Project Cyberveilige Gemeenten

Richtlijnen SSD

Aan de slag met Secure Software Development

Dit document kwam tot stand in kader van het **Project Cyberveilige Gemeenten** in samenwerking met minister van Binnenlands Bestuur Bart Somers, het Agentschap Binnenlands Bestuur, Audit Vlaanderen en Digitaal Vlaanderen. De Richtlijnen Secure Software Development werden uitgewerkt in samenwerking met de taskforce bij het project, op basis van bestaand bronmateriaal van lokale besturen en externe partners.

Een speciaal woord van dank gaat uit naar het **Centrum Informatiebeveiliging en Privacybescherming**, wiens bronmateriaal herbruikt en herwerkt mocht worden voor het opstellen van dit document. Voor het opstellen van dit bronmateriaal werkte het CIP samen met professionals uit verschillende organisaties binnen het CIP-netwerk, zowel uit de overheid als de private sector.

Richtlijnen voor Secure Software Development	
Datum	November 2021
Status	Versie 2.2
Opdrachtgever	Centrum Informatiebeveiliging en Privacybescherming, VVSG
Regime	Becommentarieerde praktijk
Auteurs	Marcel Koers (CIP), Rob van der Veer(SIG)
Reviewers	Domeingroep SSD van het CIP, VVSG (Taskforce Cyberveilige Gemeenten)

Voorwoord

De Richtlijnen voor Secure Software Development opgezet vanuit het perspectief van een lokaal bestuur die regisseert en stuurt op beveiliging tijdens ontwikkeling van software, zonder te willen inbreken in het ontwikkelproces van interne/externe softwareleveranciers/opdrachtnemers. Zo verrijkt deze procesbeschrijving de internationaal erkende modellen voor softwareontwikkelingsprocessen. Dit document is tot stand gekomen door nauwe samenwerking tussen verschillende partijen (opdrachtgevers, opdrachtnemers en adviesorganisaties) en gebaseerd op uiteenlopende ervaringen en kennis uit de praktijk en literatuur.

De bestaansreden van dit document

Het blijkt meer dan eens een uitdaging om als lokaal bestuur te voorzien in de nodige informatiebeveiliging bij het ontwikkelen van software en andere relevante IT-projecten. Zij het door het ontbreken van interne expertise of een gebrek aan tijd een middelen. Uitbesteding van ontwikkeling, onderhoud en beheer aan externe leveranciers maakt deze sturing complexer. Tussen opdrachtgever (lokaal bestuur) en opdrachtnemer (leverancier) zijn er immers dikwijls onuitgesproken verwachtingen rondom informatiebeveiliging. De opdrachtgever verwacht een deskundige opdrachtnemer die spontaan de juiste maatregelen treft. Daarentegen verwacht de opdrachtnemer dat de opdrachtgever precies specificeert wat er moet gebeuren. Door het ontbreken van expliciete afspraken worden systemen opgeleverd met kwetsbaarheden die niet of te laat worden ontdekt.

De SSD-methode omvat drie pijlers:

1. **Centrale beveiligingsuitgangspunten**: Deze vormen de basis voor de eisen die aan de opdrachtnemer worden gecommuniceerd;
2. **Contactmomenten.** De geëigende momenten om als opdrachtgever sturing te geven aan het softwareontwikkelproces;
3. **SSD-processen**. Dit zijn processen aan opdrachtgeverskant, al dan niet als extra activiteit binnen bestaande processen. Deze hebben onder meer als doel het bijhouden van risico's, centrale beveiligingsuitgangspunten en het volwassenheidsniveau van de opdrachtgevende organisatie. Daarnaast hebben de Richtlijnen Secure Software Development uiteraard invloed op de processen aan de kant van opdrachtnemer/ontwikkelorganisatie.

Belangrijkste termen

Term	Uitleg
Opdrachtnemer	Een interne of externe partij die de software ontwikkelt (omvat ontwerp, bouw en testen) en oplevert.
Hostingpartij	De partij of faciliteit die de technische infrastructuur levert waarop de software draait. Er zijn variaties in plaatsing (intern/extern) en mate van technische ontzorging (bijv. hardware, platform, SaaS).
Software	Product en werkzaamheden ten behoeve van opdrachtgever, zoals nieuwbouw/herbouw/renovatie van web/mobiel informatiesysteem/ (web)applicatie, een nieuwe release/grote wijziging op een bestaand informatiesysteem.
Ontwikkelaar	Softwareprogrammeur
Opdrachtgever	De rol/persoon die verantwoordelijk is voor het uitgeven van de opdracht tot ontwikkeling. Binnen dit document nemen we het lokaal bestuur als voornaamste opdrachtgever, de doelgroep van deze richtlijnen.
Ontwikkelproces	Het voortbrengingsproces van software aan de kant van de opdrachtnemer.
IT-systeem	Het samenstel van één of meerdere IT-functies die een gezamenlijke functie aanbieden. De afbakening is afhankelijk van relevante grenzen in de bedrijfsvoering.
Zwakheid	Een zwakheid is een tekortkoming die mogelijk tot een beveiligingsincident kan leiden. Voorbeeld: wachtwoorden worden onversleuteld opgeslagen in een bestand. Mocht het een kwaadwillende lukken om dat bestand te bemachtigen, dan liggen alle wachtwoorden op straat.
Kwetsbaarheid	Een kwetsbaarheid is een misbruikbare zwakheid in het systeem; een tekortkoming waarvan zeker is dat deze tot een beveiligingsincident kan leiden. Voorbeeld: men is vergeten toegangscontrole in te bouwen op een webpagina met vertrouwelijke gegevens waardoor een kwaadwillende direct toegang kan krijgen door het adres in te typen op het internet.
Dreiging	De mogelijkheid dat een aanvaller of situatie zorgt dat een kwetsbaarheid tot een beveiligingsincident leidt. Algemeen gezien kan dit ook een situatie zijn waarbij een systeem in ongewenste staat raakt (bijv. grootschalige stroomuitval leidt tot systeemuitval).
Risico	Een inschatting van een ongewenste gebeurtenis op basis van een dreiging met inschatting van kans en impact.
Architecten en officers	Een architect is hier een technisch specialist met een ontwerp/bewaak rol in de organisatie. Denk hierbij aan een systeembeheerder, IT-diensthoofd, functionaris gegevensbescherming (DPO) of veiligheidsconsulent (CISO).
Security/informatiebeveiliging	Dit is een verzamelnaam die moet worden gelezen in een brede context en omvat ook de eisen voor privacybescherming (bijvoorbeeld de AVG/GDPR).
DPIA	Data Protection Impact Assessment is een risicoanalyse in de context van privacybescherming. Letterlijk vertaald als gegevensbeschermings-effect-beoordeling.

1. Inhoudsopgave

2. Inleiding en doelstelling

1.1 Waarom deze methode? Omdat “veilige software” niet vanzelfsprekend is

Veilige software is een breed technisch gebied met veel afhankelijkheden. Daarom zijn verwachtingen tussen opdrachtgever en opdrachtnemer niet vanzelfsprekend. Wat voor de één veilige software is, is voor de andere niet veilig genoeg. Een dialoog met de opdrachtnemer is daarvoor noodzakelijk. De Richtlijnen voor Secure Software Development beschrijven hoe die gezamenlijke verantwoordelijkheid verdeeld is: eisen opstellen, communiceren, daarover in overleg gaan, opvolgen en toetsen. De doelmatigheid van het uitgeschreven proces vatten we zo samen:

- Duidelijke dialoog met een opdrachtnemer is nodig voor het balanceren van wensen en technische consequenties, die kunnen veranderen door voortschrijdend inzicht.
- Vroege sturing en afspraken zijn goedkoper, leiden tot veiligere software en een hechtere relatie tussen opdrachtgever en opdrachtnemer. Kwetsbaarheden te laat ontdekken is pijnlijk, door beveiliging en afspraken zo vroeg mogelijk in te brengen kunnen kostelijke en lastige herstelwerken vermeden worden.

1.2 Doelstelling

Dit document presenteert een methode voor afstemming tussen opdrachtgever en opdrachtnemer als het gaat om realiseren van veilige software, zonder daarbij de precieze invulling van de ontwikkelprocessen bij de opdrachtnemer voor te schrijven. De methode is toepasbaar bij verschillende ontwikkelmethodieken (waterval, agile) en is geschikt voor maatwerk- én standaardpakketten.

1.3 De Grip op SSD-familie van documenten

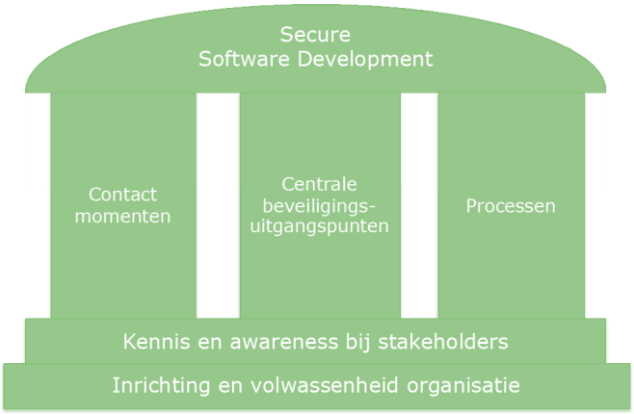
De richtlijnen Secure Software Development werden opgesteld op basis van de Grip op SSD-documenten, die geraadpleegd kunnen worden op https://www.cip-overheid.nl/

- Grip op SSD: De methode - over de samenwerking opdrachtgever-opdrachtnemer
- Grip op SSD: De normen - basis technische beveiligingseisen om toe te passen in de methode, met per aspect wie daarvoor zorg moet dragen
- Grip op SSD: De normen voor mobiele apps - basis technische beveiligingseisen om toe te passen in de methode, specifiek voor mobiele applicaties.
- Grip op SSD: Testraamwerk - over hoe de normen kunnen worden getoetst
- Agile securitymanagement: hoe een agile ontwikkelproces kan zorgen voor het inbouwen van informatiebeveiliging. Deze is het meest relevant voor ontwikkelaar

3. Globaal overzicht

Om te komen tot veilige software, zonder in te hoeven grijpen in het ontwikkelproces voor de software, kent deze SSD-methode drie pijlers:

