



Grip op Secure Software Development (SSD)

Beveiligingseisen voor (web)applicaties

Versie: 2.0

Opdrachtgever	A. Reuijl	CIP
Auteurs	M. Koers W. Tewarie	CIP UWV
Classificatie	Publiek	
Status	CIP categorie 'Becommentarieerde Practice'	
Datum	5 oktober 2014	
Filenaam	Grip op SSD Beveiligingseisen v20.docx	



© Centrum voor Informatiebeveiliging en Privacybescherming.
Voort dit werk geldt een [Creative Commons Naamsvermelding GelijkDelen 4.0 Internationaal-licentie](http://creativecommons.org/licenses/by-sa/4.0/)
verleend door het CIP. Zie <http://creativecommons.org/licenses/by-sa/4.0/>

Voorwoord en leeswijzer

Het is voor organisaties een uitdaging om als opdrachtgever te komen tot veilige IT-diensten, waarbij de gebruikte software voldoet aan de beveiligingsvereisten. Om te komen tot veilige software moet al tijdens het ontwikkelen sturing gegeven worden aan de IT-projecten en het doorvoeren van veranderingen op bestaande systemen. Uitbesteding van ontwikkeling, onderhoud en beheer aan meerdere externe leveranciers maakt dit sturingsvraagstuk extra complex. Over en weer zijn er onuitgesproken verwachtingen rondom informatiebeveiliging en privacybescherming. De methode "Grip op SSD" geeft aan hoe de opdrachtgever sturing kan geven en verwachtingen kan expliciteren en de uitvoering kan bewaken.

Een belangrijk onderdeel van de besturing is het gebruik van een hanteerbaar aantal normen. Voor u liggen de normen die bij de methode "Grip op SSD" gebruikt kunnen worden om de verwachtingen tussen de betrokken partijen te sturen; ook als er sprake is van uitbesteding van ontwikkeling, onderhoud en beheer aan meerdere externe leveranciers. De normen houden rekening met de onderlinge verwachtingen tussen de betrokken partijen en benoemen daartoe de onderlinge verantwoordelijkheden om te kunnen voldoen aan de normen.

Daar waar 'leverancier' of 'hostingpartij' staat geschreven kan ook de 'interne ontwikkelafdeling' of de 'interne IT Afdeling' worden gelezen, want bij de interactie daarmee bestaan dezelfde uitdagingen. Voor de actoren zijn generieke namen gebruikt, namelijk 'opdrachtgever', 'softwareontwikkelaar' en 'hostingpartij', terwijl die in een specifieke situatie andere namen kennen. De taken voor deze actoren zijn, daar waar van belang, steeds beschreven in de uitwerking van de normen. Bij deze beschrijving is steeds uitgegaan van het drievoudig model voor beheer van Looijen (1997), dat uitgaat van de keten: functioneel beheer, applicatiebeheer en technisch beheer.

Het document beschrijft de scope van de normen, in het kort hoe om te gaan met de normen, waarbij verwezen wordt naar de methode "Grip op SSD" en de normen zelf. Bij de nummering van de normen is de oorspronkelijke nummering aangehouden, zodat de nummering consistent is in het dashboard, zoals dat gebruikt wordt in de methode "Grip op SSD".

Dit document is tot stand gekomen door nauwe samenwerking tussen verschillende partijen binnen de overheid en het bedrijfsleven. Deze samenwerking is tot stand gekomen in de SSD-practitionersgroep. De auteurs willen met name hun dank uitspreken voor de ondersteuning door diegenen die een actieve bijdrage hebben geleverd bij het samenstellen van dit document:

- Cor Rosielle, UWV
- Ad Kint, UWV
- Jan Breeman, BKWI
- Marc Schellekens, Capgemini
- Marek Veselka, CGI
- Martijn van Schie, Unisys
- Rob van der Veer, SIG
- Wicher Spanninga, IBM

Amsterdam, oktober 2014

Inhoud

1. Inleiding	4
1.1. De scope: (web)applicaties	4
1.2. Comply or Explain	4
1.3. De betrokken partijen	4
2. Uitleg van de opzet van de beveiligingseisen.....	5
2.1. Gebruikte template	5
2.2. De beheerketen geeft duidelijkheid over verantwoordelijkheden	5
2.2.1. Het gehanteerde model	5
2.2.2. Een keten van verantwoordelijkheden	6
2.2.3. Een keten van afspraken.....	7
2.2.4. Verantwoordelijkheid voor de compliancy aan de eisen	7
3. Beveiligingseisen voor de (web)applicatie	9
3.1. SSD-1: Hardening van technische componenten	9
3.2. SSD-2: Gebruik van veilige cookies of apps.....	12
3.3. SSD-3: Beveiliging van mobile code	15
3.4. SSD-4: Sessie versleuteling	17
3.5. SSD-5: Vaststellen identiteit van een externe gebruiker	19
3.6. SSD-6: Vaststellen identiteit van een interne gebruiker	22
3.7. SSD-7: Functiescheiding	27
3.8. SSD-8: Least Privilege.....	30
3.9. SSD-9: Registreren van Unsuccessful Login Attempts.....	32
3.10. SSD-10: Concurrent Session Control.....	33
3.11. SSD-11: System Use Notification	34
3.12. SSD-12A: Session lock.....	35
3.13. SSD-12B: Session termination	36
3.14. SSD-13: Onweerlegbaarheid	37
3.15. SSD-14: Borgen van Sessie Authenticiteit	39
3.16. SSD-15: Scheiding van Presentatie, Applicatie en Gegevens	40
3.17. SSD-16: Netwerkozoning	41
3.18. SSD-17: Beheerinterface.....	42
3.19. SSD-18: HTTP validatie	44
3.20. SSD-19: Invoer normalisatie.....	46
3.21. SSD-20: Codering van dynamische onderdelen	48
3.22. SSD-21: Geparametriseerde queries	50
3.23. SSD-22: Invoervalidatie	52
3.24. SSD-23: File includes.....	54
3.25. SSD-24: Beperking van te versturen HTTP-headers	56
3.26. SSD-25: Beperken van te tonen HTTP-header informatie.....	58
3.27. SSD-26: HTTP-methoden	60
3.28. SSD-27: Error handling	62
3.29. SSD-28: Commentaar(regels).....	64
3.30. SSD-29: Directory listing.....	66
3.31. SSD-30: Applicatie logging	68
3.32. SSD-31: Standaard stack	72
Bijlage: De SIVA-methode voor het opstellen van beveiligingseisen	74

1. Inleiding

Dit document beschrijft voor organisaties de belangrijkste beveiligingseisen die van toepassing zijn bij de ontwikkeling en aanschaf van (web)applicaties. Samen met de methode "Grip op SSD", waarin de aanpak "hoe grip erop te krijgen" is beschreven, wordt met de eisen de opdrachtgever een complete oplossing geboden om tot veilige software te komen. De eisen beperken zich daartoe tot de applicatielaag van een systeem. Beveiligingseisen die gesteld worden aan bijvoorbeeld de infrastructuur, de werkplek of het personeel zijn niet meegenomen. Hiervoor kunnen bestaande frameworks voor informatiebeveiliging gebruikt worden, zoals ISO 27002.

Om blijvend de belangrijkste bedreigingen te kunnen afdekken is het van belang dat onderhoud op de lijst plaatsvindt. De lijst is en wordt daarom samen door opdrachtgevers en de leveranciers die software ontwikkelen actueel gehouden.

Door het hanteren van juist een beperkte lijst is voorkomen dat er een overkill aan eisen is ontstaan. Zodoende is een goede governance mogelijk geworden. De wijze waarop governance mogelijk wordt is in de methode 'Grip op SSD' aangegeven.

1.1. *De scope: (web)applicaties*

Wanneer dit document spreekt over een (web)applicatie gaat het om een applicatie die bereikbaar is via een webbrowser of via een andere cliënt(applicaties), zoals apps, die ondersteuning biedt voor het Hypertext Transfer Protocol (HTTP), maar ook applicaties, zoals legacy-applicaties, die gebaseerd zijn op de klassieke cliënt/server protocollen. De kern van deze definitie is dat een webapplicatie altijd bereikbaar is op basis van HTTP of de versleutelde vorm hiervan, namelijk HTTP Secure (HTTPS).

1.2. *Comply or Explain*

Ten aanzien van de gestelde beveiligingseisen geldt het principe 'pas toe of leg uit'.

Een maatregel behorende bij een beveiligingseis is niet van toepassing, indien kan worden aangetoond dat:

- op basis van een risicoanalyse de maatregel niet in verhouding staat tot de te maken kosten;
- de overige geïmplementeerde maatregelen het aan de eis ten grondslag liggende risico tot een acceptabel niveau hebben beperkt.

Belangrijk is steeds dat de genomen maatregelen en de risico's die geaccepteerd worden steeds inzichtelijk zijn en aansluiten op de "risk appetite" van de opdrachtgever en dus bepaakt wordt in een governance proces.

De in dit document beschreven beveiligingseisen zijn een handreiking (best practice) en geven aan hoe de maatregel ingevuld zou kunnen worden. Afhankelijk van de situatie kunnen mogelijk alternatieve maatregelen beter op hun plaats zijn. De voorgestelde maatregelen zijn daarom op zichzelf geen harde vereiste. Wel moet steeds de bij de eisen genoemde risico's zijn afgedekt.

1.3. *De betrokken partijen*

Bij de beveiligingseisen zijn de volgende rollen omschreven:

- De opdrachtgever voor een applicatie;
- De interne of externe softwareleverancier, die het ontwerp, de ontwikkeling, het testen en vaak ook het implementeren verzorgt;
- De hostingpartij, die voor de productie en het technisch beheer zorgt;
- De ontvangende partij, namelijk de gebruikersorganisatie die de applicatie in gebruik neemt en voor het functioneel beheer zorgt. Veelal is dit de opdrachtgever, daarom is bij de normen niet het onderscheid ontvangende partij en opdrachtgever aangehouden.

Hoe de betrokken partijen een keten vormen is in paragraaf 2.2 en 2.2.4 uitgelegd.

2. Uitleg van de opzet van de beveiligingseisen

2.1. Gebruikte template

Deze tweede versie van de SSD-normen is gebaseerd op de SIVA-methode [Tewarie, 2014]. Hoe dit is gebruikt is beschreven in de bijlage. Om gestructureerd antwoord te geven op de vraag "wie doet wat en waarom?" is een template gehanteerd.

Het gebruikte template voor de normen is:

SSD-nr Onderwerp van de norm				
<i>Criterium (wie en wat)</i>	Wat (xxxxxx) <werkwoord> xxxxx <u>trefwoorden</u> xxxxx			
<i>Doelstelling (waarom)</i>	De reden waarom de norm gehanteerd wordt.			
<i>Risico</i>	Het risico dat de aanleiding vormt om de norm te hanteren.			
<i>Referentie</i>	Bron 1	Bron 2	...	

Ieder trefwoord vormt een indicator, waaraan voldaan moet worden. Om die reden is ieder trefwoord uitgewerkt. Het gebruikte template voor trefwoorden is:

SSD-nr Onderwerp van de norm	
	<i>indicatoren</i>
/01	<u>trefwoord</u>
/01.01	indicator 1.1
/01.01	indicator 1.2
...	...

De trefwoorden (/01, /02, etc) en de invalshoeken zijn genummerd (/01.01, /01.02, etc), zodat in de toelichting hiernaar gerefereerd kan worden.

Bij de referenties wordt, voor zover relevant, aangegeven waar in de volgende standaarden en richtlijnen additionele informatie is te vinden:

- **NCSC:**
'ICT Beveiligingsrichtlijnen voor webapplicaties', deel 1 en 2, NCSC, januari 2012;
- **NIST:**
Special Publication SP800-53 'Recommended Security Controls for Federal Information Systems', NIST;
- **ISO27002:**
NEN-ISO/IEC 27002 'Code voor informatiebeveiliging', 2005. Voor de relevante referenties kan ook gebruik worden gemaakt van de 'Baseline Informatiebeveiliging Rijksdienst' (BIR).

2.2. De beheerketen geeft duidelijkheid over verantwoordelijkheden

2.2.1. Het gehanteerde model

De beveiligingseisen zijn eisen die aan software worden gesteld. Bij de realisatie van een informatievoorziening zijn echter meer partijen betrokken dan alleen een partij die de software maakt. De verschillende partijen vormen een keten. De interactie tussen de partijen kan in verschillende modellen worden weergegeven. Hiervoor bestaan meervlaksmodellen, zoals het negenvlakmodel (SAME - Strategic Alignment Model Enhanced) en het drievlakmodel van

Looijen. In het drievlaksmodel worden drie vormen van beheer onderscheiden: functioneel beheer, applicatiebeheer en technisch beheer.

De SIVA-methode vraagt om antwoord te geven op de vraag "wie" verantwoordelijk is. Om aan te geven wie voor welk deel van een beveiligingseis voor een applicatie verantwoordelijk is, kan uitgegaan worden van het drievlaksmodel van Looijen.

2.2.2. Een keten van verantwoordelijkheden

Het drievlaksmodel gaat ervan uit dat de applicaties, nadat ze zijn ontwikkeld en opgeleverd, moeten worden beheerd. Om een applicatie in beheer te kunnen nemen moet worden bepaald of de applicatie aan de functionele eisen voldoet, waarna de applicatie in een rekencentrum beschikbaar wordt gesteld aan de gebruikers.

- **Functioneel beheer**

Functioneel beheer is namens de gebruikersorganisatie verantwoordelijk voor het realiseren van de functionaliteit van een applicatie, waarbij deze applicatie optimaal moet aansluiten bij het bedrijfsproces. Functioneel beheer treedt op als opdrachtgever van de afgenomen (ICT-)dienst en dus als opdrachtgever voor applicatie- en technisch beheer.

Het model gaat er dus vanuit dat het functioneel beheer door de opdrachtgever in eigen beheer wordt gehouden.

- **Applicatiebeheer**

Onder applicatiebeheer wordt verstaan het ontwikkelen en onderhouden van de applicatie. Hierbij worden de instandhouding en aanpassing van de applicatieprogrammatuur en de structuur van de gegevensverzamelingen geregeld.

