INFORMATIEBEVEILIGING
MAGAZINE

THEMA: ARCHITECTUUR

- Bouwen aan digital architecture met Lego Serious Play
- Drie loodgieterslessen voor security
- Het beste framework voor security-architectuur
- Berry eindelijk aan de Android?

JAARGANG 19 - 2019 - EDITIE 5

To design or not to design

Deze vraag lijkt ten grondslag te liggen aan de beslissing van het Nederlands Architectuur Forum (NAF) om het door haar georganiseerde Landelijke Architectuur Congres (LAC) te stoppen en daarvoor in de plaats te starten met de Digital Architecture Design Day (DADD).

Het is een bekend gegeven dat een niet-aanpassing (lees niet-verandering) leidt tot extinctie. Blijkbaar geldt dit ook voor congressen. Het NAF wenst – zo is althans onze indruk – van een statische activiteit te switchen naar een dynamische en participatieve digitale architectuur-ontwerp, met aandacht voor verwante vakgebieden.

De iB-redactie vindt dit belangrijk en zet daarom deze ontwikkeling binnen onze Nederlandse IT-wereld in de schijnwerpers en heeft een aantal auteurs benaderd om hun licht te laten schijnen hoe zij de IT-architectuur benaderen dan wel zien. Natuurlijk hebben wij ook het bestuur van het NAF geïnterviewd. Daar zijn wij verrijkt van teruggekomen.

We belichten in dit nummer onderwerpen zoals `Lego Serious Play' en brengen cognitieve- alsook gedragsindicatoren (RTTI & OMZA) en de meningen van de specialisten in kaart. Wij hopen dat onze visie op het belang van een kwalitatief hoogwaardig architectuur-proces daardoor mede gestalte zal krijgen.

Naast dit hoofdthema komen ook de gebruikelijke nieuwtjes uit onze beveiligingswereld weer aan de orde. Daarbij komen wij ook even terug op het Annual Security Report van de eerder gehouden Security Bootcamp, geplaatst naast het Data Breach Investigation Report (DBIR) van Verizon.

Wij wensen jullie veel lees- & ervaringsplezier toe.

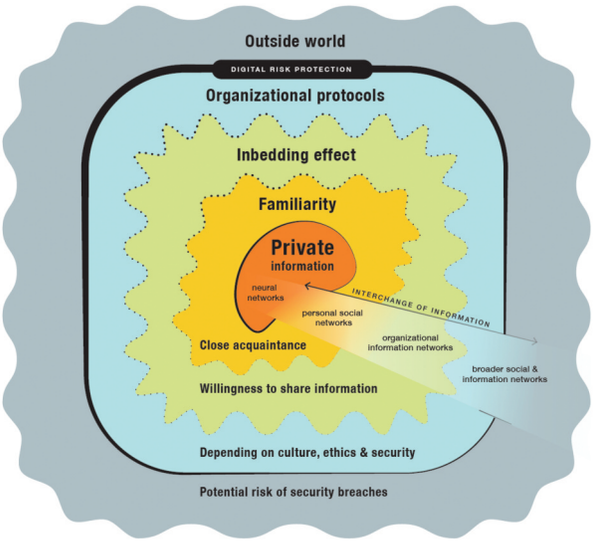
Lilian Knippenberg en Chris de Vries

IN DIT NUMMER

Erik Schoppen is merkexpert, neurowetenschapper, gedragsonderzoeker, innovator, designer, auteur en een veelgevraagd internationaal spreker over vertrouwen en (merk)leiderschap. Erik is bereikbaar via info@erikschoppen.com.

Vertrouwen gaat niet over veiligheid maar over vrijheid

Vertrouwen is een vraagstuk dat vooral ter tafel komt bij de inbedding binnen kwetsbare veiligheidssystemen. Hoe kunnen we het vertrouwen in een systeem vergroten zonder dat de integriteit en veiligheid binnen zo'n systeem schade wordt aangedaan? Aan de ene kant wil je een systeem zo veilig en beregeld mogelijk houden, aan de andere kant geef je gebruikers de vrijheid en autonomie om hier zelfstandig mee om te gaan. Dat vraagt om vertrouwen.



Figuur 1 - From neural networks to digital information networks.

Zonder vertrouwen in gebruikers worden systemen onbruikbaar, mede door de eindeloze protocollen die dan gevolgd moeten worden om informatie te ontsluiten binnen en buiten een systeem. Gebruikersgemak en vertrouwen gaan daarom hand in hand. Een eerste voorbeeld hiervan is de 'single sign-on', een inlogmethode waarmee gebruikers met één ID en wachtwoord toegang krijgen tot verschillende, al dan niet verbonden, systemen binnen een organisatie. Als deze inlog gehackt wordt, liggen alle aangesloten applicaties binnen de organisatie open. Het is vaak niet praktisch en zeker niet gebruiksvriendelijk om voor alles aparte inloggegevens aan te maken. Ons brein is 'cognitief lui' en kiest voor de makkelijkste weg - dus gebruiken mensen vaak dezelfde wachtwoorden. Volgens recent onderzoek van SplashData (2019) was het meest gebruikte wachtwoord nog steeds '123456.' Twee-staps-verificatie verdient daarom de voorkeur, ook omdat er dan ook een bewuste cognitieve handeling tussen zit die routinegedrag of fraude voorkomt. Als je persoonlijk je smartphone moet pakken om een code te activeren, is dit ook een bevestiging die je erop attendeert dat je op dat moment inlogt in een mogelijk kwetsbaar systeem – moeilijker te hacken dus.

Veiligheid begint in ons brein

Het is de uitdaging om in een digitale architectuur een zodanig systeem te ontwerpen dat zowel veilig en betrouwbaar als toegankelijk en gebruiksvriendelijk is. Maar hiervoor moeten we wel eerst begrijpen hoe vertrouwen als mentaal mechanisme een rol speelt in onze besluitvorming. Op basis waarvan nemen mensen hun beslissingen en besluiten ze vervolgens wel of geen informatie te delen? Daarom eerst een introductie over hoe vertrouwen werkt in ons brein.

Evolutionair vertrouwen

Vertrouwen ontstaat in ons brein op moleculair niveau in onze zenuwcellen. Boodschappers in deze cellen zorgen voor informatieoverdracht naar andere cellen die gezamenlijk weer neurale netwerken vormen. Door deze netwerken kunnen we informatie verwerken, onthouden en opslaan. Hoe sneller we toegang krijgen tot bepaalde informatie in deze netwerken - die bijvoorbeeld een aanname bevestigen - des te sneller ontstaat er vertrouwen. Deze neurale 'snelwegen' vormen het fundament voor ons vertrouwen. Ze spelen een rol bij 'familiarity' (het snel kunnen herkennen van informatie waarmee we goed

Digitaal vertrouwen stelt je in staat het gedrag van anderen te voorspellen

bekend zijn of personen die we vertrouwen) en het 'inbedding-effect' (de omgeving waarin informatie wordt aangeboden is bepalend of we deze informatie vertrouwen en of we bereid zijn informatie uit te wisselen), zie ook figuur 1. Hierdoor kunnen we snel beoordelen of we op onszelf moeten vertrouwen of juist iemand anders kunnen vertrouwen. Dit miljoenen jaren oude systeem is niet perfect, maar werkt in de meeste gevallen prima, mede omdat het snel is. Denk aan de eerste indruk die we van een persoon kunnen hebben. We vertrouwen dan op wat we soms al in 200 milliseconden kunnen waarnemen. Hoewel deze beoordeling niet altijd correct is, is deze vorm van (patroon)herkenning vanuit de evolutie goed verklaarbaar. Als er een kwaadwillend persoon op je afkomt, is snel kunnen beslissen van levensbelang. Vaak speelt de omgeving hierin ook een rol – een donker steegje met daarin een verdacht opgesteld individu loop je 's avonds ook niet in. Maar in onze digitale wereld bestaan helaas geen donkere straatjes; we krijgen constant de illusie van verlichting en veiligheid voorgeschoteld. De digitale industrie draait primair om het creëren van virtuele veiligheid, maar cybersecurity en het begrijpen van menselijk gedrag blijft noodzakelijk. Een onbedachtzame handeling is online snel gedaan, vooral omdat één klik al grote consequenties kan hebben en authenticatie zo lastig valt te verifiëren. De tijd die betrouwbare identificatie kost op de digitale snelweg is veelal langer en ingewikkelder dan de kwart seconde die ons brein gemiddeld nodig heeft voor (h)erkenning in de reële wereld.

Netwerkvertrouwen

Vertrouwen in jezelf of in de ander draait er vooral om of je jezelf kwetsbaar durft op te stellen. Deze vorm van sociaal vertrouwen zorgt er van oudsher voor dat we kunnen samenwerken en taken kunnen verdelen. Het uitwisselen van gevoelige informatie binnen digitale netwerksystemen werkt tegenwoordig op exact dezelfde wijze. Weten wat mensen denken en voelen, maar vooral waarop ze vertrouwen, is de sleutel tot het begrijpen van hun huidige gedrag en het voorspellen van hun toekomstige gedrag binnen virtuele omgevingen. Ook omdat deze omgevingen vaak een afspiegeling zijn van de organisatie in de reële wereld. Hoe meer ze hierop lijken, des te sneller ze erop vertrouwen. Vertrouwen ontstaat hier vooral door bekendheid van omgeving en het secuur omgaan met (privacy)gevoelige data en het delen van elkaars waarden en kennis binnen deze netwerken. Als dit overeenstemt, voelt een relatie binnen zo'n systeem vertrouwd en motiveert het mensen om meer informatie over zichzelf te delen. Er is dus sprake van zowel familiarity als een positief inbeddingeffect. Zowel de beveiligde netwerkomgeving als de betrouwbaarheid van anderen in dit netwerk leidt dan tot netwerkvertrouwen. Dit is het vertrouwen dat we in elkaar stellen binnen grotere netwerken waarbij mensen kennisdelende schakels vormen in het netwerk. Elke schakel is belangrijk om interpersoonlijk vertrouwen tot stand te brengen. Daarom is toegang tot goede toegangsprotocollen en een gezamenlijk gedeelde bedrijfsbeveiligingscultuur zo belangrijk. We delen letterlijk elkaars (neurale en sociale) netwerken. Hoe sterker de organisatiecultuur, des te sterker het vertrouwen in de organisatie en daarmee vaak samenhangend het beveiligingsniveau.

Systeemvertrouwen

Ook zijn organisaties en samenwerkingsverbanden in het digitale tijdperk steeds groter en complexer geworden - resulterend in steeds grotere systemen en bijbehorende architectuur. Dit vraagt om vertrouwen in iets wat ons

menselijk brein niet meer kan overzien. Mensen denken het liefst in voorspelbare factoren en nabijheid. Hoe dichter iets mentaal of fysiek bij ons staat, des te meer aandacht we hieraan schenken en hoe sterker het vertrouwen is. Hier maken hackers gebruik van middels familiarity en inbedding. Maar hoe werkt dit in de onlinewereld waarin via globale netwerkstructuren alles met alles aan elkaar verbonden is? In digitale onlinesystemen ben je altijd en overal kwetsbaar voor aanvallen. Dit vereist een ander veiligheidsniveau. Vertrouwen in een systeem kan namelijk uitgroeien tot een wereldwijd niveau bij miljarden mensen.

Systeemvertrouwen draait dan om de kredietwaardigheid en reputatie van allerlei digitale netwerken die gezamenlijk met elkaar verbonden zijn tot één herkenbare entiteit die qua gevoel weer dicht bij mensen staat of hun zorgen wegneemt. Een voorbeeld van dit systeemvertrouwen is als je bijvoorbeeld een volledig verzorgde vakantie boekt. De aanbieder (de entiteit waarop je vertrouwt) boekt voor jou de tickets bij een internationale vliegmaatschappij (die weer afspraken heeft met vliegvelden in verschillende landen), de huurauto bij een verhuurbedrijf in het land van bestemming en het huisje en restaurant op het vakantiepark (die ook weer afspraken hebben met lokale leveranciers). Hier zijn dus allerlei partijen bij betrokken waarmee je geen rechtstreeks contact meer hebt als klant. Je vertrouwt er dan op dat al deze samenwerkende organisaties, die op een of andere manier met elkaar verbonden zijn, de juiste informatie veilig met elkaar delen om jou zorgeloos de dienst te kunnen leveren. Al die partijen vertrouwen erop dat zij beveiligde toegang krijgen tot (delen van) elkaars informatie, want voor alle betrokken partijen in dit systeem geldt: geen veilige en betrouwbare informatie betekent geen vertrouwen en geen transactie. Net zoals dat op microschaal in de neurale netwerken binnen in je brein gebeurt, gebeurt dit op grotere schaal in de wereld daarbuiten. Hoe houd je hier als security officer het overzicht? De principes van familiarity en inbedding gaan hier immers niet meer op. Evenmin de controleerbare identificatieprotocollen binnen een organisatie.

Vertrouwen in (cyber)security

Waar bedrijven ooit zijn begonnen om hun interne netwerken binnen de organisatie te beschermen middels Digital Risk Protection-programma's (denk aan firewalls en identification authentication and authorization) richten zij zich nu steeds vaker op de beveiliging van netwerken en systemen buiten de organisatie. Gezien de complexiteit en schaalgrootte is dit door mensen niet meer te doen. Mede hierdoor vertrouwen we in cybersecurity steeds vaker op kunstmatige intelligentie, simpelweg omdat AI-systemen afwijkingen sneller en beter kunnen herkennen. Maar om het vertrouwen van mensen te krijgen moeten deze systemen hen eerst beter leren begrijpen. Daarvoor bouwen ze dezelfde neurale netwerken na als in jouw hersenen, zodat ze niet alleen sneller patronen kunnen herkennen in je gedrag, maar tegelijkertijd ook dezelfde neurale paden kunnen aflopen. Hierdoor kunnen ze net wat sneller voorspellen wat de gedachten zijn van hacker en slachtoffer en kunnen deze systemen hier adequater op inspelen. In de toekomst zullen AI-systemen steeds beter in staat zijn om familiarity en inbedding te simuleren, wat niet alleen de veiligheid en regelgeving binnen netwerken van organisaties ten goede komt, maar ook de vrijheid en autonomie van hun gebruikers.

In digitale onlinesystemen ben je altijd kwetsbaar voor aanvallen

Niek de Visscher is CEO bij Digital Innovation Benelux. Digital Innovation's Lab/Tech-team werkt als specialist voor o.a. Adidas, Lufthansa en Philips. Niek is bereikbaar via niek.devisscher@digital-innovation.com.

Microservice-architectuur

Voor- en nadelen ten opzichte van monolythische software concepten en een beveiligingsperspectief. Acht praktische beveiligingstips.

Microservices zijn dé trend in de technologiesector. Origineel ontwikkeld door bedrijven als Netflix, Google en Twitter overwegen veel ondernemingen de voordelen van het gebruik van een op microservices gebaseerde architectuur. In een recent onderzoek zei 36% van de CIO's dat ze al microservices hadden geïmplementeerd, terwijl nog eens 26% beweerde dat ze overwogen om microservices in gebruik te gaan nemen (1). Gezien de groeiende populariteit van microservices zou je verwachten dat men zou weten wat de best practices van microservices zijn, maar er blijft veel verkeerde informatie rondgaan over microservices, over het verschil t.a.v. klassieke (monolythische) software architectuur, omtrent beveiliging van microservice architectuur en of deze architectuur goed kan passen bij de eigen organisatie.

Voor een start-up is het uiteraard eenvoudig om vanaf het prille begin van microservices gebruik te maken, omdat start-ups geen last hebben van legacy-applicaties die moeilijk te moderniseren zijn en voor een 'lock-in' zorgen. Voor de grotere onderneming, met veel verschillende toepassingen, elk met hun eigen geschiedenis en doel; kan het uiterst ontmoedigend zijn om het overgangsproces van een monolithische architectuur naar een op microservices gebaseerde architectuur te starten.

De belofte van microservices

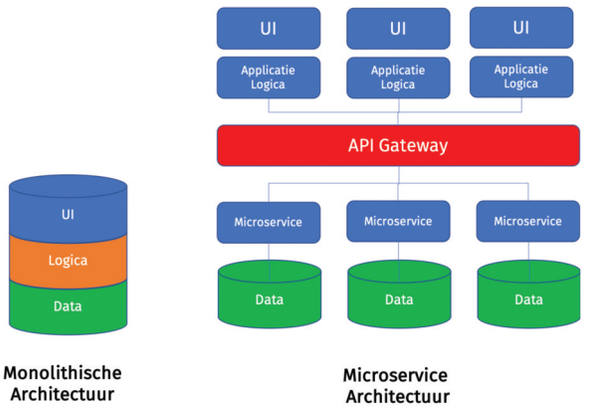
Tien jaar geleden waren onder meer Netflix, Amazon, Twitter, Google en eBay de vroege pioniers van de microservices. Sindsdien genieten microservices de voorkeur bij steeds meer ontwikkelaars om geavanceerde softwaretoepassingen te bouwen. Microservices hebben hun voor- en nadelen, maar ze zijn op grote schaal in gebruik genomen, omdat ze de volgende voordelen bieden:

Wendbaarheid

Compartimentering en gedistribueerde functionaliteit stellen applicatieontwikkelaars in staat om hun functies continu te hergebruiken en te implementeren, onafhankelijk van andere bedrijfseenheden en applicatieteams.

Keuzevrijheid

Ontwikkelaars kiezen hun eigen framework, waardoor ze sneller functionaliteit bouwen en implementeren.



Figuur 1 – Diagramarchitectuur.

Veerkracht

Microservices zijn ontworpen voor het verlagen van redundantie en met het concept van isolatie in het achterhoofd, wat toepassingen robuuster maakt.

Efficiëntie

Ondernemingen die functionaliteit ontkoppelen en microservices gebruiken kunnen aanzienlijke besparingen bereiken. Om beter te begrijpen of een op microservices gebaseerde ontwikkeling en implementatie kan passen, kijken we eerst naar de verschillen tussen de meer traditionele monolithische benadering en microservices.

Monolithische architecturen

Monolithische en microservice georiënteerde architecturen bouwen op zeer verschillende manieren applicaties en producten, elk met hun eigen voor- en nadelen. Tot voor kort, vóór de opkomst van cloudapplicaties, was een monolithische architectuur quasi de enige manier om applicaties te bouwen.

Monolithische architecturen zijn relatief gemakkelijk te begrijpen en te overzien: veel apart 'bewegende' onderdelen ontbreken en alles is meestal opgenomen in één basiscode. Dat is geweldig bij het ontwikkelen van een gloednieuwe applicatie, vooral wanneer de basiscode en het team dat eraan werkt relatief klein zijn. Het ontwikkelt ook op een snelle manier het product en brengt dat snel op de markt, omdat er geen andere afhankelijkheden zijn om over na te denken.

Wanneer de basiscode groeit, wordt het ontwikkelen van software steeds uitdagender

De meeste functies baseren zich op dezelfde Basiscode, die meestal communiceert met één database voor het opslaan van 'stateful'-objecten (de status van een object op een bepaald moment in tijd). Voor het schalen van een monolithische toepassing introduceert men een front-end load-balancer (verdeling van services) die het verkeer distribueert naar elke service van de monolithische toepassing. Het was gebruikelijk dat de front-end-inhoud ook door de server werd gegenereerd, totdat duidelijk werd dat een dergelijke architectuur de backend servers te veel zouden belasten, wat het op zijn beurt moeilijker zou maken om verder te schalen. Dat was in de tijd dat applicaties begonnen met het implementeren van een aantal clientside functionaliteiten. Ad-hoc, backend API's werden geïntroduceerd die door de clients konden worden gebruikt om een deel van het werk van de servers weg te halen. Niet zo lang geleden werden AJAX en JSON de "de facto standaard" voor client-servercommunicatie, vooral met betrekking tot browser-based clients. JSON, in tegenstelling tot XML, parst veel makkelijker in Javascript, genereert makkelijker de serverkant en leest beter voor een mens. Met de toenemende implementatie van clients was er een manier nodig om externe partners aan boord te krijgen en/of een ontwikkelaarsplatform te bouwen. API's werden de standaardmanier om software te bouwen volgens de 'API first'-manier van denken. Als de basiscode klein is, zijn ook de eventuele uitdagingen die voortvloeien uit deze architecturale keuze klein. Naarmate de codebasis groter wordt, ontstaan er echter problemen. Monolithische applicaties zijn een goede manier om met een nieuwe applicatie te beginnen en deze naar de markt te brengen. Iteraties gaan relatief snel, omdat ontwikkelaars via eenvoudige 'blue-green deployment' continu verbeteringen naar de eindgebruikers kunnen verzenden. Wanneer de basiscode groeit, wordt het ontwikkelen van software steeds uitdagender. Zelfs kleine wijzigingen vereisen volledige herschikkingen van de gehele applicatie en als er iets misgaat, kan dit vaak de hele applicatie beïnvloeden. In dit scenario wordt wat een voordeel was een nadeel: de bedrijfslogica is gebundeld en ontwikkelaars kunnen de wijzigingen niet goed isoleren en compartimenteren naarmate de basiscode groter en groter wordt. Een groeiende basiscode komt met eigen uitdagingen: het bijhouden van schone code-abstracties, gedocumenteerde code en good practices kunnen de 'technical debt' in de loop van de tijd helpen verminderen, maar goed codebeheer wordt lastiger. Nieuwe teamleden proberen hun eigen code te pushen naar productie en kunnen zo de fundamenten van monolithische systemen in de loop van de tijd doorbreken en het proces frustreren, omdat functies steeds moeilijker te bouwen worden - of lijken te mislukken - zodra ze in productie gaan. De hoofdoorzaak van deze problemen is dat het erg moeilijk is om een grote basiscode georganiseerd en schoon genoeg te houden. Ook ontbreekt er een manier om code en bedrijfslogica te isoleren dan wel gecompartimenteerd feature-versiebeheer te garanderen. Dit terwijl verschillende teamleden tegelijk bijdragen aan verschillende delen van dezelfde basiscode, die uiteindelijk in één keer in alle omgevingen worden ingezet.