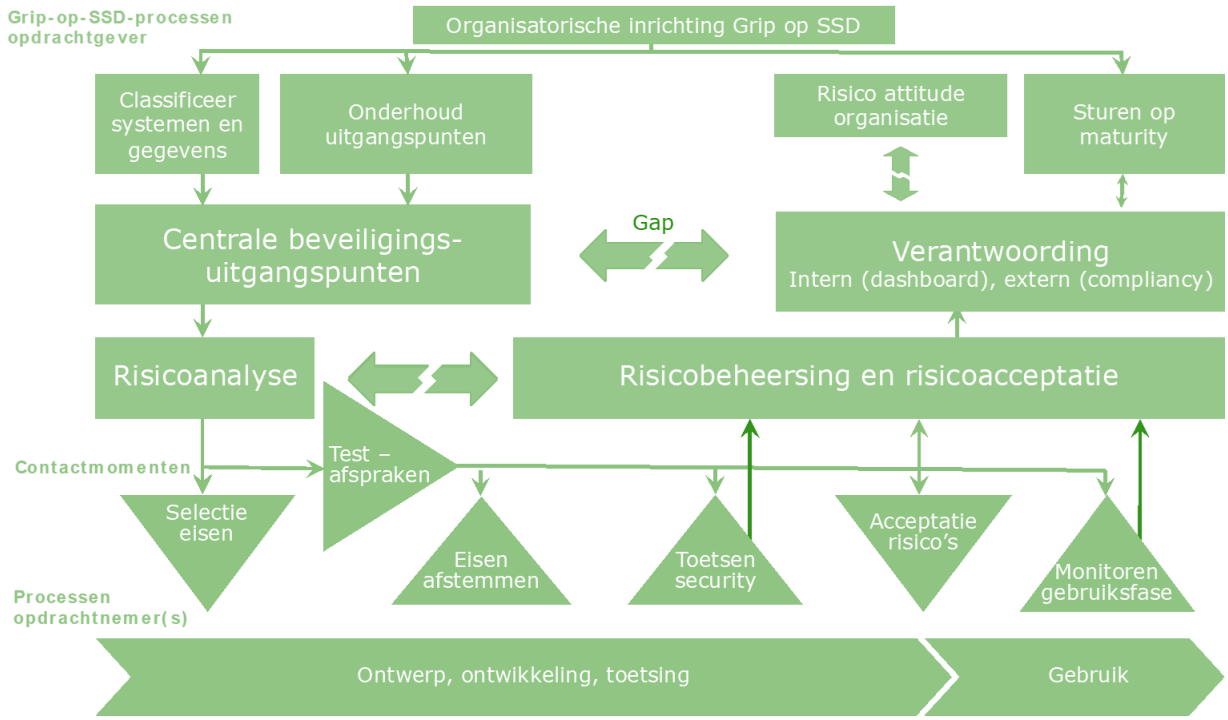
- De centrale beveiligingsuitgangspunten (hoofdstuk 5),
- De contactmomenten (hoofdstuk 6),
- De SSD-processen (hoofdstuk 7).



Het fundament onder de pijlers is kennis en awareness bij de beveiligingsadviseurs, procesdeskundigen, architecten, ontwerpers en testers. Via een groeiproces wordt awareness/bewustzijn gecreëerd bij de stakeholders/belanghebbenden en wordt kennis en ervaring opgedaan over de te hanteren eisen en processen.

Hierbij dient de opdrachtgever te toetsen/meten of de kennis en bewustzijn daadwerkelijk groeien. Voor dit meetproces zijn volwassenheidsniveaus gedefinieerd die overeenkomen met die van het Capability Maturity Model (CMM). Voor de meeste organisaties is CMM-niveau 3 voor SSD afdoende, namelijk ‘vastgesteld proces (established)’.

De relatie van de pijlers met het ontwikkelproces is als volgt:



Afbeelding 2: relatie pijlers van de SSD-methode met het ontwikkelproces

Bij de klassieke aanpak van softwareontwikkeling (‘waterval’) geeft de opdrachtgever een volledig gespecificeerde opdracht aan de opdrachtnemer en ontvangt na enige tijd het complete product. Na een acceptatieproces wordt dit product daarna in productie genomen. Voor informatiebeveiliging is deze aanpak in de praktijk zeker niet ideaal, zeker niet in de context van web-ontwikkeling waarin snelle reactietijden nodig zijn voor het oplossen van beveiligings- en gebruikersklachten.

In een ontwikkelproces volgens ‘agile’ uitgangspunten gebeuren de taken rond ontwerp, ontwikkeling en toetsing in korte cycli. Vereenvoudigd gezien worden bijvoorbeeld ontwerpen niet systeembreed uitgewerkt vooraleer de eerste regel code geschreven wordt (de ‘klassieke’ ontwikkelwijze). De contactmomenten uit de Richtlijnen Secure Software Development blijven voor een agile proces in de kern gelijk. Voor ‘security toetsen’ mag een opdrachtgever meer tussentijdse en voorlopige resultaten verwachten. De korte cycli die bij agile horen betekenen ook dat niet elke test of toets bij elke iteratie wordt uitgevoerd. Testen die automatiseerbaar zijn mogen zo vaak mogelijk door het constructieproces (build pipeline) lopen als praktisch haalbaar, maar duurdere, specifieke testen zoals penetratiests en code reviews worden nog steeds op specifieke basis toegepast.

Toepassing op bestaande systemen en standaardpakketten

Voor bestaande systemen en standaardpakketten geldt dat er per definitie geen ruimte is om eisen voorafgaand en tijdens de ontwikkeling te communiceren. Daarnaast zijn de mogelijkheden voor toetsing (bijvoorbeeld automatische testrapporten of code review) bij standaardpakketten typisch beperkt. Als bestaande systemen of standaardpakketten in een laat stadium niet blijken te voldoen aan bepaalde eisen dan kan de opdrachtnemer typisch niet verantwoordelijk worden gehouden. Toch is het toepassen van de Richtlijnen Secure Software Development aan te raden voor bestaande systemen en standaardpakketten, met de volgende doelstellingen:

- **Adressering van de eisen met opdrachtnemer**: Het communiceren van en afstemmen over eisen maakt in elk geval bespreekbaar wat beide partijen doen als verwachtingen niet volledig voldaan worden. Een mogelijk gevolg is het herzien van (eventueel contractuele) afspraken in de huidige samenwerking voor nieuw ontwikkelwerk. Eventueel meerwerk kan dan in kaart worden gebracht en geprioriteerd worden met opdrachtgever. Risico’s door eisen waar niet aan kan worden voldaan, kunnen mogelijk met andere tegenmaatregelen worden gecontroleerd – technisch of organisatorisch, zoals bijvoorbeeld het inregelen van meer strikte monitoring (operationeel volgen van systeemgedrag).
- **Interne adressering van de eisen:** Een deel van de eisen kan betrekking hebben op zaken die onder regie van de eigen organisatie vallen, zoals bepaalde onderdelen van de infrastructuur en het beheer.
- **Selectie van standaardpakketten:** Als een bestaand systeem aangeschaft moet worden, kunnen deze richtlijnen ondersteunen bij de keuze. Dit betreft het inbrengen, afstemmen en toetsen van en bespreking van wat de opdrachtnemer wil en kan doen met afwijkingen.

4. Beginnen met SSD

De Richtlijnen Secure Software Development zijn niet bedoeld voor een ‘big bang’, een grote verandering ineens, aangezien meerdere rollen binnen de organisatie betrokken zijn. De groei gaat als organisatie “samen”, door ontwikkelen van ervaring, routines, standaard werkprocessen en opbouw van gedocumenteerde kennis (bijvoorbeeld een overzicht van risico’s en standaardmaatregelen). De SSD-volwassenheidsniveaus helpen bij doelstellingen van projectvoorstellen.

Ook zijn volwassenheidsniveaus meetpunten om te bepalen hoever de organisatie staat met betrekking tot het verkrijgen en inzetten van veilige software.

Stap 1: Nulmeting

Een nulmeting betekent vaststellen waar de organisatie nu staat: in hoeverre worden zaken nu al effectief aangestuurd en waar zitten de gebreken?

Deze analyse is een typische taak van architecten of business analisten: rollen die het ‘nu’ en ‘toekomst’ van de organisatie schetsen en het verschil daartussen. In architectentermen wordt vaak gesproken over Ist, Soll en Gap. De Gap, de vastgestelde achterstand op wat de organisatie wil, vertelt wat er ‘ongeveer nodig is’. Dat kan worden opgenomen in een (herhalend) informatieveiligheidsplan en de initiatieven/projecten daarvoor.

Bij het opstellen van het tijdschema is voorzichtigheid geboden, aangezien SSD deels is gebaseerd op een cultuuromslag. Die kan men alleen geleidelijk realiseren.

Stap 2: Toekennen rollen

Binnen de doelorganisatie moeten de juiste personen worden gekoppeld aan de taken voor SSD en zij moeten voldoende mandaat (besliskracht) hebben.

De SSD-methode is primair gericht op ondersteuning van de opdrachtgever, het lokaal bestuur. Hierbij moet de opdrachtnemer begrijpen hoe de opdrachtgever tot het formuleren van de eisen is gekomen en welk belang hij of zij daaraan hecht. Daarnaast zijn er de adviespartijen die met specifieke kennis de kwaliteit van de processen van zowel opdrachtgever als opdrachtnemer verder kunnen helpen verbeteren.

Stap 3: Pilootproject

Start de kennismaking met de Richtlijnen Secure Software Development door middel van een pilot voor één systeem. Selecteer hiervoor een belangrijk systeem zodat meteen waarde voor de organisatie wordt gecreëerd. Bij meerdere opdrachtnemers: prefereer degene waar al zoveel mogelijk sprake is van wederzijds vertrouwen zodat de kans op complicaties door de relatie bij de pilot minimaal is. Een systeem waar de tijdslijnen kort zijn is niet ideaal, omdat ervaring opdoen met de verschillende onderdelen uit dit proces de eerste keer meer tijd kost.

Eénmaal geselecteerd: ga voor dit systeem aan de slag met de handreikingen uit de drie pijlers (zie de volgende hoofdstukken): leg de uitgangspunten vast, maak contact met de opdrachtnemer en richt de nodige processen in.