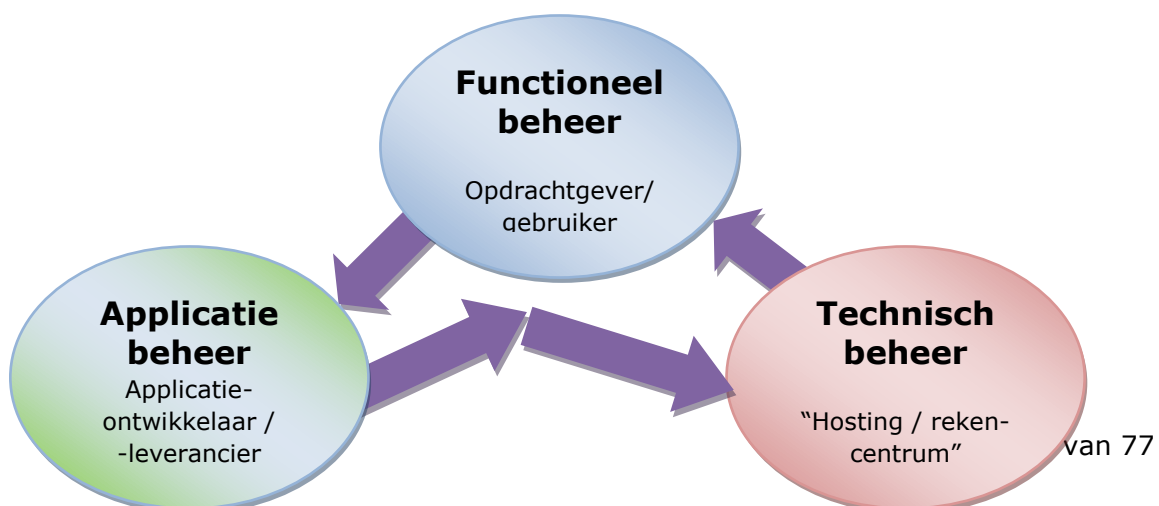
Het applicatiebeheer kan zijn uitbesteed en aangestuurd worden met behulp van contracten of binnen de eigen organisatie liggen. Ook wanneer het applicatiebeheer in eigen hand ligt is nog steeds sprake van een keten en daarmee van een keten van verantwoordelijkheden.

- **Technisch beheer**

Het technisch beheer is verantwoordelijk voor de beschikbaarheid van de (ICT-)dienst en instandhouding van de technische infrastructuur. Op die infrastructuur stelt het technisch beheer ook de applicaties en de bestanden beschikbaar. De technisch beheerder heeft de zorg voor de totale technische infrastructuur (hardware, inclusief de netwerk- en werkplekinfrastructuur, systeemprogrammatuur, ontwikkelhulpmiddelen, enz.).

Ook het technisch beheer kan zijn uitbesteed en aangestuurd worden met behulp van contracten (in ITIL-terminologie: "underpinning contracts") of binnen de eigen organisatie liggen. Ook hier is steeds sprake van een keten en daarmee van een keten van verantwoordelijkheden.

De onderstaande afbeelding toont hoe de drie beheervormen en de daarbij betrokken partijen een keten vormen en hoe verantwoordelijkheden kunnen worden belegd.



2.2.3. Een keten van afspraken

Door het opdelen van de verantwoordelijkheden ontstaat er bij de betrokken partijen een informatiebehoefte. Hierbij moet de "onzekerheidsreductie" van Galbraight worden nagestreefd. Galbraight stelt dat processen in de uitvoering zo min mogelijk moeten worden gestoord door gebrek aan informatie. Door de afspraken tussen de betrokken partijen goed te kiezen en deze aan te laten sluiten op de competenties van de betrokken partijen wordt de overdracht van informatie geminimaliseerd en worden de effectiviteit en efficiëntie gemaximaliseerd. Dit betekent dat, hoewel de opdrachtgever in een regierol zit, moet worden voorkomen dat de opdrachtgever in de rol komt dat hij de technische producten, zoals installatieprocedures en/of configuratierichtlijnen, van de applicatieontwikkelaar moet beoordelen en goedkeuren.

1. Functioneel beheer – Applicatiebeheer

In de uitwerking van de beveiligingseisen in hoofdstuk 3 is een keten van verantwoordelijkheden gehanteerd. In de gehanteerde keten wordt ervan uitgegaan dat de opdrachtgever beveiligingseisen stelt aan de applicatie. Omdat de beveiligingseisen een meer technisch karakter hebben, zijn de beveiligingseisen opgesteld met in achtneming van de competenties van de betrokken partijen. In deze keten is daarbij sprake van een samenwerkingsverband. De practitionersgroep van CIP voor "Grip op SSD" is een samenwerkingsverband dat het mogelijk maakt dat de betrokken partijen hun kennis in de beveiligingseisen inbrengen.

2. Applicatiebeheer – Technisch beheer

De door de applicatieontwikkelaar opgeleverde applicatie wordt ter acceptatie aangeboden aan de opdrachtgever. Om de overdracht van informatie te minimaliseren en de effectiviteit en efficiëntie te maximaliseren wordt de technische documentatie die nodig is voor de installatie (in de beveiligingseisen aangeduid als configuratierichtlijnen) ter goedkeuring aan technisch beheer aangeboden.

3. Technisch beheer – Functioneel beheer

De afspraken tussen hostingpartij en de opdrachtgever liggen vast in de underpinning contracts en zijn ten aanzien van beveiliging uitgewerkt in een beveiligingsplan. In het beveiligingsplan wordt beschreven hoe aan de beveiligingseisen, zoals die in hoofdstuk 3 zijn beschreven, wordt voldaan en hoe controle, door bijvoorbeeld het uitvoeren van pentesten, kan plaatsvinden.

Deze keten van afspraken vormt de basis voor het opstellen van de beveiligingseisen in hoofdstuk 3.

2.2.4. Verantwoordelijkheid voor de compliance aan de eisen

De primaire verantwoordelijkheid voor het implementeren van de beschreven eis ligt bij de interne of externe softwareleverancier. Als onderdeel van de oplevering van een release, die klaar is voor productie, dient deze leverancier te rapporteren over de wijze waarop aan de gestelde eisen is voldaan.

De controle en toezicht op de werking en implementatie van de gestelde beveiligingseisen is belegd bij de leverancier, de hostingpartij en de opdrachtgever. In de praktijk zijn bij de opdrachtgever een test team en een security-team verantwoordelijk voor de controle en het toezicht.

Deze controle gebeurt, afhankelijk van de eis, op een van de volgende manieren:

Controlemomenten	Opdr. gever	Appl. Ontw.	Hosting- partij
Governance binnen "Grip op SSD", inclusief de beveiligingseisen	V		
Controle of het ontwerp aan de beveiligingseisen voldoet	V	A	
Controle van de code op de compliance met de beveiligingseisen en hierover te rapporteren aan de opdrachtgever		V	A
Controle van de werking van de specifieke functionaliteit	V	A	
Controle door te verifiëren of de applicatie kwetsbaar is voor aanvallen en de rapportage hierover aan de opdrachtgever		V	V
Controle door te verifiëren of de configuratie van de applicatie voldoet aan de beveiligingseisen en hierover te rapporteren aan de applicatieontwikkelaar en de opdrachtgever		A	V
Toetsen van de compliancy-rapportage aan de beveiligingseisen	V		

V= verantwoordelijk, A = adviserend

3. Beveiligingseisen voor de (web)applicatie

3.1. SSD-1: Hardening van technische componenten

SSD-1 Hardening					
<i>Criterium (wie en wat)</i>	De software voldoet aan het <u>hardeningbeleid</u> . De software en het platform zijn daartoe geconfigureerd volgens de bijbehorende <u>hardeningrichtlijn</u> . Het configureren is <u>procesmatig en procedureel</u> ingericht.				
<i>Doelstelling (waarom)</i>	De geleverde dienst beveiligen tegen misbruik via zwakheden door het beperken van de functionaliteit tot hetgeen noodzakelijk is voor het correct functioneren van de diensten.				
<i>Risico</i>	Aansprakelijkheid in geval van incidenten, doordat (verwijtbaar) niet voldaan wordt aan de actuele en gangbare beveiligingspraktijk, aangezien vermijdbare kwetsbaarheden niet zijn gedeactiveerd.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B0-6 B0-7				

Toelichting

Het is mogelijk om het aantal potentiële aanvallen te verminderen door het systeem te ont-
doen van onder andere software, gebruikersaccounts en diensten die niet gerelateerd en ver-
eist (strikt noodzakelijk) zijn voor het functioneren van het systeem. Wanneer dat niet moge-
lijk is, moeten alle niet strikt noodzakelijke faciliteiten zijn uitgeschakeld.

Bij het hardenen van het systeem worden de communicatiemogelijkheden van het systeem
tot een minimum (het strikt noodzakelijke) beperkt. Eén van de manieren om dit te bereiken
is door onnodige services onbereikbaar te maken door ze te verwijderen of uit te schakelen.
Door benodigde services in kaart te brengen en vervolgens de afhankelijkheden te bepalen,
kom je tot een minimale lijst van services die op het systeem moeten staan. Alle overige ser-
vices kun je het beste verwijderen of uitschakelen. Bedenk dat niet-actieve maar wel aanwe-
zige services op een systeem uiteindelijk toch tot een kwetsbaar systeem kunnen leiden. Vei-
liger is het daarom om onnodige services volledig van het systeem te verwijderen.

Conformiteitsindicatoren

/01 hardeningbeleid

Het informatiebeveiligingsbeleid bevat een onderdeel expliciet gericht op hardening. De
meeste computer- en netwerkapparatuur en softwarepakketten bevatten meer functionaliteit
dan een organisatie nodig heeft voor het doel waarvoor deze is aangeschaft. Het hardening-
beleid beschrijft hoe de organisatie hiermee wil omgaan. Ter voorkoming dat onveilige (ver-
ouderde) protocollen of functies worden gebruikt beschrijft het hardeningbeleid ook hoe dit
door middel van patching kan worden voorkomen.

SSD-1 Hardening	
	<i>indicatoren</i>
<u>/01</u>	<u>hardeningbeleid</u>
/01.01	Er zijn voorschriften voor hardening en patching van ICT-componenten.

/02 hardeningrichtlijn

Hardeningrichtlijnen geven voorschriften en/of aanwijzingen voor het veilig configureren van
platformen en webserver.

SSD-1 Hardening	
	<i>indicatoren</i>
<u>/02</u>	<u>hardeningrichtlijn</u>
/02.01	De softwareleverancier stelt de software beschikbaar met een actueel overzicht van de noodzakelijke protocollen, services en accounts.
/02.02	Tijdens de hosting zijn alleen de noodzakelijke protocollen, services en accounts actief, andere protocollen, services en accounts zijn gedeactiveerd of verwijderd.
/02.03	Tijdens de hosting zijn de beveiligingsconfiguraties van netwerkservices en protocollen aangepast conform richtlijnen.

Toelichting

/02.01 Zorg dat dit document onderdeel is van het proces wijzigingsbeheer.

Het document:

- heeft een eigenaar;
- is voorzien van een datum en versienummer;
- bevat een documenthistorie (wat is wanneer en door wie aangepast);
- is actueel, juist en volledig;
- is door het juiste (organisatorische) niveau vastgesteld/geaccordeerd.

/02.03 Het 'tunen' van de TCP/IP-stack kan helpen in het beveiligen tegen (D)DoS-aanvallen.

/03 procesmatig en procedureel

Door procesmatig en procedureel te werk te gaan wordt van alle ICT-componenten – nieuwe zowel als bestaande – geborgd dat deze gehardend zijn en blijven.

SSD-1 Hardening	
	<i>indicatoren</i>
<u>/03</u>	<u>procesmatig en procedureel</u>
/03.01	ICT-componenten zijn (aantoonbaar) volgens de instructies en procedures van de leverancier ingericht.
/03.02	ICT-componenten ondersteunen alleen die services die vanuit het ontwerp noodzakelijk zijn, andere services zijn gedeactiveerd of verwijderd.
/03.03	ICT-componenten zijn voorzien van de meest recente, relevante patches.
/03.04	Alleen ICT-componenten die voldoen aan de hardeningrichtlijnen zijn in productie genomen.
/03.05	Wanneer niet voorkomen kan worden dat de inrichting van een ICT-component afwijkt van de hardeningrichtlijn, dan is dit gedocumenteerd, gerapporteerd en geaccordeerd.
/03.06	Periodiek wordt getoetst of de in productie zijnde ICT-componenten niet meer dan de vanuit het ontwerp noodzakelijke services bieden (statusopname). Afwijkingen worden hersteld of gedocumenteerd (zie /01.05).

Toelichting

/03.01 Als alternatief kan de controle door een geautomatiseerd proces plaatsvinden. In dat geval volstaat het noteren van het versienummer van de gebruikte software en eventuele parameters.

/03.01 Neem in de werkinstructies het toepassen van de instructies en procedures van de leverancier op. Houdt tijdens het inrichten van een component een checklist bij en teken deze af na voltooiing van het inrichten van de component.

/03.02 TIP: Leg alle gevonden services vast, met de vermelding of ze actief, uitgeschakeld of verwijderd zijn. Op die manier is het eenvoudiger vast te stellen of nieuwe services zijn geïntroduceerd.

- /03.02 Gebruik het ontwerp om te bepalen welke services nodig zijn. Schakel alle andere services uit, ook wanneer deze 'leuk' of 'handig' zijn. Verwijder zo mogelijk deze services van de ICT-component (de-installatie).
- /03.03 Sluit aan bij een notificatieservice van de leverancier om op de hoogte te blijven van patches en releases van de software. Vergelijk de documentatie van nieuwe patches met de services die op de ICT-component actief zijn. Bepaal of een patch noodzakelijk is en plan het uitvoeren van noodzakelijke patches.
- /03.06 Automatische controles kunnen een onderdeel van deze periodieke toetsing zijn, maar zijn in veel gevallen niet voldoende.
- /03.06 Uitvoering van de toetsing dient geregistreerd en afgetekend te worden.

3.2. SSD-2: Gebruik van veilige cookies of apps

SSD-2 Gebruik van veilige cookies of apps					
<i>Criterium (wie en wat)</i>	In cookies en de applicatie op het front-end zijn de <u>vertrouwelijke gegevens</u> afgeschermd door toepassing van <u>cryptografische technieken</u> .				
<i>Doelstelling (waarom)</i>	Gebruik, inzage, wijziging of verlies van vertrouwelijke gegevens in een niet-veilige omgeving door onbevoegden wordt voorkomen.				
<i>Risico</i>	Indien de vertrouwelijke gegevens (inclusief de authenticatiegegevens) niet worden afgeschermd is de vertrouwelijkheid van gegevens niet gewaarborgd en is de toegang tot de aangeboden dienst niet afgeschermd tot de geautoriseerde gebruikers.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-16 B5-4				

Toelichting

Gevoelige (vertrouwelijke) gegevens worden beschermd door gebruik te maken van cryptografische technieken. Welke gegevens gevoelig of vertrouwelijk zijn, moet door de organisatie op basis van een risicoanalyse worden vastgesteld.