Microservice-architecturen

Het lijkt daarom logisch dat het bouwen van kleinere applicaties (in plaats van één enkele grote applicatie) een oplossing voor deze uitdagingen vormt. De vereisten zijn ook duidelijk: deze componenten moeten onafhankelijk gebouwd en ingezet worden. Het allerbelangrijkst is dat elk teamlid, dat aan een component werkt, niet noodzakelijk de nuances van andere componenten hoeft te kennen, aangezien elk implementatiedetail wordt verzorgd door dat dedicated team dat aan die specifieke functie werkt. Omdat elk applicatie-onderdeel zich niet bewust is van de implementatie van andere componenten, moet er een manier zijn om met interfaces of API's te werken. En aangezien ontwikkelaars elk versieverzoek moeten kunnen routeren en zelfstandig moeten kunnen implementeren alsook schalen, gebruikt men deze interfaces via een netwerk. Dit is het punt waarop 'componenten' 'diensten' worden. Omdat elk van hen een specifieke functie heel goed zal doen, vallen zij vaak veel kleiner uit. Dit is de reden

waarom het 'microservices' worden genoemd. Zoals we hebben gezien, zijn monolithische en microservice architecturen twee fundamenteel verschillende manieren om software te ontwerpen en te bouwen, elk met eigen voor- en nadelen. In deze sectie zullen we deze plussen en minnen samenvatten. Laten we beginnen met monolithische architecturen.

De voor- en nadelen van een monolithische aanpak

Het grootste voordeel is dat er weinig bewegende onderdelen zijn, wat voor kleine basiscodes ook betekent dat er eenvoudiger mee kan worden omgegaan. Meestal wordt de toepassing achter een load-balancer uitgevoerd en elke keer dat een ontwikkelaar een wijziging aanbrengt, werkt men de basiscode bij en implementeert men de volledige toepassing opnieuw. Het schalen van de applicatie kan horizontaal worden gedaan door meer codes achter de load-balancer te plaatsen. Teams kunnen updates effectief herhalen en snel implementeren. Ze besluiten om één of meerdere keren per dag te implementeren in vaste schema's. Terwijl het team blijft itereren en de basiscode verbetert, kunnen ze die wijzigingen continu aan de eindgebruiker leveren met blue-green-implementaties of canary releases. Het hebben van maar enkele bewegende onderdelen maakt het eenvoudig om integratietests uit te voeren in een monolithische applicatie, omdat alles op dezelfde plek staat en het team niet te veel afhankelijkheden moet testen om functies volledig te beproeven. Hoogstwaarschijnlijk is de database de enige afhankelijkheid. Al deze voordelen verdwijnen snel als de applicatie complexer wordt. Een groeiende basiscode is moeilijker om mee om te gaan. Dit geldt vooral voor nieuwe medewerkers, omdat we nu het grote geheel ineens moeten begrijpen, waardoor de leercurve steiler wordt dan gewenst.

Zelfs als men de code goed beheert (met technische good practices en een goede algemene 'engineeringcultuur'), wordt het inzetten van een grote basiscode steeds langzamer, omdat we ervoor moeten zorgen dat alles bij elkaar blijft. Ook moeten we controleren of onze wijzigingen niet onverwacht van invloed zijn op andere gebieden van de basiscode. Experimenteren met nieuwe technologieën, zoals nieuwe databases of talen, wordt bijna onmogelijk, omdat ontwikkelaars de experimenten niet goed kunnen isoleren en daarmee de hele applicatie in gevaar brengen. Om deze redenen innoveren monolithische toepassingen steeds moeilijker als ze in omvang toenemen. Zelfs een kleine wijziging in de basiscode vereist een volledige herdistributie van de gehele applicatie en dat geldt ook als men bibliotheken maakt om de basiscode te ontkoppelen. Elke wijziging aan de bibliotheek vereist nog steeds een volledige implementatie. Daarnaast leiden negatieve factoren bij het bouwen of verzenden van een product uiteindelijk ook tot frustratie voor zowel het management als het technische team, omdat de releasecycli langer worden en de innovatiesnelheid vertraagt. De algemene ervaring voor het team is dat elke nieuwe verandering of functie exponentieel moeilijker te implementeren en uit te brengen is. Het moreel en momentum van het team is een sleutelfactor bij het ontwikkelen van een goed product en als de architecturale keuzes een negatieve invloed hebben op hoe het team werkt en de software verzendt, dan is dat iets om voor uit te kijken.

Voor- en nadelen van microservices

De ware gedachte - achter op microservices gerichte architecturen - is om verschillende gebieden van de bedrijfslogica te delegeren aan afzonderlijke services die onafhankelijk kunnen worden gemaakt, ingezet en geschaald. Bij het bouwen van een applicatie is het mogelijk om verschillende gebieden of grenzen te identificeren die te maken hebben met afzonderlijke bedrijfslogica. Bijvoorbeeld: in een e-commerceproduct vindt je aparte functies die te maken hebben met: order-, factuur-, gebruikersbeheer, voorraad enzovoort. In een op microservice gerichte toepassing begint een architect met het loskoppelen van elk van deze functies als een afzonderlijke service, zodat men een nieuwe functionaliteit apart kan implementeren, zonder alle andere functies te beïnvloeden. Door voor elk van deze functies verschillende services te ontwikkelen, kunnen teams nu onafhankelijk beslissen wanneer ze hun eigen services implementeren en opschalen. Deze services communiceren met elkaar via een interface, zoals een HTTP/RPC API. Zolang de interface van een andere service kan worden gebruikt, kan het team microservices in verschillende talen bouwen, verschillende datastores gebruiken en binnen de context van een service experimenteren met nieuwe technologieën zoals een nieuwe database, een nieuw framework of een nieuwe taal. Microservices maken daarom ook een meer geleidelijke leercurve voor nieuwe medewerkers mogelijk , omdat ze, zoals de naam al aangeeft, relatief klein zijn en zich toespitsen op het uitvoeren van maar een paar taken.

Tenslotte bouwden microservices in de architectuur isolatie en schaalbaarheid in. Als één service uitvalt, blijven de

andere services nog steeds actief zonder de hele applicatie offline te halen. Als een dienst opeens een toename of afname van verzoeken ervaart, kan deze horizontaal worden op- of neergeschaald zonder de andere services te beïnvloeden. Op dezelfde manier wordt een service die is aangetast, geïsoleerd en offline gehaald zonder de hele applicatie te beïnvloeden of te infecteren. In microservice georiënteerde architecturen ruimen grote teams het veld voor kleinere 'pizzateams'. Deze naam komt voort uit het idee dat je een dergelijk team één grote pizza te eten moet kunnen geven. Dit leidt meestal tot een teamgrootte van zeven of acht ontwikkelaars. Elk team draagt nu de verantwoordelijkheid voor het bouwen, verzenden en schalen van de services die ze onafhankelijk van elkaar onderhouden. Natuurlijk genereren deze voordelen ook extra kosten waar architecten zich bewust van moeten zijn. Microservices introduceren veel meer bewegende onderdelen die moeten samenwerken om het eindresultaat te leveren, net als een orkest. De orkestratievereisten worden een stuk belangrijker, omdat het team ervoor moet zorgen dat al deze services correct worden gebruikt. Monitoren wordt moeilijker en ingewikkelder dan het monitoren van maar één applicatie. Terwijl gegevens zich verplaatsen tussen elke service, is het cruciaal om de consistentie van statussen en de beschikbaarheid van de gegevens te waarborgen. Hoewel het eenvoudiger is om een specifieke service te testen, wordt het moeilijker om het volledige beeld te testen, omdat al deze services tegelijkertijd moeten werken.

Microservices en security

Door het gebruik van microservices worden sommige beveiligingsuitdagingen echter wel nog een stuk lastiger om aan te pakken. Er zijn daarnaast ook enkele functies van microservices die de beveiliging juist kunnen versterken. Ondanks microservices is 'het netwerk' nog steeds een risico. Een aspect zoals toegangscontrole, die we voor monolithische toepassingen al goed begrijpen en beheersen, veroorzaakt een nieuw, bijna onverwacht, niveau van complexiteit. Dit maakt de weg vrij voor discussies over de vraag of een microservices-architectuur daadwerkelijk meer problemen oplost dan dat deze creëert. Een beslissing om microservices te gebruiken, moet daarom altijd voorwaardelijk zijn: wanneer u uw vooronderzoek hebt gedaan en hebt besloten dat microservices geschikt voor u zijn, moet u ervoor zorgen dat aan alle beveiligingsvereisten van uw applicaties wordt voldaan. Hier zijn acht praktische tips voor het beveiligen van uw microservices.

1. Gebruik OAuth voor gebruikersidentiteit en toegangscontrole

De overgrote meerderheid van de applicaties zal een bepaald niveau van toegangscontrole en autorisatie moeten uitvoeren. OAuth/OAuth2 is zo goed als de industriestandaard voor wat betreft gebruikersautorisatie. Hoewel het bouwen van je eigen aangepaste autorisatieprotocol uiteraard een optie is, is dat niet aan te raden, tenzij je daarvoor goede en zeer specifieke redenen hebt. Alhoewel OAuth2 niet perfect is, is het dus een algemeen aanvaarde en mature norm. Het voordeel hiervan is dat u kunt vertrouwen op bibliotheken en platformen die uw ontwikkelingsfase en betrouwbaarheid/robuustheid van uw product aanzienlijk verbeteren en versnellen.

2. Identificeer je belangrijkste applicatieonderdelen en bescherm ze middels defense in depth-beveiligingsmaatregelen

Ervan uitgaan dat een firewall op uw netwerkgrens voldoende is om uw software te beschermen, is een grote fout. 'Defense in depth' wordt gedefinieerd als: 'Een concept voor informatiebeveiliging waarbij meerdere lagen van beveiligingscontroles (defensie) in een informatietechnologiesysteem worden geplaatst.' Dit betekent dat u moet identificeren wat uw meest gevoelige services zijn en een aantal verschillende beveiligingslagen daarop moet toepassen, zodat een potentiële aanvaller die één van uw beveiligingslagen kan exploiteren, nog steeds een manier moet achterhalen om al uw andere verdedigingen op uw kritieke services te breken. Beveiliging is een taak die beter aan deskundigen kan worden overgelaten. Een werkende en juiste strategie voor 'defense in depth' is beter haalbaar als deze wordt opgezet door mensen die echt weten wat ze doen. Het mooie van microservices is dat ze het gemakkelijker maken om deze strategie op een zeer gedetailleerde en strategische manier uit te werken - door uw beveiligingsinspanningen op specifieke microservices te richten. De architectuur maakt het ook eenvoudiger voor u om de verschillende beveiligingslagen te diversificeren die u op elke microservice wilt toepassen. Hierdoor kan een aanvaller die één van uw services kan exploiteren, niet noodzakelijkerwijs weten hoe hij de tweede kan openbreken.

3. Schrijf geen eigen cryptocode

In de loop der jaren investeerden veel mensen ongelooflijke hoeveelheden geld, tijd en middelen in het bouwen van bibliotheken die codering en decodering uitvoeren. Als het

gaat om beveiliging, is het beter om niet eigen nieuwe oplossingen en algoritmen te bedenken, tenzij er goede en specifieke redenen zijn om dit te doen - en de capaciteiten hebt om dit uit te kunnen voeren. In de meeste gevallen kan NaCl/libsodium gebruikt worden voor codering. Het is snel, gemakkelijk te gebruiken en veilig. Hoewel de oorspronkelijke implementatie van NaCl in 'C' is geschreven, ondersteunt het ook C++ en Python en dankzij de libsodiumvork zijn verschillende adapters voor andere talen zoals PHP, Javascript en Go beschikbaar. Dit gedeelte zou niet compleet zijn zonder de enorm
populaire Bouncy Castle-bibliotheek te vermelden. Als u met Java of C# werkt, is deze bibliotheek het onderzoek en overwegen waard.

4. Gebruik automatische beveiligingsupdates
Als u wilt dat uw microservices-architectuur tegelijkertijd veilig maar ook schaalbaar is, is het een goed idee om - in een vroege ontwikkelingsfase - een manier te vinden om al uw software-updates te automatiseren of op zijn minst onder controle te houden. Het gebruik van geavanceerde, diepgaande en automatische testing is hier belangrijker dan ooit tevoren. Telkens wanneer een deel van uw systeem wordt bijgewerkt, wilt u ervoor zorgen dat u problemen zo vroeg als mogelijk detecteert en zo goed mogelijk oplost. Zorg ervoor dat uw platform 'atomair' is. Dit betekent dat alles in containers is verpakt, zodat het testen van uw toepassing met een bijgewerkte bibliotheek of een aangepaste taalversie gewoon een kwestie is van het opzetten van een andere container. Als de bewerking mislukt, is het terugdraaien van alles relatief eenvoudig en, belangrijker nog, kan het worden geautomatiseerd. CoreOS, Atomic Linux van RedHat en Snappy Core van Ubuntu zijn in deze context projecten die je in de gaten wilt houden, omdat ze proberen hetzelfde concept op OS niveau tot stand te brengen.

5. Gebruik een gedistribueerde firewall met gecentraliseerde controle
Voor het grootste deel is dit nog onbekend terrein, maar een firewall waarmee gebruikers meer gedetailleerde controle over elke microservice hebben, is een goede manier waarop we firewallconcepten voor microservices kunnen realiseren.

6. Haal uw containers uit het openbare netwerk
Amazon, met hun AWS API-gateway, heeft dit hele idee mainstream gemaakt en is gemakkelijk te gebruiken. Een API-gateway zorgt voor één toegangspunt voor alle aanvragen van alle clients. De gateway weet vervolgens hoe het een interface ophaalt voor alle aan die gateway verbonden microservices. Door deze techniek te gebruiken, beveiligt u al uw microservices achter een firewall, zodat de API-gateway externe service-aanvragen kan verwerken en vervolgens met de microservices achter de firewall kan communiceren. Bovendien is het gebruik van een API-gateway een geweldige manier om aanvragen op basis van de client te optimaliseren, vooral in het geval van mobiele apparaten.

7. Gebruik beveiligingsscanners voor uw containers
Het is logisch en aangewezen om - in uw geautomatiseerde Testtooling - beveiligingsscans voor al uw containers op te nemen. Iets anders om rekening mee te houden is dat de containerimage (software met alle codes, tools & resources) zelf niet noodzakelijkerwijs kan worden vertrouwd, tenzij de handtekening is geverifieerd.

8. Monitor alles: met een tool
U kunt het zich niet veroorloven om een gedistribueerd systeem te gebruiken zonder een solide, geavanceerd en betrouwbaar monitoringplatform te gebruiken. Er zijn hiervoor verschillende open source en commerciële oplossingen beschikbaar.

Vind het wiel niet opnieuw uit

Hoewel het bovenstaande niet bedoeld is als een uitputtende lijst, gaat het in op de problemen waarmee u waarschijnlijk te maken krijgt bij het bouwen van applicaties op basis van een microservices-architectuur. Als het gaat om beveiliging, is "het wiel opnieuw uitvinden" zelden een goed idee. Onderzoek altijd de best practices die door de industrie zijn overgenomen en door experts worden voorgesteld.

Referenties:

(1) Onderzoek Digital Innovation AG, juni 2018 op een populatie van 78 CIO'suit Duitsland, Oostenrijk & Zwitserland.

Pascal de Koning is senior cybersecurity consultant en is werkzaam voor onder andere de Johan Cruijff ArenA en Vattenfall. Pascal is te bereiken via pascal@dekoning-cybersecurity.nl.

Het beste framework voor security-architectuur

Er bestaan verschillende frameworks voor security-architectuur. De belangrijkste zijn: SABSA (1), O-ESA (2) en OSA (3). Elk framework heeft zijn eigen sterke punten, waardoor ze elkaar goed kunnen aanvullen. Het ontwikkelen van een effectieve security-architectuur is een uitdaging. Als je gaat zoeken in bestaande frameworks, dan blijkt die vaak nét niet te passen in de eigen situatie. Het begrip security-architectuur kent vele gedaantes en elk framework heeft zijn eigen focus en sterktes. Hoe maak je uit deze diversiteit de beste security-architectuur?

De belangrijkste eerste stap is het vaststellen welke knelpunten je denkt op te lossen met de security-architectuur. Wat is het doel? Wanneer is het een succes? Wie moet het gaan gebruiken? Dit soort vragen zijn essentieel om een security-architectuur te laten slagen. Hier komt een vierde framework van pas: TOGAF (4). De genoemde vragen worden uitgebreid behandeld in de preliminary phase. Hier besluit je of het architectuurproject de moeite waard is. Het is in mijn ogen de allerbelangrijkste fase van het project. Het maakt nogal verschil of een security-architectuur een manier is om jarenlang de securityrisico's te beheersen, of dat het een one time exercitie is om koers te bepalen en na afloop wordt weggegooid.

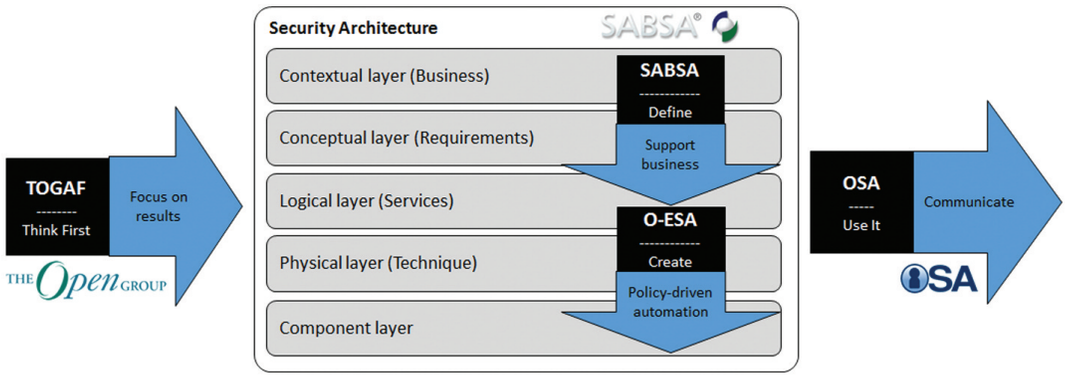
Figuur 1 - Verschillende architectuurframeworks in perspectief geplaatst op basis van hun sterke punten.

Selecteer de delen die je gaat ontwikkelen

Een mooi overkoepelend kader voor een enterprise security-architectuur wordt gegeven door SABSA. Het heeft een holistische benadering: van businessdoelstellingen tot aan het laatste bitje. Dit leent zich prima als kapstokmodel, waarmee in de preliminary phase kan worden bepaald welke componenten wel of niet nodig zijn. Het gaat hier nog puur om de wat-vraag, nog niet om de hoe-vraag, die komt later. De benodigde componenten worden geplot op het architectuurlagenmodel. Dit lagenmodel uit SABSA is zeer sterk door eenvoud en is herkenbaar voor veel mensen. Op deze wijze wordt een eerste houtskoolschets van de security-architectuur gemaakt.

Business alignment in de bovenste lagen

Voor het daadwerkelijk invullen van de lagen of het ontwikkelen van de componenten in de security-architectuur biedt SABSA ook hulp, met name in de bovenste lagen (van businesscomponenten tot logische componenten). Uniek en sterk is het model om security requirements te managen: het Business Attribute-model. Eveneens sterk en onmisbaar voor business commitment is de drang om de security business-driven in te regelen. "Wij doen security om jou te laten slagen in je bedrijfsdoelstelling", wordt er gezegd. Met deze insteek wordt security gelinkt aan business en is het nooit moeilijk om akkoord te krijgen voor de securityplannen.



Figuur 2 - Overzicht van samenhang van security-architectuurraamwerken.

Technische uitwerking in de onderste lagen

De onderste lagen in de security-architectuur hebben betrekking op functionaliteit en techniek van de security-maatregelen. Hier gaat het om de 'echte' security, zoals toegangsbeheersing, systeem-hardening, securityscans, security-awareness, enzovoort. Het framework wat op dit vlak echt onderscheidend is, is O-ESA.

O-ESA zoomt in op policy-driven architectuur, een uitwerking van de visie dat operationele beveiliging digitaliseert. Operationele risicobeslissingen worden steeds meer dynamisch genomen en daarom is het nodig om security policies te automatiseren. Veel aandacht is er voor toegangsbeheersing en securitymonitoring. Anders dan SABSA, waar een security policy een maatregel is die nog gedefinieerd kan worden, gaat O-ESA ervanuit dat er al een security policy bestaat. Ondanks zijn naam is O-ESA dus een framework voor de onderste architectuurlagen.

Communicatie met de rest van het bedrijf

Als de security-architectuur eenmaal staat, inclusief de operationele security services, dan is er nog een laatste stap te zetten: zorgen dat deze gebruikt gaat worden door de rest van de organisatie. Security zou (net als bijvoorbeeld finance) een integraal onderdeel moeten zijn van alle andere processen, diensten, architectuur, enzovoort. Om dit te bereiken is communicatie nodig. Mensen zijn visueel ingesteld. Hier is in OSA goed gebruik van gemaakt. OSA bevat bijvoorbeeld een icon library die helpt om uit te beelden welke beveiligingsmaatregelen nodig zijn in een bepaalde IT-context. Bijvoorbeeld wanneer een klant een website bezoekt welke verbinding heeft met de backoffice. De icon library is terecht het meest populaire component van OSA. Eén plaatje zegt meer dan 80 bladzijden tekst, vooral als het bedoeld is voor mensen die security niet zo heel interessant vinden (ja, die zijn er ook en meer dan je denkt).