Stap 4: Evalueer en groei

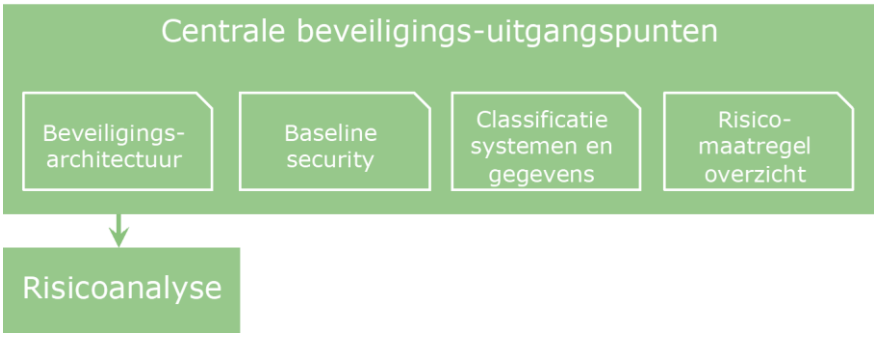
Nadat de pilot is geëvalueerd, kan verder worden geïtereerd over het IT landschap en al lerend stappen worden gezet naar verdere volwassenheid. Eén van de SSD processen is het sturen op volwassenheid (zie 0). Daarin staan de processen uitgewerkt voor verschillende stappen in de groei.

5. Pijler 1: Centrale beveiligingsuitgangspunten

De centrale beveiligingsuitgangspunten vormen de kennisbron binnen de Richtlijnen Secure Software Development. Ze dienen om te voorkomen dat bij elk project weer opnieuw alles over beveiliging moet worden bedacht. Ze vormen de basis voor het samenstellen van de juiste **specifieke** beveiligingseisen per situatie.

De centrale beveiligingsuitgangspunten dienen te bestaan uit:

- Beveiligingsarchitectuur, of 'architectuurblokken';
- Baseline security (een minimumniveau van beveiliging in de vorm van een set standaard eisen);
- Classificatie van systemen en gegevens rond Beschikbaarheid, Integriteit en Vertrouwelijkheid;
- Risico-maatregel overzicht.



Afbeelding 1: pijler 1 - Centrale beveiligingsuitgangspunten

Een beveiligingseis kan bijvoorbeeld zijn dat twee-factor authenticatie verplicht is voor toegang tot een portaal waar klantgegevens inzichtelijk zijn. De "tweede factor" naast een wachtwoord is dan typisch een geregistreerde telefoon/telefoonnummer waarop een gebruiker een bericht of "challenge" ontvangt om de toegang te bevestigen.

Belangrijk onderscheid functionele en non-functionele eisen

Het is belangrijk om het onderscheid te kennen tussen functionele eisen en non-functionele eisen. Een functionele beveiligingseis gaat over wat het systeem voor de gebruiker moet doen - bijvoorbeeld twee-factor authenticatie. Deze wordt typisch als functionele specificatie opgenomen en bijvoorbeeld in een agile ontwikkelproces tot een story vertaald. Een non-functionele beveiligingseis definieert niet een gebruikersfunctie maar de kwaliteit van de implementatie - bijvoorbeeld dat gegevens die getoond worden op een website worden geschoond van eventuele ongewenste code. Deze eisen vinden hun weg als criteria voor het ontwikkelwerk, die bijvoorbeeld in een agile ontwikkelproces worden getoetst via de acceptatiecriteria van een story.

Invloed operatie van systeem op beveiligingseisen en benodigde kennis opdrachtgever

Met “operatie” bedoelen we de technische infrastructuur waarop een systeem geïnstalleerd is. Een systeem “komt tot leven” of “draait” op die infrastructuur. Dat zijn o.m. servers, raamwerken specifiek voor applicaties en webservers en de “periferie” die netwerktoegang regelen, zoals firewalls. De operatiewijze heeft invloed op de taken, verantwoordelijkheden en vereiste expertise bij de opdrachtgever. Zie de tabel hieronder.

Wijze van operatie	Taken en verantwoordelijkheden	Vereiste expertise opdrachtgever
Eigen beheer	Zelf installeren, runnen, monitoren.	Hoog
Publieke cloud	Zelf configureren, beheren en monitoren.	Hoog
Centrale/vertrouwde (overheidsfaciliteit)	De mate van ontzorging en garanties zijn afhankelijk van de “breedte” van het technische aanbod (bijvoorbeeld IAAS/PAAS/SAAS/FAAS).	Laag-hoog
Uitbesteed	Uitbesteden taken met afgesproken garanties (in SLA). Volwassenheid opdrachtnemer is van invloed op vereiste deskundigheid en sturing. Van te voren verwachtingen en eisen definiëren.	Laag-middel

Beveiligingsarchitectuur

In de context van de Richtlijnen Secure Software Development, betreft de beveiligingsarchitectuur een beschrijving van de al bestaande technische beveiligingsmaatregelen. Meestal zijn dit specifieke groepen componenten in de infrastructuur. Dergelijke componenten voorkomen dat bedrijfsbrede beveiligingsmaatregelen versnipperd worden ingericht en beheerd. Voorbeelden hiervan zijn een centraal mechanisme voor toegangscontrole (bijvoorbeeld een IAM zoals Active Directory), de netwerkcomponenten en diverse vormen van middleware en platformen. De risico’s die al zijn afgedekt met standaard beveiligingsmaatregelen hoeven dus niet nogmaals te worden afgedekt in de te bouwen applicatie. Wel blijven ze deel van de beveiligingseisen omdat anders niet controleerbaar is of ze juist zijn toegepast.
In ketenverband kunnen sommige van de componenten ook buiten de eigen organisatie liggen, zoals de voorzieningen voor digitale identiteit (zoals DigID of eID).

Baseline security

De baseline security beschrijft een set relevante standaardeisen, die een ‘minimaal niveau van beveiliging’ vertegenwoordigen. De baseline security wordt afgestemd met de opdrachtnemer bij wijze van verificatie, bijvoorbeeld door te toetsen ‘Is deze set van normen haalbaar? Zijn er problemen voorzien om deze te realiseren?’.

Een overzichtelijke lijst van standaardeisen is te vinden in het werk ‘Grip op SSD - de normen’ op https://www.cip-overheid.nl/. Zie bijlage Bijlage: SSD Normen voor een beknopt overzicht. Naast deze lijst bestaat ook een versie van deze normen specifiek voor mobiele applicaties.

De classificatie van systemen en gegevens (zie □o) kan resulteren in inzichten waarmee bepaalde standaardeisen kunnen vervallen en eventueel nieuwe standaardeisen worden toegevoegd, afhankelijk van de specifieke situatie. Voorbeeld: direct uitloggen zodra een gebruiker klaar is bij een applicatie die werkt met zeer vertrouwelijke gegevens.
Enkele andere bronnen met relevante beveiligingseisen zijn:

- **ISO/IEC 27002:2013,** welke documentatie- en procescontroles beschrijven.
- **OWASP Application Security Verification Standard:**
  De OWASP ASVS biedt een basis om de beveiligingsmaatregelen van (web)applicaties te toetsen; Een overzicht van eisen en hoe ze binnen OWASP, NIST en Mitre worden beschreven is te vinden op https://www.opencre.org/
- **Specifiek voor België**:
  - Richtsnoeren met betrekking tot de informatiebeveiliging van persoonsgegevens, van de Gegevensbeschermingsautoriteit (België)
  - Minimale normen informatieveiligheid en privacy, van de KSZ (België)

Classificatie van systemen en gegevens

De classificatie van systemen en gegevens is het resultaat van het classificeringsproces dat wordt uitgevoerd door de eindverantwoordelijke voor een bedrijfsproces. Hierin worden de gegevensstromen, informatiesystemen en gegevensverzamelingen in kaart gebracht en geclassificeerd voor de kwaliteitsaspecten Beschikbaarheid, Integriteit en Vertrouwelijkheid (BIV). Voor iedere klasse geldt een minimale basis van de te stellen beveiligingseisen en de te nemen beveiligingsmaatregelen, zoals het wel of niet versleutelen van gegevens.

Risico-maatregel overzicht

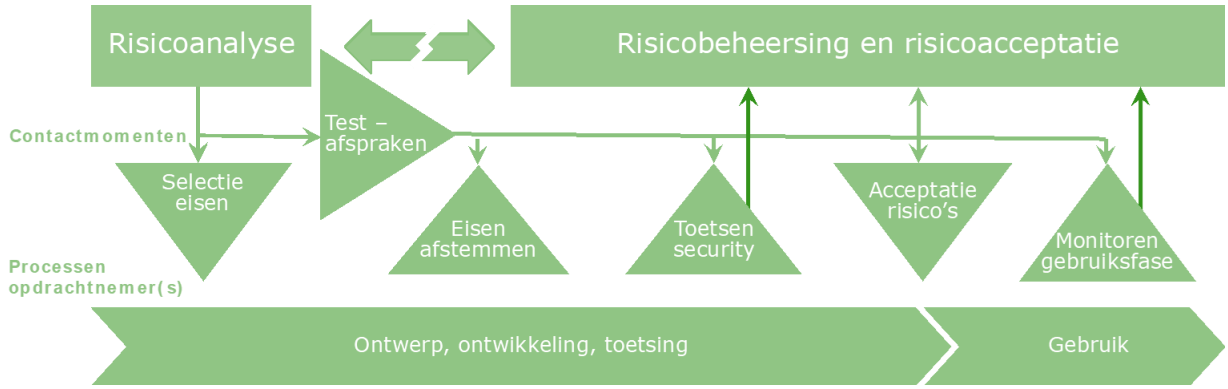
Het risico-maatregel overzicht verzamelt eerder geïnventariseerde risico's en de daarvoor bedachte maatregelen. Door deze centraal in de organisatie bij te houden, kunnen nieuwe risicoanalyses (zie 0) voortbouwen op bestaand werk. Daarmee ontstaat een samenhangende risicobeheersing en een overzicht van gepaste standaard maatregelen (best practices).

6. Pijler 2: Contactmomenten

De SSD-methode gebruikt contactmomenten tussen opdrachtgever en opdrachtnemer voor, tijdens en na de ontwikkeling van de software om ervoor te zorgen dat de software voldoet aan de eisen. Voorafgaand aan de bouw stelt de opdrachtgever de beveiligingseisen op. Tijdens de bouw wordt tussen opdrachtgever en opdrachtnemer afgestemd over de eisen en tijdens / na de bouw wordt gecontroleerd of de software voldoet. Op de contactmomenten kan, wanneer nodig, tijdens de ontwikkeling worden bijgestuurd.

De zes contactmomenten zijn:

1. Selectie eisen;
2. Testafspraken;
3. Eisen afstemmen;
4. Toetsen security;
5. Acceptatie van risico's;
6. Monitoren in de gebruiksfase.