Een cookie of een cliëntdeel van een applicatie (zoals een app) bevat doorgaans informatie over de sessie die een gebruiker heeft met een bepaalde (web)applicatie en mogelijk ook authenticatiegegevens. Door tijdens het bezoek aan een (web)applicatie continu deze gegevens aan de (web)applicatie te tonen, weet de (web)applicatie dat een gebruiker al geauthenticeerd is en dat dit niet opnieuw hoeft te gebeuren. Misbruik van deze informatie leidt ertoe dat kwaadwillenden zich ongeautoriseerd toegang verschaffen tot de aangeboden dienst. Een kwaadwillende kan dit doen door de informatie op te vangen (door het netwerkverkeer te sniffen) of door achtergebleven informatie van een werkstation te halen (bijvoorbeeld in een internetcafé). Veel van deze problemen kunnen worden voorkomen door de gegevens te versleutelen of door extra controles uit te voeren op de container (zoals een cookie) waarin de informatie zich bevindt. Een container met de informatie (cookie) moet versleuteld worden met een sleutel die alleen bekend is bij de webapplicatie. De gebruiker (en ook de kwaadwillende) kan hierdoor vrijwel niets met de container (de cookie) (hij beschikt immers niet over de geheime sleutel) behalve dan de container/cookie aan te bieden aan de (web)applicatie. Een restrisico is dat een kwaadwillende zich op basis van de container/cookie toegang verschafft tot de applicatie (authentication replay). Om dit restrisico te verkleinen kan de (web)applicatie werken met dynamische sleutels. Door bijvoorbeeld elke 2 minuten een nieuwe sleutel te genereren, blijft de container/cookie op het werkstation ook maar 2 minuten geldig. Zolang de gebruiker ingelogd blijft op de (web)applicatie, ontvangt deze elke 2 minuten een nieuwe. Op het moment dat een kwaadwillende een dergelijke container/cookie in handen krijgt, is de geldigheid van de sleutel naar alle waarschijnlijkheid al verlopen en kan deze niets meer met deze container/cookie beginnen.

De flag 'HttpOnly' op een cookie zorgt ervoor dat de cookie uitsluitend via een HTTP en HTTPS verbindingen kan worden gebruikt en niet via bijvoorbeeld een javascript.

De flag 'secure' limiteert de communicatie van cookies tot een beveiligde verbinding via HTTPS en voorkomt dat de inhoud van een cookie voor onbevoegden zichtbaar wordt. Door daarbij de inhoud van een cookie te versleutelen, waarbij de sleutel alleen bekend mag zijn bij de webapplicatie, is de inhoud van de cookie afgeschermd en daarmee de vertrouwelijkheid geborgd.

Conformiteitsindicatoren

/01 vertrouwelijke gegevens

Van gegevens is de gevoeligheid / vertrouwelijkheid bekend, zodat een passende afscherming kan worden geboden.

SSD-2 Gebruik van veilige cookies of apps	
	<i>indicatoren</i>
<u>/01</u>	<u>vertrouwelijke gegevens</u>
/01.01	De opdrachtgever specificeert de classificatie van de vertrouwelijkheid van de gegevens in de container/cookie
/01.02	Indien van een gegeven niet de classificatie van de vertrouwelijkheid is vastgesteld, wordt het gegeven (per default) versleuteld.

Toelichting

/01.01 Er is een BIA uitgevoerd. De classificatie geeft duidelijkheid over welke gegevens afgeschermd moeten worden; in dit geval door versleuteling.

/01.02 Niet voor ieder attribuut zal in de praktijk bekend zijn wat de classificatie is, zeker als het een technisch attribuut betreft en daarom niet in een BIA of risicoanalyse is meegenomen. Als de vertrouwelijkheid van een attribuut is bepaald dan geldt per default dat versleuteling plaatsvindt.

/02 cryptografische technieken

Gegevens worden beschermd tegen ongeautoriseerde kennisname en/of manipulatie door toepassing van cryptografische technieken.

SSD-2 Gebruik van veilige cookies of apps	
	<i>indicatoren</i>
<u>/02</u>	<u>cryptografische technieken</u>
/02.01	De vertrouwelijke gegevens in cookies/ gegevenscontainers zijn afgeschermd door versleuteling.
/02.02	De softwareleverancier stelt de software beschikbaar met bijbehorende configuratierichtlijnen.
/02.03	Communicatie van vertrouwelijke gegevens (van en naar containers/ cookies) is versleuteld.
/02.04	De hosting draagt zorg voor de configuratie conform de configuratierichtlijnen van de softwareleverancier.

Toelichting

/02.01 Cookies zijn bijzondere gegevensbestanden(-containers). Ze worden onder meer gebruikt voor het onderhouden van sessies. Cookies dienen daarom versleuteld te zijn.

/02.03 Dit zal in de regel SSL/TLS zijn. De volgende practice is van toepassing om SSL/TLS veilig te gebruiken¹:

Algemeen:

- Zorg voor een up-to-date implementatie van SSL/TLS. De library die dit protocol implementeert moet van recente datum zijn. Als er een update beschikbaar is, installeer die dan.
- Schakel ondersteuning voor SSL 1.0 en SSL 2.0 uit. Deze versies hebben bekende kwetsbaarheden die ze onveilig maken.

¹ - NCSC – Factsheet 'FS 2011-09 Kwetsbaarheid in SSL en TLS', versie 1.2, de dato 31 januari 2012- <https://www.ncsc.nl/dienstverlening/expertise-advies/kennisdeling/factsheets/factsheet-over-een-kwetsbaarheid-in-ssl-en-tls.html>

- Als de implementatie daarin voorziet, schakel dan via de instellingen de categorieën 'weak ciphers' en 'medium ciphers' uit.

Specifiek:

- Schakel cipher suites uit met de volgende elementen: ADH (Anonymous Diffie-Hellman), NULL (biedt geen versleuteling), EXPORT (biedt onvoldoende veiligheid), RC4 en DES.
- Schakel compressie op SSL/TLS-niveau uit. SSL/TLS-compressie maakt een website vatbaar voor de CRIME-aanval.
- Schakel, indien mogelijk, HTTP-compressie uit. HTTP-compressie maakt een website vatbaar voor de BREACH-aanval.
- Wees bewust van de BEAST-aanval. Deze is goed te mitigeren aan de zijde van de client (de browser dus), maar niet aan de zijde van de server. Alle grote browserbouwers bieden bescherming tegen de BEAST-aanval in recente versies.

Indien mogelijk, doe ook het volgende:

- Ondersteun TLS 1.1 en TLS 1.2. Deze versies van SSL/TLS bieden betere veiligheidsvoorzieningen.
- Biedt ECDHE- en/of DHE-ciphers aan. Deze bieden forward secrecy. Dat is een mechanisme dat voorkomt dat een aanvaller versleutelde communicatie opslaat om deze in de toekomst (met krachtiger computers) te kraken.

/02.03 Als niet alle informatie via een beveiligde verbinding (zoals HTTPS) wordt uitgewisseld vindt er een redirect plaats van HTTP naar HTTPS op het moment dat een (contact) formulier wordt opgevraagd of vertrouwelijke/gevoelige gegevens worden uitgewisseld.

3.3. SSD-3: Beveiliging van mobile code

SSD-3 Beveiliging van mobile code					
<i>Criterium (wie en wat)</i>	Applicaties maken gebruik van <u>geautoriseerde mobile code</u> .				
<i>Doelstelling (waarom)</i>	Compliance met dit criterium zorgt dat enkel ondertekende code wordt gebruikt, waarbij de bron wordt geverifieerd en het risico op kwaadaardige code wordt beperkt.				
<i>Risico</i>	Mobile code, zoals plug-ins, kan virussen en malware bevatten. Zonder een beveiligingsmechanisme kan een applicatie kwetsbaar zijn voor verscheidene vormen van aanvallen en virussen via mobile code.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
		SC-18	10.4.1 10.4.2		

Toelichting

Mobile code is programmatuur die kan worden overgedragen van de ene naar de andere computer, automatisch wordt uitgevoerd en een specifieke functie verricht zonder of met weinig tussenkomst van de gebruiker. Voorbeelden van technologieën voor mobile code zijn Java, javascript, ActiveX, Postscript, PDF, Shockwave-films, Flash, en VBScript.

Conformiteitsindicatoren

/01 geautoriseerde mobile code

Bij gebruik van mobile code wordt zekerheid verkregen over de veiligheid van de code.

SSD-3 Beveiliging van mobile code		
	<i>indicatoren</i>	
<u>/01</u>	<u>geautoriseerde mobile code</u>	
/01.01	De applicatie maakt geen gebruik van mobile code, waarvan de oorsprong en de veiligheid niet is vastgesteld.	
/01.02	Voor het gebruik van de applicatie door de gebruiker wordt geen (downloaden van) mobile code vereist, waarvan de oorsprong en de veiligheid niet is vastgesteld.	
/01.03	Indien gebruik van een mobile code vereist is, waarvan de oorsprong of de veiligheid niet met zekerheid kan worden vastgesteld, wordt deze mobile code alleen in een besloten virtuele omgeving, mede gescheiden van vertrouwelijke informatie, uitgevoerd.	

Toelichting

- /01.01 De bron is bekend en er zijn van de gebruikte versie geen zwakheden (tenzij deze aantoonbaar geen bedreiging vormen) bekend binnen het vakgebied.
- /01.01 Vertrouw geen closed source componenten van derde partijen, tenzij precies bekend is wat de componenten doen. Van open source componenten moet een analyse zijn uitgevoerd, waaruit blijkt dat er de software niet als onveilig kan worden gezien.

- /01.02 Voor de software die bij de bouw van een applicatie is gedownload geldt ook de eis in /01.01 en moet de bron bekend zijn en moeten van de gebruikte versie geen zwakheden (tenzij deze aantoonbaar geen bedreiging vormen) bekend zijn binnen het vakgebied.
- /01.03 De softwareleverancier geeft in het ontwerp aan hoe dit is gerealiseerd en bij de hosting moet worden geconfigureerd.

3.4. SSD-4: Sessie versleuteling

SSD-4 Sessie versleuteling					
<i>Criterium (wie en wat)</i>	De applicatie past <u>encryptie</u> toe op de communicatie van <u>vertrouwelijke gegevens</u> over niet veilige netwerken.				
<i>Doelstelling (waarom)</i>	Het waarborgen dat de vertrouwelijkheid, de integriteit en eventueel de onweerlegbaarheid van gegevensleveringen of transacties wordt geborgd.				
<i>Risico</i>	Het ongeautoriseerd toegang tot diensten en onthullen of muteren van gegevens.				
Referentie	NCSC	NIST	ISO27002		
	B5-2	SC-8 SC-9 SC-13	10.6.1 10.8.1 10.9.1		

Toelichting

Encryptie van een sessie tussen het serverdeel van de applicatie en het werkplekdeel van de applicatie beschermt vertrouwelijke gegevens die worden getransporteerd. Encryptie is nodig over netwerken die als niet veilig moeten worden bestempeld. Niet veilige netwerken zijn netwerken die niet afgeschermd zijn tegen ongeautoriseerde toegang. Dit zijn bijvoorbeeld publieke netwerken en die bedrijfsnetwerken op kantoren die niet aangetoond fysiek (en niet elektronisch) zijn afgeschermd. Of een netwerk afgeschermd is, moet formeel, bijvoorbeeld in een audit, zijn bepaald. Met andere woorden, als het niet duidelijk is of een netwerk veilig is moet het als onveilig worden beschouwd.

Conformiteitsindicatoren

/01 encryptie

Met encryptie wordt afscherming van vertrouwelijke informatie verkregen.

SSD-4 Sessie versleuteling		
	<i>indicatoren</i>	
<u>/01</u>	<u>encryptie</u>	
/01.01	De softwareleverancier gebruikt alleen protocollen en cryptografische technieken die als veilig worden bestempeld.	
/01.02	De applicatie versleutelt de communicatie tussen het serverdeel en het werkplekgedeelte van de applicatie.	
/01.03	De applicatie versleutelt de communicatie tussen het servers en de applicatie.	
/01.04	Bij de aanroep van de diensten is in de aanroep de vertrouwelijke informatie versleuteld.	
/01.05	De softwareleverancier stelt de software beschikbaar met bijbehorende configuratierichtlijnen.	
/01.06	De hosting draagt zorg voor de configuratie conform de configuratierichtlijnen van de softwareleverancier.	

Toelichting

- /01.01 Van de gebruikte versies zijn geen zwakheden (tenzij deze aantoonbaar geen bedreiging vormen) bekend binnen het vakgebied.
- /01.01 Op X.509-certificaten (of vergelijkbaar) gebaseerde encryptie is in de meeste gevallen genoeg.
- /01.04 Dit geldt bijvoorbeeld voor privacygevoelige informatie in URL's.

/02 vertrouwelijke gegevens

Van gegevens is de gevoeligheid / vertrouwelijkheid bekend, zodat een passende afscherming kan worden geboden.

SSD-4 Sessie versleuteling	
	<i>indicatoren</i>
<u>/02</u>	<u>vertrouwelijke gegevens</u>
/02.01	De opdrachtgever specificeert de classificatie van de gegevens die worden uitgewisseld.
/02.02	Per default geldt hierbij de classificatie, waarvoor versleuteling plaatsvindt.

Toelichting

- /02.01 Er is een BIA uitgevoerd. De classificatie geeft duidelijkheid over welke gegevens afgeschermd moeten worden; in dit geval door versleuteling. Als er geen BIA is uitgevoerd of geen risicoanalyse waar het een attribuut betreft voor het laten werken van de techniek, dan geldt per default dat versleuteling plaatsvindt.
- /02.02 Niet voor ieder attribuut zal in de praktijk bekend zijn wat de classificatie is, zeker als het een technisch attribuut betreft.

3.5. SSD-5: Vaststellen identiteit van een externe gebruiker

SSD-5 Vaststellen identiteit van een externe gebruiker					
<i>Criterium (wie en wat)</i>	(Web)applicaties stellen de identiteit van <u>externe gebruikers</u> vast op basis van een <u>mechanisme voor identificatie en authenticatie</u> , waarbij de authenticatiegegevens in een <u>geconsolideerde authenticatievoorziening</u> worden beheerd.				
<i>Doelstelling (waarom)</i>	Het voorkomen van ongeautoriseerde externe toegang tot applicaties, zodat alleen geautoriseerde handelingen worden uitgevoerd en deze handelingen aantoonbaar herleidbaar zijn naar een natuurlijk persoon.				
<i>Risico</i>	Het risico is misbruik van identiteit. De gevolgen kunnen zijn ongeautoriseerde onthulling van informatie, ongeautoriseerde modificatie of onterechte invoer van transacties uit naam van valide gebruikers.				
Referentie	NCSC	NIST	ISO27002		
	B0-12 B3-2	IA-1 IA-2	15.1.1 11.2.3 11.4.2 11.5.2		

Toelichting

Identificatie en authenticatie heeft tot doel alle handelingen te kunnen herleiden tot natuurlijke personen. Voor de herleidbaarheid tot natuurlijke personen zijn twee of drie stappen nodig:

- 1. Identificatie:**
Identificatie is het kenbaar maken van de identiteit van een persoon. (Dit gebeurt in beginsel via persoonlijk contact).
- 2. Authenticatie van de persoon:**
Nadat de identiteit is bepaald moet er altijd zekerheid verkregen worden dat de persoon die een handeling wil verrichten degene is die hij of zij wordt verondersteld te zijn. Dit gebeurt met een meerfactor authenticatie (zie ook de toelichting bij /02.01) en geschiedt door een authenticatieservice.
- 3. Authenticatie van een service of applicatie:**
Voor de handeling die in opdracht van een natuurlijk persoon door een service of applicatie wordt uitgevoerd is altijd authenticatie van de service of applicatie nodig. Dit geschiedt door de directoryservice binnen het domein waar de applicatie wordt aangeboden.