Performance en risicobeheersing

Als het doel is om over langere tijd risico's te beheersen, dan zal een metric-component moeten worden toegevoegd. Dit zal in ieder geval rekening moeten houden met impact, dreiging en kwetsbaarheid. In de Conceptual Layer heeft SABSA de Key Risk Indicators gepositioneerd, waarbij de link met businessimpact wordt gemaakt. De link met het dreigingenlandschap is echter nauwelijks uitgewerkt. De kwetsbaarheid van de omgeving zou mijns inziens kunnen worden bepaald aan de hand van performancemetingen van security services op de Logical Layer. Mogelijk wordt hiervoor in de toekomst nog eens een praktisch framework gemaakt.

Conclusie

Door van tevoren goed na te denken over doelstellingen van de security-architectuur en wat er nodig is om dat te bereiken, wordt een helder beeld gegeven over de vorm ervan. Vervolgens kan worden gekeken naar de bestaande frameworks welke dit het sterkst invult. Er is geen silver bullet framework, maar door goed voor ogen te houden welk probleem je wilt oplossen, is er altijd wel een bruikbaar framework te vinden die sterk is waar jij het nodig hebt en die je helpt om het doel te bereiken.

Referenties:

(1) www.sabsa.org
(2) https://www2.opengroup.org/ogsys/jsp/publications/-PublicationDetails.jsp?publicationid=12380
(3) www.opensecurityarchitecture.org
(4) www.togaf.org

Dit artikel is geschreven door Chris de Vries, Lilian Knippenberg en Sandra Kagie. Chris de Vries en Lilian Knippenberg zijn redactielid van iB-Magazine. Sandra Kagie is freelance tekstschrijver (Sanscript Tekstproducties).

Fotograaf: Fotoproducties Arjan Smalen.

INTERVIEW

Bas van Hengstum en Peter Beijer van NAF:

‘Dé IT-architect bestaat niet meer’

Veranderde dynamiek moet op DADD tot uiting komen

"Na twintig jaar was het nodig te vernieuwen", stelt Bas van Hengstum namens het Nederlands Architectuur Forum (NAF). En dus gaat het NAF samen met congresorganisator CKC Seminars op 14 november iets nieuws doen. Samen organiseren ze de Digital Architecture Design Day (DADD), als opvolger van het Landelijk Architectuur Congres (LAC). NAF-voorzitter Peter Beijer en bestuurslid Bas van Hengstum, tevens lid van de programmacommissie DADD, kijken uit naar de eerste editie van DADD in Den Bosch.

"De dynamiek van ons vak als architect in de digitale wereld, de IT-architect, is veranderd. We hebben binnen organisaties steeds vaker te maken met digitale transformaties waar 'agile teams' en 'lean werken' de norm is. Dat vraagt om een ander soort architect. Een architect die uit zijn ivoren toren komt en bereid is met zijn laarzen in de klei te staan. Die veranderde dynamiek moet op DADD tot uiting komen", trapt Peter Beijer af.

Hij verwijst in zijn betoog naar de theorie van Simon Wardley, die met zijn Wardley Maps de evolutie in volwassenheid van innovaties beschrijft aan de hand van drie fases: pioneers, settlers en town planners. De afgelopen twintig jaar hebben de meeste architecten het vak volgens Beijer benaderd als 'town planner'. Dat geldt volgens hem ook voor veel informatiebeveiligers. "Regelgeving, policies, procedures, processen, functionarissen. Alles keurig geordend." Maar de 24/7 digitale economie, gebaseerd op agile en lean, vraagt wat hem betreft dus om een heel ander soort IT-architect.

Spanningsveld in de praktijk

In de praktijk ziet Beijer binnen bedrijven een groot spanningsveld tussen de IT- en informatie-afdeling enerzijds en de business mensen anderzijds. "De IT- en informatie-afdeling worden gezien als traag, log, traditioneel en als de clubs van de grote projecten. Terwijl de businessmensen snel iets willen kunnen uitproberen."

Om dit spanningsveld te doorbreken heb je volgens hem een IT-architect nodig die snelheid in het verhaal brengt. "Een anarchist binnen boundaries", typeert hij. "Je hebt enige discussie nodig om uiteindelijk daar te landen waar iedereen achter staat. Dan kan een zeurende IT-architect soms heel irritant zijn. Maar hij kan ook ogen openen", vult Bas van Hengstum aan.

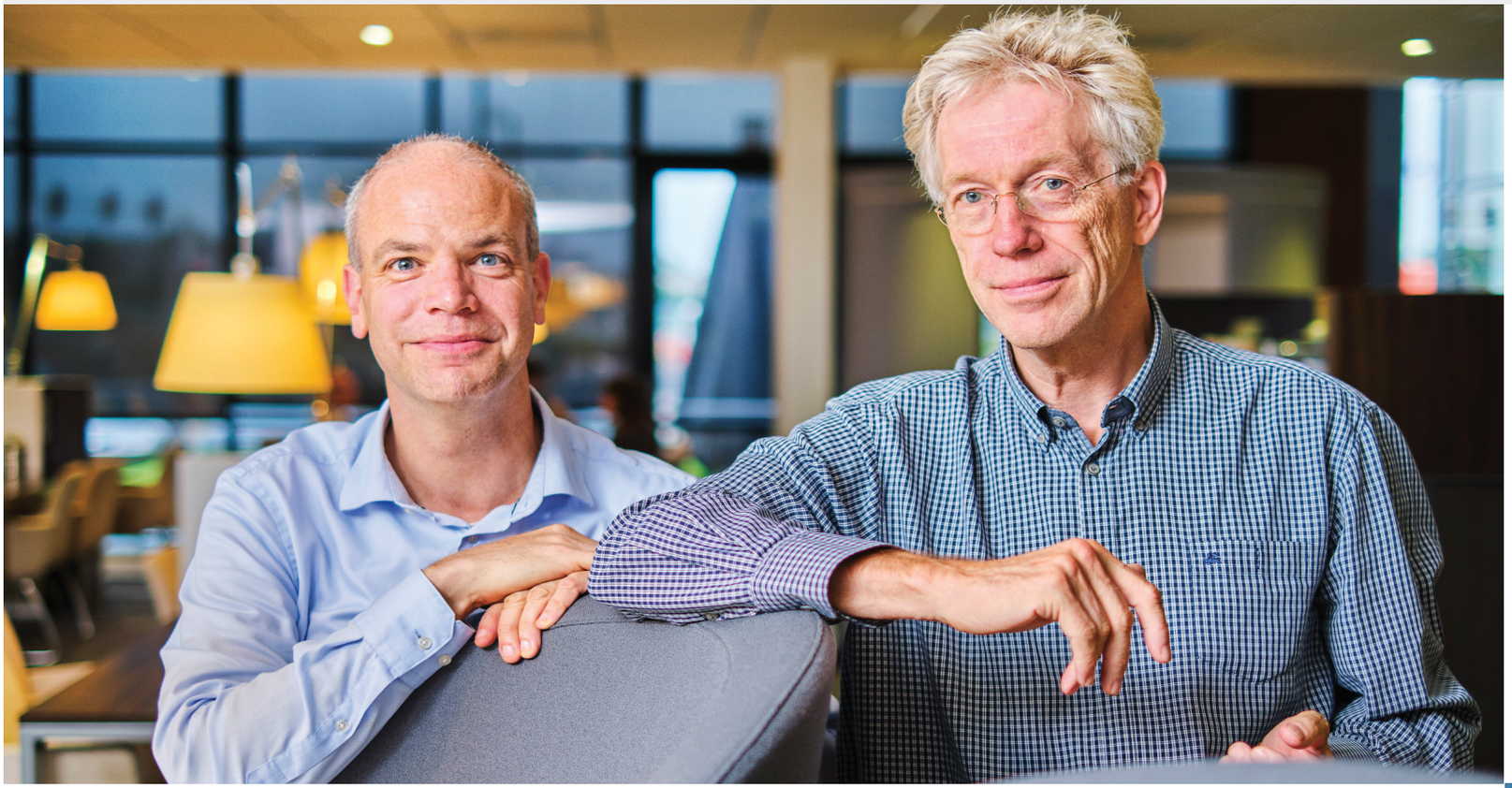
Beijer ziet overigens geen rol voor IT-architecten in de eerder genoemde pioneerfase. Dat is wat hem betreft 'contrasterend met het architectuurdenken'. Maar hij ziet dus wel mogelijkheden voor IT-architecten die aan de slag gaan met een idee om daar een minimal viable-product van te maken. "En als er dan opgeschaald moet worden qua productiecapaciteit dan geven zij het stokje over aan een enterprise architect." Met andere woorden: dé IT-architect bestaat niet meer, er zijn meerdere architectuurrollen van belang.

"In agile teams zit vaak geen fulltime architect", haakt Van Hengstum in. "Maar je ziet wel dat andere teamleden bewust of onbewust architectuurachtige rollen op zich nemen. Die mensen willen we ook meenemen in het architectuurvakgebied. We richten ons met DADD dus niet alleen op architecten, maar juist ook op aanpalende rollen", benadrukt hij. "We willen begrip voor elkaar kweken. En we hopen dat DADD daaraan een bijdrage kan leveren door nieuwe interacties tot stand te brengen."

Wanneer is DADD geslaagd?

"Een andere dialoog op gang brengen in Nederland over architectuur, wanneer we daarin slagen, is DADD wat mij betreft geslaagd", concludeert Beijer. "Een dialoog over digitalisering en de digitale economie. Een dialoog met andere vakgebieden ook. Van architect naar architectuur", vat hij samen.

"Meer diversiteit, een bredere bezoekersgroep qua achtergrond en leeftijd, en meer vrouwen. En in plaats van alleen



hoorcolleges juist meer interactieve werkvormen tijdens het congres. Wat de invulling van de dag betreft willen we dus echt de creatieve kant op. Waarbij we natuurlijk hopen dat de trouwe LAC-bezoeker zich hierbij ook thuis en welkom voelt. Wanneer we die verschillende doelen weten te bereiken, is DADD wat mij betreft een succes", stelt Van Hengstum.

Samenwerking als rode draad

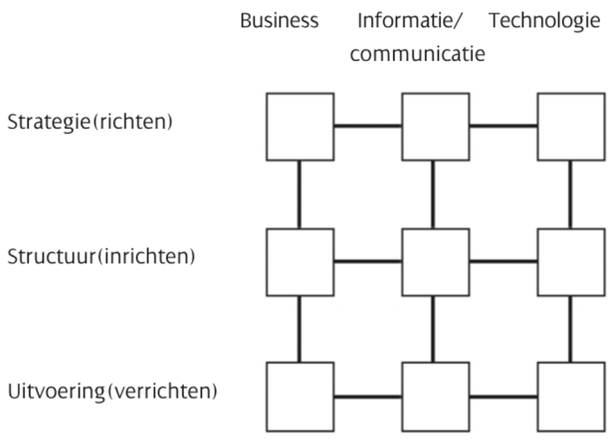
Samenwerking binnen het vakgebied én met andere vakgebieden, het is een thema dat als rode draad door het gesprek met beide heren loopt. Het is volgens hen dan ook een belangrijke doelstelling van het NAF om middels intervisie, kennisdeling, een digitaliseringsbenadering én door de ontwikkeling van nieuwe bedrijfsmodellen te bouwen aan een verdere professionalisering van het vakgebied architectuur. Met als grote ambitie het vak van IT-architect of digitaal architect erkend te krijgen, zoals het vak van een architect in de bouw. Zo wil het NAF bijvoorbeeld toe naar een register waarin onderhouden van kennis en kunde geborgd wordt, vergelijkbaar met het BIG-register in de geneeskunde. "Nu noemen veel mensen zich IT-architect terwijl ze het niet zijn. En andersom. Het profiel is nog niet vast omkaderd en dat zou het wel moeten zijn", vindt Beijer. Heel belangrijk hierin is wat hem betreft samenwerking met de wetenschappelijke onderzoeks- en onderwijswereld. "Samen moeten we aan de slag om het vakgebied van IT-architect verder te verduidelijken. En uiteraard met de ontwikkeling van opleidingen die hierbij aansluiten."

Maesbruggen-thematiek

Tot slot haken we met de heren van het NAF nog in op de Maesbruggen-thematiek, waarover we als PvIB inmiddels drie themabijeenkomsten hebben georganiseerd (de meest recente was afgelopen juni). Bijeenkomsten waarin we hebben geprobeerd oplossingen te vinden voor de communicatiekloof die bestaat tussen het management enerzijds en zij die het werk doen anderzijds (de operatie). Je kunt dit verbeelden als een diagonale kloof door het Negenvlaksmodel van Maes heen, van linksonder naar rechtsboven.

Kan een IT-architect een rol spelen in het overbruggen van deze kloof is de vraag die we de heren voorleggen. "Deze architect zou die bruggenbouwer moeten zijn", denkt Van Hengstum. "Maar door de operatie zal hij vaak nog worden

gezien als iemand die aan de andere kant van de kloof staat. In plaats van dat hij hem overbrugt."
Het model van Maes staat in de 'nieuwe wereld' volgens Beijer onder druk. Het traditionele 'layered-denken' maakt in zijn ogen binnen nieuwe type organisaties, digitale businesses, steeds meer plaats voor Domain Driven Design. "De Amazon-manier van werken waarbinnen een klein clubje van mensen gewoon alles doet. Als naar de buitenwereld maar dezelfde taal gesproken wordt. Dit middels een Application Programming Interface (API) met schaalbaarheid als fundamenteel succescriterium", legt hij uit. "Bedenken betekent binnen dit nieuwe type organisaties dus ook repareren. Terwijl het model van Maes juist onderscheid maakt tussen richten, inrichten en verrichten. Kom op zeg! Dit nieuwe denken vraagt om een heel andere cultuur."



Figuur 1 - Het negenvlaksmodel.

"Een cultuur die in Japan 'catch ball' wordt genoemd. Kenmerkend hierbinnen is de korte link tussen het upper management en de medewerkers op de werkvloer. Het senior management zegt dat ze een idee hebben. Een paar individuen binnen de organisatie snappen het en zij gaan er vervolgens mee aan de slag, waarbij management en de werkvloer elkaar durven te challengen. Wanneer je mensen op zo'n manier bij elkaar brengt, krijg je heel snelle bewegingen. Dat moet je eens met het een organisatie volgens het Negenvlaksmodel proberen."

TSTC

ICT en Security Trainingen

Fast Track Certified Information Systems Security Professional CISSP
9–13 december 2019

Fast Track Certified Cloud Security Professional CCSP
2–6 december 2019

Fast Track Certified Data Protection Officer CDPO
9–13 december 2019

Fast Track Certified Chief Information Officer C|CISO
4–8 november 2019

www.tstc.nl

Vereenvoudig je risicoanalyse

Al je documentatie op 1 plek

Optimaliseer je operationele planning

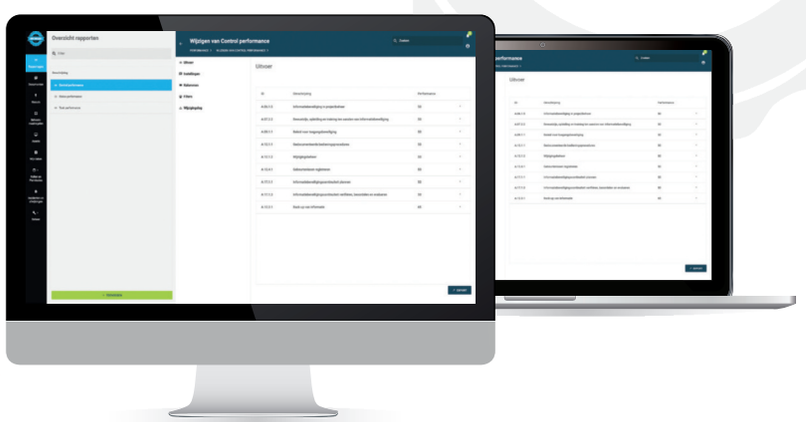
Maak flexibele rapportages

Dit en nog veel meer is mogelijk met ISOToolkit
Kijk en ervaar het gemak zelf via:

ISOTOOLKIT.NL
Probeer nu **30 dagen gratis**

ISO TOOLKIT

ISOTOOLKIT:
Complete en eenvoudige software voor je ISMS



COLUMN PRIVACY

Mr. Rachel Marbus
@RACHELMARBUS OP TWITTER

Privacy doe-het-zelven

Na een heerlijke lange zomer met veel zon in binnen- en buitenland zak ik weer ontspannen achter de laptop om een beetje door het nieuws te scrollen. Ook privacy heeft zomervakantie gehad en heel veel nieuws is er niet. De grote zaken waren al wel doorgedruppeld tijdens het smeren van de zonnebrand en het nippen van een cocktail. Ik zoek naar die kleine pareltjes waarvan je een glimlach om je mond krijgt. Die berichtjes die de voorpagina zelden halen, zelfs in de zomer niet als de komkommers rijp zijn.

Er is een prachtige wet, een steeds meer aanwezige toezichthouder en de meeste (zelf respecterende) organisaties hebben tenminste wel een iemand in dienst die iets van privacy weet en klanten, burgers en werknemers kan helpen. Maar ook die hulpmensen kunnen niet altijd de beste weg naar een bevredigend resultaat bewerkstelligen. Want zo nu en dan helpt de wet niet echt en kun je beter een beetje gaan doe-het-zelven. Zo dacht ook Zihni Özdil, voormalig Tweede Kamerlid, toen hij merkte dat de Kamer van Koophandel (KvK) nog steeds heel hardnekkig de privacy van zzp'ers schendt.

Nu heeft de KvK een ongelooflijk beroerde reputatie als het op de privacy van haar zelfstandige klanten aankomt en heeft de Autoriteit Persoonsgegevens daar al een flinke tik op de vingers voor gegeven. Maar de hunkering naar geld blijft zeer hardnekkig, Özdil kreeg na flink wat doordrammen te horen dat de KvK jaarlijks 45 miljoen euro verdient met het doorverkopen van persoonsgegevens. Dat drammen maakte deel uit van het privacy doe-het-zelven van Özdil. Het voormalig Tweede Kamerlid wilde zich als zzp'er inschrijven bij de KvK en drong erop aan dat zijn gegevens geheim moesten blijven. Er zijn zeer ernstige reële bedreigingen aan zijn adres dat geheim is in het GBA en voorzien is van een rode vlag door het NCTV. De jurist van de KvK vond het allemaal wel meevallen met die dreiging en keurde het verzoek tot geheimhouding af. Pas na een flinke Twitter-campagne en het verschijnen van Özdil in diverse media trok de KvK haar besluit in.

Ook buiten Nederland werd het heft in eigen hand genomen. Porn Wikileaks onderhield een website over pornoacteurs en actrices en vermeldde daarbij ook een enorme berg aan zeer persoonlijke gegevens over de acteurs zoals hun echte namen, woonadres en gegevens over gezinsleden. Pornoproducent Bang Bros had het helemaal gehad met de privacyschendende website en kocht de hele santekraam op. De site werd acuut offline gehaald en de harde schijven werden ceremonieel in brand gestoken. Voelt het rechtvaardig dat de gegevens en de veiligheid van de acteurs en hun familie gegijzeld werden? Nee, natuurlijk niet. Maar de boel opkopen en in rook laten opgaan was het enige snelle effectieve middel. (Amerikanen hebben ook nog eens geen GDPR, dus daar hadden ze nog minder dan hier wat zou kunnen helpen). Soms moet je zelf drastische middelen inzetten om gehoord te worden en effect te verkrijgen. Waar het kan help ik graag, maar soms kan ik heel weinig betekenen en dan juich ik het doe-het-zelven graag toe!

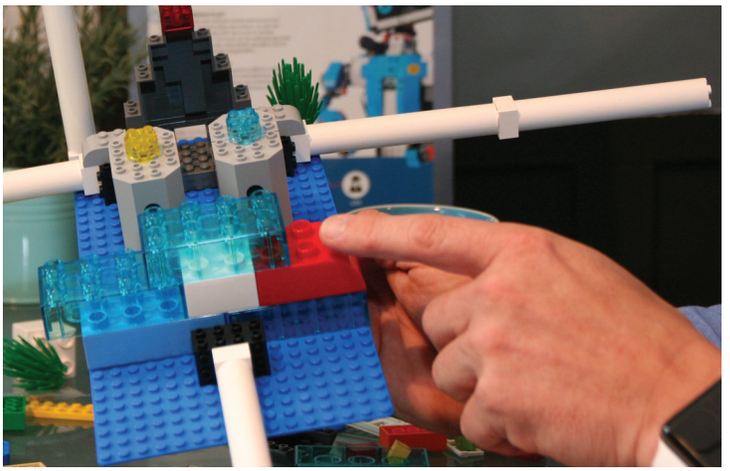
Rachel

Martin van der Poel is met zijn onderneming Zeta IT & Security actief op gebied van ICT/IT. Maarten van der Boon helpt vanuit zijn bedrijf Novitek bedrijven in de ontwikkeling van strategie, innovatie en communicatie.
Hij is gecertificeerd facilitator van de Lego Serious Play-methode. Maarten is bereikbaar via maarten@novitek.nl.

Bouwen aan digital architecture met LEGO SERIOUS PLAY

Hoe een internationale methodiek bijdraagt aan een 3D-visualisatie van netwerkarchitectuur.

Organisaties zijn omgevingen waarin mensen volgens allerlei processen en systemen werken. Dat was vroeger al zo en dat is nu nog steeds zo. Het verschil is dat er dankzij de digitale revolutie veel geautomatiseerd is met inzet van hardware, software en randapparatuur. Van omvangrijke ERP-systemen tot handige apps en van energieslurpende servers tot aan handzame smartphones. Wanneer iets bijdraagt aan het proces (of nog beter: de productiviteit) dan zijn er direct mensen te vinden voor de toepassing en het gebruik ervan. De afgelopen 15 jaar is de complexiteit rond digitale infrastructuur enorm toegenomen.
De digitale transitie is in volle gang.



Lego verbindt hoofd en handen naar een voor iedereen begrijpelijk verhaal (foto: Maarten van der Boon).

Individuele bouwmodellen worden met elkaar verbonden tot een sluitende visie, strategie of digital architecture (foto: Maarten van der Boon).

