Vlaanderen
is ondernemen

CYBERSECURITY

Uitdagingen in de API-economie

AGENTSCHAP
INNOVEREN &
ONDERNEMEN

DistriNet

Online, webgebaseerde softwarediensten worden steeds vaker aangeboden als *Application Programming Interface* (API). Deze evolutie, ook gekend onder de naam *API-economie,* is over vele sectoren een booming trend en creeert de opportuniteit voor kleine spelers om een grote business-impact te hebben ondanks beperkte middelen. Het dwingt ook bestaande, eventueel grotere spelers om mee te evolueren. Deze evolutie is van groot belang voor de hele softwaresector en voor de digitaliserings(r)evolutie. Online, webgebaseerde softwarediensten worden daarom steeds vaker aangeboden als API. Deze API-gebaseerde aanpak is dus geen niche maar wordt op vele plaatsen in vele sectoren toegepast.

Deze evolutie veroorzaakt een uitbreiding van bestaande markten, connecteert bedrijven met aanleunende markten, en leidt vaak tot nieuwe disruptieve businessmodellen. Dit gegeven creeert grote kansen voor Vlaamse bedrijven. De fintech-sector wordt in deze context als toonaangevend beschouwd. Vlaanderen kent daar een steile opmars van startups en scaleups. De sector wordt gekenmerkt door enkele grote financiële spelers met daarrond een groot netwerk van technologie-startups en KMO's. Deze fintech startups en KMO's zijn de softwarebouwers en software-dienstenaanbieders voor financiële processen bij heel wat grote bedrijven.

Cybersecurity is een inherente vereiste om succesvol te zijn. Om end-to-end bescherming van de gebruiker te bieden, moeten alle partnersystemen de nodige authenticatie, autorisatie, gegevensbescherming, en systeemintegriteit aanbieden. De APIs vormen de basisstructuur waarop alle verdere connectiviteit tussen systeem gebouwd is. Het implementeren van een hoogwaardige beveiliging is hier absoluut noodzakelijk.

De vereiste kennis om een veilig systeem te implementeren is niet geconsolideerd en gefragmenteerd. Er bestaat ook geen *one-size-fits-all* oplossing. De complexiteit vereist trade-offs en technologieselectie, met een correcte vertaling naar specifieke platformen en technologieomgevingen.

Een belangrijk probleem voor de evolutie naar en adoptie van een API-economie is dus de complexiteit van de securitytechnologieën voor het ontsluiten van de softwarediensten als een veilige, web-gebaseerde API. Deze technologieën omvatten geavanceerde authenticatie, geëxternaliseerde autorisatie, en ook het beveiligen van zowel de client als de server subsystemen.

Businessnoden en opportuniteiten

Een grote uitdaging situeert zich in de urgente businessnood voor het veilig ontsluiten van softwarediensten en data in de context van een digitale revolutie waar de API-gebaseerde aanpak razendsnel aan belang wint. Het regelgevende landschap verandert en de sector wordt verplicht van zijn diensten open te stellen naar derde partijen (PSD2). Dit vraagt echter dat deze sector zijn interne en publieke API's beschikbaar stelt in een B2B- en B2C-context: aan partners, klanten, eindgebruikers, en andere afdelingen binnen de organisaties.

API-security is een belangrijke hefboom voor deze ontsluiting. Een dringende versnelling is nodig in de adoptie van security-technologieën, maar de complexiteit van de technologieën belemmert dit. Orthogonaal hierop is de complexiteit van de brede waaier aan clients die ondersteund moeten worden, zoals web, mobile, en machine-to-machine (M2M).

Complexiteit van het technologielandschap

Op korte tijd is er technologisch veel geëvolueerd omtrent API-security met nieuwe technologieën, standaarden en oplossingen (zoals bv. JWT, OAuth2, B2C identity management, API

In een analyse van consultancybedrijf PwC en het *Open Data Institute* wordt geschat dat de verplichting van banken om hun diensten aan te bieden aan externen zal leiden tot een omzetopportuniteit van £7.2 miljard tegen 2022. Deze extra omzet komt uit nieuwe producten en diensten, maar houdt ook rekening met het verlies van inkomsten voor partijen die zich niet aanpassen.