Conformiteitsindicatoren

/01 externe gebruikers

SSD-5 Vaststellen identiteit van een externe gebruiker	
	<i>indicatoren</i>
<u>/01</u>	<u>externe gebruikers</u>
/01.01	Externe gebruikers zijn geregistreerde gebruikers, die niet aansluiten via het interne vertrouwde netwerk of de kantoorautomatisering.
/01.02	De externe gebruiker heeft alleen toegang tot vooraf gedefinieerde en vastgelegde functionaliteit op basis van vastgelegde rollen.

Toelichting

/01.01 Een netwerk wordt pas als vertrouwd netwerk gezien als dit op CxO-level formeel is vastgesteld.

- /01.01 Externe gebruikers zijn, in tegenstelling tot interne gebruikers, geregistreerde gebruikers, die standaard niet aansluiten via het interne netwerk of de kantoorautomatisering, maar via openbare netwerken.
- /01.02 Voor interne- of externe medewerkers die zich niet via het interne vertrouwde netwerk authenticeren zijn de autorisaties vastgelegd in rollen die, afhankelijk van de risico's van de externe toegang, een inperking kunnen zijn van de rollen.
- /01.02 Voor burgers beperken de autorisaties zich tot de aan de burger aangeboden functionaliteit. Deze autorisaties zijn op een onloochenbare wijze vastgelegd. Burgers hebben daarbij alleen toegang tot de persoonsgegevens, tenzij de toegang door middel van mandatering is uitgebreid.
- /01.02 De autorisaties die aan een gebruiker worden toegekend zijn op een onloochenbare wijze vastgelegd.

/02 mechanisme voor identificatie en authenticatie

SSD-5 Vaststellen identiteit van een externe gebruiker	
	<i>indicatoren</i>
<u>/02</u>	<u>mechanisme voor identificatie en authenticatie</u>
/02.01	De applicatie maakt uitsluitend gebruik van het door het opdrachtgever gespecificeerde mechanisme voor identificatie en authenticatie.
/02.02	De configuratie van de identificatie en authenticatievoorziening waarborgt dat de geauthenticeerde persoon inderdaad de geïdentificeerde persoon is.

Toelichting

- /02.01 Wanneer welke wijze van authenticatie en welk authenticatiemechanisme moet worden toegepast om een (web)applicatie te beschermen moet gebaseerd zijn op het toegangsvoorzieningsbeleid. Binnen het toegangsvoorzieningsbeleid is de wijze van authenticatie gebaseerd op een risicoanalyse, waarbij het toegangspad (bijvoorbeeld toegang via het interne netwerk of juist via het Internet) en de classificatie van de vertrouwelijkheid leidend is.
De wijze van authenticatie wordt bepaald door de verschillende 'factoren' waaruit een authenticator kan bestaan. Een factor beschrijft op welke manier een gebruiker zich moet authenticeren. Dit kan op basis van:
 - Iets dat de gebruiker weet (bijvoorbeeld een wachtwoord of pincode).
 - Iets dat de gebruiker heeft (bijvoorbeeld een token of smartcard).
 - Iets dat de gebruiker is (bijvoorbeeld een vingerafdruk).
 Bij applicaties met vertrouwelijke informatie die toegankelijk zijn via het Internet wordt de toegang op basis van de eerste twee factoren (iets dat de gebruiker weet of heeft) het meest toegepast. Gebruik van biometrische kenmerken voor authenticatie tot webapplicaties is nog geen gemeengoed.
- /02.02 Belangrijk is dat de authenticatiemechanismen eenvoudig toegepast kunnen worden in verschillende technologieën en protocollen. Hierbij is een rol weggelegd voor de Identity & Access Management (IAM)-voorziening. Deze voorziening moet ondersteuning bieden aan zowel HTML- als XML-applicaties, zonder dat hiervoor grote inspanningen nodig zijn. Denk hierbij bijvoorbeeld aan de implementatie van authenticatie op basis van digitale (X.509-)certificaten: wanneer je dit hele proces hebt ingericht voor een webapplicatie op basis van HTML, moet het hele authenticatiemechanisme eenvoudig te gebruiken zijn door een webservice op basis van XML.

/03 geconsolideerde authenticatievoorziening

De toegang van gebruikers tot applicaties moet worden geregistreerd voor controledoeleinden. Er moet kunnen worden vastgelegd, met name in fraudegevoelige omgevingen, welke gebruiker op basis van zijn of haar autorisatie welke acties heeft uitgevoerd.

SSD-5 Vaststellen identiteit van een externe gebruiker	
	<i>indicatoren</i>
<u>/03</u>	<u>geconsolideerde authenticatievoorziening</u>
/03.01	Er wordt voor de authenticatie van burgers gebruik gemaakt van een federatieve voorziening binnen Nederland.
/03.02	Er wordt voor de authenticatie van interne- of externe medewerkers gebruik gemaakt van een bedrijfsbrede authenticatievoorziening.
/03.03	Het beheren en onderhouden van identiteiten en autorisatie vindt op onloochenbare wijze plaats.
/03.04	De applicatie maakt gebruik van uniforme en flexibele ² authenticatiemechanismen.

Toelichting

- /03.01 Voor burgers wordt daarbij gebruik gemaakt van de binnen Nederland aangewezen authenticatievoorziening (DigiD).
- /03.02 Consolideer gebruikersgegevens (authenticatie- en autorisatiegegevens) zoveel mogelijk in dezelfde dataverzameling. Creëer geen aparte databases met gebruikers voor verschillende applicaties, maar consolideer deze zoveel mogelijk in één database.
- /03.02 Voor burgers wordt daarbij gebruik gemaakt van de binnen Nederland aangewezen authenticatievoorziening (DigiD).
- /03.03 Maak gebruik van strikte authenticatie(mechanismen). Het is cruciaal dat de authenticatie tot systemen zeer strikt wordt ingeregeld. Afhankelijk van het type besturingsstelsel kunnen hier verschillende maatregelen voor worden getroffen.
- /03.03 Het ondersteund systeem maakt gebruik van mechanismen om activiteiten vast te leggen (loggen). Zie verder logging en monitoring (SSD-30).
- /03.04 Een applicatie dient gebruik te maken van bestaande, uniforme en flexibele authenticatiemechanismen. De authenticatietechnologieën zoals Federated Identity, Security Assertion Markup Language (SAML), Web Services Trust (WS-Trust), et cetera moeten eenvoudig ingebouwd kunnen worden in verschillende protocollen en technieken zoals HTML³ en XML⁴.

² Het gaat hier om toekomstvastheid zodat mogelijke toekomstige risico's beperkt worden – bijvoorbeeld door technologiewijziging.

³ HTML = HyperText Markup Language

⁴ XML = Extensible Markup Language

3.6. SSD-6: Vaststellen identiteit van een interne gebruiker

SSD-6 Vaststellen identiteit van een interne gebruiker					
<i>Criterium (wie en wat)</i>	(Web)applicaties stellen de identiteit van <u>interne gebruikers</u> vast op basis van een <u>mechanisme voor identificatie en authenticatie</u> , waarbij de authenticatie- en autorisatiegegevens in een <u>geconsolideerde authenticatie- en autorisatievoorziening</u> worden beheerd.				
<i>Doelstelling (waarom)</i>	Het voorkomen van ongeautoriseerde interne toegang tot applicaties, zodat alleen geautoriseerde handelingen worden uitgevoerd en deze handelingen zijn aantoonbaar herleidbaar naar een natuurlijk persoon.				
<i>Risico</i>	Het risico is misbruik van identiteit. De gevolgen kunnen zijn ongeautoriseerde onthulling van informatie, ongeautoriseerde modificatie of onterechte invoer van transacties uit naam van valide gebruikers.				
Referentie	NCSC	NIST	ISO27002		
	B0-12 B3-2	IA-1 IA-2	15.1.1 11.2.3 11.4.2 11.5.2		

Toelichting

Identificatie en authenticatie heeft tot doel alle handelingen te kunnen herleiden tot natuurlijke personen. Voor de herleidbaarheid tot natuurlijke personen zijn twee of drie stappen nodig:

1. **Identificatie:**
Identificatie is het kenbaar maken van de identiteit van een persoon. (Dit gebeurt via persoonlijk contact; in principe voor externe- en interne medewerkers éénmalig bij het in dienst treden. De vastlegging geschiedt in het personeelssysteem.
2. **Authenticatie van de persoon:**
Nadat de identiteit is bepaald moet er altijd zekerheid verkregen worden dat de persoon die een handeling wil verrichten degene is die hij of zij wordt verondersteld te zijn. Dit gebeurt met een meerfactor authenticatie (zie ook de toelichting bij /02.01) en geschiedt voor medewerkers door een authenticatieservice.
3. **Authenticatie van een service of applicatie:**
Voor de handeling die in opdracht van een (natuurlijk persoon) door een service of applicatie wordt uitgevoerd is altijd authenticatie van de service of applicatie nodig. Dit geschiedt door de directoryservice binnen het domein waar de applicatie wordt aangeboden.

De wijze waarop de identiteit van een gebruiker moet worden vastgesteld, moet plaatsvinden op basis van het identiteits- en access management beleid van de opdrachtgever. Het identiteits- en access management beleid geeft regels en voorschriften voor de organisatorische en technische inrichting van de toegang tot ICT-voorzieningen, bijvoorbeeld applicatie (gebruikers) en ICT-componenten (beheerders). Het identiteits- en access management beleid beschrijft onder andere de manier waarop de organisatie omgaat met identiteits- en toegangsbeheer. Het identiteits- en access management beleid moet zijn vastgesteld door het verantwoordelijke hoger management (CxO-niveau) om een effectief en consistent gebruikersbeheer, identificatie, authenticatie en toegangscontrole mechanismen voor heel de organisatie te waarborgen.

Conformiteitsindicatoren

/01 interne gebruikers

SSD-6 Vaststellen identiteit van een interne gebruiker	
	<i>indicatoren</i>
<u>/01</u>	<u>interne gebruikers</u>
/01.01	Interne gebruikers zijn geregistreerde gebruikers, die aansluiten via het interne vertrouwde netwerk.
/01.02	De interne gebruiker heeft alleen toegang tot vooraf gedefinieerde en vastgelegde functionaliteit en gegevens op basis van vastgelegde rollen.

Toelichting

/01.01 Een netwerk wordt pas als vertrouwd netwerk gezien als dit op CxO-level formeel is vastgesteld.

/01.01 Interne gebruikers zijn, in tegenstelling tot externe gebruikers, geregistreerde gebruikers, die standaard niet aansluiten via openbare netwerken, maar via het externe netwerk.

/01.02 Voor externe- of interne medewerkers zijn de autorisaties vastgelegd in rollen.

/01.02 De autorisaties die aan een gebruiker worden toegekend zijn op een onloochenbare wijze vastgelegd.

/02 mechanisme voor identificatie en authenticatie

SSD-6 Vaststellen identiteit van een interne gebruiker	
	<i>indicatoren</i>
<u>/02</u>	<u>mechanisme voor identificatie en authenticatie</u>
/02.01	De applicatie maakt uitsluitend gebruik van het door het opdrachtgever gespecificeerde mechanisme voor identificatie en authenticatie.
/02.02	De configuratie van de identificatie en authenticatievoorziening waarborgt dat de geauthentiseerde persoon inderdaad de geïdentificeerde persoon is.

Toelichting

/02.01 Wanneer welke wijze van authenticatie en welk authenticatiemechanisme moet worden toegepast om een (web)applicatie te beschermen moet gebaseerd zijn op het toegangsvoorzieningsbeleid. Binnen het toegangsvoorzieningsbeleid is de wijze van authenticatie gebaseerd op een risicoanalyse, waarbij de classificatie van de vertrouwelijkheid leidend is.

De wijze van authenticatie wordt bepaald door de verschillende 'factoren' waaruit een authenticator kan bestaan. Een factor beschrijft op welke manier een gebruiker zich moet authentifieren. Dit kan op basis van:

- Iets dat de gebruiker weet (bijvoorbeeld een wachtwoord of pincode).
- Iets dat de gebruiker heeft (bijvoorbeeld een token of smartcard).
- Iets dat de gebruiker is (bijvoorbeeld een vingerafdruk).

Bij applicaties die toegankelijk zijn via het interne vertrouwde netwerk geldt, tenzij een risicoanalyse anders vereist, als minimum de eis van een wachtwoord.

/02.02 De volgende aanbevelingen zijn van toepassing op vrijwel alle besturingssystemen met betrekking tot de authenticatie(mechanismen):

- Zorg dat het systeem niet toegankelijk is op basis van 'anonieme' accounts zoals een gastaccount.
- Maak gebruik van persoonsgebonden gebruikersdefinities.

- Beperk de toegang op afstand tot accounts met gelimiteerde rechten. Zorg dat de toegang op basis van root- of beheerderaccounts niet mogelijk is. Beheerders moeten op afstand inloggen met een gelimiteerd beheeraccount en vervolgens lokaal, daar waar nodig, gebruik maken van verhoogde rechten via mechanismen als sudo (Unix, Linux, MacOS) en RunAs (Windows).
 - Overweeg de invoering van sterke authenticatiemechanismen voor de toegang tot systemen. Deze mechanismen kenmerken zich door het gebruik van twee factoren voor authenticatie.
 - Beperk het aantal groepen waartoe een gebruiker behoort (groepslidmaatschappen). Machtigingen en rechten die aan een groep worden toegekend, gelden ook voor de leden van die groep.
 - Implementeer een strikt wachtwoordbeleid. In een wachtwoordbeleid worden de minimale wachtwoordlengte, lengte van de wachtwoordhistorie, complexiteit van het wachtwoord en account lock-outs vastgelegd.
 - Voorkom dat wachtwoorden in leesbare vorm worden opgeslagen door middel van het gebruik van hashing (in combinatie met salts).
 - Verwijder of deactiveer ongebruikte accounts en standaard aanwezige accounts. Hernoem 'bekende' accounts die niet verwijderd kunnen worden (zoals 'administrator') of maak gebruik van sterke wachtwoorden.
- /02.02 Belangrijk is dat de authenticatiemechanismen eenvoudig toegepast kunnen worden in verschillende technologieën en protocollen. Hierbij is een rol weg gelegd voor de Identity & Access Management (IAM)-voorziening. Deze voorziening moet ondersteuning bieden aan zowel HTML- als XML-applicaties, zonder dat hiervoor grote inspanningen nodig zijn. Denk hierbij bijvoorbeeld aan de implementatie van authenticatie op basis van digitale (X.509-)certificaten: wanneer je dit hele proces hebt ingericht voor een webapplicatie op basis van HTML, moet het hele authenticatiemechanisme eenvoudig te gebruiken zijn door een webservice op basis van XML.

/03 geconsolideerde authenticatie- en autorisatievoorziening

De toegang van gebruikers tot applicaties moet worden geregistreerd voor controledoeleinden. Er moet kunnen worden vastgelegd, met name in fraudegevoelige omgevingen, welke gebruiker op basis van zijn of haar autorisatie welke acties heeft uitgevoerd.