Afbeelding 2: pijler 2 - Contactmomenten

Organisatorische eisen aan de opdrachtnemer

De Richtlijnen Secure Software Development veranderen het ontwikkelproces niet, maar brengt er een uitbreiding op aan met de contactmomenten. Daarmee vereist de methode dat er een gestructureerd ontwikkelproces bestaat bij de leverancier waardoor het mogelijk is eisen in te brengen, daar gevolg aan te geven en daarop te toetsen. Hoe eerder de leverancier in het ontwikkelproces stuurt op de eisen, hoe eerder afstemming over eisen kan plaatsvinden en onnodig herstelwerk in een latere fase kan worden voorkomen. Voor dit aspect wordt ook wel de term 'shift left' gebruikt: meenemen van eisen en toetsen meer 'links' in de tijd, dus eerder in het ontwikkelproces. Aangezien de opdrachtgever hierbij gebaat is, kan de leverancier gevraagd worden om te onderbouwen hoe hier procesmatig aan wordt voldaan of eventueel hierop worden geaudit. Daarnaast is het bij maatwerksoftware van belang om de software zelf te kunnen auditen, dus het recht op inzage van de broncode en de mogelijkheid om de software werkend te testen.

Andere onderwerpen voor organisatorische afspraken zijn volwassenheid van werkprocessen, relevant (veiligheids)beleid en formele accreditatie (bijvoorbeeld ISO

27xxx). Daarnaast kan toetsing van betrokken personen worden geëist (certificering, integriteitstoetsen). Zie bijlage A voor een overzicht van organisatorische eisen. Deze spelen een rol bij bestaande opdrachtnemers en uiteraard ook bij selectie van nieuwe leveranciers.

Tot slot: het is belangrijk om vast te stellen dat de opdrachtnemer ook kritisch kijkt naar haar externe leveranciers en de producten daarvan. Uiteindelijk bestaat moderne software vaak uit een samenstelling van componenten uit een keten van leveranciers. Open source raamwerken en bibliotheken zijn een goed voorbeeld hiervan.

Selectie eisen

De beveiligingseisen voor een specifiek systeem bestaan uit:

1. de baseline security;
2. de vereisten volgens de classificatie van betrokken gegevens en andere systemen (specifieke maatregelen die passen bij een classificatie van data, bijvoorbeeld het type versleuteling of toegangscontrole voor een bepaalde bron gegevens);
3. eventueel additionele eisen die voortkomen uit een risicoanalyse voor het systeem.

Alle overige informatie daaromheen is belangrijke context voor de opdrachtnemer: wat is belangrijk om te beveiligen, wat is de mate waarin dat moet gebeuren en welke beveiligingsarchitectuur is al aanwezig.

Het opstellen van beveiligingseisen gebeurt doorgaans eenmalig per project of deelproject. Deze eisen worden meegenomen in een Project Start Architectuur (PSA) of in het Functioneel Ontwerp (FO). Bij een aanbesteding worden de eisen opgenomen in het Programma van Eisen (PvE).

Maak testafspraken met opdrachtnemer

De testafspraken gaan over hoe men zal vaststellen dat het systeem voldoet aan de beveiligingseisen. Daarbij wordt niet alleen gekeken naar de toets voor acceptatie, maar naar tests in de gehele ontwikkel- en gebruiksfase. Bij korte ontwikkelfases (‘sprints’) is het gebruikelijk dat er frequent testresultaten beschikbaar zijn.

“Acceptatie” bedoelen we hier breed, inclusief het accepteren van niet-functionele eisen, dus niet alleen functionele acceptatie (typisch bekend als FAT).

Een voorbeeld van testafspraken, is dat de opdrachtnemer tijdens het ontwikkelproces inzage verschaft in de resultaten van testverslagen, waarbij onder meer gebruik wordt gemaakt van statische code analyseprogramma’s. Daarnaast wordt voor de oplevering nog een codereview uitgevoerd door de opdrachtnemer zelf en een pentest door een extern bureau.
Opdrachtgever en opdrachtnemer spreken af welke testaanpak wordt toegepast en welke testresultaten naar de opdrachtgever worden gecommuniceerd en hoe. Zo is het mogelijk af te spreken dat een opdrachtgever aanwezig is bij bepaalde sessies van de opdrachtnemer waar functionele en nonfunctionele toetsing besproken wordt, zoals een demosessie in scrum agile. Van een opdrachtnemer mag verwacht worden dat deze actief

aantoont welke maatregelen genomen zijn om non-functioneel systeemgedrag te garanderen (met name security, betrouwbaarheid, performance).

Blijf eisen afstemmen met opdrachtnemer

Eisen worden typisch gesteld in de vorm van *comply or explain (pas toe of leg uit)*: het kan zijn dat door technisch voortschrijdend inzicht tijdens ontwerp en ontwikkeling duidelijk wordt dat een eis onevenredig veel werk met zich meebrengt, of onevenredige nadelen brengt, zoals bijzonder ongemak voor gebruikers. Ook kan blijken dat het achterliggende risico op een betere manier kan worden afgedekt. Dit vereist wederzijdse afstemming. Als de opdrachtnemer niet wil of kan voldoen aan een bepaalde eis, moet opdrachtgever een proces voor risicoacceptatie doorlopen, met de afwegingen van de opdrachtnemer als input.

Het is aanbevolen om met de opdrachtnemer(s) vóór de contractering te overleggen over de inhoud van de centrale beveiligingsuitgangspunten. Daaruit komen de specifieke beveiligingseisen voort. Hierdoor weten opdrachtnemers beter welke beveiligingseisen zij kunnen verwachten. Bij hun prijsstelling kunnen zij daar rekening mee houden. Ook kunnen opdrachtnemers vaak waardevolle aanvullingen of feedback geven op beveiligingseisen, die de kwaliteit en effectiviteit van de eisen kunnen verbeteren.

(Laat) security toetsen

Het contactmoment Security toetsen heeft als doel voor de opdrachtgever om vertrouwen te krijgen dat de software aan de beveiligingseisen voldoet. Dit kan de opdrachtgever doen door het (laten) beoordelen van testresultaten van de opdrachtnemer, door het zelf uitvoeren van tests, of door het laten uitvoeren daarvan. Daarnaast kan worden gekozen voor een audit van het ontwikkelproces.

Testen kunnen plaatsvinden tijdens en/of na realisatie (bij acceptatie). Eventuele afwijkingen/bevindingen uit een securitytoets worden vastgelegd in een afwijkingsrapportage (exception report) en worden meegenomen in de risicoacceptatie.

Soorten toetsen

Beveiligingseisen kunnen op verschillende manieren worden getoetst. Ruwweg zijn dit zes manieren:

1. **Met een automatische dynamische test** die naar typische openingen of kwetsbaarheden zoekt in een werkend systeem (vulnerability scan). Deze manier van testen kan een klein deel van de kwetsbaarheden vinden - typisch configuratiefouten. Omdat dit soort scans goed herhaalbaar zijn, zijn ze vaak deel van een overeenkomst om periodiek uit te voeren.
2. **Met een handmatige securitytest (pentest)** waarbij een systeem in operatie op de proef wordt gesteld. Hieronder worden de verschillende vormen nader toegelicht.
3. **Met een automatische statische test** die broncode en configuratie scant naar bekende kwetsbaarheden, te vergelijken met een spellingschecker: bepaalde

soorten fouten kunnen goed worden gevonden, maar deze aanpak kan niet de precieze werking doorgronden.

4. **Met een code/design review**. Daarbij zoeken specialisten naar kwetsbaarheden in code, ontwerp, configuratie en documentatie, door de precieze werking te doorgronden met ondersteuning van analysetools. Vaak gaat dit samen met een architectuur review, waarbij het technische ontwerp van de software en operatie (logische en deployment architectuur) worden nagelopen op gepaste beveiliging.
5. **Red team/blue team** constructies worden soms gebruikt in grotere organisaties om aanvals- en verdedigingsteams te laten concurreren met elkaar.
6. **Procesanalyse/consistentie systeeminrichting met bedrijfsprocessen.** Het ontwerp en de operatie van een systeem moeten in lijn zijn met verwachtingen en taken van externe medewerkers en tooling, want anders is er geen opvolging bij problemen. Bijvoorbeeld: worden logs geanalyseerd in geval van afwijking/is deze verantwoordelijkheid belegd? Zijn er escalatieprocessen in geval van uitzonderingen/fouten/incidenten? Zijn hiervoor taken en verantwoordelijkheden afgesproken?

Verschillen pentest en code review

Over het algemeen vullen penetratietests en code reviews elkaar aan, omdat ze verschillende dingen beoordelen. Vereenvoudigd gezien zijn de belangrijke verschillen tussen een code/design review en een pentest:

- **Een pentest** zorgt voor inzichten in kwetsbaarheden door een werkend systeem te benaderen. Daarom heeft een pentest eerst werkende software nodig. Bevindingen uit een pentestresultaat hebben veel zeggingskracht omdat misbruik is gedemonstreerd. Het is niet altijd duidelijk waarom een kwetsbaarheid bestaat, dat vraagt soms extra onderzoek.
- **Een code review** kan naast kwetsbaarheden ook zwakheden vinden. Dat zijn potentiele kwetsbaarheden die een risico kunnen vormen. Bijvoorbeeld: wachtwoorden worden niet versleuteld opgeslagen, of mislukte inlogpogingen worden niet gelogd. Code review kan worden toegepast in elke fase van de ontwikkeling. In een code review is bij een bevinding duidelijk waar de oorzaak zit.

Voor een overzicht van verschillende vormen van toetsen, zie het ‘Whitepaper securitytesten’ van het NCSC.

Verschillende types pentests

De gebruikelijke vormen van een pentest zijn:

- **Black box testing:** hierbij probeert een specialist de software aan te vallen zonder kennis van de infrastructuur en werking van de software op voorhand, om een echte hacker te simuleren. Hierdoor is het niet altijd mogelijk om die problemen te vinden die zich diep in de software bevinden. Black box testen zijn minder effectief voor grote, complexe systemen;

- **Grey box testing:** hierbij probeert een specialist de software aan te vallen met rudimentaire kennis van de infrastructuur en de interne werking van de software op voorhand. Deze vorm leidt typisch tot meer bevindingen;
- **White box testing of crystal box testing:** hierbij probeert een specialist de software aan te vallen vanuit aanwijzingen uit broncode en operatie. Deze vorm geeft typisch het meest diepgaande resultaat;
- **Fuzzing:** een speciale tactiek waarbij willekeurig interacties worden uitgevoerd met de software op zoek naar een kwetsbaarheid. Fuzzing werkt als een soort stresstest: hoe gedraagt het systeem zich als deze grote hoeveelheden willekeur moet verwerken. Fuzzing richt zich wel minder op de typische aanvals-scenario's.