In de stormachtige ontwikkelingen binnen het IT-landschap is de toe te passen netwerkarchitectuur een veelbesproken onderwerp. Het belang van een toereikend netwerk is evident aanwezig voor iedere organisatie. Toch lees je regelmatig berichten over ontoereikende netwerken: netwerken die de snelle ontwikkelingen niet aankunnen, onveilige netwerken of over implementatietrajecten die budgetten ruimschoots overschrijden. Wanneer er wat dieper doorgevraagd wordt, ligt in nagenoeg alle gevallen de oorzaak niet in het netwerk an sich. Vaak zijn de boosdoeners te vinden in achterliggende keuzes en besluiten.

Het roept de vraag op via welke weg de juiste informatie is te ontvangen om een passend en toekomstgerichte netwerkarchitectuur te ontwerpen. Juist in deze zoektocht kan Lego Serious Play een belangrijke bijdrage leveren. De auteurs van dit artikel presenteren u op praktische wijze hoe de werelden van digital architecture en die van de 'Serious Play' succesvol kan worden samengevoegd. Het resultaat? Duidelijke en nagenoeg tijdloze richtlijnen en toetsstenen waarop architectuur ontwikkeld en ontworpen wordt.

Enterprise architecture is organiseren van voorwaarden

Het is misschien een open deur, maar waar staat het woord architectuur eigenlijk voor? En welke relatie heeft het dan naar IT? Het woordenboek Van Dale omschrijft architectuur kort maar krachtig als 'bouwkunst'. Andere bronnen beschrijven het als kunst en wetenschap van het ontwerpen van gebouwde omgevingen. Niet iets waarbij je direct denkt aan IT, toch? Om die link te maken, is het goed nog iets verder terug te gaan in de geschiedenis. Daar lees je over de Romein Vitruvius en zijn verhandeling Architectura.

Vitruvius probeerde het ethos van de architectuur te pakken door te verklaren dat de kwaliteit afhangt van de maatschappelijke relevantie van het werk van de kunstenaar en niet van de vorm of de afwerking van het werk zelf. Volgens de Romein is architectuur gestoeld op drie principes: 'schoonheid' (venustas), 'stevigheid' (firmitas) en 'bruikbaarheid' (utilitas). Architectuur wordt door hem omschreven als de balans tussen deze drie elementen. Hij leert ons vervolgens dat geen van drieën overheersend is ten opzichte van de andere twee. En die stelling wordt vandaag de dag nog steeds volop gebruikt in een vrijere vertaling: 'Goed bouwen heeft drie voorwaarden: stevigheid, gebruiksgemak en genoegen.' De parallel met de enterprise netwerkarchitectuur mag hiermee duidelijk zijn.

Enterprise architecture draait om balans

Onder het begrip 'enterprise architecture' is een samensmelting gaande van de verschillende disciplines als bedrijfskunde, informatiekunde en informatica. Enterprise architecture is daarmee niet alleen een technische aaneenschakeling van hard- en software en communicatiesystemen. Het gaat ten diepste om de juiste balans tussen techniek, functionaliteit en mens, maar dan wel een balans onder drie voorwaarden: betrouwbaar, beschikbaar en bedienbaar.

Enterprise architecture vereist strategie en organisatie

Een goede architectuur heeft een gelaagde opbouw. Vaak in haar eenvoud uitgelegd als een piramide met vier lagen. De

piramidepunt vormt de business architectuur. De onderste laag geeft de technische architectuur weer. Daarboven volgen de applicatie- en data-architecturen als lagen. Deze weergave toont duidelijk aan dat de technische architectuur volledig afhankelijk is van de functionele wensen, geformuleerd in de 'business'. Kortom, gebaseerd op dat wat 'de organisatie' strategisch nodig heeft om succesvol te opereren.

Het is dus van belang om vanuit verschillende gezichtspunten strategisch naar de organisatie te kijken en inzicht te verkrijgen in de manier waarop deze is georganiseerd en functioneert. En dat in een context van demografische, economische, sociale, technologische, ecologische en politieke ontwikkelingen (DESTEP). Strategie en organisatie gaan hand in hand met de missie, visie en propositie(s) van een organisatie. Een goede, toekomstvaste architectuur is dus niet begrensd op techniek, maar laat zich voeden door duidelijke enterprise strategy. En hier is een directe verbinding te maken naar de internationale methodiek van Serious Play.

Het verhaal achter LegoSerious Play

Het idee achter deze methode is ontstaan in 1996, toen de twee professoren Johan Roos en Bart Victor bij IMD in Zwitserland en Kjeld Kirk Kristiansen, CEO en eigenaar van Lego, op zoek gingen naar nieuwe instrumenten en systemen rond strategievorming. Jarenlang onderzoek, analyseren, testen en documenteren volgden. Steeds scherper kwam voor ogen hoe Lego ingezet kan worden om vanuit 3D-modellen businessvraagstukken te beantwoorden. En in 2010 werd besloten de methodiek in een community-based-model aan te bieden. Gedurende de jaren van onderzoek zijn er verschillende publicaties uitgebracht over de resultaten en methodiek. Op de website is een overzicht te vinden. Overigens, de eerste publicatie werd gedaan door een onafhankelijk onderzoeksbureau: Imagination Foundation Lab. Tot op de dag van vandaag verschijnen er regelmatig publicaties die de kracht van de methode aantonen.

Actief en innovatief ontwikkelen

Elk gezond bedrijf is bezig met een scala aan vraagstukken in het heden, de nabije - of verdere toekomst. Denk hierbij aan knel- of verbeterpunten, de organisatiestrategie, de innovatiekracht of vraagstukken van een totaal andere aard. Bijvoorbeeld rond de persoonlijke ontwikkeling van medewerkers of teams. Natuurlijk mag het ontwerpen van een enterprise architecture ook in deze opsomming genoemd worden. Wat deze vraagstukken onderling gemeen hebben, is dat er uren over gepraat, vergaderd en gerapporteerd kan worden. De hier besproken methode brengt daar verandering in. Met beter, leuker en vooral blijvend resultaat.

Ontwapenend boeien, motiveren en binden

De kracht van Lego Serious Play is 100% deelname en inbreng van iedereen aan de tafel en 100% draagvlak in de resultaten van de tafel. Iedere deelnemer, ongeacht rang of stand, is in staat om zijn of haar kennis en mening in te brengen met behulp van de bekende Deense bouwsteentjes. Ideeën, gedachten of wensen worden zo in 3D uitgebouwd en onderling gedeeld.

De laagdrempeligheid van deze methode zorgt ervoor dat deelnemers hun onderbewuste gaan aanspreken. Medewerkers gaan op een speelse wijze elkaar informatie delen. Dat doet een ieder onderling, in verhaalvorm, over alleen datgene wat ze zelf gebouwd hebben en wat de ander dus kan zien. Zo wordt ons natuurlijke 3D-beelddenken vertaald naar heldere verhalen met 3D-ondersteuning vanuit het bouwwerk. Het bijkomende voordeel: dat wat gedeeld wordt, blijft ongekend lang 'hangen' in het collectieve geheugen. De bouwmodellen maken onderwerpen en vraagstukken concreet, tastbaar en veranderbaar. De gestructureerde gefaciliteerde manier van werken zorgen voor het inhoudelijk zien én delen van wat er écht bedoeld wordt. Het leidt tot een set van 'guiding principles' waaraan elk volgend idee, plan, ontwerp – en in de context van dit artikel de architectuurkeuze - aan getoetst kan worden. Voldoet het aan alle principles, dan is met zekerheid te zeggen dat een design sustainable is.

Met Lego bouwen aan digital architecture

Eerder is de stelling neergelegd dat de basis van digital architecture ligt in een duidelijke enterprise strategy. Dit - nader onderzocht - betekent dat er duidelijkheid moet zijn over welke ambities de organisatie nastreeft. Allerlei bedrijfskundige vragen dienen dan beantwoord te zijn met de daarbij horende duidelijke bedrijfsprocessen. Is digital architecture dan alleen een kwestie van vertalen en updaten wat je al hebt? Dan gaat de volgende uitspraak op: 'Als je doet wat je deed, krijg je wat je kreeg.' Om een goede digital architecture te krijgen heb je meer nodig. Meer in de kennis van de ontwikkelende organisatie in relatie tot de eerder genoemde DESTEP-factoren.

Concreet betekent dit dat er antwoorden nodig zijn. Antwoorden op vragen in gebruikers-, management- en bedrijfsomgevingen en de daarbij horende bedrijfskritische factoren. Om antwoorden te krijgen, zijn er duidelijke vragen nodig. En personen die deze vragen beantwoorden. Ter illustratie vertalen we dit naar een onderwerp als 'veiligheid'.

Of een architectuur veilig is of niet hangt af van allerlei diepere afwegingen. Er wordt in veiligheid gesproken over een scala aan deelonderwerpen als toegankelijkheid, beschikbaarheid en stabiliteit. Maar ook over de keuze van hard- en software op netwerkniveau alsmede de uitgebreide keuzemogelijkheden in randapparatuur. Bij veel organisaties wordt dit omgeven met allerhande government policies, waar rekening mee moet worden gehouden.

Architecture follows structure follows strategy

Het vraagstuk over veiligheid, wat in de vorige alinea's geschetst staat, zou met Lego Serious Play uitermate goed behandeld kunnen worden. In de praktijk betekent dit het samenstellen van een organisatiebreed team van betrokkenen. Omdat de techniek niet centraal staat, maar het gaat om het krijgen van inzicht en het achterhalen van wensen, is kennis van het ICT/IT-werkveld geen voorwaarde om deel te nemen.

Het start met het uitbouwen van de organisatiestrategie. Wanneer het strategiemodel op tafel staat, is de verdiepingsslag te maken. Eerst door op hoger abstractieniveau op onderzoek uit te gaan wat de strategische uitleg is van 'veiligheid'. Om vervolgens nog dieper de randvoorwaarden te bepalen op de genoemde deelonderwerpen. Op tafel ontstaat er na verloop een omvangrijk bouwwerk van Lego. Door een sterk gefaciliteerd programma omvat het een 3D-visualisatie van de benodigde en gewenste functionaliteiten binnen de architectuur. Iedere deelnemer aan tafel weet exact wat er bedoeld wordt. Door het toepassen van serious gaming-technieken worden er scenario's uitgespeeld op het bouwwerk. Eventuele zwakke plekken in de architectuur worden direct zichtbaar en kunnen afdoende hersteld worden.

Handvat naar gedragen digital architecture

Als laatste stap binnen de methodiek worden de guiding principles achterhaald waarop het ontwerp van de architectuur wordt gebaseerd. Meestal zijn dit 4 tot 7 principles, vrij vertaald: de gedeelde waarden. Op deze waarden zijn ontwerpkeuzes te maken. Iedereen heeft meegeholpen om die te bepalen. Dat maakt de acceptatiegraad significant groter bij de verdere uitvoering en design van de architecture. Tegelijkertijd dienen de principles als waarschuwing. Als iets in de uitvoering niet aan alle guiding principles voldoet, weet je vooraf dat je er niet aan moet beginnen. Doe je dit wel, dan is de kans groot dat er later problemen ontstaan in de acceptatie en implementatie van de architectuur. Deze vorm van Serious Play faciliteert effectief én efficiënt alle processen om tot scherpe en duidelijk geformuleerde ontwerpuitgangspunten te komen. Met het resultaat kan de volgende stap in het opzetten van digital architecture gezet worden.

100% betrokkenheid en belang

In tegenstelling tot vele andere methodes om de ontwerpeisen van een digital architecture te bepalen, betrekt Lego Serious Play iedere betrokkene voor de volle 100%. Ieders eigen belang komt ook op tafel. In logische stappen wordt een architectuurontwerpvraagstuk beantwoord. De deelnemers bouwen steeds met de steentjes hun gedachten uit in een model en lichten dit toe. Iedereen ziet en begrijpt wat iemand bedoelt. Elk individu spreekt vanuit zijn of haar eigen belang. Het gaat niet meer over iets abstracts. En alles heeft betekenis. Elk kleurtje, steentje, vormpje of stapeling. Bewust of onbewust. Als iemand een poppetje met grijs haar neerzet, kan hij verwijzen naar oudere werknemers. Hij kan ook zeggen dat hij af en toe grijze haren van iets of iemand krijgt. In groepsverband wordt er zo gewerkt aan een 'gedeeld model' dat door alle deelnemers wordt gedragen. De individuele bouwwerken voegen we samen tot één bouwmodel dat iedereen begrijpt. Je weet dus nooit van tevoren waar je uitkomt. Serious play is dus niet bedoeld als inbeddingsinstrument voor een visie die de directie of het managementteam heeft bepaald.

Combinatiekracht

De methode-ontwikkelaars ontdekten in de praktijk dat wanneer beleid of besluiten 100% voldoen aan de gedeelde waarden er ook 100% draagvlak is. Bij alle werknemers, zowel in betrokkenheid als in het persoonlijk belang. Ook werd er ontdekt dat er – via de gedeelde waarden – veel sneller en gerichter gehandeld werd bij grote veranderingen. Naast creativiteit en verbeeldingskracht riep de funkant een positief gevoel op. Dat zagen de ontwikkelaars als een bijkomend voordeel in het beantwoorden van bedrijfsvraagstukken. Beleid vormen werd leuk in plaats van de soms saaie en lange vergaderingen, vermoeiende discussies en duimdikke rapporten.

In het ontwerpen van een digital architecture ligt het niet direct voor de hand om Lego in te zetten via deze Serious Play-methode. Maar wil je als organisatie voorkomen dat digitale infrastructuur begrensd wordt in een technische discussie over wat nodig is, biedt de methode wel een gefundeerd alternatief. Een alternatief wat zorgt dat er zichtbare verbindingen en duidelijke relaties worden gemaakt. Relaties tussen de organisatiestrategie, haar informatie- en applicatieomgevingen en de benodigde technische architectuur. En dat in de juiste balans en toekomstvast. Betrouwbaar, beschikbaar en bedienbaar.

Chris de Vries is redacteur van iB-Magazine.
Hij is bereikbaar via impuls@euronet.nl.

Twee actuele visies op security risks

Tijdens de Security Bootcamp keek Eward Driehuis, chief research officer (CRO) van SecureLink, naar de trends van de afgelopen jaren. Tot 2013 toonde het cybersecuritylandschap zich redelijk rustig met misdadigers die achter de banken aangingen en banken die zich daartegen probeerden te verdedigen.

In 2013 veranderde dat door de introductie van ransomware, opgevolgd in 2017 door Wannacry (eerste verdachte: Noord Korea) en Notpetya (eerste verdachte: Rusland), waarbij in de laatstgenoemde gevallen data niet teruggegeven werd. Eward Driehuis merkte daarover op: "Het waren vermoedelijk staatsactoren en het ging dus om spionage."

SecureLink constateert op basis van haar cijfers over 2018 dat er circa 21.240 veiligheidsincidenten plaats hadden gevonden bij haar klanten, waarbij 'malware' 45% en netwerk- applicatie-afwijkingen 36% uitmaakten. Een aan te halen conclusie: grotere organisaties zijn veiliger, omdat kleinere bedrijven makkelijker meegenomen worden in breedschalige aanvallen en vaak minder goed verdedigd zijn, zie navolgend overzicht.

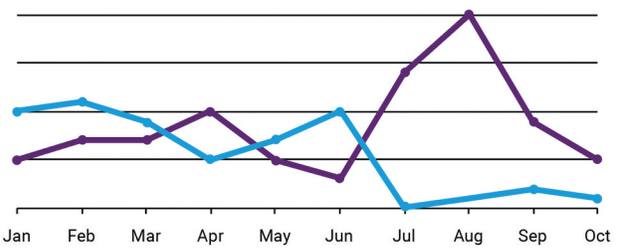
Figuur 1 - Overzicht aanvalsfactoren per 100 medewerkers.

In de maanden november en december 2018 veranderden bovenstaande cijfers aanmerkelijk. Kleinere organisaties zagen hun aanvallen per 100 medewerkers stijgen van 6,8 tot 9,1 en daarmee waren de incidenten bij kleinere ondernemingen 6 keer hoger dan bij de grotere!

Andere constateringen:

- Cryptojacking - ook wel kwaadaardige cryptomining- is een online bedreiging die zich verbergt op een computer of mobiel apparaat en de rekenkracht gebruikt om vormen van online geld te 'delven' het niveau van ransomware nadert.
- Ervaren criminelen stappen eerst in het netwerk. Daar observeren zij en zetten pas ransomware in na vernietiging van de online back-up en er sprake is van een toename van geopolitieke risico's.

Dit laatste verklaart dan ook de recente discussie rondom het 5G-netwerk en HuaWei. In de VS is dat geen discussie meer: de overheid verbiedt het en het bedrijfsleven twijfelt daar niet aan.



Figuur 2 - Overzicht ransomware versus cryptocurrency miner aanvallen.

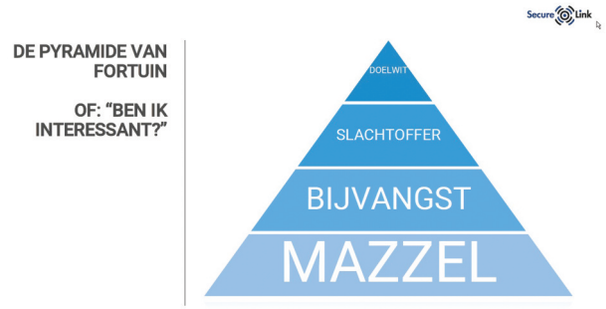
Voor de gewone man (MKB-bedrijf) onder ons is het belangrijk te realiseren dat hij wellicht niet het direct interessante, beoogde doel is, maar wel leuke bijvangst is. Figuur 3 van SecureLink verduidelijkt dit. Twee belangrijke oorzaken met betrekking tot de hier bovengenoemde incidenten/aanvallen:

- niet installeren van patches (bekend euvel);
- single factor-authenticatie in de cloud (inbreuk garantie).

Verdere constateringen uit het rapport: technologie verandert en de risico diversiteit neemt toe. De eindconclusie luidde dat men in de basis moet investeren en dat er beter in de architectuur gesegmenteerd dient te worden. Het rapport van SecureLink is overzichtelijk en goed leesbaar.

Data Breach Investigations Report (DBIR)/Verizon

Een persbijeenkomst met Alex Pinto, de nieuwe leider van het DBIR team. Op 26 juni was er een klein select gezelschap uitgenodigd voor een bijeenkomst over het al 12 jaar bestaande DBIR en het kennismaken met de nieuwe chef Alex Pinto op de mooie locatie Amsteldijk te Amsterdam in het nieuwe kantoorpand Amsteldok. Samen met de



Figuur 3 - Piramide van fortuin.

regiomanager Benelux, David van den Berg, werd er een ronde tafel presentatie gegeven over de ontwikkelingen in de cybercriminaliteit, enkele trends en de werkprocessen bij Verizon.'

Verizon detecteert netwerkbedreigingen door geautomatiseerde analyse van de afwijkingen in het netwerk voor bedrijven over de gehele wereld. Het betreft een concentratie van Amerikaanse bedrijven, omdat daar de meldingswetgeving steviger is. Sinds de invoering van het GDPR in Europa is ook hier een stijging van meldingen te constateren. Kernsectoren: gezondheidszorg en de overheidsadministratie.

De aard van de incidenten verschuift zich van pure geldkwesties naar meer privacygevoelige zaken. Omgevingsfactoren worden bij de analyse meegenomen. Op basis van de dataregistraties zijn de onderwerpen van discussie bepaald op: actoren, acties, gecompromitteerde bezittingen en attributen zoals integriteit, beschikbaarheid, enzovoort.

Het rapport 2019 is gebaseerd op 41.686 veiligheidsincidenten, waarvan 2.013 bevestigde data-inbreuken. De belangrijkste slachtoffers zijn: MKB-bedrijven (43%), gevolgd door de publieke sector (16%), gezondheidsorganisaties (15%) en de financiële sector ('slechts' 10%).

Op middellange termijn zoekt Verizon strategische partnerschappen met verzekeringsmaatschappijen. Bij het MKB zoekt Verizon het vooral in franchise partnerschappen en dan met name in de leveringsketen alwaar de secundaire slachtoffers te vinden zijn. Voor 2020 is de benadering van deze markt hun grootste doelstelling. Daarbij zal de 'supply chain' dus onderzocht worden naar specifieke patronen in relatie tot de cloudvraagstukken.

Qua incidenten constateert Verizon nog een overtrekking van de rapportages vanuit de gezondheidszorg en dan met name aangaande ransomware. Dat omdat de gezondheidszorg alles te melden heeft (groot en klein, alsook indien het geen echte inbreuk was, maar een fout) en dat leidt tot een 70% score. Verizon stelt in haar rapport dat ransomware effectiever zal blijken te zijn dan cryptomining, maar dat dit laatste risico wel groeiende is.

Als belangrijkste trend constateren zij de incidentengroei vanuit de cloud-based e-mails. Men verwacht wel een drievoudige stijging en ziet deze ontwikkeling terug bij alle partners op eenzelfde niveau. Bij deze incidenten spelen webapplicaties en het gebruik van gestolen credits een rol.

Verizon constateert als doelen met betrekking tot veiligheidsinbreuken: financiële motieven (71%) en het behalen van strategisch voordeel/winst (spionage - 25%). Daarnaast werden inbreuken in 56% van de gevallen pas na meer dan één maand ontdekt, was phising in 32% van de gevallen het toegangsmiddel en werd in 29% van de gevallen de identiteit/authenticatie-gegevens gestolen. De algemene aanbeveling die door Alex Pinto tijdens dit gesprek is meegegeven is: verdergaande en intensievere bewustmaking van medewerkers voor phising dreigingen.

Een andere dreiging is spionage, waarvan 80 tot 90% waarschijnlijk toe te schrijven is aan staatsactoren. In het rapport komt geen specificatie voor met betrekking tot de staten (overheden) die zich hier vooral mee bezighouden. Wel wat hun motieven zijn zoals financieel (fluctuaties tussen 95% tot gemiddeld rond de 70% over de jaren 2010 tot en met 2018) en spionage (van nagenoeg 0% tot stijgende naar boven de 25% over eenzelfde periode).