SSD-6 Vaststellen identiteit van een interne gebruiker	
	<i>indicatoren</i>
<u>/03</u>	<u>geconsolideerde authenticatie- en autorisatievoorziening</u>
/03.01	Er wordt voor de authenticatie van externe- of interne medewerkers gebruik gemaakt van een bedrijfsbrede geconsolideerde authenticatie en autorisatievoorziening.
/03.02	Het beheren en onderhouden van authenticatie- en autorisatiegegevens vindt op onloochenbare wijze plaats.
/03.03	De applicatie maakt gebruik van uniforme en flexibele ⁵ authenticatie- en autorisatiemechanismen
/03.04	Er is een actueel overzicht van accounts en de personen die daar gebruik van maken: - service-accounts; - beheeraccounts; - gebruikersaccount; (web)applicatie-accounts.

⁵ Het gaat hier om toekomstvastheid zodat mogelijke toekomstige risico's beperkt worden – bijvoorbeeld door technologiewijziging.

SSD-6 Vaststellen identiteit van een interne gebruiker	
/03.05	In aanvulling op de centrale autorisatievoorziening kunnen binnen de applicatie meer fijnmazige autorisaties worden beheerd.

Toelichting

/03.01 Tenzij anders aangegeven door de opdrachtgever wordt er geen gebruik gemaakt van een authenticatievoorziening binnen de eigen applicatie(voorziening).

/03.01 Het consolideren van gebruikersgegevens betekent voor de applicatie dat gebruik gemaakt wordt van de bestaande generiek beschikbaar gestelde (centrale) mechanismen voor authenticatie en toegangsrechten. Er is geen logica voor identiteit- en toegangsbeheer in de applicatie ingebouwd. De applicatie vertrouwt volledig op centraal belegde functionaliteiten op dit gebied.

/03.01 Consolideer gebruikersgegevens (zowel authenticatie- als autorisatiegegevens) zoveel mogelijk in dezelfde dataverzameling. Creëer geen aparte databases met gebruikers voor verschillende applicaties, maar consolideer deze zoveel mogelijk in één database.

De keuze voor een consolideerde authenticatievoorziening biedt:

- Een integrale informatievoorziening, zodat deze zich voor de gebruikers als één geheel gedraagt, Single Sign On⁶ en Single Sign Out mogelijk is en er door het ontbreken van interne grenzen geen (of minder) kans op "wrong door" is.
- Het voorkomen van het repliceren of synchroniseren van authenticatiemechanismen.
- Een efficiënte, effectieve en transparante voorziening die eenvoudig te auditen is.
- De mogelijkheid tot het gebruik van federatieve authenticatievoorzieningen buiten de eigen organisatie. Een groot deel van deze nieuwe mogelijkheden vindt zijn weg in het federatieve identiteitsbeheer. Hierbij worden samenwerkingsvormen mogelijk gemaakt, waarbij de dienstverlening over organisatiegrenzen heen een belangrijke factor is. Federatief identiteitsbeheer vindt – ten opzichte van 'normaal' identiteitsbeheer - bij uitstek plaats daar waar uitwisseling van identiteitsgegevens over organisatiegrenzen heen noodzakelijk is.

/03.01 Bij de centrale manier van identiteit- en toegangsbeheer (er is geen logica voor identiteit- en toegangsbeheer in de webapplicatie ingebouwd) geldt dat het tool vaak een overkoepelend systeem is over alle platformen en applicaties heen.

Het inzetten van eigen voorziening 'vermindert' de complexiteit van (web)applicaties, omdat het authenticeren en autoriseren van gebruikers los wordt gekoppeld van de applicatie. Door de authenticatie los te koppelen van de applicatie, is het eenvoudiger om in de toekomst een andere authenticatievoorziening in te zetten voor het beveiligen van de applicatie. Hiervoor worden wijzigingen doorgevoerd in de voorziening en hoeft de achterliggende applicatie hier 'in principe' niets van te merken.

/03.02 Het is cruciaal dat de authenticatie tot systemen zeer strikt wordt ingeregeld. Afhankelijk van het type besturingssysteem kunnen hier verschillende maatregelen voor worden getroffen.

⁶ Bij Single Sign On logt de gebruiker slechts eenmalig in om op andere systemen geauthentiseerd te zijn. Dit is mogelijk als de andere systemen de authenticatie vertrouwen van het systeem, waarop de gebruiker zich authentiseert. De op het andere systeem voor de authenticatie benodigde gegevens (zogenaamde credentials) worden daartoe tussen beide systemen uitgewisseld. Hiertoe wordt:

- Tussen Windows-omgevingen Single Sign On geboden door het toepassen van een Windows Trust-relatie
- Tussen omgevingen, waarbij trust-relaties niet mogelijk zijn wordt gebruik gemaakt van Kerberos-certificaten.

- /03.02 Het ondersteund systeem maakt gebruik van mechanismen om activiteiten vast te leggen (loggen). Zie verder logging en monitoring (SSD-30).
- /03.03 Een applicatie dient gebruik te maken van bestaande, uniforme en flexibele authenticatiemechanismen. De authenticatietechnologieën zoals Federated Identity, Security Assertion Markup Language (SAML), Web Services Trust (WS-Trust), et cetera moeten eenvoudig ingebouwd kunnen worden in verschillende protocollen en technieken zoals HTML⁷ en XML⁸.
- /03.05 De applicatie kan ook in aanvulling op de centrale authenticatie- en autorisatievoorziening:
1. De fijnmazige autorisaties die binnen de applicatie plaatsvinden, worden alleen toegekend binnen de meer grofmazige autorisaties in de centrale voorziening.
 2. Een eigen voorziening bieden. Hierbij wordt nog steeds gebruik gemaakt van de centrale mechanismen voor authenticatie en toegangsrechten om toegang tot de applicatie te krijgen. De applicatie biedt daarmee een eigen extra beveiliging.
 3. Het vaststellen van de identiteit vertrouwt de webapplicatie toe aan een centraal mechanisme, maar het uitdelen van (meer gedetailleerde) rechten aan deze identiteit worden uitgegeven door de applicatie. Dit heeft een aantal voor- en nadelen.
- Voordelen:
- I. Rechten kunnen meer dynamisch tijdelijk op basis van binnen een workflow toegekende taak worden toegekend.
 - II. Er kan gebruik gemaakt worden van de toekenning van gedetailleerde rechten diep in de techniek, zoals de toegang tot op veldniveau in een database.
- Nadelen:
- III. Audits op uitgegeven rechten moet plaatsvinden op een niet geconsolideerde administratie.
 - IV. De uitgifte van strijdige rollen binnen de eigen organisatie kunnen niet of moeilijk worden voorkomen.

⁷ HTML = HyperText Markup Language

⁸ XML = Extensible Markup Language

3.7. SSD-7: Functiescheiding

SSD-7 Functiescheiding					
<i>Criterium (wie en wat)</i>	De autorisaties van gebruikers (inclusief beheerders) binnen een (web)applicatie zijn zo ingericht dat <u>autorisaties</u> kunnen worden toegewezen aan <u>organisatorische functies</u> en <u>scheiding van niet te verenigen autorisaties</u> mogelijk is.				
<i>Doelstelling (waarom)</i>	Het effectief en veilig inrichten van de toegang tot functionaliteit en (vertrouwelijke) gegevens.				
<i>Risico</i>	Het ontbreken van een adequate implementatie van functiescheiding kan leiden tot een verhoogde kans op fraude of misbruik van bedrijfsmiddelen bij kritische of fraudegevoelige taken.				
Referentie	NCSC	NIST	ISO27002		
		AC-5	10.1.3 10.6.1 10.10.1		

Toelichting

Functiescheiding voorkomt dat binnen één functie (door één persoon) taken worden uitgevoerd, waarbij de som aan bevoegdheden de medewerker zoveel handelingsvrijheid heeft, dat dit schadelijk kan zijn voor de organisatie. Voorbeeld hiervan is de scheiding van het bepalen van de hoogte van vergoedingen en het uitvoeren van betalingen. Toegekende autorisaties moeten daarom altijd correleren met de taken binnen één organisatorische functie en dus niet strijdig zijn met de functiescheiding op organisatorisch niveau.

Conformiteitsindicatoren

/01 autorisaties

SSD-7 Functiescheiding	
	<i>indicatoren</i>
<u>/01</u>	<u>autorisaties</u>
/01.01	In de applicatie zijn de autorisaties op een beheersbare wijze geordend. De applicatie maakt daartoe gebruik van autorisatiegroepen.
/01.02	Bij het definiëren van autorisaties is extra aandacht voor accounts met hoge privileges, zoals functioneel beheeraccounts en platformaccounts. In het ontwerp en de bouw van de applicaties is hier rekening mee gehouden.
/01.03	Er bestaat een proces voor het definiëren, toekennen en onderhouden van de autorisaties; bij de oplevering de applicatie wordt in de configuratiebeschrijving hiervan gebruik gemaakt of naar verwezen.

Toelichting

/01.01 De autorisatiegroepen zijn zodanig opgezet dat zij eenvoudig en beheersbaar te koppelen zijn met taken en organisatorische functies.
Toegangsbeheer betreft alle activiteiten die systemen moeten uitvoeren om de autorisaties voor webapplicaties in te regelen en af te dwingen.
De manier waarop rechten op het systeem beperkt kunnen worden, is afhankelijk van het besturingssysteem en het beleid met betrekking tot toegangsbeheer. Hieronder volgen enkele aanwijzingen voor het inperken van rechten op twee populaire typen besturingssystemen:
Microsoft Windows:
Microsoft Windows biedt de mogelijkheid om rechten toe te passen op onder andere

bestanden, directories en het register. Via zogenaamde Group Policy Objects (GPO) kunnen beheerders de rechten van gebruikers centraal via de Active Directory (AD) beheren. Een GPO maakt het mogelijk om centraal beperkingen af te dwingen voor verschillende Windows-onderdelen en -systemen.

Linux/UNIX:

Linux/UNIX biedt commando's zoals `chmod` (change mode), `chown` (change owner), `umask`, `setuid` (Set UserID) en `setgid` (Set GroupID) om toegangsbeheer in te richten. Wie de autorisaties verleent (aanvraag goedkeuren), vormt een ander aandachtspunt, bijvoorbeeld de geveenseigenaar of een manager. Het toekennen van toegang moet gebaseerd zijn op classificatie van de vertrouwelijkheid en risicoanalyse. Bij het bepalen van de classificatie van de vertrouwelijkheid moet rekening worden gehouden met de zakelijke behoefte om informatie te verspreiden of verspreiding te beperken en met mogelijke schade bij bijvoorbeeld ongeautoriseerde toegang tot de informatie.

/01.02 Een applicatie maakt vaak direct of indirect gebruik van een aantal verschillende platformaccounts op een systeem. Door aan de applicatie platformaccounts toe te wijzen met beperkte rechten, wordt de schade die een aanval kan toebrengen beperkt.

Bij het inperken van rechten van platformaccounts moet je bij webapplicaties denken aan de volgende typen platformaccounts:

- Accounts voor het opzetten van verbindingen tussen de webapplicatie en (webapplicaties op) de data laag zoals databases en Lightweight Directory Access Protocol (LDAP)-stores. Inperken van rechten beperkt mogelijke schade van injectieaanvallen Structured Query Language (SQL-) injectie, LDAP-injectie).
- Platformaccounts waaronder de webserver, de databaseserver en de applicatieserver draaien. Inperken van rechten beperkt de schade van buffer overflows en injectieaanvallen.
- Platformaccounts voor toegang tot het bestandssysteem. Vaak gebruikt een webapplicatie voor toegang tot het bestandssysteem het account waaronder de webapplicatie draait. Het is daarom verstandig om op bestandsniveau de rechten van dit account te beperken, waardoor dit account niet de mogelijkheid heeft om bijvoorbeeld nieuwe bestanden aan te maken of bestaande bestanden te wijzigen. Dit is niet altijd mogelijk als de applicatie deze rechten nodig heeft om te functioneren.

/01.02 De rechten zijn beperkt tot die welke strikt noodzakelijk zijn voor het uitvoeren van de toegekende taken en verantwoordelijkheden. Dit conform SSD-8.

/01.03 De toegewezen autorisaties moeten periodiek worden gecontroleerd. Dit proces is afgestemd met de business. Dit kan op basis van een overzicht van alle autorisaties voor de webapplicatie. De webapplicatie-verantwoordelijke beoordeelt of er gebruikers en autorisaties aanwezig zijn die niet meer nodig zijn of die gewijzigd moeten worden. Belangrijk is wel dat het overzicht te 'behappen' is voor een webapplicatie-verantwoordelijke.

Om dit te kunnen uitvoeren moet wel aan de volgende randvoorwaarden worden voldaan:

- er is een verantwoordelijke (eigenaar) per webapplicatie vastgesteld;
- er is een actueel overzicht van alle gebruikers en autorisaties voor de webapplicatie;
- de verantwoordelijke voor de (web)applicatie beoordeelt of er gebruikers en autorisaties aanwezig zijn die niet meer nodig zijn of die gewijzigd moeten worden. Hieronder vallen ook ongewenste opeenhopingen van autorisaties.

Er worden verbeteracties gedefinieerd indien geïmplementeerde maatregelen niet voldoen aan de gestelde eisen en/of verwachtingen of tekortkomingen opleveren. Per verbeteractie is een verantwoordelijke vastgesteld en de status van deze verbeteracties wordt periodiek geactualiseerd.

Uitvoering van de toetsing dient geregistreerd en afgetekend te worden.

/02 organisatorische functies

De taken binnen het beheer zijn bekend en verdeeld in verschillende groepen, de organisatorische functies. Deze profielen zijn bedoeld om enerzijds tot een effectief takenpakket te komen, anderzijds tot een adequate functiescheiding. De organisatorische functies komen tot uitdrukking in de autorisaties van gebruikers, inclusief die van beheerders.

SSD-7 Functiescheiding	
	<i>indicatoren</i>
<u>/02</u>	<u>organisatorische functies</u>
/02.01	Op basis van taken, verantwoordelijkheden en bevoegdheden zijn in het ontwerp en de applicatie de verschillende functies, taken en autorisaties geïdentificeerd.
/02.02	Er is een ingevulde autorisatiematrix, waarin aan de taken autorisaties zijn toegekend.

Toelichting

- /02.01 Naast de diverse inhoudelijke taakgebieden is hierin expliciet aandacht voor ongewenste combinaties van bevoegdheden.
- /02.02 De functiescheiding geeft bijvoorbeeld aan dat één enkele beheerder niet in staat is – direct of indirect – volledige controle over alle functies te verwerven.

/03 scheiding van niet te verenigen autorisaties

De taken binnen het functioneel en technisch beheer zijn bekend en verdeeld in verschillende groepen, de functieprofielen. Deze profielen zijn bedoeld om enerzijds tot een effectief takenpakket te komen, anderzijds tot een adequate scheiding van niet te verenigen autorisaties. De functieprofielen komen tot uitdrukking in de autorisaties van gebruikers, inclusief die van beheerders.