Volgens PwC vindt 92% van de banken partnerschappen met fintech-bedrijven belangrijk, en zijn 71% van de banken overtuigd dat samenwerking nodig zal zijn om het tempo van innovatie bij te kunnen houden.

gateways, enz.). Ook zijn er nieuwe evoluties in de integratie van derde partijen, extern identiteitsbeheer alsook gecentraliseerde en geëxternaliseerde autorisatie (bijv. gefedereerde autorisatie naast gefedereerde authenticatie)

Zowel voor web als mobile zijn er complexe oplossingen voor geavanceerde authenticatie. Het resultaat is een brede waaier van heel veel verschillende alternatieven waarin het niet evident is de juiste trade-offs en keuzes te maken voor de beveiligingsarchitectuur.

Technische uitdagingen

Financiële services zijn een aantrekkelijk doelwit voor aanvallers. Dreigingen zoals virussen, malware, man-in-the-middle, phishing en Denial-Of-Service-aanvallen worden vergroot wanneer interne systemen ontsloten worden via een API.

Om te beschermen tegen dergelijke aanvallen, moet een effectieve cybersecurity-aanpak uitgerold worden in het hele ecosysteem. Dit omvat onder andere een gedetailleerde monitoring van het API-gebruik door partners. API's vormen doorgaans een abstractielaag over de interne systemen. Ze hoeven geen informatie te geven over de onderliggende systemen. Door partners enkel toegang te bieden tot deze systemen via de welomschreven regels van de API, kan de toegang en het gebruik nauwlettend worden gevolgd.

De verantwoordelijkheid voor de beveiliging van de API rust volledig bij de aanbieder ervan. Misbruik kan dramatische gevolgen hebben voor het bedrijf en de klanten. Het is dus belangrijk dat de API-provider een goed zicht heeft op de risico's die verbonden zijn met het openstellen van een API, en dat er voldoende beschermende maatregelen getroffen zijn.

Autorisatie

Het verlenen van toegang tot een API gebeurt via een *autorisatieproces*. De ontwikkelaar van de client-applicatie doorloopt een aanvraagprocedure waarna de API-provider toegang kan verlenen. Afhankelijk van de gevoeligheid van de API kan deze procedure eenvoudig zijn tot zeer complex. Voor financiële services geldt meestal een strenge onboarding-procedure waarbij de gebruiker van de API bepaalde vereisten moet kunnen aantonen.

Volgens een rapport van Akamai uit 2019 maken API-aanroepen 83% van alle webverkeer uit. Het aandeel blijft nog groeien. Gartner berekende dat API's 40% van het aanvalsoppervlak van webapplicaties beslaan, en verwacht dat dit tegen 2021 zal stijgen tot 90%. In 2022 zouden aanvallen op API's de meest voorkomende aanvalsvector worden.

Na het autorisatieproces succesvol doorlopen te hebben, krijgt de ontwikkelaar toegang tot de API voor een bepaalde applicatie. Bij elke aanroep van de API zal de applicatie deze toelating moeten kunnen aantonen. Veelal gebeurt dit door het meeleveren van een *API key* of een gebruikersnaam/paswoord dat de ontwikkelaar gekregen heeft als resultaat van het autorisatieproces. Voor extra veiligheid kan ook gebruik gemaakt worden van een *digitaal certificaat*. Dit biedt een betere bescherming tegen afluisteren of diefstal, aangezien de geheime sleutel de applicatie nooit verlaat.

Indien de API-toegang geeft toe gebruikersgegevens, is naast de autorisatie van de API-provider ook een autorisatie van de gebruiker nodig. In dit geval moet de applicatie een tweede autorisatieproces doorlopen, waarbij de gebruiker expliciet toegang verleent tot zijn gegevens aan de applicatie. Een populair autorisatieframework dat hiervoor gebruikt kan worden is *OAuth 2.0*. De gebruiker wordt dan in een browser (eventueel ingebed in een mobiele/desktopapplicatie) doorgestuurd naar een authenticatie- en autorisatieserver die vertrouwd wordt door de API-provider. De gebruiker krijgt een overzicht te zien van de rechten die de applicatie vraagt, en kan kiezen om deze toegang al dan niet te geven.