De belangrijkste actoren bij veiligheidsincidenten zijn volgens Verizon: georganiseerde misdaad (de belangrijkste, maar ogenschijnlijk relatief afnemend tot rondom 40%), de staat-gelieerde partijen (stijgende tot circa 30%), systeemadministrateurs (stijgende tot net onder de 20%), activisten en kassiers (beide laatstgenoemden nagenoeg 0%). Verizon rapporteert verder dat er een fundamentele wijziging optreedt en e-commerce meer en meer het doel wordt ten gunste van betalingskaarten. De weg naar de e-commerce gaat over de ATM en benzinestation dan wel POS-skimming. Dat toont zich door een daling van de niet-WebApp-serverinbreuken van nagenoeg 100% tot ergens tussen de 60 en 70% versus de WebApp-serverinbreuken welke stijgen van 0% naar bijna 50% en dat alles over de periode van 2015 tot en met 2018

Het DBIR presenteert ons een ander beeld van de risico's dan het SecureLink-rapport. Dit heeft deels te maken met een andere marktoegang alsook een andere benadering van de wijze waarop en de detaillering met betrekking tot onderwerpen waarover gerapporteerd wordt. Het maakt juist de vergelijking tussen beide rapportages interessant. Volgend jaar zullen wij als redactie een meer directe vergelijking trachten na te nastreven. Vaak is het gezamenlijk geschetste beeld het meest interessante.

Digitally architected

In the world of digital transformation, things are moving quickly. The concept of zero trust has matured significantly. Now we can see a much more holistic approach, often referred to as the 'software defined perimeter'. One of the key features is the use of secure APIs for machine-to-machine communications. Many web applications are now bundles of microservices rather than a single core service. The move away from monolithic applications was driven by the need for agility of the business to avoid digital disruption. It has also become an advantage in securing the applications by segregating each and every function and procedure with its own API. This API-driven architecture has little human intervention. Automation of app-to-app interaction allows applications to embed services from other sources and to expose raw data via the API. It is a very powerful technique in cloud environments, both private and public. 'Compare-the-Market' type web services are built like this, drawing data from multiple sources and collating it to present product and service ratings to human users. One of the best examples is TripAdvisor. The power of this architectural technique can also be its downfall. LandMark White is (was?) Australia's largest independent property valuation firm. In February 2019 it announced that one of its online platforms had been compromised and that more than 100,000 records had been accessed by unauthorised third parties. The records related to property valuations, information on borrowers, lenders, homeowners and property agents. The reputational and financial damage has been huge. The problem was that an API-based architecture had been poorly designed and implemented. With so many cloud applications being of this type it is essential to use a zero trust model in which nothing is taken for granted.

External parties can participate in your applications through APIs but without direct access. The API is an application-level access broker. User access may be via direct web browsing or app-to-app via APIs with no explicit user present.

The principles of zero trust that are needed to make this work are:

- There is never any implied or absolute trust – in every instance you must calculate the trust level based on a variety of factors.
- All environments must be considered public and hostile – exposed to anyone and everyone.
- Strong identification, authentication and authorisation must be applied to all entities, whether human or not.
- Contextual authorisation must be used and must depend on dynamically changing risk factors, including identity, role, group membership, user attributes (such as trained and qualified), consent information (such as age), behaviour, history, time and day, device type, connection type, location, sensitivity of the data, reputation of the application, how frequently, how much, session life and refreshment period, application-specific rules (such as bank employee and customer) and more.
- Trust is dynamic – a contract between two or more entities that want to do business, valid only in the current instance – not static – shifting as time passes and context changes.
- Consider endpoint security: device type and inventory, device health, device reputation, device management.
- Consider network security – dynamic routing, micro segmentation, software defined perimeters, traffic introspection.
- Consider workload security – API security, context aware authorisation, web application security in the business logic. Need to identity context for API traffic to authenticate every packet.
- Treat all APIs as if external and public – accessible by hostile actors from outside your domain.
- Transaction security: transaction verification, continuous session validation and security. Different transaction types might have dynamic changes in digital risk and trust levels as you move through the business process logic.
- Data security – at rest and in transit - confidentiality and integrity checking using crypto, user privacy and consent, data loss prevention, embedded data security policies and enforced policy execution.
- Assurance framework – auditing, event logging, reporting, forensics.
- Smart threat detection – machine learning and adaptive logic.

The list above suggests a holistic approach to architecture, not simply a collection of controls. The two major considerations are the mobility and changing context of both users and data. We shall need to move towards standard protocols to provide a next generation, token-based, authorisation service (such as OAUTH). Such a service will include the issue of both 'bearer tokens' (taken on face value) and 'bound tokens' (linked to the users through some provable secret).

You need a vendor-agnostic architecture framework to do all this. You need SABSA.

The Attributer

Henk Stomphorst is de bedenker en (mede)oprichter van OpenUp Technologies.
Henk is bereikbaar via hstomphorst@openuptech.com.

Groei van het menselijk kapitaal en digitaal architectuurontwerp

Mensen laten groeien of in hun kracht zetten, daar draait het om bij OpenUp Technologies, deelnemer op de Kenniscampus Ede. Te beginnen bij de developers. Ook developers ontwikkelen zich het beste in een veilige omgeving. Een omgeving waar het niet erg is om fouten te maken, als je er maar van leert. Een omgeving waar de groei van de developers centraal staat.

Volgens mij kun je alleen maar een steeds beter product verwachten van een team - en daaronder ook een beter digitaal architectuurontwerp - in een werksituatie waarin iedereen persoonlijk groeit. Er wordt daarbij vaak vooral veel aandacht besteed aan trainingen qua kennis en vaardigheden, maar werkelijke persoonlijke groei baseert zich op voldoende ruimte voor kansrijke leerinterventies en ontwikkeling van het leerproces zelf.

Zinvolle indicatoren bij opleiding en evaluatie

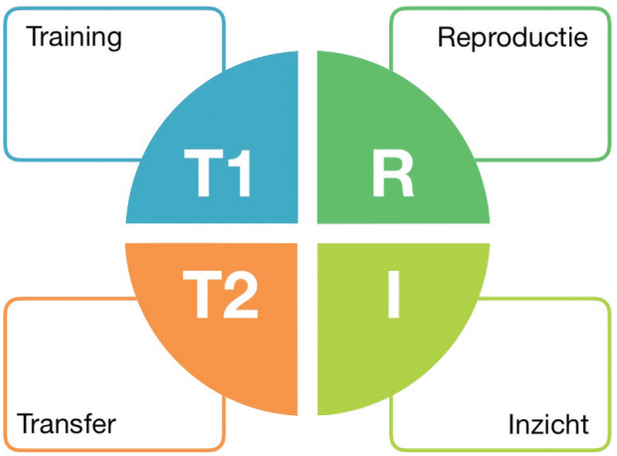
RTTI en OMZA zijn bewezen effectieve middelen voor formatieve evaluatie, de kern en het vliegwiel voor kwaliteitszorg in het ouderwijs. Leerlingen worden gestimuleerd op hun individuele niveau en kunnen daarmee steeds de volgende stap in het ontwerpproces zetten. Docenten verbeteren met deze middelen doorlopend hun onderwijs. Zoals in het hierna te beschrijven 3D-platformmodel, waar sprake is van een 'proven technology', dat gebruikers optimaal het OpenUp-platform laat aansturen en traint.

RTTI en OMZA zijn cognitieve- en gedragsindicatoren. Zij geven inzicht in het denken en gedrag van gebruikers van het platform en biedt een instrument om op dat gebruik te anticiperen. Deze indicatoren worden nu met name ingezet in het onderwijs, maar zouden ook hun inzetbaarheid kunnen bewijzen in de digitale architectuur. Niet voor niets zijn autisten vaak zeer goede probleemoplossers en digitale forensisch specialisten. Ook bij de opleiding van hen en het opdoen van werkervaring zijn deze indicatoren nuttig.

RTTI staat voor:

R = Reproductie
T1 = Training
T2 = Transfer
I = Inzicht en innovatie

'Reproductievragen' (R) mikken op memorisatie van cruciale en relevante leerstof en vormen de kennisbasis. 'Trainingsgerichte toepassingsvragen' (T1) zijn er om via een getrainde procedure tot oplossingen te komen in herkenbare situaties. 'Transfergerichte toepassingsvragen' (T2) doen een beroep op het toepassen en/of combineren van de lesstof in een situatie waarin een nieuwe transfer vereist is. Bij de 'Inzicht- en innovatievragen' (I) moet de oplossing gerealiseerd worden door zelf de context en methode te construeren.



Figuur 1 - RTTI schematisch weergegeven.

OMZA staat voor:

O = Organisatie
M = Meedoen
Z = Zelfvertrouwen
A = Autonomie

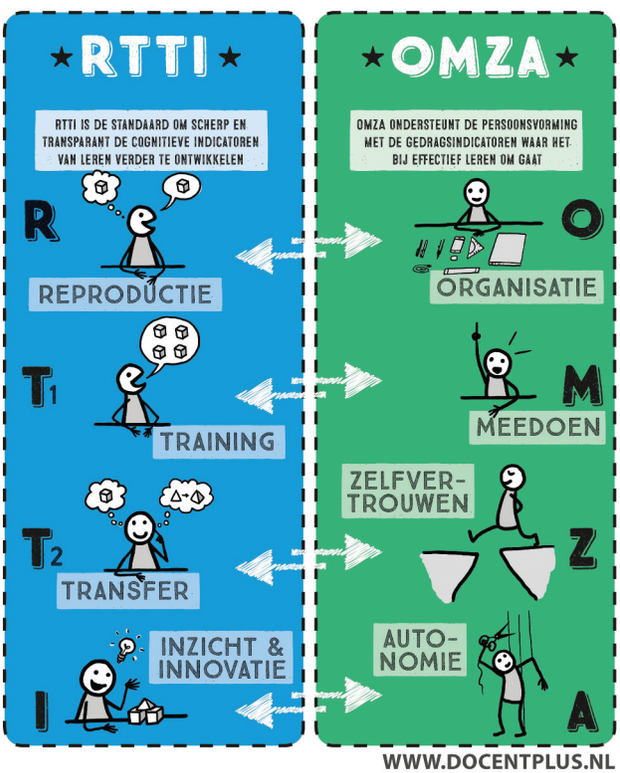
Bij de gedragsindicatoren staat de 'O' voor het 'Organisatievermogen' (overzicht & planning) en bestaat er een samenhang met de cognitieve indicator Reproductie (R). De 'M' van 'Meedoen' vraagt om betrokkenheid, resonantie en het inzetten van andere werkvormen, hier de samenhang met Trainingsgerichte toepassingsvragen (T1). Durf, lef, het laten opdoen van succeservaringen en het geven van complimenten over de actieve houding leiden tot 'Zelfvertrouwen', de 'Z', en koppelt dat aan de cognitieve indicator Transfergerichte toepassingsvragen (T2). Assertiviteit, reflectie, kritische vragen (en dat binnen het kader van de leeranalyse) leidt tot de gedragsindicator 'Autonomie', de 'A', welke koppelt met de Inzicht- en innovatie-indicator (I).

Wij streven ernaar dat iedereen, met behulp van ons OpenUp-platform, steeds makkelijker zijn/haar eigen virtuele 3D-wereld kan bouwen. Een wereld die gebruikt kan worden om mensen in hun kracht te zetten, om hen inzicht te verschaffen en te trainen maar ook voor vermaak. Deze werelden maken gebruik van speciale devices zoals de Microsoft HoloLens of de Oculus Quest. Echter, ook door middel van de eigen iOS- of Android-smartphone (of tablet). De gebouwde werelden moeten in principe device-onafhankelijk zijn.

De slag van kennisoverdracht naar de IT'er om een oplossing te kunnen bouwen moet dus (zoveel mogelijk) overgeslagen worden. Continue platformoptimalisatie bereikt men door ervoor te zorgen dat juist door niet-IT'ers eenvoudig gebouwd kan worden met en aan het platform. Participatie staat dus centraal, actief meedoen is het credo.

Maar hoe doen we dat? Hoe creëren we een omgeving waar zo eenvoudig gebouwd kan worden? Door het mogelijk te maken dat de gebruiker op een simpele wijze virtuele 3D-objecten in zijn wereld kan (ver)plaatsen of wijzigen. Via de 3D-modelbibliotheken beschikt hij/zij over enorm veel objecten. Bijvoorbeeld met behulp van een legodatabase waarmee (binnen het platform) virtueel met elk (ooit ontwikkeld) legoblok(je) gebouwd kan worden. Daarnaast kan dit met behulp van bouwelementen die voortkomen uit de natuur, de cultuur en uit onze menselijke omgeving.

Vervolgens kan aan ieder 3D-model een bijhorende gedraging meegegeven worden. Hoe reageert een object wanneer ernaar gekeken wordt als het in de buurt komt van een ander object en het botst? De combinatie van meerdere,



Figuur 2 - Een toepassing in de praktijk.

eenvoudige gedragsregels resulteert al gauw in complexe gedragssimulaties. Wat daarmee allemaal gebouwd kan worden, wordt enkel begrensd door ons menselijk voorstellingsvermogen.

Stelt u - als voorbeeld van een dergelijke toepassing - eens de navolgende 'healthsectoroplossing' voor. Een patiënt is aan rolstoel gekluisterd of kan zijn bed niet verlaten. Zijn omgeving verandert doordat de kamer langzaam volloopt met zeewater. Een schildpad komt binnenzwemmen en kijkt de aanwezigen aan. Door bijvoorbeeld met de rolstoel rond te rijden komt een interactie met de schildpad tot stand. Hoe zal de schildpad daarop reageren? Deze omgeving kan door de patiënten worden aangepast en worden uitgebreid. Een casus welke in ontwikkeling is bij een grotere zorginstelling.

Digitaal architectuurontwerp en scholing

Het is dus feitelijk basale, digitale architectuur en dat kan een rijke bron van ontwikkeling zijn voor onze kinderen en onze werknemers. Het oogmerk is dan ook deze technieken in te gaan zetten vanaf de basisschool. Daar zou je iedere leerling kunnen leren rekenen door hen objecten te laten tellen welke gebaseerd zijn op hun persoonlijke interesse. Voor de een zijn dat paarden, voor de ander muziek en een derde gaat voor gele ballen. Een physics engine maakt het mogelijk om objecten natuurlijk te laten bewegen. Hierdoor wordt het mogelijk om bijvoorbeeld een telopdracht te geven is, zoals: "Tel het aantal gele ballen welke van de tafel rollen en die op de vloer stuiteren."

Dit maakt dat het (digitaal) architectuur denken en het ontwerpen (en zeker niet te vergeten met betrekking tot de API) een basisvaardigheid kan gaan worden. Denk eens aan de betekenis van dit voor het digitaal architectuurontwerp!

Het OpenUp-platform ontwikkelt zich door voorgaande aanpak op een organische wijze. Regelmatige evaluatie borgt dat de ontwikkelingen op het goede pad blijven. Zijn we daarmee voldoende voorbereid op nieuwe technologie die eraan zitten te komen? Dat weet je nooit zeker, maar door de actiegerichte en regelmatige aanpak verbetert de architectuur en zijn wij allen steeds beter voorbereid op de toekomst.

Als wij de resultaten analyseren, dan zijn wij ook tevreden ten aanzien van het tempo in de platformontwikkeling, de groei van onze medewerkers en de kwaliteit van de IT-oplossingen. En dat toont zich bijvoorbeeld in oplossingen waarbij een warmtepompinstallatie door leerling-monteurs in elkaar wordt gezet met de opdracht deze te vullen met water om vervolgens de installatie in te regelen.

Het is daarom dat wij ons herkennen in het initiatief van de Digital Architecture Design Day (DADD). Wij trachten binnen ons bedrijf actief vorm te geven aan deze wijze van denken en door stage- en betaalde werkplaatsen een bijdrage te leveren aan de scholing van medewerkers die wellicht anders moeilijk tot een carrière kunnen komen.

Conclusie

De voorzichtige conclusie die wij trekken, is dat het digitale architectuurontwerp - gekoppeld aan de juiste indicatoren - een kansrijke toekomst lijkt te zijn. Wij menen al een aardig eind op die weg gevorderd te zijn. Het initiatief van het Nederlands Architectuur Forum (NAF) om te komen tot een Digital Architecture Design Day (DADD) sluit daarbij goed aan.

Natuurlijk is geen enkele werkwijze of procedure het enig zaligmakende. Verleggen van de aandacht van de hoogwaardige IT-oplossing (misschien niet altijd goed te begrijpen door de passieve gebruiker/ontwerper) naar het proces waarbij een ieder actief participeert, voelt intuïtief wel goed aan.

De ontwerper is in onze optiek niet alleen diegene die zich zo noemt, maar juist elke actieve participant die op een of andere wijze een ontwerpersrol claimt en het lef heeft ervoor te gaan. Daarbij beperkt zich de ontwerpersrol zich niet alleen tot de IT-techniek, maar heeft ook de 'business' een belangrijke stem.

Het is niet alleen het proces van digitaliseren, maar ook het proces om te komen tot een digitaal bedrijfsmodel. Daarbij vervult de manager en het managementteam een initiatorrol. De te ontwikkelen API's en de schaalbaarheid zijn fundamenteel voor het bedrijfssucces. Samenwerking met andere vakgebieden alsook het koppelen aan andere vaardigheden, zoals het eerdergenoemde onderwijs, maar ook (wetenschappelijk) onderzoek, vormen daarbij belangrijke randvoorwaarden. Wij zien in deze aanpak een goede toekomst vorm gegeven worden.

Thursday November 14, 2019 | Congress centre 1931, 's-Hertogenbosch, The Netherlands

DIGITAL ARCHITECTURE DESIGN DAY 2019

Conference Organisation

Naf Nederlands Architectuur Forum voor de digitale wereld

ckcseminars

Putting strategy into practice

Digital Architecture Design Day is a community driven event that will reveal the best ways of putting strategy into practice for information intensive, medium to large sized ecosystems that deal with a complex environment. Our goal is achieving sustainable change and coherent solutions by exchanging visions from global thought leaders, best practices, learnings and ideas. Digital Architecture Design Day is a one day conference festival incl. evening program and walking dinner: playground of creativity, innovation and hands-on experiences. People come for the insight, but stay for the networking and fun!

Target Audience DADD

Digital Architecture Design Day is the key meeting place for 'Users' (business) and 'Creators' (IT) of digital architecture, involved in translating strategy into practice, from medium to large ecosystems involved in complex collaboration. Such as: digital architects, digital designers, senior developers / engineers, senior management (CIO's, CTO's), IT managers, product owners, product managers, innovation managers, business analysts, data analysts, technical project managers, tech leads, lead architects, enterprise architects, business architects, domain architects, information architects, solution architects.

Host and keynote speakers

Your host of Digital Design Day 2019
Annemarie van Campen
Digital Project Manager

3 new key insights that urge you to rethink your data transformation journey
He-Yu Hua and Bram Nauts Associate Vice President respectively Enterprise Data Advisor, ABN AMRO Bank

Open key to your career
Eric D. Schabell
Global Technology Evangelist and Portfolio Architect Director, Red Hat

The X factor of Customer Centricity
Nancy Rademakers
Partner, Nexxworks

Is technology your master or your slave?
Rens van der Vorst
Head of IT Innovation, Fontys University and Technophilosopher

INTERESTED IN VISITING DIGITAL ARCHITECTURE DESIGN DAY?
Register now and ensure your place at this unique conference. When you register via **PvIB**, you receive a **€ 50,- discount**.
REGISTER WITH DISCOUNT CODE: PvIB50D1419

Silver Partners

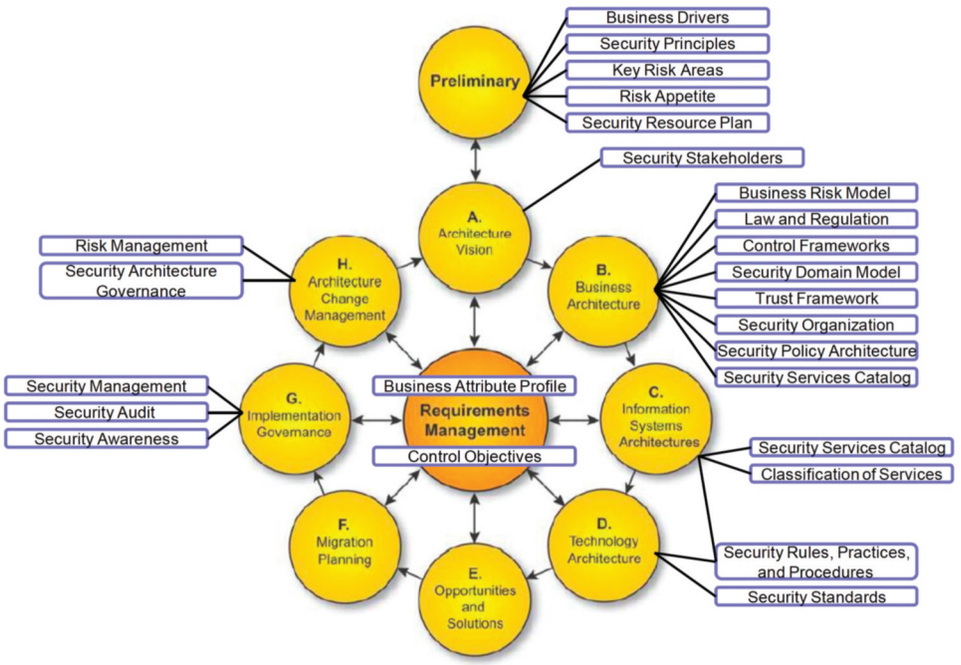
ArchiXL | Info Support | Le Blanc Advies

More information and registration at www.digitalarchitecturedesignday.com

Renato Kuiper is principal consultant/security-architect bij Tesorion. Hij is gastdocent aan de Cyber Security Academy op het gebied van security en architectuur. Renato is bereikbaar via renato.kuiper@tesorion.nl.