SSD-7 Functiescheiding	
	<i>indicatoren</i>
<u>/03</u>	<u>scheiding van niet te verenigen autorisaties</u>
/03.01	Op basis van taken, verantwoordelijkheden en bevoegdheden zijn de niet te verenigen taken en autorisaties geïdentificeerd.
/03.02	Er is een ingevulde autorisatiematrix, waarin de functiescheiding tot uitdrukking komt.
/03.03	Naast het bestaan van een gevulde matrix moet er een uitleg zijn over de ondersteuning van de functiescheiding.
/03.04	Er bestaat een proces voor het definiëren en onderhouden van de autorisaties.

Toelichting

- /03.01 Naast de diverse inhoudelijke taakgebieden is hierin expliciet aandacht voor ongewenste combinaties van bevoegdheden.
- /03.02 De functiescheiding geeft bijvoorbeeld aan dat één enkele beheerder niet in staat is – direct of indirect – volledige controle over alle functies te verwerven.

3.8. SSD-8: Least Privilege

SSD-8 Least privilege					
<i>Criterium (wie en wat)</i>	De (web)applicatie dwingt de door de opdrachtgever voorgeschreven beperkende set van <u>rechten en privileges</u> af met alleen de voor de gebruiker noodzakelijke toegang.				
<i>Doelstelling (waarom)</i>	Een medewerker krijgt alleen die rechten die nodig zijn voor de aan hem of haar opgelegde specifieke taken, zoals beschreven in de functieomschrijving.				
<i>Risico</i>	Als een gebruiker over te hoge rechten beschikt kan daar bedoeld of onbedoeld misbruik van worden gemaakt. Tevens is er een risico dat bij compromittering van een gebruikersaccount of systeemaccount onnodige schade ontstaat indien dit account over te hoge rechten beschikt.				
Referentie	NCSC	NIST	ISO27002		
		AC-5	10.1.3 10.6.1 10.10.1		

Toelichting

Risico's van systeemmisbruik kunnen aanzienlijk worden verminderd door rechten op een systeem te beperken. Principes die in het beleid gehanteerd kunnen worden, zijn bijvoorbeeld gebaseerd op 'standaard geen toegang', 'least privilege', 'need-to-know' of functiescheiding (Separation of Duties). Volgens het principe van 'Least privilege' worden de rechten van gebruikers gelimiteerd tot de minimale set van rechten die nodig zijn om de functie naar behoren uit te voeren. Dit principe is naast de rechten van medewerkers (inclusief beheerders) ook van toepassing op applicaties en processen.

In het ontwerp van de applicatie moet daarom rekening gehouden zijn met het principe van least privilege. Een gebruikersaccount of user-id wordt vaak op veel plekken gedefinieerd en wordt daar gekoppeld aan autorisatiegroepen. De wijze hoe dit gebeurt is beschreven in SSD-8.

Conformiteitsindicatoren

/01 rechten en privileges

SSD-8 Least privilege	
	<i>indicatoren</i>
<u>/01</u>	<u>rechten en privileges</u>
/01.01	De opdrachtgever stelt, op basis van een risicoanalyse, vast welke taken strijdig zijn voor de gebruikers.
/01.02	In het ontwerp is rekening gehouden met niet-verenigbare rechten.
/01.03	De (web)applicatie zorgt via de toekenning van autorisaties dat niet meer dan de toegewezen rechten en privileges beschikbaar komen voor de gebruikers.
/01.04	Webapplicaties draaien op de onderliggende servers met "least privileges". Door de hostingpartij is de inperking van de rechten geïmplementeerd, zoals die door de softwareleverancier zijn meegegeven in de configuratiehandleiding.
/01.05	Webapplicaties en onderliggende servers zijn zodanig geconfigureerd dat ook security-gerelateerde events worden vastgelegd.

Toelichting

- /01.01 Het betreft hier taken die in de functiebeschrijving zijn vastgelegd (of in een TVB-matrix, waarbij taken tot in detail zijn toebedeeld aan functionarissen).
- /01.03 Hierbij wordt gebruik gemaakt van de autorisatiegroepen, zoals in de toelichting van SSD-7/01.01.
- /01.05 Door de hostingpartij is de logging en monitoring van de security-gerelateerde events geïmplementeerd, zoals die door de softwareleverancier zijn meegegeven in de configuratiehandleiding.
- /01.05 In de security-gerelateerde events zijn die handelingen opgenomen, waarbij gebruik wordt gemaakt van rechten die verder gaan dan least-privilege.

3.9. SSD-9: Registreren van Unsuccessful Login Attempts

SSD-9 Registreren van Unsuccessful Login Attempts					
<i>Criterium (wie en wat)</i>	De (web)applicatie registreert <u>mislukte login-pogingen</u> .				
<i>Doelstelling (waarom)</i>	Het tijdig signaleren van een aanval, zodat daarop actie ondernomen kan worden.				
<i>Risico</i>	Indien <u>mislukte login-pogingen</u> niet worden gesignaleerd en bewaakt, is er een risico dat onbevoegden ongezien in de systemen kunnen binnendringen.				
Referentie	NCSC	NIST	ISO27002		
		AC-7	11.5.1		

Toelichting

Door tijdig te acteren op mislukte login-pogingen kan een mogelijke aanval, zoals een brute force attack, worden tegengegaan. Een brute force attack is een aanvalsmethode die bestaat uit het proberen van alle mogelijke opties tot een inbraak, net zolang tot er een optie is gevonden die succesvol is.

Door de mislukte login-pogingen vast te leggen, te monitoren en hierop te reageren kan schade worden voorkomen of worden beperkt.

Conformiteitsindicatoren

/01 mislukte login-pogingen

SSD-9 Registreren van Unsuccessful Login Attempts	
	<i>indicatoren</i>
<u>/01</u>	<u>mislukte login-pogingen</u>
/01.01	Alleen daar waar geen gebruik gemaakt kan worden van centrale authenticatievoorzieningen, zoals beschreven in SSD 5 en SSD 6, vind authenticatie en logging door de (web)applicatie plaats.
/01.02	De (web)applicatie en de (web)server leggen mislukte login-pogingen vast in logbestanden.
/01.03	De logbestanden worden voor real time monitoring beschikbaar opengesteld voor de opdrachtgever of voor de door de opdrachtgever aangewezen derde partij.
/01.04	De opdrachtgever zorgt voor het monitoren en analyseren van de logrecords met mislukte login-pogingen om dreigingen adequaat aan te pakken.

Toelichting

/01.01 Om te voorkomen dat door de (web)applicatie logging moet plaatsvinden wordt bij voorkeur gebruik gemaakt van centrale authenticatievoorzieningen.

/01.03 In de configuratiehandleiding is beschreven hoe de logbestanden gekoppeld zijn met monitoringvoorzieningen.

3.10. SSD-10: Concurrent Session Control

SSD-10 Concurrent Session Control					
<i>Criterium (wie en wat)</i>	Een (web)applicatie is voorzien van Concurrent Session Control, die slechts <u>één sessie per gebruiker</u> toestaat, tenzij onderbouwd is dat meer noodzakelijk is.				
<i>Doelstelling (waarom)</i>	Het gebruik van accounts en sessies van en met gebruikers op een beheersbare wijze laten verlopen, door het aantal sessies te beperken.				
<i>Risico</i>	Het ongecontroleerd toestaan van meerdere gelijktijdige sessies verhoogt de kans op fouten door gebruikers en maakt de logica van de (web)applicaties ingewikkeld, doordat deze logica rekening moet houden met handelingen via meerdere sessies met dezelfde gebruiker.				
Referentie	NCSC	NIST	ISO27002		
		AC-10			

Toelichting

Het technisch afdwingen van de mogelijkheid om het aantal gelijktijdige sessies binnen eenzelfde applicatie te beheersen wordt aangeduid als 'Concurrent Session Control'.

De ontwerper stelt via een functionele analyse vast of er meer dan één gelijktijdige sessie van een gebruiker nodig is en hoe risico's ervan worden voorkomen. Indien er geen aantoonbare noodzaak toe is, wordt volstaan met het beperken van het aantal tot één gelijktijdige sessie per gebruiker.

Conformiteitsindicatoren

/01 één sessie per gebruiker

SSD-10 Concurrent Session Control	
	<i>indicatoren</i>
<u>/01</u>	<u>één sessie per gebruiker</u>
/01.01	De ontwerper stelt via een functionele analyse vast of er meer dan één gelijktijdige sessie van een gebruiker nodig is en hoe risico's ervan worden voorkomen.
/01.02	De ontwerper legt de argumentatie vast indien er meer dan een sessie per gebruiker nodig blijkt te zijn.
/01.03	De (web)applicatie limiteert het aantal gelijktijdige sessies per gebruiker conform de specificatie van de ontwerper.

Toelichting

/01.01 Het aantal toegestane sessies kan verschillen per type account of per account.

3.11. SSD-11: System Use Notification

SSD-11 System Use Notification					
<i>Criterium (wie en wat)</i>	De (web)applicatie toont tijdens het interactieve inlogproces aan de gebruiker een <u>te conformeren melding</u> , waarmee de gebruiker op de rechten en plichten wordt gewezen.				
<i>Doelstelling (waarom)</i>	De melding wijst de gebruiker op zijn wettelijke plichten, zorgt voor een ontmoediging voor ongewenste handelingen en voor een juiste basis om te kunnen loggen en monitoren.				
<i>Risico</i>	Het risico van het ontbreken van een System Use Notification is dat een kwaadwillende achteraf kan beweren dat deze niet op de hoogte was van de regels voor het correcte gebruik van de (web)applicatie, of bezwaar kan maken tegen het vastleggen van de uitgevoerde handelingen.				
Referentie	NCSC	NIST	ISO27002		
		AC-8	11.5.1 15.1.5		

Toelichting

Een System Use Notification is een meldtekst die vereist is voor (web)applicaties met een interactieve login interface. Een gebruiker die wil inloggen krijgt een tweeledige melding:

1. Benadrukt wordt dat de (web)applicatie alleen mag worden gebruikt voor activiteiten die door de opdrachtgever zijn toegestaan.
2. Benadrukt wordt dat de uitgevoerde handelingen worden vastgelegd en door de opdrachtgever kunnen worden geanalyseerd.

Conformiteitsindicatoren

/01 te conformeren melding

SSD-11 System Use Notification	
	<i>indicatoren</i>
<u>/01</u>	<u>te conformeren melding</u>
/01.01	De opdrachtgever stelt de tekst vast die wordt opgenomen in de System Use Notification.
/01.02	De (web)applicatie toont de System Use Notification tijdens de interactieve inlogprocedure aan de gebruiker.
/01.03	De (web)applicatie zorgt dat de gebruiker een handeling moet verrichten om de System Use Notification van het scherm te verwijderen, bijvoorbeeld door een Enter te geven.

Toelichting

/01.01 In de System Use Notification staat dat bijvoorbeeld:

- Een melding waarin gebruikers erop wordt gewezen dat zij toegang krijgen tot een systeem van de Nederlandse overheid en/of de (betrokken overheids-) organisatie;
- Het gebruik van het systeem mag worden gecontroleerd, bijgehouden en gecontroleerd;
- Ongeoorloofd gebruik van het informatiesysteem verboden is en onderworpen aan strafrechtelijke en civielrechtelijke sancties.

/01.03 De System Use Notification blijft op het scherm totdat de gebruikers de inhoud erkent en expliciet actie neemt om in te loggen of op het systeem verdergaat.

3.12. SSD-12A: Session lock

SSD-12A Session lock

Vervalt als eis (staat geheel los van de applicatie)

3.13. SSD-12B: Session termination

SSD-12 Session termination					
<i>Criterium (wie en wat)</i>	De (web)applicatie beëindigt een sessie na een <u>vooringestelde periode</u> van inactiviteit van de gebruiker via <u>automatische session termination</u> .				
<i>Doelstelling (waarom)</i>	Het voorkomen dat een sessie slechts langer dan gedurende een beperkte tijd onbeheerd toegankelijk is voor andere personen, nadat de gebruiker de sessie onbeheerd heeft achtergelaten.				
<i>Risico</i>	Het ontbreken van een session termination kan als gevolg hebben dat de reeds geopende sessie door een kwaadwillende wordt misbruikt.				
Referentie	NCSC	NIST	ISO27002		
		AC-12	11.3.2		

Toelichting

Als een sessie met een (web)applicatie blijft openstaan, kan hiervan misbruik worden gemaakt door andere personen. Session termination zorgt ervoor dat de sessie wordt beëindigd na een door de opdrachtgever voorgeschreven tijdsinterval van inactiviteit.

Conformiteitsindicatoren

/01 vooringestelde periode

SSD-12 Session termination	
	<i>indicatoren</i>
<u>/01</u>	<u>vooringestelde periode</u>
<u>/01.01</u>	Als default wordt 15 minuten aangehouden, tenzij de functionaliteit anders noodzakelijk maakt.

/02 automatische session termination

SSD-12 Session termination	
	<i>indicatoren</i>
<u>/02</u>	<u>automatische session termination</u>
<u>/02.01</u>	De (web)applicatie activeert automatisch session termination na een door de opdrachtgever voorgeschreven tijdsinterval van inactiviteit.
<u>/02.02</u>	Session termination komt overeen met het uitloggen van de gebruiker en de (web)applicatie vernietigt daarbij dienovereenkomstig de sessie.

3.14. SSD-13: Onweerlegbaarheid

SSD-13 Onweerlegbaarheid					
<i>Criterium (wie en wat)</i>	De (web)applicatie ondersteunt de onweerlegbaarheid voor daartoe <u>aange- wezen transacties via cryptografische technieken.</u>				
<i>Doelstelling (waarom)</i>	Het onweerlegbaar vastleggen van transacties, waarvan duidelijk moet zijn dat die door een specifieke gebruiker daadwerkelijk zijn uitgevoerd.				
<i>Risico</i>	Indien een aangewezen transactie niet onweerlegbaar aan een persoon kan worden gekoppeld, kan die later zijn of haar betrokkenheid bij de transactie ontkennen.				
Referentie	NCSC	NIST	ISO27002		
		AU-10	10.8.2 10.9.1 12.3.1		

Toelichting

Voor bepaalde transacties moet onweerlegbaar worden vastgelegd wie deze heeft uitgevoerd, zoals financiële transacties of transacties die bepaalde rechtsgevolgen hebben. Voor de vaststelling van de identiteit van de gebruiker en de onweerlegbaarheid van de transacties wordt een cryptografische techniek gebruikt met een elektronische handtekening. Bij deze norm kan er sprake zijn van het verwerken van persoonsgegevens, waarbij wettelijke verplichtingen in acht dienen te worden genomen.

Conformiteitsindicatoren

/01 aangewezen transacties

SSD-13 Onweerlegbaarheid	
	<i>indicatoren</i>
<u>/01</u>	<u>aangewezen transacties</u>
/01.01	Tijdens de risicoanalyse wordt door de opdrachtgever bepaald voor welke transacties onweerlegbaarheid nodig is. Denk hierbij onder andere aan het verwerken van financiële, persoongerelateerde en andere vertrouwelijke gegevens.
/01.02	Tijdens de ontwerpfasen worden de aangewezen transacties nader gedefinieerd en wordt bepaald op welke wijze de onweerlegbaarheid wordt geïmplementeerd.
/01.03	De (web)applicatie dwingt de onweerlegbaarheid af voor de aangewezen transacties.