Accepteer de oplevering en bijbehorende risico's

Na de bouw vindt de formele acceptatie plaats van de software en de risico's. Dit valt onder de verantwoordelijkheid van de opdrachtgever. Risicoacceptatie is relevant wanneer aan beveiligingseisen onvoldoende voldaan is. De afweging voor acceptatie kan bijvoorbeeld als de ernst van de bevindingen laag is of de kosten van een eventueel herstel niet opwegen tegen de opbrengst.

De opdrachtgever maakt, in afstemming met relevante actoren, de volgende afweging:

- **De applicatie voldoet en wordt geaccepteerd**
- **De software voldoet niet en moet worden aangepast**
- **De software voldoet niet, maar wordt tijdelijk gedoogd:** De software voldoet niet aan de beveiligingseisen, maar het oplossen van de afwijkingen is minder belangrijk dan de noodzaak de software in productie te nemen. Het mankement wordt tijdelijk gedoogd en er wordt een plan opgesteld om de afwijking te herstellen en/of een mitigerende maatregel in te voeren. Bij gedogen gelden de volgende afspraken: Er is een uitgewerkt, goedgekeurd plan (inclusief budget) met een business case beschikbaar, waarin wordt beschreven hoe en wanneer een formeel goedgekeurde situatie zal ontstaan.
- **De beveiligingseisen worden niet geaccepteerd:** De beveiligingseisen sluiten bij nader inzien niet aan op de eisen van de business. Dan worden de specifieke beveiligingseisen aangepast. De opdrachtgever bepaalt of de software desondanks toch in gebruik mag worden genomen en de gewijzigde eisen in een volgend release worden meegenomen, of dat de software eerst moet worden aangepast.

Monitor systeemgedrag tijdens de gebruiksfase

Tijdens de gebruiksfase kunnen zich incidenten voordoen of kunnen nieuwe risico's ontstaan door bijvoorbeeld aanpassing van configuratie, een nieuwe kwetsbaarheid in een open source component dat gebruikt wordt, of een hacker ontdekt een nog onbekende kwetsbaarheid in het systeem.

Incidenten kunnen worden waargenomen via de helpdesks en door het loggen en monitoren van de productiesystemen. In een Service Level Agreement (SLA, of diensten-niveau overeenkomst) dienen hiertoe afspraken te zijn gemaakt voor het afhandelen van de incidenten in de incident management processen. Een SLA vertelt bijvoorbeeld wat de

verwachte reactie- en oplostijden zijn van verschillende typen problemen en op welke manier er wordt gerapporteerd.

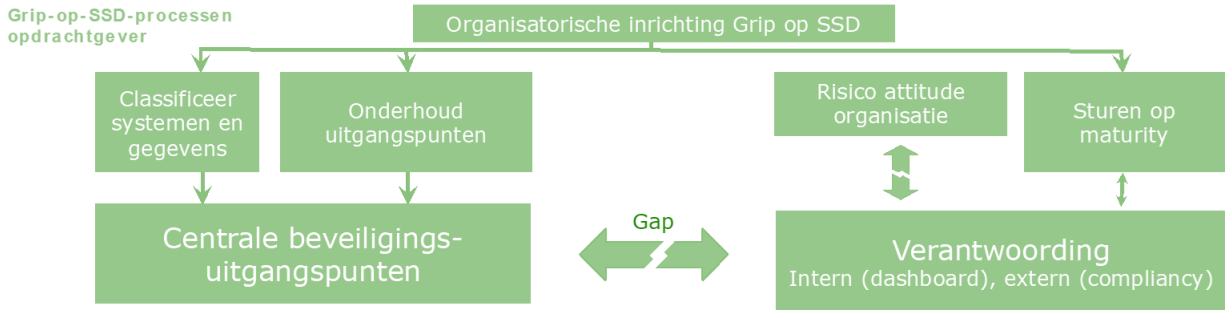
Monitoren, het volgen van systeemgedrag tijdens gebruik, kan met tooling grotendeels automatisch gevolgd worden. Belangrijk is om vast te stellen welke kenmerken relevant zijn, wat beschouwd wordt als een uitzondering, en wat daarmee moet gebeuren (wie doet wat).

Nieuwe risico's kunnen worden gedetecteerd door bijvoorbeeld regelmatig automatische vulnerability scans uit te voeren en met enige regelmaat een penetratietest (pentest) of code review uit te voeren. Verder is het belangrijk om te weten welke (open source) componenten worden gebruikt in een systeem en welke versies, zodat kan worden gemonitord op de noodzaak voor het patchen of vervangen van een component, bijvoorbeeld als nieuwe kwetsbaarheden bekend worden.

7. Pijler 3: SSD-processen

De processen/activiteiten voor SSD worden in principe ingericht bij de opdrachtgever. De opdrachtnemers moeten wel deelnemen aan de contactmomenten en inhaken op de wensen van de opdrachtgever. Bij de opdrachtgever moeten processen worden ingericht of bestaande processen worden uitgebreid voor:

- Het classificeren van systemen en gegevens;
- Het opbouwen en onderhouden van de Centrale beveiligingsuitgangspunten;
- Risicobeheersing en risicoacceptatie;
- Het afleggen van verantwoording;
- Het sturen op maturity/volwassenheid van SSD.



Afbeelding 3: pijler 3 - SSD-processen

Classificeer systemen & gegevens

Om te bepalen hoe moet worden omgegaan met de beveiliging van specifieke systemen en gegevens is het zaak dat de proceseigenaar die systemen en gegevens classificeert naar het benodigde beveiligingsniveau voor Beschikbaarheid, Integriteit en Vertrouwelijkheid (BIV). Een deel hiervan bestaat uit een Business Impact Analyse, waarin IT-middelen (systemen en gegevens) worden geinventariseerd, samen met de algemene dreigingen en bijhorende risico's. Op de website van de Vlaamse overheid vind je een bruikbaar raamwerk voor het classificeren van gegevens, met beslissingsbomen die je helpen om tot een score te komen voor elk van de luiken beschikbaarheid, integriteit en vertrouwelijkheid.

Een voorbeeld van een classificatie: via de nieuwe software kan men toegang verkrijgen tot biometrische gegevens. Dergelijke gegevens vallen al zeker onder klasse 4 'geheim'. De software vraagt in dit geval dus om passende extra beveiligingseisen, bijvoorbeeld: autorisaties moeten zijn beperkt tot specifieke rollen, logs moeten 2 jaar bewaard worden en deze gegevens mogen niet worden gebruikt in testomgevingen.

De definities van de drie BIV kwaliteitsaspecten zijn:

- **Beschikbaarheid:**
  Beschikbaarheid betreft het zorgen voor een ongestoorde voortgang van de informatievoorziening voor geautoriseerde gebruikers.
- **Integriteit:**
  Integriteit betreft het waarborgen van de juistheid, tijdigheid, actualiteit en volledigheid van informatie en de verwerking daarvan. Een onderdeel van Integriteit

is onweerlegbaarheid (non-repudiation). Dit is de mate van zekerheid dat een gebeurtenis heeft plaatsgevonden en/of opzettelijk was.

- **Vertrouwelijkheid:**
  Vertrouwelijkheid betreft het waarborgen dat informatie alleen voor geautoriseerden toegankelijk is.

Specifiek voor de ontwikkeling van software geldt een vierde kwaliteitsaspect, namelijk:

- **Controleerbaarheid:**
  Controleerbaarheid betreft de mate waarin bovenstaande kwaliteitskenmerken (BIV) gecontroleerd kunnen worden. Dit betreft met name logging, documentatie en openheid voor audits (bijvoorbeeld het beschikbaar stellen van broncode voor onderzoek).

Classificatieschema's zijn vaak voorgeschreven of een best practice binnen de organisatiesoort van de opdrachtgever[1]. Een typische constructie daarbij is dat een klasse die 'Midden' is kan volstaan met de baseline aan securitymaatregelen en dat voor hogere klassen een risico-anyse moet plaatsvinden om additionele maatregelen te bepalen.

Stappenplan voor het classificeren

1. Beschrijf de doelstelling van het bedrijfsproces.
2. Bepaal de scope van het bedrijfsproces: welke stappen worden doorlopen en welke informatiestromen/informatieverzamelingen worden daarbij aangesproken.
3. Stel zakelijke randvoorwaarden (beleid en architectuur) en juridische eisen (relevante wet- en regelgeving) vast.
4. Stel de algemene risico's' vast voor de betrokken systemen en gegevens op de aspecten BIV (bijvoorbeeld een hack, uitval van een rekencentrum, lekken van een database, technische storing, of een netwerkaanval die een systeem onbeschikbaar maakt). Start hiervoor met het centrale risico-maatregel overzicht om een eerste selectie te maken. Bedenk daarna nieuwe risico's die specifiek zijn voor het onderzochte bedrijfsproces. Beschrijf de impact/consequentie voor die risico's. Dat kan bijvoorbeeld zijn: reputatieschade, onderbreking van dienstverlening, boetes, schade bij individuën of derden. Bijbehorende overwegingen zijn:
   - Op welke technische wijze kan een bedrijfsproces verstoord of onderbroken worden? Als een deel van de techniek verstoord wordt (bijvoorbeeld een portaal, technische koppeling of database), werkt die verstoring dan door naar andere systemen of processen?
   - Hoe lang kan het bedrijfsproces onderbroken of verstoord zijn voordat dit leidt tot ernstige consequenties voor de organisatie?
   - Welke compenserende maatregelen zijn beschikbaar om een onderbreking of verstoring van een systeem/informatiestroom tijdelijk of permanent op te vangen?

[1] Binnen de Nederlandse overheid wordt de BIO toegepast en in Vlaanderen wordt gebruik gemaakt van het Informatieclassificatieraamwerk.

- Van welke gegevens is de vertrouwelijkheid van belang en in welke mate?