Ontwikkelingen rondom security in architectuur

Dit artikel beschrijft de ontwikkelingen die hebben plaatsgevonden in de laatste 20 jaar rondom het begrip 'security-architectuur'. De voorloper van het PvIB, het GvIB, heeft in 2005 en 2006 twee expertbrieven geschreven over securityprincipes (1) en security-architectuur (2). Destijds werd ervan uitgegaan dat security een integraal onderwerp zou worden van architectuur en dat het onderwerp security een integraal onderdeel zou worden van het opleidingsprogramma van architecten. Nu, bijna 15 jaar later, zien we dat dit 'helaas' nog niet het geval is.



Figuur 1 - TOGAF SABSA-integratie.

In 2005 en 2006 bestonden een beperkt aantal raamwerken voor security-architectuur. De bekendste daarvan zijn: SABSA (3), OSA (Open Security Architectuur) (4) en security als onderdeel in TOGAF (5). Laten we eens kijken hoe de volgende onderdelen zich ontwikkeld hebben door de jaren:

- architectuurraamwerken;
- securityprincipes;
- opdrachtgeversrol;
- security-architectuurraamwerk en -methoden;
- security-architect;
- inhoud van de security-architectuur.

Architectuur raamwerken

TOGAF en DYA zijn de meest bekendste raamwerken voor architectuur, toegepast in Nederland. TOGAF als architectuur benoemde destijds security als onderdeel van de technische architectuur. DYA noemt het een kwaliteitsaspect in de architectuur, maar heeft het nooit geadresseerd. Een positieve ontwikkeling toonde zich in 2011. Integratie van SABSA in TOGAF (waarover later meer) werd als een gezamenlijk initiatief door de de Open Group en het SABSA Institute opgepakt en beschreven in een white paper (5) van de Open Group. In deze white paper werden de SABSA-artifacts beschreven en aangegeven waar deze in de ADM van TOGAF een plek konden krijgen, zoals weergegeven is in figuur 1.

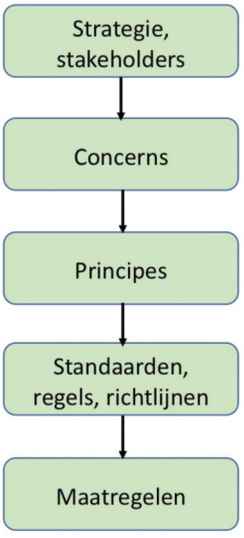
Destijds geloofden de security-architecten dat security - op de SABSA-wijze - integraal in de TOGAF 9.1 release opgenomen zou worden in het TOGAF-raamwerk (en methode). Nu, 8 jaar later, weten we dat dit nog steeds niet het geval is. Met andere woorden: security heeft nog steeds niet die plaats in architectuurmethoden die we als security community wel noodzakelijk achten. Dit is een ontwikkeling waarvan ik helaas meer had verwacht, dus dat is nog geen goede ontwikkeling voor ons securityvakgebied.

Securityprincipes

Securityprincipes bestaan al heel lang. In de periode van de expertbrieven werd hoofdzakelijk gekeken naar 'General accepted information security principles' (GAISP) (6). Deze principes waren herkenbaar voor architecten en toepasbaar voor architecturen. Wat deze principes misten, waren goed onderbouwde rationales, gebaseerd op concerns en daarbij natuurlijk ook de impact van het principe 'wat moet ervoor gebeuren', weergegeven in figuur 2. Zelfs nu komen

we securityprincipes tegen in architecturen zonder een goede onderbouwing waarom en welke impact het heeft om dit principe te implementeren.

Securityorganisaties zoals ISF, ISC2 en ISACA hebben de handen in elkaar geslagen en hebben nagedacht over het onderwerp securityprincipes. Zij hebben in 2010 een set van 12 securityprincipes (7) geformuleerd die herkenbaar en bespreekbaar zijn op hoger managementniveau. Hierbij zijn drie belangrijke aandachtsgebieden geformuleerd: 'Bescherm de business', 'Ondersteun de business' en 'Creëer een securitycultuur' (zie figuur 3). Deze principes helpen bij het opstellen van de businessgedreven securityprincipes, hierbij wordt naar het verplichte 'het moet vanwege risico's' ook daadwerkelijk gekeken naar 'hoe de business ondersteund wordt om haar doelstellingen te behalen'. Ook bij deze principes moet wel gekeken worden welk probleem ze nu oplossen. De impact moet alsnog zelf bepaald worden. Al met al een positieve ontwikkeling op het gebied van securityprincipes.



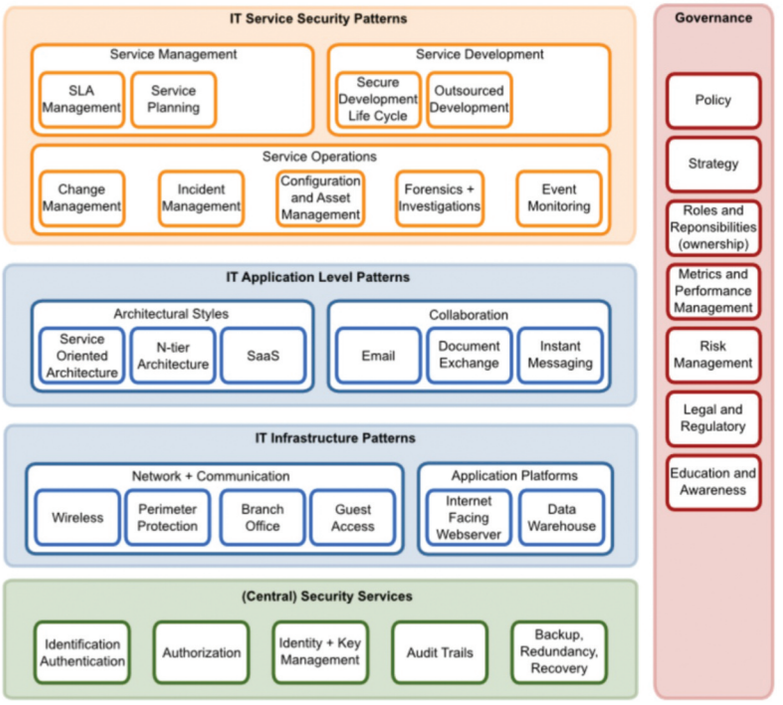
Figuur 2 – Van concern naar maatregelen.

Opdrachtgeversrol

De opdrachtgever heeft ook voor security een belangrijke rol in het definiëren van wat het wil en het sturen op die ontwikkeling. In 1997, toen ik mijn eerste security-architectuur ontwikkelde, werd eenmalig de volgende opdracht gegeven: ontwikkel een (security)architectuur. De architecten gingen daar destijds mee aan de slag en ontwikkelde een security-architectuur vanuit een aanbiedersgedachte. Hiermee bedoel ik: de architecten formuleerden de security-architectuur en securitybouwstenen waarvan zij dachten dat goed voor de organisatie was. Hierbij werd voorbijgegaan aan de businesscase voor security, deze bestond gewoon niet. Securitybouwstenen zijn gemaakt om een mogelijk probleem op te lossen, zonder dat er aan de opdrachtgever daadwerkelijk werd gevraagd waar hij/zij nu wakker van lag. Momenteel wordt de security-architectuur veel meer ontwikkeld om invulling te geven aan de concerns van de opdrachtgever(s). De opdrachtgever heeft zijn rol gepakt en de organisatie wordt daadwerkelijk uitgedaagd vanwege de securityrisico's die aanwezig zijn. Dit is vanuit security dus een goede positieve ontwikkeling.

A. Support the business	A1 Focus on the business
	A2 Deliver quality and value to stakeholders
	A3 Comply with relevant legal and regulatory requirements
	A4 Provide timely and accurate information on security performance
	A5 Evaluate current and future information threats
	A6 Promote continuous improvement in information security
B. Defend the business	B1 Adopt a risk-based approach
	B2 Protect classified information
	B3 Concentrate on critical business applications
	B4 Develop systems securely
C. Promote responsible security behaviour	C1 Act in a professional and ethical manner
	C2 Foster a security-positive culture

Figuur 3 – Securityprincipes van ISF, ISC2 en ISACA.



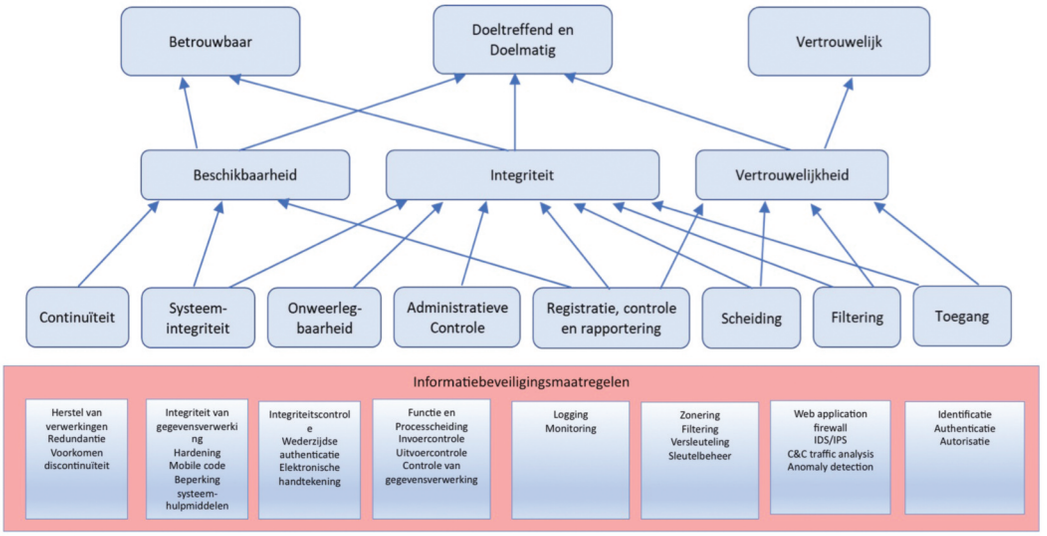
Figuur 4 – OSA-patronen.

Security architectuur raamwerk

Architectuurraamwerken voor security bestaan ook al ruim 20 jaar. De meest belangrijkste hiervan zijn SABSA en OSA op internationaal niveau en PvIB-patronen op lokaal niveau (Nederland). Bij SABSA wordt ervan uit een risicogedreven aanpak (vanuit de business) aangegeven wat belangrijk is. Dit wordt vervolgens doorvertaald naar governance-aspecten, securityprocessen en securityservices die gerealiseerd moeten worden. SABSA is een 'zware' methode voor de ontwikkeling van een securityarchitectuur. Het wordt in de markt als complex gezien. Misschien is de belangrijkste reden wel dat de architect worstelt om te bepalen welke onderdelen nu belangrijk zijn en dat een volledige SABSA life cycle te complex en een iteratie teveel is. Dit is vergelijkbaar met ITIL. De eerste ITIL-implementaties pakte gelijk alle ICT-beheerprocessen op terwijl de bedoeling was om het toe te snijden op de organisatie. Het zou denk ik wenselijk zijn dat SABSA een lightversie ontwikkelt die toegankelijker en sneller toepasbaar is.

OSA is op zich geen security-architectuur, maar een verzameling van patronen (patterns). Een patroon is een generieke oplossing die bewezen geïmplementeerd is, gebaseerd op een bekend securityprobleem. Als (security) architect heb je dus guidance hoe je oplossingen kunt bouwen, waarbij nog wel steeds geldt: welk probleem los je nu op? OSA is weergegeven in figuur 4.

De PvIB-securitypatronen (8) zijn ontwikkeld in de periode 2011-2013 door een groep security-architecten. Hierbij is ook - op basis van ervaring en bewezen toepassingsgebied - een set van patronen opgesteld. Deze patronen hebben zelfs een plek gekregen in de 'Operationele handreiking' (BIR-OH) van de BIR: Baseline Informatiebeveiliging Rijksdienst (9). Deze patronen hebben nog steeds bestaansrecht, maar behoeven wel aanpassingen om de nieuwste technische ontwikkelingen beter te ondersteunen. Zelf heb ik twee jaar aan de PvIB-patronen meegewerkt en het wordt tijd dat er een nieuw generatie opstaat om het PvIB-werk te updaten. Omdat raamwerken als SABSA erg zwaar zijn, zie je steeds meer de ontwikkeling van security-architecturen op basis van eigen 'ervaringsmethoden' ontstaan. Hierbij wordt cherry picking gedaan uit methoden als SABSA, OSA en PvIB-patronen. Al met al is dit deel voor het vakgebied nog niet voldoende uitgewerkt in eenvoudig toepasbare modellen, dus work to be done.



Figuur 5: Securityfuncties op basis van BIV-classificatie.
In de moderne security-architecturen worden de diverse architectuurlagen ondersteund met risicomanagement en security-aspecten. Een voorbeeld hiervan is weergegeven in figuur 6.

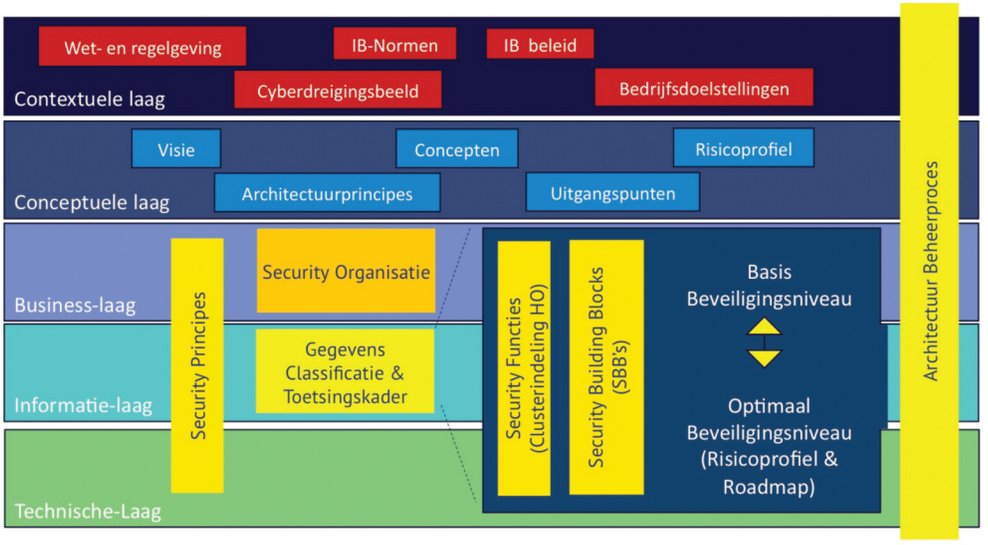
Security-architect

Bij het opstellen van de expertbrief 'Security principes' in 2005 kwam de discussie boven tafel: is een security-architect een noodzakelijke functie? In de expertgroep werd destijds gezegd: de security-architect als functie bestaat nog 5 jaar, in die periode zullen de architecten in opleiding het onderwerp security integraal meenemen vanuit een business gedachte (risicomanagement en compliancy-eisen). Nu, bijna 15 jaar later, is dit nog steeds niet het geval. Opleidingen voor architectuur benoemen security wel, maar werken het veel te beperkt uit. Vanuit het vakgebied architectuur is nu ook onderkend dat de security-architect een bijzondere functie is met bestaansrecht. In het onderwijs komt het langzamerhand op de agenda, zowel op hbo- als wo-niveau (10)(11). Zelf vervul ik de rol van security-architect al 20 jaar en doceer daar ook al ruim 7 jaar in. Het grootste risico is nu wel dat security nog steeds als een losstaand onderdeel gezien wordt van de architectuur en nog geen integraal onderdeel wordt. De vacatures voor security-architect in Nederland zijn met de bestaande architecten niet te vullen. Het is voor een security-architect een goede functie met voldoende uitdagingen voor de aankomende jaren.

Security-architectuurinhoud

De inhoud van de security-architectuur is aanzienlijk veranderd. In de periode 2000-2010 bevatte de security-architectuur, met name securityfuncties, vooral technische georiënteerde elementen (met de ondersteunende processen zoals weergegeven is in figuur 5). Hierbij worden securityfuncties en de daar uitvloeiende maatregelen getroffen op basis van slecht de BIV-classificatie. Hierbij wordt niet expliciet gekeken naar de risico's en de omgeving waarin de gegevens zich bevinden.

In de hedendaagse security-architectuur zie je onderdelen zitten in de contextuele en conceptuele laag die doorgaans niet van de architect zijn. Wet- en regelgeving is een gegeven, input komt vanuit legal (juridische zaken) en compliance is voor de architect. Een cyberdreigingsbeeld is input vanuit de business met de CISO, het informatiebeveiligingsbeleid is input vanuit de CISO. De bedrijfsdoelstellingen zijn input vanuit de business, die verankerd moeten worden in de 'Enterprise Architectuur' (en daarmee input worden van de security-architectuur). De securityfuncties die weergegeven zijn in figuur 5 zien we nu wel terugkomen in de moderne architectuur, verticaal gepositioneerd over de



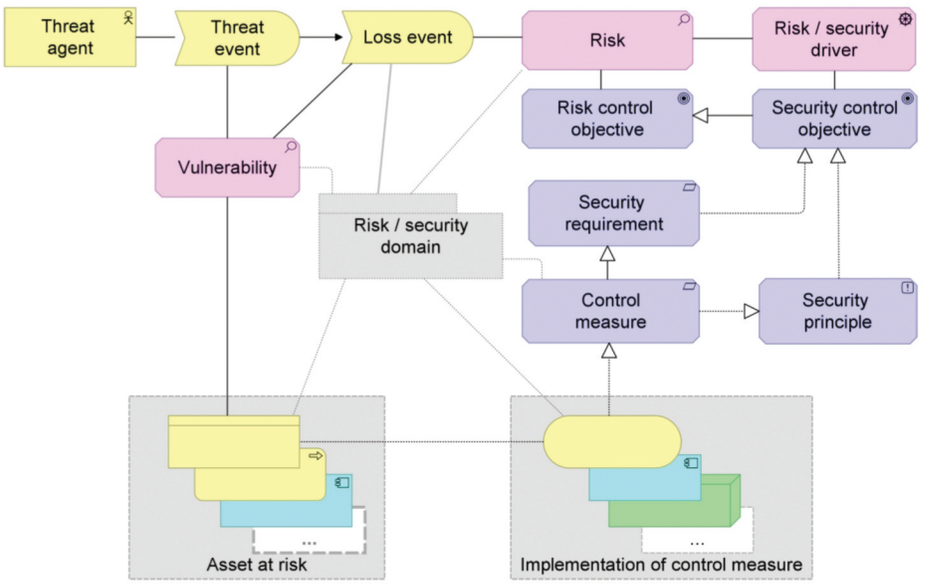
Figuur 6: Inhoud security-architectuur.

BIT-lagen heen. In de periode 2000-2010 werden securityfuncties vormgegeven met een preventief karakter. Inmiddels weten we wel beter, we kunnen niet alles beveiligen en daarom zien we steeds meer securityfuncties ontstaan met preventieve maatregelen, ook de detectieve, response en recoverymaatregelen. Deze functies en processen zijn uitgewerkt in het NIST CSF: Cyber Security framework (12), waarvan figuur 7 een interpretatie van de verschillende securityfuncties geeft.

Framework Core		
Functions	Categories	Subcategories
IDENTIFY	Asset management	Risk Assessment
	Governance of Risk	Risk Management
PROTECT	Identity and Access Management	Awareness & Training
	Data protection	Protective technologies
DETECT		SIEM
	Continuous monitoring	Security operations
RESPOND	CSIRT(team)	Crisismanagement
RECOVER	BCM	Crisis Communication

Figuur 7 – NIST CSF.

In de periode van 15 jaar is er al best het nodige gebeurd om security een plaats te geven in architectuur.



Figuur 8 – Security- en riskmodelleren.

Security krijgt een duidelijke plaats in architectuur als het ook gemodelleerd kan worden. Architecten gebruiken doorgaans een tool waarmee bijvoorbeeld in Archimate gemodelleerd kan worden. Door nu ook security en risico's te modelleren wordt het onderwerp security ook geborgd over alle architectuurlagen heen. Een voorbeeld van het modelleren hiervan is weergegeven in figuur 8.

Conclusies

In de periode van 15 jaar is er al best het nodige gebeurd om security een plaats te geven in architectuur. Echter, de integrale benadering van security als onderdeel van de enterprise architectuur is nog zeker geen gemeengoed. De rol van de security-architect wordt steeds belangrijker. Security als onderwerp in architectuurcolleges heeft zijn intrede gedaan. Het zou mooi zijn als de Open Group weer effort steekt in de integratie van security in TOGAF. Misschien was de integratie whitepaper TOGAF en SABSA te groot en complex, de behoefte bestaat er nog steeds. Alles bij elkaar opgesteld over de verschillende besproken aspecten denk ik dat we wel goede stappen gemaakt hebben, maar we zijn er nog zeker niet zijn.

Referenties

(1): GvIB Expertbrief – Security Principes: Informatiebeveiliging op de management agenda, ISSN 1872-4876, Jaargang 1 – Nr. 3, December 2005

(2): GvIB Expert Brief – Security architectuur: Nieuwe hype voor specialisten of nuttigcommunicatiemiddel? ISSN 1872-4876, Jaargang 2 – Nr. 4, December 2006.

(3) SABSA: https://sabsa.org/

(4) OSA: http://www.opensecurityarchitecture.org/cms/index.php

(5) TOGAF® and SABSA® Integration, How SABSA and TOGAF complement each other to create better architectures, october2011, Opengroup, https://sabsa.org/sabsa-togaf-integration-white-paper-download-request/

(6) GAISP: http://www.gaisp.org/

(7) Principles for Information Security Practitioners Principles for Information Security Practitioners, http://www.isaca.org/KnowledgeCenter/BMIS/Pages/Security-Principles.aspx

(8) PvIB security patronen: https://www.pvib.nl/kenniscentrum/documenten/201301-ib-patronen

(9) BIR Operationele Handreiking: https://www.earonline.nl/images/earpub/5/5c/BIR_Operationele_Handreiking_v1_0.pdf

(10) Security architectuur en SABSA, Haagse Hoge School (http://www.hhs.nl).