Toelichting

- /01.01 De organisaties moet de sterkte van de binding tussen de persoon die de transactie uitvoert en de informatie bepalen en goedkeuren. Hij doet dit op basis van de aard van de informatie en risicofactoren die in een risicoanalyse worden bepaald.
- /01.03 In het ontwerp is vastgelegd hoe de onweerlegbaarheid wordt gerealiseerd.
- /01.03 De applicatie valideert en waarborgt de binding tussen de persoon die de transactie uitvoert en de informatie. De validatie van bindingen worden bereikt, bijvoorbeeld door het gebruik van cryptografische checksums ofwel hashing.
- /01.03 Onweerlegbaarheid kan worden afgedwongen door het gebruik van verschillende technieken of mechanismen, zoals digitale handtekeningen of digitale ontvangstberichten zoals bij S/MIME.

/02 cryptografische technieken

SSD-13 Onweerlegbaarheid	
	<i>indicatoren</i>
<u>/02</u>	<u>cryptografische technieken</u>
/02.01	De opdrachtgever stelt eisen aan cryptografische technieken, zodat die als veilig kunnen worden bestempeld voor het borgen van de onweerlegbaarheid.
/02.02	De opdrachtgever specificeert op welke wijze het beheer van sleutels en certificaten wordt ingericht voor de gebruikte cryptografische techniek.
/02.03	De (web)applicatie gebruikt voor de aangewezen transacties cryptografische techniek die aan de eisen van de opdrachtgever voldoet.
/02.04	De softwareleverancier legt de wijze van sleutelbeheer vast in de configuratiehandleiding.
/02.05	De hostingpartij draagt zorg voor het beheer volgens de vastgelegde processen.

Toelichting

/02.01 Het beleid is vastgelegd en goedgekeurd door de opdrachtgever.

/02.03 Ten behoeve van auditdoeleinden is binnen de applicatie, maar eventueel ook in de vastlegging buiten de applicatie, herkenbaar hoe het bewijsmateriaal is en wordt opgebouwd, met daarin minimaal informatie over:

- de identiteit van de persoon die de transactie heeft uitgevoerd,
- de datum en tijd waarop de transactie plaatsvond,
- het doel waarvoor de transactieoverdracht plaatsvond.

3.15. SSD-14: Borgen van Sessie Authenticiteit

SSD-14 Borgen van Sessie Authenticiteit					
<i>Criterium (wie en wat)</i>	De (web)applicatie hanteert bij de sessienummering op een <u>onvoorspelbare wijze van nummeren</u> en bij het uitloggen van de gebruiker wordt de sessie actief beëindigd.				
<i>Doelstelling (waarom)</i>	Het voorkomen van misbruik van een nog openstaande sessie, die door de oorspronkelijke gebruiker niet meer wordt gebruikt.				
<i>Risico</i>	Als een andere persoon een nog openstaande sessie kan oppakken, geeft dit de mogelijkheid van misbruik van de identiteit van de oorspronkelijke gebruiker.				
Referentie	NCSC	NIST	ISO27002		
	B4-2	SC-23			

Toelichting

Een sessie (via http) tussen de (web)applicatie en de gebruiker krijgt een unieke sessie-ID. Na het uitloggen van de gebruiker dient de sessie actief te worden beëindigd door de webapplicatie om te voorkomen dat een andere persoon de nog openstaande sessie kan oppakken en hiermee verder kan werken.

In dit kader wordt ook aan het sessie-ID eisen gesteld, onder andere dat deze onvoorspelbaar is. Hiertoe wordt een nummer gebruikt met voldoende lengte en wordt een volgend sessie-ID random gekozen.

Conformiteitsindicatoren

/01 onvoorspelbare wijze van nummeren

SSD-14 Borgen van Sessie Authenticiteit	
	<i>indicatoren</i>
<u>/01</u>	<u>onvoorspelbare wijze van nummeren</u>
/01.01	Een sessie-ID is voldoende sterk, namelijk een lang random nummer, om deze onvoorspelbaar te maken.
/01.02	De webapplicatie genereert steeds een nieuw random sessie-ID bij het inloggen en het opnieuw inloggen van een gebruiker.

Toelichting

/01.02 In de praktijk is de standaard functionaliteit voor het genereren van sessie-ID in een applicatieontwikkelomgeving voldoende.

/02 actief beëindigd

SSD-14 Borgen van Sessie Authenticiteit	
	<i>indicatoren</i>
<u>/02</u>	<u>actief beëindigd</u>
/02.01	De webapplicatie vernietigt aan de serverzijde actief de sessie bij het uitloggen van een gebruiker op de applicatie.

3.16. SSD-15: Scheiding van Presentatie, Applicatie en Gegevens

SSD-15 Scheiding van presentatie, applicatie en gegevens					
<i>Criterium (wie en wat)</i>	De architectuur van een (web)applicatie is gebaseerd op een <u>gelaagde structuur</u> door de presentatie-laag, de applicatielaag en de gegevens te scheiden, zodat de lagen beschermd kunnen worden binnen de netwerkzones.				
<i>Doelstelling (waarom)</i>	Het waarborgen van de vertrouwelijkheid en/of de integriteit van de applicatieloga en/of gegevens.				
<i>Risico</i>	Het compromitteren van de applicatieloga en/of gegevens.				
Referentie	NCSC	NIST	ISO27002		

Toelichting

(Web) applicaties die aan een niet vertrouwde omgeving worden aangeboden, maar waarvan de vertrouwelijkheid en/of de integriteit van de applicatieloga en/of gegevens moeten worden gewaarborgd, worden door middel van zonering beschermd. Voor Internet vindt dat op netwerkniveau plaats door middel van een DMZ. Door deze compartimentering of gelaagdheid ook toe te passen in de architectuur van (web)applicaties, wordt het compromitteren voorkomen van een server, applicatie of gegevens, waarvan de vertrouwelijkheid en/of de integriteit van de applicatieloga en/of gegevens moeten worden gewaarborgd. Hiertoe worden de presentatie-laag, de applicatie-laag en de gegevens zoveel mogelijk van elkaar gescheiden.

Conformiteitsindicatoren

/01 gelaagde structuur

SSD-15 Scheiding van presentatie, applicatie en gegevens	
	<i>indicatoren</i>
<u>/01</u>	<u>gelaagde structuur</u>
/01.01	De scheiding van de presentatie, applicatie en gegevens maken de waarborging van de vertrouwelijkheid en/of de integriteit van de applicatieloga en/of gegevens mogelijk.
/01.02	Bedrijfskritische applicatieloga, vertrouwelijke gegevens of gegevens waarvan de onweerlegbaarheid moeten worden gewaarborgd worden buiten de DMZ opgeslagen.
/01.03	In de configuratiebeschrijving staat beschreven in welke zone welke delen van de applicatie komen.
/01.04	De hostingpartij past voor de zones security regimes toe, die gelden voor die specifieke zone.

Toelichting

/01.01 De DMZ wordt als niet vertrouwde omgeving gezien, waardoor deze niet geschikt is voor de opslag van vertrouwelijke gegevens.

3.17. SSD-16: Netwerkzoning

SSD-16 Netwerkzoning

Vervalt als eis (staat geheel los van de applicatie)

3.18. SSD-17: Beheerinterface

SSD-17 Beheerinterface					
<i>Criterium (wie en wat)</i>	Beheeractiviteiten vinden plaats via een van de standaard gebruikersinterfa- ce <u>gescheiden beheerinterface</u> .				
<i>Doelstelling (waarom)</i>	Het voorkomen van onbedoeld gebruik of misbruik van de beheerfunctionaliteit.				
<i>Risico</i>	Door het ontbreken van afdoende afscherming kunnen gewone gebruikers beheerdersautorisaties verwerven of de beheerder als gebruiker onbedoeld onjuiste beheerhandelingen uitvoeren.				
Referentie	NCSC	NIST	ISO27002		
	B1-2	SC-2	11.4.5		

Toelichting

Op applicaties, gegevensverzamelingen en platformen wordt beheer uitgevoerd. Voor de activiteiten van beheerders dient een beheerinterface aanwezig te zijn, dat ervoor zorgt dat de beheeractiviteiten op een gecontroleerde en gestructureerde wijze plaatsvinden. Door het onderscheid tussen beheer- en productiefunctie wordt voorkomen dat productie en beheerhandelingen door elkaar gaan lopen.

Door de beheerinterface alleen via het interne netwerk toegang te geven tot de applicatie wordt het risico voorkomen dat de bijbehorende beheerfunctionaliteit vanaf het Internet kan worden gecompromitteerd.

Conformiteitsindicatoren

/01 gescheiden beheerinterface

SSD-17 Beheerinterface	
	<i>indicatoren</i>
<u>/01</u>	<u>gescheiden beheerinterface</u>
/01.01	Het ontwerp en de applicatie waarborgen dat de beheerinterface via autorisaties en vooraf gedefinieerde functionaliteit volgens de gespecificeerde functiescheiding wordt gerealiseerd.
/01.02	Het ontwerp en de applicatie waarborgen dat alle beheerfunctionaliteit is ondergebracht in een als beheerinterface herkenbare interface.
/01.03	Het ontwerp en de applicatie waarborgen dat de beheerinterface niet beschikbaar is voor reguliere gebruikers.
/01.04	De beheerinterface maakt alleen gebruik van veilige protocollen.
/01.05	De hostingpartij maakt de externe toegang tot de beheerinterface alleen mogelijk via een geautoriseerde en beveiligde verbinding.

Toelichting

/01.01 Er is gebruik gemaakt van de functiescheiding, zoals in SSD-7 is beschreven.

/01.04 Onderstaande aandachtspunten/overwegingen dienen als input voor de scheiding tussen beheer en productie. De gemaakte beslissingen moeten worden onderbouwd, op het juiste (organisatie) niveau worden vastgesteld, zijn gedocumenteerd en worden onderhouden zodat altijd over een actueel ontwerp/inrichting van het netwerk wordt beschikt. Het uitgangspunt moet steeds zijn, wat minimaal noodzakelijk (hoogst nodig) is om de gewenste functionaliteit te kunnen bieden.

Stel vast welke beheermechanismen toegepast worden:

Maak gebruik van bewezen standaardprotocollen die geen beveiligingsrisico's bevatten of waarvan de beveiligingsrisico's bekend en beheersbaar zijn. Het is dan ook

noodzakelijk om vooraf vast te stellen welke beheermechanismen juist wel en welke juist niet toegepast mogen worden. Maak gebruik van versleutelde beheermechanismen en verbied verbindingen die de informatie in clear text (in onversleutelde vorm) over het netwerk versturen. Voorbeelden van veilige verbindingen zijn:

- Secure Shell (SSH) in plaats van Telnet.
- Secure Copy (SCP), SSH File Transfer Protocol (SFTP) of FTP over SSL (FTPS) in plaats van File Transfer Protocol (FTP).
- HTTPS in plaats van HTTP voor webinterfaces.

/01.05 Stel vast hoe beheerders toegang krijgen tot het beheergedeelte:

Er moet vastgesteld worden hoe beheerders toegang krijgen tot het beheergedeelte. Hier zijn verschillende mogelijkheden voor:

- Implementeer beheer-clients in het beheergedeelte, die alleen te gebruiken zijn in een afgeschermd ruimte. Beheer over de omgeving kan alleen plaatsvinden via deze fysiek afgeschermd beheer-clients.
- Implementeer beheer-clients in het beheergedeelte die op basis van een remote interface (bijvoorbeeld Citrix of Microsoft RDP) te benaderen zijn voor een beperkte groep werkstations in het interne netwerk. Beheerders maken vanaf hun workstation in het LAN een verbinding met de beheer-clients en kunnen vervolgens via deze beheer-clients het beheer over de omgeving uitvoeren.
- Implementeer een apart beheer-LAN binnen het interne netwerk en sta verbindingen richting het beheergedeelte alleen toe vanuit dit beheer-LAN.
- Implementeer een VPN tunnel op het moment dat het beheer remote via het internet wordt uitgevoerd. Een VPN tunnel kan natuurlijk ook toegepast worden als het beheer vanaf het bedrijfsnetwerk wordt uitgevoerd.

3.19. SSD-18: HTTP validatie

SSD-18 (HTTP) validatie					
<i>Criterium (wie en wat)</i>	De (web)applicatie controleert de inhoud van een (HTTP-)request door deze te <u>valideren</u> alvorens die te gebruiken.				
<i>Doelstelling (waarom)</i>	Compliance met dit criterium zorgt dat alleen gecontroleerde invoer wordt verwerkt.				
<i>Risico</i>	Inzage, wijziging, verlies, of misbruik van gegevens door bijvoorbeeld manipulatie van de webapplicatielogica.				
Referentie	NCSC	NIST	ISO27002		
		SC-2	11.4.5		

Toelichting

De (web)applicatie ontvangt invoer van de gebruiker en van andere applicaties. Hierbij is een belangrijke vuistregel dat de webapplicatie geen enkele invoer mag vertrouwen. De webapplicatie moet daarvoor 'defensief' worden geprogrammeerd, waarbij alle via HTTP-requests ontvangen inhoud eerst wordt gevalideerd, alvorens die wordt gebruikt.

De verdeling van controles op de specifieke kenmerken is afhankelijk van de webserver en de webapplicatie(s) waarmee de deze samenwerkt.

Conformiteitsindicatoren

/01 valideren

SSD-18 (HTTP) validatie	
	<i>indicatoren</i>
<u>/01</u>	<u>valideren</u>
/01.01	De webapplicatie controleert de correcte authenticatie en autorisatie van de initiator van een HTTP-request.
/01.02	De webapplicatie controleert een HTTP-request op type, lengte, formaat, karakters en patronen.
/01.03	De webapplicatie controleert de binnenkomende: • URL's; • Query parameters (variabelen die de client via een GET doorgeeft); • Form parameters (variabelen die de client via een POST doorgeeft); • Cookies; • HTTP-headers; • XML (hieronder vallen ook protocollen zoals SOAP, JSON en REST); • Bestanden.
/01.04	Voor andere protocollen dan HTTP worden met HTTP vergelijkbare controles uitgevoerd.
/01.05	De webapplicatie verwijdert ongewenste invoer en maakt risicovolle karakters uit de invoer onschadelijk.
/01.06	In de configuratiehandleiding van de softwareleverancier is beschreven welke controles moeten worden uitgevoerd en welke ongewenste invoer door de webserver moeten worden verwijderd.
/01.07	Door de hostingpartij is de configuratie uitgevoerd conform de configuratiehandleiding.

Toelichting

/01.01 Dit zal in de regel een producteigenschap van de desbetreffende webserver zijn. Hiervoor is een inspectie van de broncode (codereview) nodig of een garantie van de softwareleverancier.

/01.02 Valideer de volgende scenario's:

- Het is niet mogelijk om een cookie te gebruiken vanaf een IP-adres anders dan het IP-adres aan wie de cookie verstrekt is.
- Het is niet mogelijk om transacties voor gevalideerde gebruikers uit te voeren vanaf een andere website dan de website waarop de gebruiker is gevalideerd.