5. Ken de gewenste klassen (beveiligingsniveaus) voor de relevant systemen en gegevens.
6. Specificeer per beveiligingsniveau de relevante beveiligingseisen (bijvoorbeeld 2 factor authenticatie voor toegang tot gegevens met de klasse ‘geheim’).
7. Het is van belang de classificatie periodiek te herevalueren, bijvoorbeeld bij grote veranderingen.

Onderhoud van de Centrale beveiligingsuitgangspunten

De Centrale beveiligingsuitgangspunten (zie hoofdstuk 5) vormen een levende kennisbron voor de richtlijnen binnen dit document:

- **Beveiligingsarchitectuur:** De beveiligingsarchitectuur documenteert de beschikbare technische faciliteiten voor de organisatie. Deze dient up-to-date te worden gehouden en regelmatig getoetst of deze marktconform is en aansluit bij eventuele ketenpartners. Mogelijk kunnen bepaalde faciliteiten worden gedeeld met andere organisaties.
- **Baseline security:** De baseline is een “basis-selectie” en kan worden aangepast door nieuwe inzichten (uit classificatie, risicoanalyse, nieuwe dreigingen, ervaringen met beveiligingseisen die wel of niet effectief blijken te zijn). Tot slot is periodiek onderhoud nodig, waarbij bijvoorbeeld jaarlijks de externe bronnen van de eisen worden nagelopen om wijzigingen te verzamelen en te beoordelen of die moeten worden verwerkt.
- **Classificatie van systemen en gegevens:** Deze is het resultaat van een classificeringsproces en bij veranderingen wordt een nieuwe classificatie uitgevoerd. Classificatie helpt bij het verantwoorden van beveiligingseisen.
- **Risico-maatregel overzicht:** Risico’s en bijbehorende maatregelen volgen uit risicoanalyses en classificering.

Stel risicoanalyse op en onderhoud deze

Een risicoanalyse beantwoordt voor een specifieke context (bijvoorbeeld een systeem) de vraag ‘welke problemen voorzien we en wat kunnen we er tegen doen?’. Het doel is om daarmee specifieke beveiligingseisen op te stellen die horen bij de risico’s (denk aan het eerder vermelde Contactmoment ‘Selectie eisen’. De bedachte combinaties risico-maatregel worden centraal bijgehouden in het risico-maatregel overzicht. Dit overzicht kan goed worden hergebruikt in volgende risicoanalyses.

Wanneer het nodig is, kan een gegevensbeschermingseffectbeoordeling (DPIA) worden uitgevoerd in het kader van de Algemene Verordening Gegevensbescherming (AVG/GDPR). Deze is vergelijkbaar met de risicoanalyse. In een DPIA wordt onderscheid gemaakt in de classificatie van type gegevens (data zoals intern administratief, financieel, persoonlijk identificerend, publieke data).

Een risicoanalyse is een levend document (wordt bijgewerkt bij nieuwe inzichten) en helpt bij besluitvorming in risicomanagement. Verantwoordelijken in een organisatie kunnen met een risicoanalyse expliciet kiezen of een risico geaccepteerd wordt of niet, maar ook hoeveel budget er beschikbaar is voor maatregelen.

Risicoanalyse kent typisch de **volgende stappen**:

1. Vaststellen van de scope voor de risicoanalyse: welke informatiesystemen, gegevensverzamelingen en andere bedrijfsmiddelen, en de context. Dit bepaalt ook hierbij de zakelijke en juridische kaders en andere relevante gegevens uit de classificering;
2. Dreigingsmodellering: identificeren van dreigingsbronnen en dreigingsgebeurtenissen die relevant zijn binnen de gekozen scope;
3. Identificeren van de bekende kwetsbaarheden;
4. Bepalen van de kans van optreden van de dreigingen, onder de conditie van het bestaande of voorziene stelsel van maatregelen en de bekende kwetsbaarheden;
5. Bepalen van de omvang van de schade bij een inbreuk op de kwaliteitsaspecten Beschikbaarheid, Integriteit of Vertrouwelijkheid;
6. Bepalen van het risico. Dit is het combineren van de kans van optreden en de omvang van de te verwachten schade, gezien over alle dreigingen.
7. Bepalen van de bijbehorende maatregelen: de beveiligingseisen voor het risico

Dreigingsmodellering

Om technische dreigingen vast te stellen wordt dreigingsmodellering (threat modelling) meestal toegepast als workshop. Dan wordt op basis van logische systeemarchitectuur geanalyseerd per relevante component welke dreigingen een rol spelen.
STRIDE is een bekende analysemethode die bij dreigingsmodellering wordt toegepast. De naam STRIDE is een afkorting van de namen van zes categorieën aan dreigingen, namelijk:

- **S**poofing: misbruik van de gebruikersidentiteit, namelijk zich als een ander voordoen);
- **T**ampering: schending van de Integriteit;
- **R**epudiation: weerlegbaarheid (een actie kunnen ontkennen);
- **I**nformation disclosure: schending van de privacy of het lekken van opgeslagen gegevens of systeemoperatie details;
- **D**enial of Service (DoS): onbeschikbaarheid uitlokken;
- **E**levation of privilege: misbruik van bevoegdheden.

Kwalificering van de risico's.

Een risico is de kans dat iets gebeurt met een bepaalde impact. Om risico's op waarde te schatten en te vergelijken zijn er kwalitatieve en kwantitatieve methoden. Het gebruik van numerieke waarden bij het inschatten van incidentschade leidt tot schijnzekerheid. Vandaar dat meestal gekozen wordt voor een pragmatische wijze van kwalificering met de categorieën 'hoog', 'middel' en 'laag'.

Zowel voor de kans van optreden als de omvang van de schade wordt nagegaan of deze ‘hoog’, ‘middel’ of ‘laag’ is. Voor het inschatten van een risico wordt de kans van optreden binnen deze methode gecombineerd met de omvang van de mogelijke gevolgen, volgens onderstaande indeling:

Risico (per dreiging)			
Kans van optreden	**Omvang van de schade door een dreiging**		
	Laag	**Midden**	**Hoog**
Laag	Laag	Laag	Midden
Midden	Laag	Midden	Hoog
Hoog	Midden	Hoog	Hoog

Eisen aan de methode voor risicoanalyse voor SSD

De risicoanalyse moet **rekening houden** met:

- Het toepassingsgebied. Dit betreft de scope, namelijk de processen en diensten die moeten worden geleverd;
- De bekende dreigingen, die volgen uit het risico-maatregel overzicht, maar ook de (nog) onbekende dreigingen voor:
    - Beschikbaarheid;
    - Integriteit;
    - Vertrouwelijkheid, inclusief privacy;
    - Controleerbaarheid.
    - Het wel of niet hergebruiken van bestaande reeds genomen maatregelen (hergebruik van bestaande architectuur);
    - De technologie die wordt ingezet en de architectuurkeuzen die worden gemaakt;
    - De technische implementatie;
    - De ontwikkelprocessen, onderhoudsprocessen en werkprocessen.

Het **resultaat** van de risicoanalyse moet aangeven:

- Welke beveiligingseisen relevant zijn voor de software, uit te splitsen naar het inrichten van preventieve, detectieve (signalerende), correctieve en repressieve beveiligingsmaatregelen;
- Welke defecten en fouten leiden tot welke beveiligingsrisico’s;
- Waar in de levenscyclus beveiligingseisen moeten worden getest en of getoetst op defecten en fouten;
- Welke artefacten moeten worden getest en getoetst;
- Welke testhulpmiddelen en testtechnieken moeten worden gebruikt;
- Welke restrisico’s blijven openstaan.

Risicobeheersing en risicoacceptatie

Tijdens het ontwikkelproces zijn er verschillende contactmomenten waarop risico's inzichtelijk worden gemaakt. Deze risico's kunnen door (aanvullende) beveiligingsmaatregelen onder controle worden gebracht of geaccepteerd worden. In dit proces wordt dit per project centraal bijgehouden. Hierdoor heeft de opdrachtgever inzicht in welke risico's "onder controle zijn" of nog "open staan". Meestal zal een openstaande status van een risico het gevolg zijn van een bewuste keuze.

Leg verantwoording af

Het afleggen van verantwoording kan zowel intern (middels een dashboard/centraal overzicht) als extern (middels een compliancy statement, een uiting van verantwoording vergeleken met heersende regels of standaarden). Voorbeeld van zo'n statement kan zijn 'onze bedrijfsprocessen zijn onafhankelijk beoordeeld en gecertificeerd naar de ISO 27001 norm'.

SSD-dashboard

Een SSD-dashboard kan management inzicht geven in de status en effectiviteit van de risicobeheersing. In het dashboard wordt de risicoclassificatie van de software afgezet tegen restrisico's: risico's die worden gelopen doordat bepaalde maatregelen niet zijn geïmplementeerd. De risicoclassificatie is gebaseerd op de BIV-classificatie van de informatiesystemen en de gegevensverzamelingen.

In het SSD-dashboard wordt met een kleurcodering bijgehouden in hoeverre applicaties aan de beveiligingseisen voldoen.

Het gebruik van kleuren in het SSD-dashboard

De rapportages zijn steeds gericht op applicaties[2]. Uitspraken of een applicatie aan de eisen voldoet geldt altijd voor de gehele applicatie. Als een deel van de applicatie (bijvoorbeeld een module) niet aan de eis voldoet, voldoet daarmee de gehele applicatie niet aan deze eis. Deze lijn doortrekkend betekent dit:

- **Groen:** De gehele applicatie voldoet aan deze eis.
- **Lichtgroen:** De eis is voor deze applicatie niet van toepassing (nvt). Hierover is een uitleg (comply) en overeenstemming verkregen.
- **Oranje:** Met de kleur oranje kan worden aangeduid dat met de applicatie-eigenaar afspraken zijn gemaakt over de eindigheid van de afwijking (de eerder vermelde gedoogaanpak).
- **Rood:** In de applicatie is op één of meerdere plekken een afwijking geconstateerd. Welke afwijking dat is, wordt in een commentaarveld opgeslagen.
- **Wit:** Er is nog geen constatering, bijvoorbeeld bij een recent toegevoegde eis of bij een applicatie die nog niet of slechts gedeeltelijk tegen de SSD-normen is aangehouden.

[2] In compliancy-termen: 'Het object of control is de applicatie.' Door in het dashboard ook de applicatie-eigenaar en de softwareleverancier bij te houden, kan ook bepaald worden in hoeverre de applicatie-eigenaar en de softwareleverancier in control zijn.