OAuth biedt ondersteuning voor meerdere *flows* aan. Een flow legt vast op welke manier de autorisatie gebeurt, en elke flow is gespecialiseerd voor een bepaalde use case. Naast de flows waarbij een eindgebruiker bij betrokken is, is er bijvoorbeeld ook een flow voor machine-to-machine-communicatie (als alternatief voor API keys).

Het resultaat van een succesvolle autorisatie door de eindgebruiker is doorgaans een *access token*. Een access token is een stuk data dat de

Er zijn drie manieren waarop een eindgebruiker een applicatie kan autoriseren. Deze zijn:

Redirection De gebruiker wordt van de applicatie omgeleid naar de autorisatieserver van de API-provider. De provider behoudt de volledige controle over het proces, en stuurt de gebruiker na voltooiing terug naar de applicatie.

Embedded De authenticatiegegevens van de klant worden via de applicatie-interface opgevraagd en doorgegeven aan de API-provider. Hierdoor kan de API-provider de klant authentiseren; de autorisatie wordt impliciet aangenomen.

Decoupled De API-provider vraagt de klant om een transactie goed te keuren via een andere applicatie of apparaat dat onafhankelijk is van de applicatie front-end.

Het OAuth 2.0 framework ondersteunt elk van deze manieren, afhankelijk van de gekozen *OAuth flow*.

toegang tot de gebruikersgegevens voorstelt. Een applicatie kan gebruik maken van een access token om gebruikersdata op te vragen, maar de applicatie weet niet *welke* gebruiker toegang gegeven heeft. Access tokens stellen dus geen gebruikers voor, maar enkel een gedelegeerd recht om gebruikersdata op te vragen.

Authenticatie

Afhankelijk van de use case van de API, kan identificatie van de gebruiker belangrijk zijn. De identiteit kan worden vastgesteld op verschillende manieren. Traditioneel moet de gebruiker bewijzen dat hij iets *weet*, zoals een paswoord of een PIN-code. In de financiële wereld wordt dit al enige tijd gecomplementeerd met het *bezitten* van een digipass of smartcard. Dankzij vingerafdrukscanners op mobiele telefoons wordt het identificeren van gebruikers aan de hand van *biometrische gegevens* ook steeds populairder.

Voor de financiële wereld vereist de PSD2-wetgeving *strong customer authentication*. Deze richtlijnen omvatten onder andere het gebruik van multi-factor authenticatie, waarbij de identiteit van de gebruiker wordt vastgesteld aan de hand van meerdere authenticatietechnieken. In het bijzonder moeten deze technieken resistent zijn tegen phishing-aanvallen. Eenvoudigweg de login en het paswoord vragen aan de gebruiker is dus niet voldoende. Afhankelijk van de actie die uitgevoerd wordt, zoals het uitvoeren van een betaling, is herauthenticatie vereist.

Binnen de Europese financiële sector wordt het gebruik van het OpenID Connect-protocol (dat gebaseerd is op het OAuth 2.0 framework) aanzien als de *best practice*. OpenID Connect behoudt de autorisatie-functionaliteit van OAuth 2.0 en voegt extra authenticatie-functionaliteit toe. Op deze manier kunnen banken de toegang tot gebruikersgegevens regelen, en kunnen de applicaties van derden gebruikers identificeren zonder dat de logingegevens van de eindgebruikers gecompromitteerd kunnen worden. Naast de Belgische banken biedt ook de Belgische overheid een OpenID Connect-service aan met de *Federal Authentication Service* (FAS).

OpenID Connect biedt ondersteuning voor *federated authentication*, waarbij een applicatie gebruik kan maken van verschillende *identity management* systemen. Dit is in het bijzonder belangrijk voor fintechs om klanten van verschillende banken te kunnen ondersteunen.