(11) Security in architecture, Executive Master Cybersecurity (http://www.csacademy.nl)

(12) NIST Cyber security framework, versie 1.1, April 2018, https://www.nist.gov/cyberframework/framework

Lilian Knippenberg is senior consultant Governance, Risk & Compilance bij Strict Consultancy.
Lilian is bereikbaar via lsjmanschot@gmail.com.

BOEKREVIEW

Survivalgids voor de Digitale Jungle door Brenno de Winter

‘Handboek voor omgang met spionnen, boeven, hackers en digibeten.’ Met deze ondertitel wordt mijn aandacht direct getrokken. Je wapenen tegen een dergelijk brede en verschillende groep is niet alleen noodzaak, maar ook best complex. Mensen zijn immers je zwakste schakel, maar ook je sterkste.

Een nieuw boek van een bekende journalist op het gebied van onder andere informatiebeveiliging is altijd interessant om nader te bekijken. In dit boek heeft Brenno de Winter zijn jarenlange ervaring op papier gezet in een tekst die niet alleen zijn visie op het vakgebied bevat, maar ook lijsten met tips voor de lezer. Het boek is geschreven in zeer toegankelijke tekst - bijna spreektaal op sommige momenten. Hierdoor is het alsof de auteur naast je staat om je tips in het oor te fluisteren of te schreeuwen. Basishygiëne is immers een noodzaak, geen tip.

Nu weet iedereen die mij kent dat ik dol ben op praktische ‘lijstjes’ en overzichten. In dit boek zijn deze erg goed op hun plaats. De lijsten en overzichten worden allen goed toegelicht en beschreven, waardoor een mooi overzicht ontstaat voor een ieder die ofwel start in het vakgebied en zoekt naar toegankelijke ingangen en uitgangspunten in de informatiebeveiliging - ofwel de ICT-professional in een aanverwant vakgebied die de uitgang niet meer ziet in de IB-jungle. Brenno de Winter behaalt mijns inziens met een positieve blik op informatiebeveiliging in dit boek zijn doel om de lezer met praktische tools te laten overleven in de digitale jungle. Als kritische informatiebeveiliger heb ik natuurlijk ook wat aanvullingen. Ik neem u graag even mee.

Brenno de Winter start met een interessante analogie naar het dierenrijk, waarbij hij de vergelijking maakt tussen informatiebeveiliging en beveiliging zoals kuddes dieren dat doen in het leven in de jungle. Ieder lid van de groep heeft haar eigen rol en taak. Dit vormt een pleidooi voor meer samenwerking en openheid om gezamenlijk de uitdagingen van spionnen, boeven en (kwaadwillende) hackers het hoofd te bieden. Hoewel ik het helemaal eens ben met de boodschap, denk ik toch dat in werkelijkheid een complicerende factor is dat informatie als zodanig een concurrentievoordeel oplevert voor de organisatie die beveiliging beter op orde heeft. Wordt jouw concurrent gehackt, dan kun je daar ook voordeel van hebben. Hier ligt mijns inziens ook een rol voor de overheid, om samenwerking te stimuleren en desnoods anoniem voorbeelden te delen (inclusief tegenmaatregelen). Het boek gaat echt van start vanaf hoofdstuk 2, waar de vraag ‘wat beveiligen we eigenlijk?’ centraal staat. Hierbij wordt aangenomen dat het duidelijk is wat ‘informatie’ eigenlijk is. Voor veel organisaties zal dat echter een moeilijke vraag zijn om te beantwoorden, waarbij een zekere mate van inzicht in de informatiestromen van de organisatie nodig is. Vervolgens is het belangrijk om te weten welke informatie op welk moment in het proces belangrijker is dan de andere informatie. Natuurlijk wordt de term ‘kroonjuwelen’ hiervoor gebruikt, maar ik kan mij zo voorstellen dat de opbouw van het hoofdstuk wat verwarrend kan zijn voor personen die niet zo thuis zijn in het vakgebied.

Het boek gaat verder met doodzonden in informatiebeveiliging, een erg leuk beschreven manier om de top vijf van maatregelen die vallen onder de noemer ‘basishygiëne’ vast te leggen. Ieder hoofdstuk had een boek op zich kunnen zijn, maar Brenno de Winter slaagt er goed in om de basis samen te vatten met aansprekende voorbeelden en tips. Dit geldt eigenlijk ook voor de vervolghoofdstukken, waarbij onder andere risicomanagement, privacy, maatregelen (en vooral manieren om het thema beheersbaar te houden) naar voren komen.

Tot slot wil ik nog toevoegen dat ik de uitdaging (die Brenno de Winter aangaat met het schrijven van dit boek) erg bewonderenswaardig vind. Ga er maar aanstaan: ons hele vakgebied op een eenvoudige manier uitleggen aan een breed publiek zonder het vakgebied te degraderen. Ik vind dat hij hier goed in slaagt, met als aantekening dat de basis helder maar summier, met her en der de noodzakelijke terminologie, beschreven is. Ook de voorbeelden zijn geschreven voor mensen waarvan verondersteld wordt dat zij wel iets van ICT en informatiemanagement afweten of hierin minstens geïnteresseerd zijn. Niet alleen voor de starter of zij-instromer is dit boek interessant, ook de meer ervaren informatiebeveiligingsprofessional vindt fijne lijstjes voor zijn basishygiëne.

Lex Borger is security consultant bij Testorion company en oud-hoofdredacteur van iB-Magazine.
Lex is bereikbaar via l.borger@i-to-i.nl.

Agile security

In oktober 2017 ontdekte het UpGuard Cyber Risk Team (1) dat er minstens vier AWS S3-buckets van Accenture geheel publiekelijk toegankelijk waren. Het meest schokkende van dit verhaal was dat de buckets belangrijke bedrijfsinformatie en klantinformatie bevatten. Deze informatie kan in de verkeerde handen een potentieel incident met een gigantische impact veroorzaken: imagoproblemen, schadeclaims, privacyschendingen. UpGuard handelde de ontdekking netjes af met een responsible disclosure-procedure, voordat ze publiek gingen met dit nieuws. Accenture kreeg de kans dit voor publicatie te repareren. Bij de ontdekking was het niet te zeggen wie deze buckets nog meer hadden ontdekt en of ze kwade bedoelingen hadden. Nu, twee jaar later, kunnen we zeggen dat het erop lijkt dat dit niet het geval was.

S3-buckets zijn een typische IT-dienst van Amazon in het cloudtijdperk. De voordelen: je huurt grote brokken opslag in de cloud voor een laag bedrag per maand, krijgt direct toegang en je betaalt alleen voor wat je gebruikt. In de context van agile werken is dit een pracht product. S3-buckets hebben echter een probleem: ze zijn niet secure by default (2). Als je agile werkt en je gebruikt S3-buckets in je softwareoplossing, dan heb je een aantal configuratietaken uit te voeren en een aantal securitydiensten op te zetten, zoals het opnemen van de buckets in je toegangsbeveiliging en je monitoring. Dit zijn allemaal taken die doorgaans uitgevoerd worden nadat in de sprintdemo getoond is dat de S3-bucket werkt. De business stakeholders lijken niet geïnteresseerd om een of twee sprintdemo's later te zien dat het ook veilig is, wat leidt tot het wegvallen van de druk bij de ontwikkelaars om die beveiliging af te maken.

Dit incident staat niet op zichzelf. Er zijn de laatste jaren tientallen AWS S3-buckets gemeld door onderzoekers. Door dit soort incidenten hoor ik regelmatig dat de security bij agile werken wel moeilijk zal zijn. Agile werken krijgt zo het imago dat het voor vrijbuiters is, zonder discipline, planning en documentatie. Zo boekt de business snel resultaten, zonder beperkt te worden door vervelende securitymanagers die lastige maatregelen verplichten. In de praktijk werkt agile security niet zo. Agile werken is een balans zoeken tussen stabiliteit en flexibiliteit. Vergelijk dit met de balans die we zoeken in security: de balans tussen maatregelen en risicoacceptatie. Agile werken gaat over overzichtelijk wijzigingen kunnen aanbrengen, de oplossing laten ontstaan: `emerging' is het sleutelwoord, flexibiliteit willen we houden en verspilling willen we tegengaan. Dit sluit niet uit dat de oplossing wel degelijk veilig is. Veel security kan ingebracht worden met de instrumenten die agile werken al gebruikt.

In dit artikel neem ik aan dat de lezer bekend is met de opzet van agile werken. Ik gebruik scrumtermen in de tekst, maar dat houdt niet in dat de aanbevelingen niet te betrekken zijn op andere manieren van agile werken. Verder is er bewust geen focus op personen die een bepaalde functie hebben, maar wordt er meer gesproken over de invulling van rollen gesproken, zonder deze per se aan een functie te verbinden.

Om te zien hoe agile security kan werken, moeten we kijken naar de inbedding van architectuur in agile werken. Hierbij put ik uit mijn eigen ervaring. Binnen agile werken vul je de architectuur in door een omgeving op te zetten waarin complexe ontwerpbeslissingen nog steeds flexibel genomen kunnen worden, maar wel goed geïnformeerd. Waarbij de richting die we opgaan duidelijk is en er voldoende inzicht is wat het volgende resultaat is waar we aan werken. Ontwikkelaars werken gefocust met de informatie die ze hebben. Omdat de prioriteit van nieuwe functionaliteit de focus op security kan verdringen, moeten we zorgen dat security voldoende in de focus van de ontwikkelaars blijft.

Richting

Het eerste fundamentele architectuurelementen zijn principes en standaarden. Deze geven ons de nodige richting. Bijvoorbeeld het principe om in een internetbankierenapplicatie wel of geen token te gebruiken voor autorisatie. En de standaard die daarbij hoort is dan dat het token bijvoorbeeld een Gemalto-paslezer of een Vasco-scanner moet zijn. Een principe kan zijn om single sign-on te gebruiken voor alle webapplicaties en de bijbehorende standaard om dat te doen kan verplichten dat met SAML of OpenID Connect security tokens te doen.

Deze principes en standaarden geven richting bij ontwerpbeslissingen; op korte termijn via designs en deliverables en op lange termijn via technical debt en rework. Zo worden de agile user stories uitgewerkt die nodig zijn en voor zover dat nodig is.

Inzicht

Richting is prima als er maar een handjevol agile teams zijn, op dezelfde locatie of in ieder geval in regelmatige connectie met elkaar. Die teams kunnen prima elkaars ontwerpbeslissingen volgen en bepalen waar onderlinge afhankelijkheden optreden. Afhankelijkheden kunnen tijdsvolgordelijk zijn: je wilt een service ontwikkelen voordat de clients deze aanroepen. Of als dit niet mogelijk is, wil je zinvolle interface stubs ontwikkelen. Ze kunnen ook inhoudelijk zijn. Het kan zijn dat een ontwerpbeslissing in een team de ontwerpbeslissingen van andere teams beïnvloedt. Als je nog niet hebt gehoord welke attributen opvraagbaar zijn uit een security token, kun je op basis hiervan nog geen concrete ontwerpbeslissingen vastleggen.

Hoe meer teams er zijn en hoe meer locaties er zijn, hoe harder het nodig is afhankelijkheden vast te leggen in documentatie. En deze documentatie vormt in feite een onderdeel van de agile architectuur.

In een grotere organisatie kun je agile teams geen vrijbrief geven om alles wat ze aan functionaliteit nodig hebben zelf aan te pakken en uit te werken. Dit zou te veel actieve kennis vergen van de ontwikkelaars, wat weer wegneemt van de focus die ze moeten hebben. Architectuur erkent abstracte lagen die gevuld moeten worden met functionaliteit die weer gebruikt kan worden door bovenliggende lagen. De applicatieve lagen kun je grofweg indelen in drie lagen:

- features (primaire functies - in teams - evenwichtig);
- support (hulpfuncties - in teams en tussen teams - heeft transparantie nodig);
- plumbing (onzichtbare infra - in teams, tussen teams met begrip van overzicht en smenhang).

Deze indeling in lagen help inzicht te krijgen. Ze kunnen hun eigen principes en standaarden toegewezen krijgen.

Architectuuractiviteiten

Architectuur maakt dus de richting duidelijk en verschaft de nodige inzichten over teams heen. Dit komt voort uit verschillende activiteiten die uitgevoerd worden:

• Anticiperen op ontwerpbeslissingen door de juiste principes en standaarden klaar te hebben.

• Toepassen van deze principes en standaarden op de backlogs door het documenteren van ontwerpbeslissingen. Deze kunnen in lijn zijn met principes en standaarden, maar kunnen ook daarvan afwijken om praktische redenen. De onderliggende gedachte is: geen BDUF (Big Design Up-Front), maar emergent design.

• Aanpassen van de backlog door de gevolgen van ontwerpbeslissingen die niet in lijn zijn met de principes en standaarden. Zo blijft bekend welk werk nu feitelijk vooruitgeschoven is, zodat het later een keer opgepakt kan worden als dat nodig is. Dit houdt de kwaliteit op peil.

Als een applicatie bijvoorbeeld in het begin niet onder een single sign-on dienst wordt gehangen, moet dat als een user story op de backlog blijven staan, met een prioriteit die gerelateerd is aan het risico als dit niet gebeurt. Als een applicatie initieel geen volledige security event logging implementeert, kan dat bij gebrek aan user story op de backlog later in productie leiden tot extra problemen bij een diagnose stellen na een hackpoging.

Security-inbreng

Hoe past security in dit geheel van agile werken met architectuur? Allereerst zijn er in het operationele landschap securitydiensten nodig, zoals encryptie, toegangsbeveiliging en monitoring en diensten die de beveiliging ondersteunen, zoals incidentmanagement, certificaatbeheer en sleutelbeheer. Deze moeten functioneel ontwikkeld en geïmplementeerd worden.

De rest van de security is onderdeel van de plumbing, de onzichtbare infra die het gewoon goed en veilig moet doen wanneer nodig. Kwaliteitstechnisch zijn dit de non-functional requirements, de 'NFR's. Voor softwareontwikkeling zijn de NFR-onderwerpen gedocumenteerd in de ISO/IEC 25010-standaard. Dit was voorheen de ISO 9126. Primair valt security in deze standaard onder

Met de NFR's moeten we anders omgaan

'reliability' (beschikbaarheid) en 'security' (vertrouwelijkheid en integriteit), maar secundair dragen de meeste andere categorieën ook wel iets bij aan integriteit of beschikbaarheid.

Deze twee vormen van security moeten dus op de juiste manier aandacht krijgen binnen het agile werken:

1. User Stories
De securityfunctionaliteit wordt gewoon in user stories beschreven, net zoals andere functionaliteit. Dit is 'business as usual' voor agile werkers, dus daar ga ik weinig extra woorden aan vuil maken. Heel veel aandacht wordt besteed om datgene wat goed moet gaan, goed te laten gaan.

Een groot gevaar hierbij is dat we ook in de gaten moeten houden dat we beschrijven wat fout zou kunnen gaan, om te zorgen dat dat niet fout gaat. Hier gaat het over iets wat je in agile-documentatie niet tegenkomt en daarom ook geen standaard naam hebben: de red light scenarios, de error cases, de abuse stories, de hacking stories. Kortom, alles wat in productie af kan leiden van een goede uitkomst. Deze stories beschrijven helpt. Soms zijn dit gekunstelde user stories: 'Als de verantwoordelijke business eigenaar wil ik dat hackers niet wachtwoorden kunnen blijven uitproberen, zodat ze op deze manier niet het systeem binnen kunnen dringen.'

Er is hier een anti-pattern mogelijk, een user story als de volgende werkt niet: 'Als de verantwoordelijke business eigenaar wil ik dat er geen OWASP top 10 kritieke risico's voorkomen in de applicatie.' Als user story zit hier hooguit een assessment aan wat je kunt uitvoeren, maar dat verloopt meteen na uitvoering. En dan vind je wellicht een heleboel werk, wat dan weer tot extra user stories leidt. Iets dat zeker niet leidt tot tevreden opdrachtgevers. Om dit te voorkomen hebben we de NFR-kant van security nodig.

2. Definition of done
Met de NFR's moeten we anders omgaan. Die kunnen we in twee categorieën indelen:

De eerste categorie wordt gevormd door die NFRs die afgeleid zijn van securityprincipes en standaarden die bijna altijd van toepassing zijn: worden er op de juiste momenten autorisatiebeslissingen gevraagd? Worden de belangrijke momenten gelogd? Die kunnen een plaats krijgen in de 'definition of done'. Tien jaar geleden zou de definition of done de aanbeveling 'has the security audit been signed off' zijn.

Dat is ook een anti-pattern, alsof we elke deliverable kunnen laten inspecteren en testen door de securit-experts. Reken erop dat in de gemiddelde organisatie een security-expert dit voor ongeveer honderd ontwikkelaars zal moeten doen. Een no-go dus.

Een self-audit op basis van de definition of done is veel praktischer, waarbij de code reviewer nog een kritisch oog mee kan laten lopen. Dit vergt wel de discipline om code reviews serieus te doen. Alle kwaliteitsaspecten zullen hier baat bij hebben, niet alleen security. Een puntje van aandacht dus voor de organisatie die agile werkt.

De andere categorie bevat NFR's - die lang niet altijd van toepassing zijn op iedere user story. Ik heb daar een checklist voor en verklaar bij het ready maken van de user stories welke securityprincipes relevant waren en welke standaarden toegepast kunnen worden. 'Geen' is zelden een acceptabel antwoord, er is altijd wel iets van security terug te vinden in een functie. En laat de code reviewer dat achteraf meereviewen. Deze code review staat ook in de definition of done. En vergeet het belang van automatisch testen niet, zodat wat werkt blijft werken en wat niet hoort te werken ook niet werkt.

Securityprincipes en -standaarden

Securityprincipes zijn eigenlijk wonderwel stabiel gebleven door de jaren heen. Goed beleid heeft maar een per een aantal jaren aanpassing nodig. En die aanpassingen vallen vaak nog best wel mee ook. Als je de BS7799 naast de huidige ISO/IEC 27001- en 27002-normen legt, dan kun je veel structuur en belangrijke elementen in beide documenten herkennen. Er zijn verschillen, maar niet drastisch. Er is maar een beleidson-

derwerp bijgekomen, dat was eind tachtiger jaren voor bescherming tegen malware. Houd principes techniek-onafhankelijk en algemeen. Documenteer ze kort en bondig en maak ze gemakkelijk vindbaar.

Een ander verhaal is het met securitystandaarden. Deze zijn wel afhankelijk van de techniek en bewegen daarom mee met het hoge tempo waarin sommige techniek veranderd. Een standaard die regelt welke encryptie toegepast moet of mag worden kan al heel snel door nieuwe ontdekkingen in de cryptologie achterhaald worden. Protocollen die encryptie gebruiken, kunnen daardoor ook aangepast moeten worden.

TLS is hier een mooi voorbeeld van. Niet alleen de versie van TLS (1.0, 1.1, 1.2, 1.3) is belangrijk, ook welke cryptografische algoritmen en sleutellengtes aangeboden worden in de opzet van een verbinding en de volgorde waarin die aangeboden worden is belangrijk.

Loop alle standaarden dus frequenter na dan de securityprincipes. Jaarlijks is een goed idee, soms sneller, als reactie op nieuwe aanvalsmethoden die bekend worden. Houd daarom het securitynieuws bij. Als een standaard aangepast moet worden, houd er dan rekening mee dat dit werk op de backlog met zich meebrengt. Er zal moeten worden uitgezocht welke functionaliteit en/of configuratie eventueel aangepast moet worden, wat ook weer voor meer user stories op de backlog zorgt.

In het geval dat de interface naar securitydiensten aangepast moet worden, wordt het nog meer werk, want dan moeten alle gebruikers van de service ook hun aanroepen wijzigen. Bij goed ontworpen services zal dit niet vaak het geval zijn, maar het is mogelijk.

Security-activiteiten

De security-activiteiten die hier genoemd worden, zullen grotendeels uitgevoerd worden door teamleden van agile teams, eventueel bijgestaan door een securitymanager of -specialist.

- Het toepassen van securityprincipes en security-standaarden.

- Het valideren van relevante securityontwerpen. Laat belangrijke elementen van de security valideren door een securityspecialist. Securityfuncties horen ook met automatische testen gevalideerd te worden, zodat er continu de zekerheid is dat ze goed blijven werken. Sowieso zullen security professionals de securityprincipes en standaarden bijhouden, in samenwerking met de ontwikkelaars.

Agile werken, architectuur en security kunnen heel goed samengaan

Het kan bij de besten fout gaan: in 2019 repareerde Apple met iOS 12.3 een mogelijkheid om apparaten te jailbreaken. De kwetsbaarheid werd echter geherintroduceerd met iOS 12.4, wat tot een noodreparatie leidde, iOS 12.4.1, om de kwetsbaarheid nogmaals te repareren. Kennelijk is het releasetraject bij Apple nu zo lang geworden dat reparaties parallel in verschillende subversies moeten worden aangebracht. Automatisch testen had dit waarschijnlijk eerder aan het licht gebracht.

- Houd je technical debt bij op de product backlog. Dit is de technical debt die voortkomt uit ontwerpbeslissingen om af te wijken van de security principes of standaarden, of uit principes of standaarden die aangepast zijn;

- Stel de juiste prioriteiten vast van de user stories op de backlog. Doe dit als nodig in overleg met een securityspecialist om de juiste impact te kunnen bepalen. En werk volgens deze prioriteiten.