Gap-analyse

Met het dashboard kan een verschil worden geconstateerd tussen vereiste en geïmplementeerde maatregelen. Dit suggereert een restrisico. Belanghebbenden moeten bepalen of dit restrisico acceptabel is voor de organisatie.

Het herhaaldelijk uitvoeren van de gap analyses maakt het mogelijk om de voortgang of achteruitgang vast te stellen met betrekking tot het informatiebeveiligingsbeleid. Ook kan worden vastgesteld of de Centrale beveiligingsuitgangspunten passen bij de beleving van risico's door de belanghebbenden in de organisatie.

Stuur met je lokaal bestuur op maturity/volwassenheid van SSD

Om de vooruitgang op vlak van SSD in kaart te brengen, lijsten we binnen deze methode enkele volwassenheidsniveaus op, waarmee een organisatie de methode stapsgewijs kan invoeren en realistische verwachtingen/groeidoelstellingen kan formuleren. De volwassenheidsdoelstellingen zijn gebaseerd op een volwassenheids-standaard model (Capability Maturity Model, CMM) en specifiek gedefinieerd voor de SSD-processen.

Volwassenheidsmodellen zijn een manier om de mate van controle en voorspelbaarheid van processen te classificeren. Als standaard wordt daarbij typisch verwezen naar CMM of haar variant CMMI. Deze staan voor Capability Maturity Model (Integration). Deze volwassenheidsniveaus zijn formeel toetsbaar met een standaard aanpak zoals SCAMPI (Standard CMMI Appraisal Method for Process Improvement).

Het doel van CMM is om overzicht te krijgen en mogelijke risico's/verbeterkansen te identificeren. Het is niet gezegd dat iedere organisatie per se op alle punten maximaal volwassen moet zijn. Maar volwassenheid is een vereenvoudigde, abstracte inschatting van hoe voorspelbaar, en met welke garanties, een organisatie haar producten en diensten kan leveren. Als we CMM toepassen op SSD geeft dit een indruk van de verschillende niveaus.

1. **De SSD-CMM niveaus voor het opstellen van beveiligingseisen**

B.01 Selectie eisen en Eisen afstemming	
De organisatie geeft beveiligingseisen mee aan de opdrachtnemers voor haar IT-diensten en stemt daarover af	
Niveau	Criterium (wie en wat)
1. Informeel uitgevoerd	De aan de opdrachtnemer meegegeven eisen bestaan uit een **kopie van een vaste lijst beveiligingseisen**.
2. Beheerst proces	De aan de opdrachtnemer meegegeven eisen bestaan uit **een selectie uit een vaste lijst beveiligingseisen**. • Deze eisen zijn in meerdere projecten toegepast, contractueel afgestemd met leverancier. • Eisen zijn in zekere mate consistent met de enterprise architectuur. • Eisen zijn nog niet gebaseerd op bedrijfsrisicoanalyse;

	• Controle op de juistheid van de eisen vindt herhaaldelijk plaats, inclusief actualisering.
3. Vastgesteld proces	In aanvulling op niveau 2 zijn de eisen **in lijn gebracht met de security-architectuur blokken en de enterprise architectuur**, waardoor er geen onnodige eisen worden meegegeven. • De eisen zijn tot stand gekomen na een risicoanalyse, waardoor de eisen aansluiten op de bedrijfsbehoefte (algemeen beleid). • Controle op de juistheid van de eisen vindt plaats door toetsen van consistentie met de enterprise architectuur blokken.
4. Voorspelbaar proces	In aanvulling op niveau 3 bestaan er **prestatie-indicatoren** van normen en metrieken die medewerkers op uitvoerend niveau in staat stellen om corrigerend op te treden c.q. tijdig maatregelen te nemen tegen beveiligingsrisico's in software. Deze normen en metrieken houden de uitvoering van dag tot dag in lijn met het beleid.
5. Geoptimali-seerd process	In aanvulling op niveau 4 bestaan er prestatie-indicatoren van normen en metrieken die het **hoger management** in staat stellen om de opdrachtnemer snel bij te sturen en consequenties voor (en van) beveiligingseisen vast te stellen. Dit is mogelijk door het hanteren van internationaal geaccepteerde prestatie-indicatoren. De specifieke en algemene eisen worden daarmee van dag tot dag aantoonbaar in lijn met de vooropgestelde doelen.

2. De SSD-CMM niveaus voor Security toetsen

B.02 Security toetsen	
De opdrachtgever voert een kwaliteitscontrole op de software (of laat dit doen)	
Niveau	Criterium (wie en wat)
1. Informeel uitgevoerd	De keuzes om wel/niet/hoe te testen worden door de applicatie-eigenaar op **ad-hoc** basis genomen (naar eigen inzicht, los van een standaard werkwijze). • Er wordt geen gebruik gemaakt van bedrijfsbrede testprocessen en testvoorzieningen. • De testsets worden per release bepaald, waarbij er geen verband is met vooraf gestelde beveiligingseisen.
2. Beheerst proces	Security toetsen gebeurt **tegen bedrijfsbreed vastgestelde beveiligingseisen**. • Afspraken hierover maken structureel onderdeel uit van het toetsplan. • Er wordt bij voorkeur gebruik gemaakt van bedrijfsbrede testprocessen en testvoorzieningen. • De resultaten van de tests zijn onderliggend vergelijkbaar. • De testsets zijn nog niet afgestemd op de eisen vanuit de bedrijfsbrede beveiligingsarchitectuur.

	• De resultaten van de tests worden doorgegeven om centraal bijgehouden te worden in het dashboard. • Er wordt nog niet bijgehouden of voldaan wordt aan bedrijfsbrede beveiligingsarchitectuur eisen. • Voor bedrijfskritische applicaties wordt risico-afhankelijk en onafhankelijke code review uitgevoerd.
3. Vastgesteld proces	Als niveau 2. Ook wordt bijgehouden **of voldaan wordt aan bedrijfsbrede beveiligingsarchitectuur** eisen. Voor bedrijfskritische applicaties wordt standaard onafhankelijke code review uitgevoerd.
4. Voorspelbaar proces	Als niveau 3 en bij het Security toetsen worden **kortcyclische processen** gebruikt, waardoor eerder voorspelbaar is of software aan de eisen voldoet.
5. Geoptimali-seerd proces	Als niveau 4 en de opdrachtnemers hanteren **dezelfde tooling en prestatie-indicatoren**, waardoor prestaties van de opdrachtnemers onderling vergelijkbaar zijn.

3. De SSD-CMM niveaus voor pentesten

B.03 Pentesten	
De organisatie toetst of de software ook tijdens het systeemgebruik geen beveiligingsrisico's bevat.	
Niveau	Criterium (wie en wat)
1. Informeel uitgevoerd	• Pentesten maken geen onderdeel uit van de beveiligingsaanpak volgens de SSD-methode. • Slechts na één of meerdere beveiligingsincidenten worden pentesten uitgevoerd.
2. Beheerst proces	Voor **bedrijfskritische applicaties** worden pentesten uitgevoerd. • Dit proces is echter niet periodiek. • Voor bedrijfskritische applicaties worden de bevindingen bijgehouden in het dashboard en zo teruggekoppeld naar de applicatie eigenaren.
3. Vastgesteld proces	Net als niveau 2. Echter, voor de **bedrijfskritische applicaties worden periodiek pentesten** uitgevoerd.
4. Voorspelbaar proces	**Direct na oplevering van een applicatie(-release) worden pentesten uitgevoerd**, zodat de bevindingen meegenomen kunnen worden in het acceptatieproces. • Deze pentesten maken onderdeel uit van het toetsplan. • Resultaten kunnen leiden tot bevriezen/terugdraaien van een release.
5. Geoptimali-seerd proces	**Na een melding van een toename in beveiligingsdreigingen wordt gericht een pentest uitgevoerd.** De resultaten worden gebruikt als criterium om de release terug te draaien of te besluiten een (applicatie)service tijdelijk uit te schakelen.

4. De SSD-CMM niveaus voor risicoacceptatie

B.04 Risicoacceptatie	
De applicatie-eigenaar van opdrachtgever heeft inzicht in de beveiligingsrisico's en accepteert tijdelijk eventuele restrisico's en neemt hierop actie.	
Niveau	Criterium (wie en wat)
1. Informeel uitgevoerd	De risicoacceptatie is beperkt tot het **informeren van de applicatie-eigenaar van de gevonden afwijkingen op de gestelde eisen.** De applicatie-eigenaar accepteert de restrisico's zonder dat daar vervolgafspraken over worden gemaakt.
2. Beheerst proces	De applicatie-eigenaar wordt geïnformeerd over de gevonden afwijkingen op de gestelde eisen. De **applicatie-eigenaar accepteert de risico's en deze worden bijgehouden in het dashboard**. • Afwijkingen op eisen uit de bedrijfsbrede beveiligingsarchitectuur zijn nog niet meegenomen en worden ook niet in het dashboard bijgehouden. • De applicatie-eigenaar accepteert de eventuele restrisico's op de bevindingen op de standaard baseline beveiligingseisen. • Doordat deze in het dashboard worden bijgehouden en afgestemd met de applicatie-eigenaren worden daar vervolgafspraken over gemaakt.
3. Vastgesteld proces	Zoals niveau 2. Daarnaast: afwijkingen op de eisen die vanuit de bedrijfsbrede beveiligingsarchitectuur worden gesteld zijn daarin (vanaf niveau 3) wel meegenomen. • De bedrijfsbrede beveiligingsarchitectuur wordt meegenomen als referentie voor het bijhouden van afwijkingen. • Doordat afwijkingen in het dashboard worden bijgehouden en afgestemd met de applicatie-eigenaren worden daar vervolgafspraken over gemaakt.
4. Voorspelbaar proces	Afstemming met de applicatie-eigenaar over de restrisico's en wegwerken van de restrisico's vindt niet alleen op basis van de bevindingen via het dashboard plaats. **De applicatie-eigenaar voorkomt dat negatieve bevindingen in het dashboard terecht komen, door deze kortcyclisch weg te werken**, dus door het versneld uitbrengen van een release, waarin de bevinding is weggewerkt.
5. Geoptimali-seerd proces	Het dashboard maakt gebruik van internationaal geldende prestatie-indicatoren. **De afstemming en het wegwerken van de restrisico's met de applicatie-eigenaar leidt nog beperkt tot bevindingen**. Bevindingen worden kortcyclisch weggewerkt. Het dashboard wordt gebruikt om aan te tonen dat de organisatie op het gebied van informatiebeveiliging optimaal functioneert.

