7 best practices om je API's te beveiligen

> In de huidige digitale wereld zijn Application Programming Interfaces (API's) onmisbaar geworden voor het uitwisselen van gegevens en het integreren van verschillende systemen en applicaties. Nieuwe API's worden aan een duizelingwekkende snelheid in gebruik genomen—een snelheid die veel voor ligt op de maturiteit van netwerk- en applicatiebeveiligingspraktijken. Dit maakt ze een belangrijk doelwit voor cyberaanvallen. Het was dus niet helemaal onverwacht toen Gartner eind 2021 voorspelde dat API's het grootste aanvalsoppervlak van een onderneming zouden worden (Gartner artikel).

In dit artikel belichten we enkele best practices om je API's te beveiligen. Door deze best practices toe te passen, wordt jouw bedrijf effectief beschermd tegen cyberaanvallen en beheers je de risico's die API's met zich meebrengen.

1. Authenticeer en autoriseer elke API-oproep

Authenticatie en autorisatie zijn belangrijk om ervoor te zorgen dat alleen geautoriseerde gebruikers toegang hebben tot een API. Authenticatie is het proces waarbij de identiteit van een gebruiker wordt vastgesteld, waarna het autorisatieproces bepaalt of een geauthenticeerde gebruiker toegang mag hebben tot specifieke resources of functies van de API. Eenmaal de identiteit en de rechten geverifieerd zijn, wordt een zogenaamd access token gegenereerd waarmee toegang tot (bepaalde delen van) de API verleend wordt.

Beperk de toegang tot beveiligde gegevens door het gebruik van rollen- en rechtenbeheer en zorg dat enkel geautoriseerde gebruikers toegang hebben tot beveiligde gegevens. Gebruik hiervoor het principe van least privilege. Ondanks het feit dat dit eenvoudig klinkt, zijn fouten in autorisatie en authenticatie respectievelijk de eerste en tweede meest voorkomende beveiligingsproblemen in API's volgens de OWASP API Security Top 10.

Wanneer enkel autorisatie vereist is, kan het populaire OAuth protocol gebruikt worden. OAuth biedt een aanpasbaar raamwerk aan dat verschillende use cases ondersteunt via zogenaamde flows. De eenvoudige client credentials flow laat applicaties toe om access tokens te genereren waarmee de API aangeroepen kan worden. De authorization code flow laat gebruikers inloggen en toegang verlenen tot persoonlijke gebruikersdata aan applicaties die dan een access token krijgen om die gebruikersdata via de API te benaderen.

OAuth biedt geen ondersteuning voor authenticatie. De gegenereerde access tokens zijn opaak en kunnen niet rechtstreeks gelinkt worden aan een persoon. OpenID Connect (OIDC) is een protocol dat gebaseerd is op OAuth 2.0 en vult deze leemte op. Naast een access token wordt er ook een identity token afgeleverd met informatie over de gebruiker die het access token afgeleverd heeft én aan welke applicatie die dat afgeleverd heeft.

2. Sla je API-sleutels veilig op

API-sleutels zijn unieke codes die toegang verlenen tot een API. Ze worden door de API gebruikt om applicaties van elkaar te onderscheiden en eventuele gebruikslimieten toe te passen. Wanneer een API-sleutel in handen komt van een malafide persoon, kan die toegang krijgen tot de API in naam van de applicatie waar de API-sleutel toe behoort. Het is dus belangrijk dat ze op een veilige manier worden opgeslagen en beheerd.

Zorg dat API-sleutels nooit rechtstreeks terecht komen in de broncode van een applicatie. Wanneer die code publiek wordt gemaakt—bijvoorbeeld in een toegankelijke source code repository—dan kan een aanvaller de API-sleutel gewoon uitlezen en gebruiken. Zo rapporteerde Github dat er in 2022 maar liefst 1.7 miljoen potentiële 'geheimen' gevonden zijn in repositories op het platform. Een betere oplossing is om geheimen lokaal op een computer te configureren (bijvoorbeeld via environment variables, of in bestanden die niet opgenomen zijn in de source code repository), of om gebruik te maken van een service die geheimen beheert.

In tegenstelling tot access tokens, zijn API-sleutels vaak long lived. Dit wil zeggen dat ze voor lange periodes geldig zijn en dus gebruikt kunnen worden om de API aan te spreken. Zorg ervoor dat API-sleutels toch op regelmatige tijdstippen gehergenereerd worden—zeker wanneer er een vermoeden van oneigenlijk gebruik is.