Agile werken, architectuur en security kunnen heel goed samengaan. Als dit goed uitgevoerd wordt volgens de werkwijze die in dit artikel is beschreven, dat zal agile werken veilige softwareproducten opleveren met een goede documentatie van de ontwerpbeslissingen en de aanpassingen die nog gedaan moeten worden.

Referenties

(1) System Shock: How A Cloud Leak Exposed Accenture's Business (https://www.upguard.com/breaches/cloud-leak-accenture)

(2) S3 Security Is Flawed By Design (https://www.upguard.com/blog/s3-security-is-flawed-by-design)

LODEWIEK JANSEN

Ik werk als IT-auditor binnen de Rijksoverheid bij de Auditdienst Rijk op het snijvlak van techniek, bedrijfsprocessen en mensen. Ik heb de master Business Information Management en de postmaster IT-audit aan de Erasmus Universiteit gedaan. Ik heb altijd al veel interesse in IT gehad. In de afgelopen negen jaar heb ik op alle departementen binnen de Rijksoverheid, IT-audits op het gebied van informatiebeveiliging uitgevoerd. Sinds een paar jaar houd ik me vooral bezig met het uitvoeren van pentesten en technische IT-audit. De omvang, complexiteit en diversiteit van mijn werkzaamheden zorgen voor een continue uitdaging. Daarnaast vind ik het leuk om presentaties te geven over hacking en security awareness en om deelnemers dat ook zelf te laten ervaren of uit te voeren.

In 2010 kwam ik als Young Professional in de YP-commissie van het PvIB terecht. Voor mij was het PvIB een kickstart in de wereld van informatiebeveiliging. Het organiseren voor Young Professionals blijft leuk om te doen en ondertussen hebben we veel verschillende bedrijven bezocht en uiteenlopende activiteiten georganiseerd: boksclinics, bezoek aan Schiphol en een hacksessie bij OWASP. Er is dus veel mogelijk voor onze Young Professionals. Aandacht voor de Young Professionals blijft heel belangrijk voor onze vereniging en zorgt voor nieuwe aanwas; zowel voor de vereniging als voor het vakgebied. Ik heb zelf in de eerste jaren dankzij het PvIB snel en veel kunnen leren over informatiebeveiliging van vakgenoten.

'Het huiskamergevoel' dat de PvIB-bijeenkomsten nu karakteriseert moeten we blijven koesteren om ook in de toekomst een ontmoetingsplek voor informatiebeveiligers te zijn. Daarnaast mag het PvIB digitaal meer uitdragen welke kennis er binnen de vereniging aanwezig is en welke leuke activiteiten we organiseren voor leden. Ook kunnen we onze leden nog beter digitaal bedienen op het gebied van in- en uitchecken bij evenementen en met een actueel, online overzicht van de behaalde PE-punten. Dat wordt in de komende maanden gerealiseerd.

De cloud, virtualisatie en containerisatie zijn ontwikkelingen in het ons vakgebied die (toekomstige) uitdagingen vormen. IT-infrastructuur wordt steeds meer een code - en daarmee een commodity. Als informatiebeveiliger en IT-auditor zal je de oude paradigma's gedeeltelijk moeten loslaten of anders toepassen. Tevens maken deze ontwikkelingen het nu mogelijk om zelf aan de slag te gaan en een hele infrastructuur op je laptop of in de cloud te verkennen. Door het zelf te doen en te ervaren leer je het meest, kun je de technieken beter begrijpen en zo de mogelijke risico's beter doorgronden. Kortom, genoeg mooie onderwerpen om ons als informatiebeveiligers de komende tijd in vast te bijten.

Robert Metsemakers schrijft op persoonlijke titel en is als ervaren IT-auditor en informatiebeveiligingsexpert beschikbaar voor security-advies en (algemene) schrijfopdrachten via robert.metsemakers@gmail.com.

Drie loodgieterslessen voor security

Securityprofessionals lijken in hun werk op loodgieters. Ook zij moeten een goede volgorde kiezen in het oplossen van problemen, bij minder leuke klussen toch gemotiveerd blijven en steeds voorbeeldgedrag tonen aan hun omgeving.

Securitymedewerkers moeten ook 'roomser dan de paus' zijn

Water en ontlasting lopen altijd naar het laagste punt. Cybercrime en securityproblemen gebeuren ook meestal op het punt met de laagste bescherming. Vaak is de menselijke schakel de zwakste in de keten en dus is awarenessverhoging meestal een goed idee. De security maturity van een afdeling is te meten op een aantal kwaliteitsaspecten. Wanneer we security-aspecten A,B,C,D en E meten en de score is 3-4-4-4-5, dan adviseer ik de 'A' te verhogen van 3 naar 4. Niet door nu al te proberen om B+C+D te verhogen naar 5 (rekenkundig gaat de totaalscore dan omhoog), maar aspect 'A' blijft dan het zwakste punt. Bijvoorbeeld wanneer 'A' de security awareness bij medewerkers, managers en systeembeheerders is. Soms is de awareness al '5' en moet het wachtwoordbeheer worden verbeterd, omdat bijvoorbeeld de password reset te gemakkelijk misbruikt kan worden. Kijk naar je zwakste punt in security en verbeter dat eerst.

Securitywerk is stressvol. Je reageert op bewuste aanvallen en onbewuste fouten (van jezelf of andere personen) en die komen per definitie onverwacht. Vaak is security een hygiënefactor. Je kunt het nooit goed doen, alleen fout. Je kunt er de commerciële wedstrijd niet mee winnen als je het goed doet, maar wel verliezen als je het fout doet. Dan is het goed je te realiseren dat aan het eind van de week (of maand, als u niet meer werkt met papieren loonzakjes) er een beloning tegenover staat. Of voor elk teamlid een eigen 'pannenkoek babi pangang' bij een on-time-and-specs afgesloten project, zoals ik meemaakte tijdens een leuke afdelingstraditie.

Een bijzondere situatie van betalen voor security is een externe penetratietest. Blackbox (zonder enige informatie over uw configuratie aan de tester en voor zeer weinig geld in een korte periode) komt dan ter sprake. Dan kun je van de externe leverancier niet veel meer verwachten dan een onervaren pentester die net een geautomatiseerd scantool kan starten en het gegenereerde rapport daarna omzet naar PDF. Je betaalt dan een beperkt bedrag voor de uitkomst – 'In twee werkdagen kan een onervaren hacker met een beperkt aantal tools en zonder voorinformatie niet binnendringen in het onderzochte systeem.' Dat is iets heel anders dan dat ze zeggen: 'De website, app of applicatie is voldoende veilig.' Terwijl dat vaak de uitspraak is waarom het niet-inhoudelijk management een externe penetratietest verplicht stelt als voorwaarde voor 'live' gaan. Penetratietesten is negatief testen: bij gevonden issues en tekortkomingen kun je wel aantonen dat een systeem onveilig is, maar het ontbreken van bevindingen uit een pentest zegt niet dat het systeem voldoende beveiligd is. Het zegt alleen dat de betreffende tester(s) in de beschikbare tijd en budget geen kwetsbaarheden konden vinden. Een beperkte pentest is wel noodzakelijk, maar niet voldoende, bedoel ik.

Nooit op je nagels bijten is logisch als je werkt met poep in en rond buizen. Securitymedewerkers moeten ook 'roomser dan de paus' zijn. Zorgen dat hen zelf geen securitynarigheid overkomt, om geen afbreuk te doen aan de organisatiebrede security awareness-campagne vanuit de afdeling Security en hun geloofwaardigheid, professionaliteit en marktwaarde te continueren. Bewaar je back-ups (!) apart van de originele datadrager. Wees alert op phishing en denk aan het Turfschip van Breda(*) bij installeren van software uit 'onduidelijke bron'. Niemand moet een gevonden USB-stick met mogelijk malware in de zakelijke laptop steken - en zeker een ervaren security-expert niet!

(*) Lang na het Paard van Troje smokkelde men in Breda - met vrijwel dezelfde list - soldaten de stad binnen om deze te veroveren op de bezetters.

Achter Het Nieuws

In deze rubriek geven enkele iB-redacteuren in een kort stukje hun reactie op recente nieuwsitems over de informatiebeveiliging. Dit zijn persoonlijke reacties van de auteurs en deze geven niet noodzakelijkerwijs het officiële standpunt weer van hun werkgever of van PvIB. Vragen en/of opmerkingen kunt u sturen naar ibmagazine@pvib.nl.

Hackers

Hackers zijn steeds vaker op verschillende manieren in het nieuws. We kennen bijvoorbeeld 'hacktivisten', hackers die informatie stelen of versleutelen en hackers die wijzigingen aanbrengen in soft- en hardware om daar zelf beter van te worden. Zo werd onlangs bekend dat het Openbaar Ministerie onderzoek gaat doen naar het hacken van rijtijds- en rijsnelheidsmeters in vrachtwagens. De datalekken vliegen ons om de oren. Daarnaast kennen we steeds meer hackersevenementen en ethische hackers - die door organisaties worden ingezet om kwetsbaarheden te vinden. Wat zijn de gevaren en voordelen van hackers?

Fook Hwa Tan

Hackers zijn noodzakelijk in onze maatschappij. We zien dat steeds meer mensen in onze samenleving aan hacking doen. Zo heb je naast de digitale hackers ook productiviteit- en dieet-hackers. Hackers zijn mensen die bestaande systemen kunnen omzeilen. Dat kan dus op elk gebied, maar laten we het hier hebben over de hackers die computersystemen omzeilen. Waarom zijn deze eigenlijk nodig?

Steeds meer organisaties vallen onder regulering. Hierbij wordt door een toezichthouder gevraagd om aantoonbaar te maken dat er aan de algemene (en soms specifieke) technische beveiligingseisen wordt voldaan. Dit is dus puur om anderen te laten zien dat de regels zijn nageleefd. Iets wat meer voor de organisatie zelf plaatsvindt, is het zoeken naar kwetsbaarheden binnen de organisatie. Het is belangrijk om te weten waar je kwetsbaar bent, zodat je kunt bepalen wat je op basis van je risicobereidheid wilt gaan doen om je risico's in te perken.

De bovenstaande twee redenen kun je (als je de resources vrij kunt maken) intern laten uitvoeren. Het is ook aan te bevelen om periodiek een externe hacker te vragen om je assets te testen. Een frisse blik kan vaak tot nieuwe inzichten leiden. Interne medewerkers kunnen na een tijd blind worden voor de eigen kwetsbaarheden. Hackers moeten daarom onderdeel zijn van processen in organisaties om informatiebeveiliging te kunnen borgen!

Maarten Hartsuijker

Terwijl ik dit schrijf zit mijn hoofd vol met Powershell en ben ik de jaarlijkse lokale pentest voor één van mijn klanten aan het uitvoeren. De beheerrechten op het hele netwerk kwamen binnen een paar uur al binnen bereik. Evenals de toegang tot de netwerken van vele andere klanten van de IT-leverancier. Maar bij een goede pentest is dat voor de tester allemaal bijzaak en slechts een middel om aanbevelingen te illustreren en kracht bij te zetten. Het zijn aanbevelingen die kwetsbaarheden moeten verhelpen (die ook kwaadwillende hackers écht misbruiken). Deze kwetsbaarheden blijven ook in veel ISO 27001-gecertificeerde omgevingen massaal links liggen.

Ethische hackers zijn onmisbaar bij het overbruggen van dit gat tussen beveiligingsbeheersing (27001) en technische veiligheid. Ze kijken op hun eigen manier naar kwetsbaarheden. Ze weten wat er makkelijk misbruikt kan worden en wat vooral een theoretisch gevaar is. Ze zullen je organisatie

over het algemeen niet snel overladen met beveiligingsmaatregelen die nuttig lijken, maar totaal niet beschermen tegen de grootste gaten in je netwerk. Heb je een leverancier op bezoek die je de 'latest and greatest' securitysnufjes komt aanbieden? Laat een hacker eens kritisch meedenken of dit product de organisatie écht veiliger maakt. Zo levert een hacker meer geld op dan dat hij kost.

Sommige organisaties zijn terughoudend met het toelaten van hackers in hun netwerk. Wees gerust: die angst verdwijnt meestal na de eerste kennismaking. Al kan het wel zijn dat deze wordt ingeruild voor een onbestendig gevoel over al die leveranciersaccounts die in je netwerk aanwezig zijn. Vaak met eenvoudige wachtwoorden of wachtwoorden die de leverancier bij al zijn klanten gebruikt. Hoe zit het eigenlijk met jouw leveranciers? Vertrouw je ze op hun blauwe ISAE3402- of 27001-ogen? Of stuur je af en toe - onder eigen regie - een hacker op ze af, zodat je een ongefilterd beeld van de beveiliging van een afgenomen dienst krijgt?

Nicole van Deursen

Binnen ons vakgebied wordt veel gesproken over ethiek en gedragscodes voor hackers en voor andere IT-specialisten met specifieke kennis en vaardigheden. Soms staan we namelijk voor lastige keuzes. Stel dat we een kwetsbaarheid vinden, waarschuwen we dan publiekelijk of waarschuwen we alleen de leverancier, zodat die een oplossing kan publiceren? Houden we ons aan de wet (AVG) of delen we wel informatie over IP-adressen waarvan we vermoeden dat er een securityvoorval speelt? Is versturen van een nep-phishingmail een nuttig instrument om mensen iets te leren of is het pesten? Is endpointmonitoring een middel om datalekken te voorkomen of is het surveillance?

In de medische wereld zijn ethische overwegingen al een verschijnsel sinds de tijd van Hippocrates. In moderne vormen van de artseneed worden ook beloftes opgenomen dat er geen misbruik van de kennis zal worden gemaakt (bijvoorbeeld tijdens oorlog of onder commerciële druk). Ook zijn er de ethische commissies die oordelen of bepaalde onderzoeken en behandelingen verantwoord zijn. Voor ons vakgebied bestaan er inmiddels ook al diverse ethische codes waaronder die van de EC-council voor hackers (1), voor ISACA-leden en gecertificeerden (2), voor (ISC)2-gecertificeerden (3), en FIRST werkt aan een ethische code voor security incident response teams (4). IT-afdelingen kunnen natuurlijk ook zelf een gedragscode opstellen die als leidraad kan fungeren wanneer er gewerkt wordt met bepaalde tools of wanneer toegang tot gevoelige informatie onvermijdelijk is voor support of testwerkzaamheden.

Ethische commissies, die zich buigen over beslissingen en dilemma's zoals in de medische sector, zijn binnen ons vakgebied echter zeldzaam. Ik denk dat het handig kan zijn om zo'n commissie achter de hand hebben. Een ethische commissie zou kunnen adviseren in situaties waarin de wet, commerciële doelstellingen, het welzijn van de medewerkers, en de risico's niet met elkaar in balans zijn. Het kan ook een nuttig instrument zijn wanneer de CIO of de IT-directeur ook de CISO is en er dus soms een 'pettenprobleem' ontstaat. Een ethische commissie kan ook adviseren wanneer commerciële druk tegenhoudt dat er securitymaatregelen worden genomen (bijvoorbeeld preventief ethisch hacken of het installeren van patches), omdat de business angst heeft voor vertraging of verstoring van kritieke processen.

Patrick Dersjant

Dit blad besteedt al een tijdje regelmatig aandacht het gebruik van pentesten en andere middelen om kwetsbaarheden te vinden in de ICT-middelen die door een organisatie worden ingezet. Een hacker - een persoon die geniet van de intellectuele uitdaging om op een creatieve en onorthodoxe manier aan technische beperkingen te ontsnappen – wordt daarbij graag 'ethisch' ingezet, zodat de kennis over de kwetsbaarheid niet wordt misbruikt maar juist ervoor om dezelfde kwetsbaarheid te dichten. Vaker wel dan niet krijgt zo'n ethisch hacker echter een opdracht: beperk de pentest tot de bovenste drie ramen. Dat je gewoon om het gebouw heen kunt lopen om toegang te verkrijgen, doet niet ter zake. Je kunt prima met pentesten de veiligheid van 2-factor-authenticatie door middel van sms onderzoeken, maar als je niet stilstaat bij de procedure van je telecomaanbieder om simkaarten te vervangen, krijg je een verkeerd beeld van je risico's. Met andere woorden: hackers zijn het meest waardevol als je ze zo vrij mogelijk laat in hun opdracht!

Referenties

(1) https://www.eccouncil.org/code-of-ethics/
(2) http://www.isaca.org/certification/code-of-professional-ethics/pages/default.aspx
(3) https://www.isc2.org/Ethics#
(4) https://www.first.org/global/sigs/ethics/

COLOFON

IB is het huisorgaan van het Platform voor InformatieBeveiliging (PvIB) en bevat ontwikkelingen en achtergronden over onderwerpen op het gebied van informatiebeveiliging.

PvIB Platform voor InformatieBeveiliging

REDACTIE
Tom Bakker
Bianca Brooijmans
Patrick Dersjant
Nicole van Deursen
Maarten Hartsuijker
Lilian Knippenberg
Rachel Marbus
Fook Hwa Tan
Chris de Vries

BLADMANAGEMENT
MOS bv
José Broekhuizen
Lisa Petersen
E ibmagazine@pvib.nl

ADVERTENTIE-ACQUISITIE
MOS bv
Jan van de Vis
E acquisitie@mos-net.nl
T 033 247 34 00

VORMGEVING
Neverseen Art & Design
Dimitri van den Berg

DRUK
VDR druk & print

UITGEVER
Platform voor Informatiebeveiliging (PvIB)
Postbus 1058
3860 BB NIJKERK
T (033) 247 34 92
E secretariaat@pvib.nl
W www.pvib.nl

ABONNEMENTEN
De abonnementsprijs in 2019 bedraagt € 118,50 (exclusief btw), prijswijzigingen voorbehouden.

ABONNEMENTENADMINISTRATIE
Platform voor Informatiebeveiliging (PvIB)
Postbus 1058
3860 BB NIJKERK
E secretariaat@pvib.nl

ISSN 1569-1063

COLUMN Berry

Berry eindelijk aan de Android?

Berry komt uit de tijd dat je - in ruil voor een kwartje - je telefoontje mocht plegen op een mooi bakelieten toestel. De jongeren onder ons zullen nu de wenkbrauwen fronzen, maar toch was dit vroeger heel normaal. Je werd niet voor gek verklaard als je dit vroeg aan je buurman. Inmiddels zijn we 50 jaar verder en is de bakelieten telefoon alleen nog in het museum te vinden en heeft iedereen zijn eigen telefoon. Een paar jaar gelden was er nog keuze in het besturingssysteem, maar dat is helaas ook niet meer zo. We hebben nog twee smaken en een paar kruimeltjes van besturingssystemen (die meer gadgets geworden zijn en in de markt niet meer meespelen).

Nee, de markt bestaat eigenlijk nog maar uit één besturingssysteem (die op 9 van de 10 telefoons staat) en dat is Android, het bedenksel van Google. Het andere type is de iPhone van Apple. Apple heeft een goed werkende telefoon op de markt gezet, maar heeft de omgeving erg gesloten gehouden. Een bewuste keur van Apple, want het bedrijf wil alleen maar apps aanbieden die getest zijn, veilig zijn en geen malware kunnen bevatten. Bovendien zijn Apple-telefoons veel duurder. Mensen die een telefoon nodig hebben, kijken op de markt en zien dat de Android-toestellen veel aantrekkelijker geprijsd zijn.

Het zal u niet verbazen dat ik wel een (te) dure iPhone heb. Apple heeft controle over zowel hardware als software en kan snel bedreigingen wegnemen door beveiligingsupdates. Google is bijna niet in staat om voor alle merken en types toestellen beveiligingsupdates uit te brengen, zodat sommige fabrikanten het ook zelf wel doen, of niet natuurlijk. Daarbij willen de meeste fabrikanten een eigen kleur geven aan hun Android-machines door software te maken die ze onderscheid van andere Android-leveranciers. Tja, dat je dan niet de beveiligingsupdates van Google kunt installeren, is dan jammer. Sterker nog, nu blijkt dat juist in die onderscheidende software vaak malware zit opgesloten die niet eens te verwijderen is. Wordt goedkoop dan toch duurkoop?

In het kader van beveiliging durf ik ook wel te stellen dat een aankoop van het merk iPhone een beter alternatief is voor je nieuwe telefoon. Zijn er dan helemaal geen apparaten die je zou moeten kopen en die onder Android werken? Jawel hoor, als je een Android wilt aanschaffen, kies dan voor de Pixel. Deze toestellen zijn door Google gemaakt en kunnen altijd de beveiligingsupdates (die wekelijks worden aangeboden) plaatsen. Dit is ook altijd het idee geweest van Google bij de ontwikkeling van Android. Als je overigens in de prijslijst kijkt, komen ze al aardig in de buurt van de iPhone.

Berry

SECURITY ACADEMY

Actief binnen de overheid? Leer alles over de BIO!

De Security Academy heeft nu de 2-daagse cursus Baseline Informatiebeveiliging Overheid in haar opleidingsaanbod.

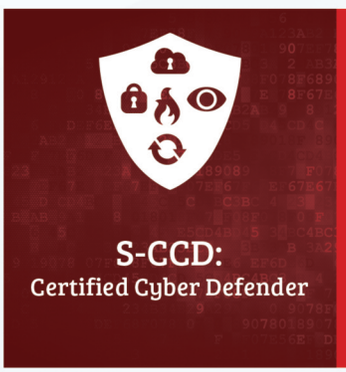
Leer in 2 dagen de specifieke implementatie- en audit en control kant van deze nieuwe baseline.

Kijk voor de cursusomschrijving op onze website.

www.securityacademy.nl info@securityacademy.nl +31(0)348-40 80 61

SECURITY ACADEMY

Je snel ontwikkelen in de wereld van Cyber Security?



Deze fast track cursus is opgebouwd uit:

- Information Security Foundation
- IT-Security Foundation
- Privacy & Data Protection Foundation
- Business Continuity Foundation
- Crisis Management Foundation

Kijk voor de cursusomschrijving op onze website.

www.securityacademy.nl info@securityacademy.nl +31(0)348-40 80 61