Onderwerp : Beveiligingsrichtlijnen voor API's en webservices

Datum : 14 november 2019

API en webservices beveiligingsrichtlijnen

API's moeten ontwikkeld worden met veiligheid in het achterhoofd en vervolgens ook gedurende productie en onderhoud veilig zijn en blijven. Ontwikkelteams/ontwikkelaars en architecten houden rekening met onderstaande punten als API's en API gateways worden ontwikkeld die vervolgens in beheer worden genomen.

Gebruik

Deze beschrijving kan op 2 manieren worden gebruikt:

1. Als indeling van een referentiearchitectuur over API-beveiliging die een architect maakt
2. Als indeling van een beschrijving waar ontwikkelaars rekening mee moeten houden en benoemen bij het documenteren van API's die zij ontwikkelen. Daarna kan deze lijst gebruikt worden om te toetsen of alle onderwerpen aan bod komen of niet.

Benoemd/gebruikt worden de volgend aspecten door architecten en ontwikkelteams in de documentatie

API design

- Documentatie is up to date
- Gebaseerd op standaarden (bijvoorbeeld RESTful style of SOAP Webservices)
- Performance eisen en maatregelen zijn vastgelegd
- API versiebeheer is geregeld en gedocumenteerd
- Privacy maatregelen zijn bekend door bijvoorbeeld een uitgevoerde DPIA en expliciet gemaakt en beschreven en geïmplementeerd
- Security eisen zijn uitgewerkt op basis van een risicoanalyse en geïmplementeerd
- Deployment eisen zijn vastgesteld en beschreven en beschikbaar

API Aandachtspunten

- Statelessness
- Latency
- Betrouwbaarheid (availability) en Toegankelijkheid (toegangsregels)
- Naamgeving
- Keuzes als HTTP method
- Encryptie

API Authenticatie en autorisatie

- Keuze voor voorkeur methode voor alle API ontwikkelingen of afhankelijk van eerdere keuzes, welke authenticatie en autorisatie standaarden gevpolg worden.

API Implementaties

- Toegangsbeheer
- Aandachtspunten voor bericht toegangs (access) attributen uitgaande van ABAC:
    - Client
    - User
    - Locatie
    - Type
    - Time
    - Size
    - Device
- Netwerktoegang attributen
    - SSL/TLS/Device
    - Netwerk address, range
- Caching
    - Bepalen of en wat er gecached kan worden en hoe dat wordt opgezet
    - Data caching
    - Token caching en token validation cache lifetime (en relatie tot API performance)
    - API gateways, integratie, proxies

API security waar de ontwikkelaar rekening mee moet houden

- Foutafhandeling:
    - API eigenaar gaat over de foutmeldingen
    - API moet de juiste foutmelding teruggeven
    - API foutmeldingen mogen geen gevoelige informatie teruggeven
    - Foutmeldingen moeten in het juiste berichtformaat zijn
    - Foutmeldingen worden centraal beheerd en onderhouden
    - Dezelfde fout moet logischerwijs dezelfde foutmelding genereren
    - Alle foutmeldingen zijn gedocumenteerd (ook in een berichtenboek)
- Encryptie (SSL / TLS)
- Logging
- Authenticatie (user) en autorisatie (client), houdt rekening met inzet in een windows omgeving, denk aan:

    - Windows Kerberos and NTLM
    - Windows Active Directory
    - Azure Active Directory
    - Azure AD B2C
    - X.509 Certificates including mutual SSL
    - WS-* for SOAP including advanced WS-Federation
    - SAML 1.1, 2.0
    - API Security Keys for REST
    - OAuth and OpenID Connect
- Replay attack protection
- Bericht grootte validatie
- SQL-injectie bescherming

Betreft het API gateways, dan zijn API gatewayfuncties tevens beschreven / aandachtspunten (bijvoorbeeld):

- Toegangscontrole is uitgewerkt
- Netwerk beveiliging (TLS) is geimplementeerd
- Bericht beveiliging (encryptie) is uitgewerkt en geimplementeerd
- Bericht validatie en transformatie is uitgewerkt (API berichtenboek?)
- Bericht routering
- API beschikbaarheid
- API Logging is uitgewerkt, er worden geen privacy gevoelige gegevens gelogd
- Bescherming tegen bedreigingen zoals SQL injectie
- Eventueel support voor messaging (MQ)
- Ondersteuning bij toegang tot databronnen (databases)
- API (PEN)testen zijn uitgevoerd en problemen zijn opgelost

Overige aandachtspunten voor cloud-omgevingen

Algemeen

- Support voor deployment in VM's in Cloudomgevingen
- Support voor double byte character set
- Support voor integratie met ESB's

Traffic Management / Run time

- Beperk load op individuele APIs (aka throttling)
- Beperk load gebaseerd op gebruiks patronen over meerdere API's (aka defensive throttling)
- Stel service parameters vast, meet en rapporteer SLA (Snelheid, volume (TPS), latency, ...)
- Lever data voor analyse
- Lever data voor auditable records
- Unified management of at least 3 deployable gateways/nodes (ie. intern, extern, leverancier)
- Regel foutopsporing en afhandeling van bericht fouten en API fouten
- Koppel met een SIEM, Usecase , Events