8. Bijlage: Organisatorische eisen opdrachtnemer

Organisatorische eisen aan opdrachtnemers spelen een rol bij selectie van externe leveranciers en bij afspraken die worden gemaakt bij bestaande opdrachtnemers. Het onderdeel informatiebeveiliging is vooral van belang als de opdrachtnemer ook zorgdraagt voor de hosting van de software, maar daarnaast ook voor de beveiliging van broncode en documentatie. Hieronder staan enkele voorbeelden van onderwerpen waar eisen over gesteld kunnen worden.

Informatiebeveiliging en beleid

- De Leverancier kan een volwassen informatiebeveiligingsbeleid aantonen i.e. certificering.
- De leverancier heeft een volwassen incidentafhandelingsplan
- De leverancier is bereid om deel te nemen aan audits i.e. code review, pentests.
- De leverancier houdt zich aan de GDPR-wetgeving en implementeert mechanismen hiervoor om dit te garanderen.
- De leverancier beschermt gevoelige informatie met de nodige mechanismen i.e. encryptie.
- De leverancier implementeert mechanismen om data te herstellen in geval van verlies.
- Operaties en toegang tot gevoelige informatie worden gemonitord en gelogd.
- Werknemers bij leverancier zijn getoetst op hun integriteit.

Toegangsbeheer

- De leverancier biedt transparantie over wie binnen het bedrijf toegang heeft tot de afgeleverde gegevens.
- De leverancier zorgt voor toegangsbeheermechanismen die enkel toegang tot het systeem/informatie verleend aan gebruikers die gemachtigd zijn.
- De leverancier integreert autorisatie en authenticatie mechanismen, gebruikmakende van de huidige industriestandaarden.
- De leverancier logt en monitort gefaalde login pogingen en gefaalde data toegangs- en aanpassingsoperaties.

Software ontwikkeling en veiligheid

- De leverancier gebruikt de laatste versie van softwarebibliotheken en softwarepakketten.
- De leverancier toetst de software tijdens ontwikkeling op kwetsbaarheden.
- De leverancier test de software gebruikmakende van penetratietests.
- Secure coding is deel van het ontwikkelingsproces gebruikmakende van bekende industriestandaarden.

Kennisdeling en opleiding

- Medewerkers krijgen jaarlijks een security awareness training.

- De leverancier heeft een team die zich bezighoudt met security en secure coding practices.
- De leverancier biedt secure coding guidelines aan ter ondersteuning van zijn developers.

9. Bijlage: SSD Normen

In onderstaande tabel staat een overzicht van de globale criteria voor de Grip op SSD normen versie 3.0 - de laatste versie ten tijde van het schrijven van dit document. Zie https://www.cip-overheid.nl/. Naast deze lijst bestaat ook een versie van deze normen specifiek voor mobiele applicaties.

De globale criteria in deze bijlage geven een beeld van de normen maar zijn door hun beknoptheid niet specifiek. Daarvoor worden ze verder in detail uitgewerkt in het desbetreffende document.

SSD-2 Veilige gegevensopslag Te beschermen gegevens worden veilig opgeslagen in databases of bestanden, waarbij zeer gevoelige gegevens worden versleuteld. Opslag vindt alleen plaats als noodzakelijk. erd.
SSD-4 Veilige communicatie De applicatie past versleuteling toe op de communicatie van gegevens die passend is bij het classificatieniveau van de gegevens, zowel over interne als externe netwerken en controleert hierop. Van te beschermen gegevens worden alleen de noodzakelijke gecommunice
SSD-5 Authenticatie van gebruikers en systemen Applicaties stellen de identiteit van gebruikers en systemen vast op basis van een mechanisme voor identificatie en authenticatie, waarbij de authenticatiegegevens in een centrale authenticatievoorziening worden beheerd.
SSD-8 Autoriseer toegang De applicatie dwingt de door de opdrachtgever voorgeschreven beperkende set van rechten en privileges af met alleen de voor de gebruiker en systemen noodzakelijke toegang.
SSD-7 Gebruikersrechtenbeheer De rechten die gebruikers hebben binnen een applicatie (inclusief beheerders) zijn zo ingericht dat autorisaties kunnen worden toegewezen aan organisatorische functies en scheiding van niet verenigbare autorisaties mogelijk is.
SSD-12 Sessie-beëindiging De applicatie beëindigt een sessie na een vooringestelde periode van inactiviteit van de gebruiker via automatische sessie-beëindiging,.
SSD-14 Borgen van Sessie Authenticiteit De applicatie hanteert voor sessie-identifiers onvoorspelbare tekenreeksen en bij het uitloggen van de gebruiker wordt de sessie actief beëindigd.
SSD-30 Applicatie logging In de applicatieomgeving zijn signaleringsfuncties (registratie en detectie) actief en efficiënt, effectief en beveiligd ingericht.
SSD-13 Onweerlegbaarheid De applicatie ondersteunt de onweerlegbaarheid voor daartoe aangewezen transacties via cryptografische technieken.

SSD-9 Registreren van (on)succesvolle login-pogingen De applicatie registreert gelukte en mislukte login-pogingen
SSD-19 Invoer-normalisatie De applicatie voorkomt manipulatie door alle ontvangen invoer te normaliseren alvorens die te valideren. De richtlijnen voor invoerbehandeling zijn van toepassing voor alle invoer die van buiten de applicatie komt. Dus niet alleen (eind)gebruikers, maar ook externe systemen en applicaties.
SSD-20 Uitvoer-schoning (output sanitization) De applicatie beperkt de uitvoer tot waarden die (veilig) verwerkt kunnen worden door deze te normaliseren naar de juiste context.
SSD-21 Beperkte commando/query-toegang De applicatie legt beperkingen aan queries en commando's op daar waar met achterliggende systemen wordt gecommuniceerd en deze communicatie wordt alleen ingericht indien strikt noodzakelijk.
SSD-22 Invoer-validatie De applicatie controleert invoer (bijvoorbeeld een HTTP-request) door deze te valideren alvorens die te gebruiken.
SSD-23 Beperkte file Includes De applicatie voorkomt de mogelijkheid van dynamische file includes.
SSD-24 Beperking van te versturen HTTP-headers De webserver stuurt bij een antwoord aan een gebruiker alleen die informatie in de HTTP-headers mee die van belang is voor het functioneren van HTTP.
SSD-27 Discrete foutmeldingen De applicatie neemt in een foutmelding geen inhoudelijke foutinformatie op die misbruikt kan worden.
SSD-28 Discreet commentaar De aan de gebruiker aangeboden scripts / code bevat geen commentaar dat tot misbruik kan leiden.
SSD-32: Bescherming tegen XML externe entiteit injectie De applicatie beperkt de mogelijkheid tot manipulatie door alle externe XML invoer te beschermen tegen entiteit injectie.
SSD-3 Veilige externe componenten Applicaties maken gebruik van veilige en actief onderhouden externe componenten.
SSD-15 Scheiding van presentatie, applicatie en gegevens De architectuur van een applicatie is gebaseerd op een gelaagde structuur door de presentatie-laag, de applicatielaag en de gegevens te scheiden, zodat de lagen beschermd kunnen worden binnen de netwerkzones.
SSD-17 Gescheiden beheerinterface Beheeractiviteiten vinden plaats via een van de standaard gebruikersinterface gescheiden beheerinterface.
SSD-1 Hardening De applicatie voldoet aan het hardeningbeleid. De software en het platform zijn daartoe geconfigureerd volgens de bijbehorende hardeningrichtlijn. Het configureren is procesmatig en procedureel ingericht.

SSD-26 Beperkte HTTP-methoden De webserver faciliteert alleen de HTTP-functionaliteiten die nodig zijn voor het functioneren van de applicatie.
SSD-29 Voorkom directory listing De aan de gebruiker getoonde informatie bevat geen directory listings, zodat die niet kunnen worden misbruikt.
SSD-31 Standaard stack De (web-)applicatie(-omgeving) maakt gebruik van systeemcomponenten en voorzieningen die onderdeel zijn van een formeel gespecificeerde stack.
SSD-33: Veilige HTTP response headers De applicatie maakt gebruik van veilige response headers.

10.Bijlage C: Referentiedocumentatie

Procesraamwerken veilige software ontwikkeling

- NIST Secure Software Development Framework: https://csrc.nist.gov/projects/ssdf
- OWASP SAMM, Software Assurance Maturity Model: https://owaspsamm.org/
- BSIMM – Building Security In Maturity Model: https://www.bsimm.com/
- Software Security: Building Security In, Gary McGraw, ISBN: 0-321-35670-5, Januari 2006

Technische eisen

- CIS controls: https://www.cisecurity.org/controls/
- OWASP ASVS: https://github.com/OWASP/ASVS
- OWASP Mobile ASVS: https://github.com/OWASP/owasp-masvs
- OpenCRE: https://www.opencre.org/
- Mitre CWE: https://cwe.mitre.org/

Divers

- Business Impact Analyse Vlaamse Overheid: https://overheid.vlaanderen.be/toolbox/toolbox-bedrijfscontinu%C3%AFteitsmanagement-bcm/stappenplan/stap/2
- Grip op Secure Software Development: https://www.cip-overheid.nl/
- Informatieclassificatieraamwerk Vlaamse Overheid: https://www.vlaanderen.be/uw-overheid/werking-en-structuur/hoe-werkt-de-vlaamse-overheid/informatie-en-communicatie/informatieveiligheid/informatieveiligheid-en-veiligheidsdiensten/informatieclassificatieraamwerk
- NCSC Whitepaper securitytesten: https://www.ncsc.nl/documenten/publicaties/2020/maart/30/whitepaper-securitytesten

Rond dreigingenanalyse en risicoanalyse

- Publicatie van de CIO Interest Group Informatiebeveiliging, CIO Platform Nederland, september 2012; https://docplayer.nl/2163428-Risicoanalyse-een-verkenning-publicatie-van-de-cio-interest-group-informatiebeveiliging.html
- Procedure Risicoanalyse, Standaard methodiek Groep ICT, RABO-groep, 4 januari 2011
- https://nl.wikipedia.org/wiki/ISO_25010
- Risicoanalyse, Een verkenning, https://www.informit.com/articles/article.aspx?p=446451
- Mitre CAPEC (dreigingen): https://capec.mitre.org/

Een uitgave van de Vereniging van Vlaamse Steden en Gemeenten
VVSG vzw • Bischoffsheimlaan 1-8 • 1000 Brussel
september 2021