3. Wantrouw de invoer van de API

Wanneer een API gegevens ontvangt van externe bronnen, zoals een webformulier of een mobiele app, kunnen deze gegevens gemanipuleerd worden door een aanvaller om ongewenst gedrag van de API uit te lokken. Dit kan leiden tot problemen zoals beveiligingslekken, fouten in de toepassing of zelfs systeemuitval. Door de invoer te valideren, kan worden verzekerd dat de gegevens die naar de API worden verzonden, voldoen aan het verwachte formaat.

De invoer van de API kan worden gecontroleerd door gebruik te maken van technieken zoals invoervalidatie, sanitizing, en whitelisting. Invoervalidatie kan helpen bij het controleren of de gegevens voldoen aan het juiste formaat, zoals een geldig e-mailadres of getal. Sanitizing kan worden gebruikt om onveilige tekens te verwijderen uit de gegevens. Whitelisting kan worden gebruikt om alleen bepaalde waarden of formaten toe te staan.

4. Beperk de netwerktoegang tot jouw API

Wanneer access tokens of API-sleutels gestolen worden, kan een aanvaller die misbruiken om toegang te krijgen tot het systeem. Een extra beschermingslaag waarbij de toegang tot een API beperkt wordt tot specifieke IP-adressen of netwerken kan helpen om de slaagkansen van zo een aanval te verkleinen. Zo wordt de aanvaller al geweerd van het systeem voordat hij de kans krijgt om de gestolen toegangssleutel bij de API aan te bieden.

Beperken van toegang tot specifieke IP-adressen kan ook helpen bij het identificeren van verdachte activiteiten of inbraken. Als een ongeautoriseerd IP-adres toegang probeert te krijgen tot de API, kan dit worden gedetecteerd, geblokkeerd, en geanalyseerd. Hierdoor kan de inbraak snel opgespoord worden en kan de situatie opgelost worden voordat er schade wordt aangericht.

Wanneer een API publiek beschikbaar wordt gemaakt, is rate limiting een belangrijke techniek om misbruik door onbevoegde gebruikers te voorkomen. Door rate limiting toe te passen, blijft de API snel reageren voor geautoriseerde gebruikers en worden tegelijkertijd gebruikers die proberen de API te overbelasten afgewezen. Dit kan bijvoorbeeld gebeuren door het beperken van het aantal verzoeken dat een gebruiker binnen een bepaalde tijdsperiode kan doen, of door het blokkeren van IP-adressen die verdachte activiteit vertonen. Rate limiting kan bovendien ook helpen om de kosten te beheren van de infrastructuur die wordt gebruikt om de API te hosten. Door het beperken van het aantal verzoeken, kan de infrastructuur efficiënter worden gebruikt.

5. Beveilig je data op het netwerk en in rust

Wanneer gegevens via een API over het netwerk worden gestuurd, kunnen deze gegevens kwetsbaar zijn voor onbevoegde toegang of manipulatie. Dit kan leiden tot ernstige beveiligingsproblemen, zoals gegevensverlies, identiteitsdiefstal, of fraude. Door gebruik te

maken van beveiligde verbindingen, bijvoorbeeld HTTPS waarbij de laatste versie van TLS gebruikt wordt, kan de connectie versleuteld en beschermd worden.

Cryptografische protocollen zoals TLS kunnen bovendien ook gebruikt worden om de communicerende partijen te authenticeren. Cryptografische certificaten kunnen gebruikt worden als veilige alternatieven voor standaard API-sleutels. Daarnaast kan het protocol ook helpen om non-repudiation te garanderen. Dit wil zeggen dat de partijen die betrokken zijn bij de communicatie niet kunnen ontkennen dat ze de gegevens hebben ontvangen of verzonden.

Ook voor interne verbindingen tussen API-services die in de backend gebruikt worden, wordt best een beveiligde connectie gebruikt. Alhoewel het bedrijfsnetwerk een veiligere plaats is dan het publieke internet, wordt in moderne zero trust beveiligingsarchitecturen ervan uit gegaan dat geen enkele component in het netwerk automatisch vertrouwd wordt. Door deze doorgedreven authenticatie, autorisatie, en dataversleuteling binnen het interne netwerk, krijgt een aanvaller die tóch toegang heeft gekregen geen enkele kans.

Net zoals data op het netwerk, moet data in rust ook voldoende beschermd worden. De beveiliging van een API kan natuurlijk omzeild worden als een aanvaller rechtstreeks aan de plek kan waar de data opslagen wordt. Zorg dat data in een database of een bestand op de schijf altijd geëncrypteerd bewaard wordt en dat enkel toegang met de nodige machtigingen toegelaten wordt.

