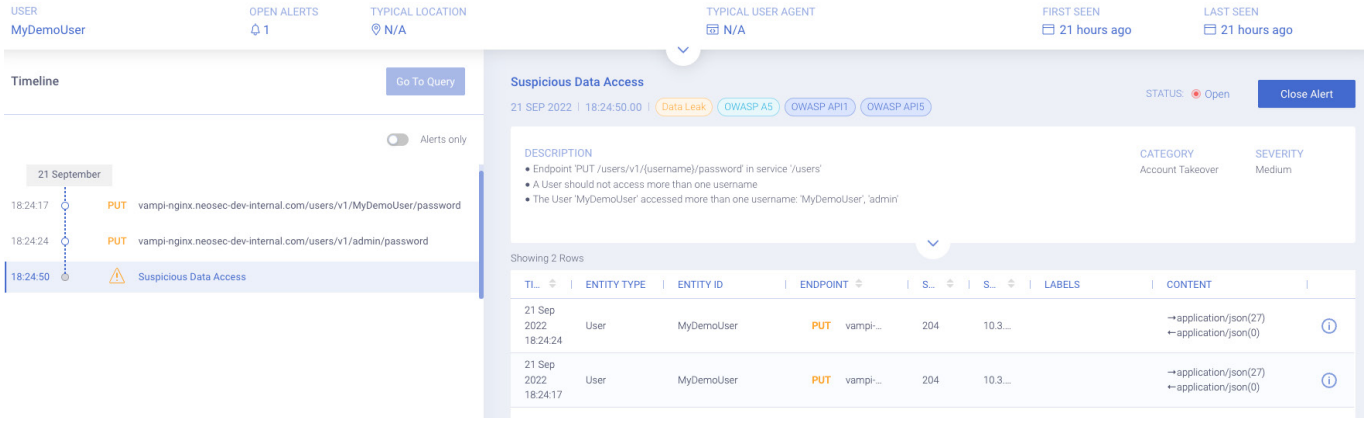
scoolio

GET/api/v3/Profile/ {ProfileID}

Header:

- Authorisierung: <MyAccessToken>

Die Entität des Akteurs ist grün und die angeforderte Ressource (die Profil-ID) ist rot hervorgehoben. Sobald diese Beziehungen verstanden wurden, können Schritte unternommen werden, um eine allgemeine Logik durchzusetzen, wie z. B. die Beschränkung des Zugriffs eines Akteurs auf eine einzelne Ressource, wenn dies angemessen ist. Dies ist alles andere als trivial, da die Beziehungen komplexer sein und 1:n Dimensionen umfassen können. Techniken wie maschinelles Lernen und Verhaltensanalyse machen dies jedoch möglich. Die erfolgreiche Erkennung einer BOLA-Schwachstelle für einen unserer Kunden sieht beispielsweise so aus:



Für dieses Beispiel wurde eine BOLA-Schwachstelle in einer Laborumgebung simuliert. Mithilfe von Entitätszuordnung und Verhaltensanalysen hat unsere Plattform die BOLA erkannt und eine Warnung mit vielen Informationen generiert. Ein Sicherheitsanalytiker oder Threat Hunter, der sich die Warnung ansieht, wird feststellen, dass MyDemoUser auf sein eigenes Nutzerprofil zugegriffen hat, um sein Passwort zu ändern – eine sanktionierte Aktion. Kurz darauf sehen wir jedoch auf der Zeitachse, dass er einen weiteren API-Aufruf durchgeführt hat, um das Administrator-Passwort zu ändern. Da dies aufgrund der Beziehung zwischen Akteur und Ressource eindeutig eine unzulässige Aktion ist, wurde die Warnung generiert.

Womit fängt eine API-Sicherheitsinitiative an?

API-Sicherheit ist für die meisten Unternehmen ein kontinuierlicher Prozess. Daher kann es schwierig sein zu wissen, wo man anfangen soll. Während die drei oben genannten Grundpfeiler einen nützlichen Ausgangspunkt darstellen, wird die Effektivität Ihres Ansatzes erheblich gesteigert, wenn Sie bei der Implementierung diese drei Empfehlungen befolgen:

1. Halten Sie Ihren API-Bestand stets auf dem neuesten Stand
2. Überwachen Sie sowohl nicht-produktive als auch produktive API-Umgebungen
3. Setzen Sie Beziehungen zwischen Entitäten durch

Sie können keine APIs schützen, die Sie nicht kennen. Ein wirksamer API-Schutz beginnt daher mit einem aktuellen API-Bestand und einer Bewertung der Sicherheitslage. Ebenso ist es wichtig, dass Sie bei der Entwicklung Ihrer API-Sicherheitsüberwachungsfunktionen diese sowohl auf produktive als auch auf nicht produktive API-Implementierungen ausdehnen. Am allerwichtigsten ist jedoch, dass Ihre API-Überwachung und -Durchsetzung über die reinen Aktionen hinausgeht und die Beziehungen zwischen den an Ihren API-Aktivitäten beteiligten Entitäten berücksichtigt. Auf diese Weise können Sie Schwachstellen und Sicherheitslücken erkennen und die Compliance der vorgesehenen API-Nutzungsmodelle durchsetzen. Wenn Sie das Verhalten innerhalb Ihrer APIs verstehen, können Sie Missbrauch erkennen.

Möchten Sie mehr über API-Angriffe und darüber, wie Sie sich davor schützen können erfahren? Sehen Sie sich unseren Artikel zu den OWASP API Top 10 an.

Akamai schützt Ihr Kundenerlebnis, Ihre Mitarbeiter, Systeme und Daten und integriert Sicherheit in alle von Ihnen erstellten Inhalte – überall dort, wo Sie sie erstellen und bereitstellen. Dank der Einblicke unserer Plattform in globale Bedrohungen können Sie Ihre Sicherheitsstrategie anpassen und weiterentwickeln, um Zero Trust zu implementieren, Ransomware zu stoppen, Anwendungen und APIs zu schützen oder DDoS-Angriffe abzuwehren. Das gibt Ihnen das nötige Vertrauen, um kontinuierlich Innovationen zu entwickeln, zu expandieren und alles zu transformieren, was möglich ist. 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.