Beveiligingsgebeurtenissen afhandelen

Telkens wanneer een beveiligingsgebeurtenis wordt gedetecteerd, moeten de juiste acties ondernomen worden om de gebeurtenis te analyseren en eventueel verder af te handelen. Dit proces gaat actief zoeken naar aanvallen of andere ongewenste interacties van buitenaf, en probeert misbruik van de API te vermijden. Op die manier wordt de kans op een succesvolle cyberaanval sterk verkleind.

Een nieuwe generatie tools, *Web Application and API Protection* (WAAP) services genoemd, bouwen verder op de bescherming die een conventionele web application firewall biedt. Het beveiligingsbeleid wordt verschoven van de buitenste perimeter, waar traditioneel alle beveiligingsbeslissingen gemaakt werden, naar de microperimeters van de onderliggende services. Deze verschuiving volgt op de overstap van vele API-implementaties naar gedistribueerde, hybrid-cloud services en de opkomst van interservice dataverkeer. Authenticatie en

autorisatie wordt zo omgevormd van een eenmalige check naar een continue monitoring op verschillende punten in de API-architectuur.

Beveiligingsgebeurtenissen worden opgeslagen in een uitgebreid logboek. Het loggen van deze gebeurtenissen kan helpen om tot een gecoördineerde oplossing te komen (eventueel tussen meerdere partners) in het geval een aanval gedetecteerd wordt. Daarnaast is een logboek nuttig om eventuele schade na een aanval te herstellen en om te kunnen rapporteren aan partners, toezichthouders, en eventueel andere belanghebbenden.

Vanuit het standpunt van de algemene verordening gegevensbescherming (AVG, Eng.: GDPR) is het logboek ook belangrijk. Eindgebruikers moeten kunnen beslissen welke data gedeeld mag worden, met wie en voor welk doel. Deze toestemming wordt bijgehouden in het beveiligingslogboek. Gebruikers moeten in staat zijn om op eender welk moment deze toestemming in te trekken en al hun data te verwijderen van de service.

API-specifieke bedreigingen

De implementatie van een API is meestal gebaseerd op bestaande webtechnologie. Toch wordt een API op een andere manier gebruikt dan een webapplicaties, waardoor er — naast de bekende web-specifieke kwetsbaarheden — ook specifieke API-kwetsbaarheden geïdentificeerd kunnen worden. De meest voorkomende problemen zijn beschreven in de *OWASP API Security Top 10*.

Bovendien zijn API's ook onderhevig aan een aantal niet-technische bedreigingen. Zo kunnen hackers zelf ook een geldige gebruikersaccount aanmaken, kunnen geldige logingegevens gestolen worden buiten de API om, of kunnen access tokens gelekt worden.

Het is belangrijk om bij de ontwikkeling van een API een set van *best practices* op te stellen en tijdens de ontwikkeling af te dwingen. Deze aanbevelingen vertrekken vanaf de onderliggende HTTP-laag, waar bijvoorbeeld het gebruik van TLS 1.2 kan afgedwongen worden. Op API-niveau moeten keuzes gemaakt worden met betrekking tot access control, privacy en gegevensbeheer.

Het testen van de beveiliging moet rekening houden met aanvallers die de API rechtstreeks aanspreken, en niet via een client. De API moet worden aangesproken op een manier dat een normale applicatie niet doet, om zo data op te vragen waarvoor de aanvaller geen toegang tot heeft.

OWASP API Security Top 10

De OWASP Foundation, een nonprofit-organisatie die werkt rond software security, heeft in 2019 een lijst gepubliceerd van de meest-voorkomende beveiligingsproblemen in API-implementaties. De top 10 bestaat uit de volgende kwetsbaarheden:

1. **Kwetsbare autorisatie op object-niveau**
2. **Kwetsbare gebruikersauthenticatie**
3. **Overmatige gegevensblootstelling**
4. **Gebrek aan beperkingen**
5. **Kwetsbare autorisatie op functie-niveau**
6. **Ontbrekende invoercontrole**
7. **Configuratieproblemen**
8. **Injectiefouten**
9. **Verborgen en verouderde API-endpoints**
10. **Ontoereikende logging en monitoring**