6. Monitor je API's en wees alert voor dreigingen

Geautomatiseerde monitoring-platformen zijn een belangrijke schakel in de beveiliging van API's. Deze systemen analyseren het gebruik van de API en kunnen inzicht geven hoe gebruikers de API consumeren. Op beveiligingsvlak kunnen AI-gedreven tools patronen herkennen in het dataverkeer en dreigingen in een vroeg

stadium opsporen. Wanneer een aanvaller naar kwetsbaarheden in de API zoekt, moet hij de API stelselmatig testen. Dit afwijkend gedrag wordt snel gedetecteerd zodat het misbruik bij de bron kan gestopt worden. Deze aanpak vereist geen vooraf gedefinieerde lijst van patronen of regels. De tool leert zelf wat het normaal gebruik van de API is, waardoor er minder nood is aan updates om ook nieuwe types aanvallen te detecteren.

Naast het monitoren van de API, moet deze informatie ook gelogd worden. Het loggen van gebeurtenissen geeft een gedetailleerd beeld van de activiteiten die plaatsvinden op een API. Het geeft beheerders de mogelijkheid om te zien wie toegang heeft gehad tot de API, wanneer dit gebeurd is, en welke acties er werden uitgevoerd. Dit laat toe om een analyse te maken van een aanval nadat die gebeurd is, om zo vast te stellen hoe de aanvaller te werk gegaan is en welke toegang hij verkregen heeft.

Een andere belangrijke reden waarom loggen belangrijk is, is dat het helpt om aan bepaalde compliance-eisen te voldoen. Veel bedrijven zijn verplicht om specifieke gegevens te loggen, zoals de gegevens van de gebruikers die toegang hebben gehad tot de API en de acties die zijn uitgevoerd. Door gebeurtenissen te loggen, kan een bedrijf aantonen dat het voldoet aan deze eisen.

7. Gebruik een management tool voor je API

Een API-gateway is een API-management tool die als toegangspunt dient tot je volledige API. API-oproepen die toekomen op de gateway worden intern doorgestuurd naar de services die ze kunnen afhandelen. Op die manier stelt een API-gateway een organisatie in staat om de toegang tot API's te regelen, te monitoren en te beschermen tegen aanvallen.

Een volwaardige API-gateway zal ondersteuning bieden voor verschillende beveiligingsfunctionaliteiten. Zo kan authenticatie en autorisatie al toegepast worden van zodra een API-oproep binnenkomt en kan het gebruik van de API gemonitord en gelogd

worden. Daarnaast kunnen ook netwerkaanvallen, zoals DDoS-aanvallen, en andere dreigingen gedetecteerd en afgeweerd worden. Dit helpt om de beschikbaarheid van de API's te garanderen, zelfs in geval van een aanval.

Documentatie van de bestaande API's is essentieel voor de goede werking van de API-gateway. Ook vanuit beveiligingsstandpunt is dit belangrijk: je kan namelijk niets beveiligen waar je het bestaan niet vanaf weet. Zorg dat je een overzicht hebt van alle API's die bestaan, waarvoor ze dienen, welke gebruikers er toegang tot hebben, en welke data er teruggestuurd wordt. De gateway kan deze informatie dan gebruiken om te controleren of API-oproepen voldoen aan de verwachtingen en eventueel maatregelen nemen wanneer dit niet het geval is.

Tot slot

Naast het doorvoeren van de juiste technologische maatregelen, moeten bedrijven blijven investeren in periodieke security assessments om de beveiliging van API's te waarborgen en te verbeteren. Dit kan bijvoorbeeld gedaan worden door middel van penetration testing, waarbij aanvallen gesimuleerd worden om gevoelige informatie of kwetsbare punten te identificeren. Wanneer het personeel met de nodige expertise niet binnen het bedrijf aanwezig is, kan beroep gedaan worden op externe beveiligingsbedrijven of ethische hackers.

Wanneer het dan toch misloopt, moet je snel kunnen handelen. Een incident management proces helpt om snel en efficiënt te reageren op een beveiligingsincident, zodat de impact hiervan beperkt kan worden. Dit proces omvat onder andere het detecteren van een incident, het analyseren van de impact, het nemen van passende maatregelen en het bijhouden van de resultaten hiervan. Bovendien helpt een incident management proces om lessen te leren uit vorige incidenten, zodat de beveiliging van de API's in de toekomst kan worden verbeterd.

Het is belangrijk om te onthouden dat API security een continu proces is, omdat de bedreigingen en de technologie voortdurend veranderen. Met een gecombineerde aanpak van de juiste technologie, uitgewerkte procedures, en interne expertise, kunnen beveiligingsincidenten vermeden worden.

Auteur: Pieter Philippaerts, DistriNet, KU Leuven

Publicatiedatum: 30 maart 2023