/01.04 Denk daarbij bijvoorbeeld ook aan interne consistentie, zoals geldige scheiders voor attachments in email (Simple Mail Transfer Protocol (SMTP)).

/01.05 De inhoud van alle onderdelen van een HTTP-request wordt gevalideerd op basis van ongewenste invoer.

Als het moeilijk is om op basis van de whitelisting alle mogelijke malafide invoer uit te filteren, dan kan de invoer aanvullend worden gevalideerd op malafide sleutelwoorden, tekens en patronen (blacklisting). Denk aan invoervelden waar de gebruiker vrije tekst kan invoeren.

De webapplicatie filtert de invoer op basis van:

- Malafide sleutelwoorden (bijvoorbeeld 'DROP' of 'rm')
- Malafide tekens (bijvoorbeeld '"' of "'")
- Malafide patronen (bijvoorbeeld '/**/' of '..\..\')

De filtering is toegespitst op de programmaonderdelen waarin de invoer wordt verwerkt. Bij het gebruik van invoer voor het samenstellen van een databasequery zijn andere filters vereist dan voor het samenstellen van een LDAP-query.

In het geval de invoer één of meerdere sleutelwoorden, tekens of patronen van de blacklist bevat, worden deze uit de invoer verwijderd alvorens deze invoer verder gebruikt wordt binnen de applicatielogica.

3.20. SSD-19: Invoer normalisatie

SSD-19 Invoer normalisatie					
<i>Criterium (wie en wat)</i>	De (web)applicatie voorkomt manipulatie door alle ontvangen invoer te <u>normaliseren</u> alvorens die te valideren.				
<i>Doelstelling (waarom)</i>	Inzage, wijziging, verlies, of misbruik van gegevens door bijvoorbeeld manipulatie van de applicatielogica.				
<i>Risico</i>	Zonder normalisatie van de ontvangen inhoud kunnen kwaadwillenden kwetsbaarheden benutten om gegevens te bemachtigen, te muteren of toe te voegen.				
Referentie	NCSC	NIST	ISO27002		
	B3-3				

Toelichting

De (web)applicatie ontvangt invoer van de gebruiker en van andere applicaties. Deze invoer kan verschillende vormen hebben. De (web)applicatie dient eerst de invoer te normaliseren, alvorens een validatie van de invoer kan worden uitgevoerd via de mechanismen voor filtering.

Normaliseren van inhoud betekent dat de inhoud gaat voldoen aan een aantal beperkende regels. Hierdoor wordt de mogelijkheid weggenomen om de validaties te omzeilen. Bovendien wordt de invoer in een zodanige representatie gebracht dat deze op alle plaatsen in de webapplicatie veilig verwerkt kan worden (eliminatie van SQL- en HTML-injecties).

Via normalisatie voorkomt de (web)applicatie dat malafide verzoeken abusievelijk de filters voor validatie kunnen passeren. Normalisatie staat ook wel bekend als anti-evasion⁹ of canonicalization¹⁰.

Conformiteitsindicatoren

/01 normaliseren

SSD-19 Invoer normalisatie	
	<i>indicatoren</i>
<u>/01</u>	<u>normaliseren</u>
/01.01	De (web)applicatie controleert de invoer op verdachte karakters of commando's.
/01.02	De (web)applicatie verzorgt onder andere: • Het omzetten van NULL karakters naar spaties; • Het coderen van bijzondere karakters naar een uniforme codering, zoals UTF-8; • Het normaliseren van pad verwijzingen zoals './.' en './..'; • Het verwijderen van overbodige spaties en regeleinden; • Het verwijderen van onnodige witruimtes; • Het omzetten van backslashes '\' naar forward slashes '/'; • Het omzetten van mixed case strings naar lower case strings.

⁹ https://www.owasp.org/index.php/Virtual_Patching_Best_Practices#Anti-Evasion_Capabilities

¹⁰ https://www.owasp.org/index.php/Canonicalization_locale_and_Unicode of <http://en.wikipedia.org/wiki/Canonicalization>

Toelichting

/01.01 De richtlijnen voor invoerbehandeling zijn van toepassing voor *alle* invoer die van buiten de webapplicatie komt. Dus niet alleen (eind)gebruikers, maar ook externe systemen en applicaties.

/01.02 Maak hierbij ook gebruik van de OWASP richtlijn voor datavalidatie
https://www.owasp.org/index.php/Data_Validation.

/01.02 Maak (minimaal) gebruik van de daarvoor bestaande standaardservices.

/01.02 Converteer alle risicovolle tekens naar een veilig formaat.

De escape (\) voorafgaand aan een teken geeft aan dat het teken letterlijk moet worden geïnterpreteerd, \"wordt ". Een andere vorm van escaping is door gebruik te maken van de hexadecimale waarde van een teken, \x22 wordt "¹¹. Dit levert echter niet de gewenste tekst op indien deze wordt gecompileerd met een ASCII¹²-incompatibele tekenset¹³.

Tekens uit de invoer die verwerkbaar en niet ongewenst zijn kunnen nog steeds risicovol zijn bij het gebruik hiervan binnen de programmalogica. Om problemen hiermee te voorkomen moeten deze tekens 'onschadelijk' worden gemaakt.

Risicovolle tekens kunnen onderdeel uitmaken van legitieme invoer. Neem onderstaand voorbeeld waarbij de plaatsnaam ('s-Gravenhage) tot een syntactische incorrecte query leidt.

```
SELECT * FROM nieuws WHERE titel LIKE '%s-gravenhage%';
```

Door een escape voor de apostrof te plaatsen, beschouwt de database de apostrof als onderdeel van de invoer en niet als onderdeel van de query. Veel programmeertalen ondersteunen standaardservices voor het escaperen van gevaarlijke tekens.

Soortgelijke conversies gelden voor bijvoorbeeld HTML, XML, et cetera.

De volgende risicovolle tekens uit de invoer worden 'onschadelijk' gemaakt:

- Voer escaping uit op de invoer na het toepassen van whitelists en eventueel blacklists.
- Escaping is toegespitst op de programmaonderdelen waar invoer wordt verwerkt.

¹¹ 22 is de hexadecimale ASCII waarde van een dubbel aanhalingsteken.

¹² <http://en.wikipedia.org/wiki/ASCII>

¹³ ASCII was tot december 2007 de meest gebruikte tekenset op het internet, daarna werd dit UTF-8.

3.21. SSD-20: Codering van dynamische onderdelen

SSD-20 Codering van dynamische onderdelen in de uitvoer	
<i>Criterium (wie en wat)</i>	De (web)applicatie beperkt de uitvoer tot waarden die (veilig) verwerkt kunnen worden door deze te <u>normaliseren</u> .
<i>Doelstelling (waarom)</i>	Voorkom manipulatie van het systeem van andere gebruikers.
<i>Risico</i>	Via uitvoer van de (web)applicatie de werking van of informatie op het systeem van anderen manipuleren.

Toelichting

Vrijwel elke webpagina bevat naast statische ook dynamische informatie. Deze dynamische informatie kan bijvoorbeeld afkomstig zijn uit databases of externe bronnen, maar kan ook gebaseerd zijn op invoer van de gebruiker. Hoe de (web)applicatie dynamische pagina-inhoud codeert is afhankelijk van de plek op de pagina waar deze dynamische inhoud verschijnt. Als een (web)applicatie onvoldoende controles uitvoert op de uitvoer die het terugstuurt naar de eindgebruiker, kan het gebeuren dat er zich onbedoelde of ongewenste inhoud in de uitvoer bevindt.

Uitvoervalidatie voorkomt dat de (web)applicatie ongewenste opdrachten geeft aan de client, bijvoorbeeld in het geval van XSS. Uitvoervalidatie kan worden geïmplementeerd door het coderen van alle dynamische inhoud van een webpagina.

Conformiteitsindicatoren

/01 normaliseren

Door de uitvoer te normaliseren wordt voorkomen dat het ontvangende systeem gemanipuleerd kan worden via de webapplicatie.

SSD-20 Codering van dynamische onderdelen in de uitvoer	
	<i>indicatoren</i>
<u>/01</u>	<u>normaliseren</u>
/01.01	Alle uitvoer wordt naar een veilig formaat geconverteerd.
/01.02	In de configuratiehandleiding van de softwareleverancier is beschreven welke controles op het verwijderen van ongewenste invoer door de web-/applicatieserver moeten worden uitgevoerd.
/01.03	Door de hostingpartij is de configuratie uitgevoerd conform de configuratiehandleiding.

Toelichting

/01.01 Uitgebreide testen kunnen aannemelijk maken dat uitvoer altijd genormaliseerd wordt. Gerichte codereviews leveren in de regel meer zekerheid op.

/01.01 Het coderen van dynamische pagina-inhoud houdt in dat de webapplicatie mogelijk 'gevaarlijke' tekens codeert. Hoe de webapplicatie deze informatie moet coderen is afhankelijk van de plek in de pagina waar deze dynamische inhoud verschijnt. Zo moet men speciale tekens in HTML, javascript, HTML-attributen en URL's allemaal op een andere wijze coderen. Neem bijvoorbeeld het 'groter dan'-teken (>). Afhankelijk van de plek waar dit teken wordt gebruikt, ziet de gecodeerde versie van dit teken er als volgt uit:

- HTML gecodeerd: <
- HTML-attribuut gecodeerd: >
- javascript gecodeerd: \x3E

- CSS gecodeerd: \3E
- URL gecodeerd: %3E

Veel scripting- en programmeertalen hebben standaard bibliotheken waarmee deze codering kan worden uitgevoerd.

3.22. SSD-21: Geparametriseerde queries

SSD-21 Geparametriseerde queries	
<i>Criterium (wie en wat)</i>	De webapplicatie communiceert alleen met onder- en achterliggende systemen op basis van statisch geconfigureerde (geparametriseerde) <u>queries en commando's</u> en uitsluitend ten behoeve van de <u>noodzakelijke functionaliteit</u> .
<i>Doelstelling (waarom)</i>	Kwetsbaarheid van de onder- en achterliggende systemen voorkomen en de integriteit en vertrouwelijkheid garanderen.
<i>Risico</i>	Via manipulatie (bijvoorbeeld commando- of SQL-injectie) kennis nemen van de inhoud van de onder- en achterliggende systemen of deze kunnen manipuleren.

Toelichting

Een webapplicatie integreert met onder- en achterliggende systemen (bijvoorbeeld het onderliggende besturingssysteem en een achterliggende database) door gebruik te maken van commando's en queries.

Grofweg bestaan er twee methoden om vanuit een webapplicatie een query of commando te genereren die gebruik maakt van invoer van gebruikers: via dynamische strings of via parameters.

Bij dynamische strings plakt de webapplicatie een vaste string (bijvoorbeeld de start van een SELECT-statement) aan een variabele (bijvoorbeeld de inhoud van de WHERE-clause). Via deze methode bestaat de mogelijkheid dat de door een gebruiker geleverde invoer de query op ongecontroleerde wijze verandert. Hierdoor kunnen gegevens bijvoorbeeld vernietigd worden of ongefilterd bij de gebruiker komen, waardoor de vertrouwelijkheid geschonden wordt. Bij gebruik van parameters gebruikt de webapplicatie een vaste string voor de query, waarbij alleen een vaste plek is ingeruimd voor variabelen. Het mechanisme dat voor invulling van de variabelen op deze vaste plek zorgt, zorgt er tevens voor dat de query niet kan veranderen.

Conformiteitsindicatoren

/01 queries en commando's

Bij het gebruik van geparametriseerde queries is de syntax van de query statisch en wordt invoer alleen gebruikt om vooraf gedefinieerde variabelen te vullen. Door te voorkomen dat de syntax van de query wijzigt, voorkomt de webapplicatie SQL-injectieaanvallen.

Geparametriseerde queries zijn ook efficiënter: doordat ze voorgedefinieerd zijn, gebruiken ze bekende tabelstructuren optimaal. Toch geven ze de eindgebruiker geen volledige vrijheid. Op dezelfde manier voorkomen statisch geprogrammeerde commando's ervoor dat de gebruiker geen mogelijkheid heeft de aard van de commando's te beïnvloeden.

SSD-21 Geparametriseerde queries	
	<i>indicatoren</i>
<u>/01</u>	<u>commando's en queries</u>
/01.01	De broncode van de webapplicatie bevat uitsluitend commando- en queryteksten die opgebouwd worden uit in de code reeds aanwezige vaste tekstfragmenten.
/01.02	Gebruikersinvoer die gebruikt moet worden in commando's en queries wordt op een zodanige manier doorgegeven, dat de beoogde werking niet wordt gewijzigd.

Toelichting

/01.01 De leverancier van de software geeft hierover een verklaring van een onafhankelijke derde af of stelt de broncode beschikbaar voor review.

/01.02 Voor zover er sprake is van gebruikersinvoer in commando's en queries wordt deze als parameters doorgegeven. Het doel hiervan is te voorkomen dat de gebruiker zelf kan bepalen welke commando's of queries worden uitgevoerd.

/02 noodzakelijke functionaliteit

Services die niet nodig zijn voor de functionaliteit van een (web)applicatie vormen een onnodig risico en dienen daarom achterwege te blijven.

SSD-21 Geparametriseerde queries	
	<i>indicatoren</i>
<u>/02</u>	<u>noodzakelijke functionaliteit</u>
/02.01	Van elke webapplicatie is bekend welke functionaliteit van backend-systemen nodig is.
/02.02	Directe data-toegang tot backend-systemen is ongewenst en alleen toegestaan indien andere opties niet voor handen zijn.

Toelichting

/02 De koppeling met backend-systemen is gedocumenteerd, inclusief de aard van de koppeling en de daarvoor noodzakelijke (gebruikers)rechten.
Denk hierbij aan File Transfer Protocol (FTP), Telnet, Post Office Protocol (POP)/SMTP (postbusfunctie), et cetera.

/02.02 De afweging om hiervan af te wijken is in de ontwerpdocumentatie vastgelegd.

Kwetsbaarheid

- SQL-injectie (OWASP 2013: A1 – Injection, CWE-89: SQL Injection)
- Commando injectie (OWASP: Command Injection, CWE-77: Command Injection)
- OS commando injectie (CWE-78: OS Command Injection)
- Code injectie (OWASP: Code Injection)

3.23. SSD-22: Invoervalidatie

SSD-22 Invoervalidatie	
<i>Criterium (wie en wat)</i>	De (web)applicatie beperkt de mogelijkheid tot <u>manipulatie</u> door de invoer te <u>valideren</u> , voordat deze invoer wordt verwerkt.
<i>Doelstelling (waarom)</i>	Voorkom (on)opzettelijke manipulatie van de (web)applicatie, waardoor de vertrouwelijkheid, integriteit en/of beschikbaarheid van de (web)applicatie aangetast worden.
<i>Risico</i>	Inzage, wijziging, verlies, of misbruik van gegevens door bijvoorbeeld manipulatie van de (web)applicatieloga.

Toelichting

Ongecontroleerde (niet-gevalideerde) invoer van gebruikers is een belangrijke dreiging voor een (web)applicatie. Als invoer van gebruikers rechtstreeks wordt gebruikt in HTML-uitvoer, cookie-waarden, SQL-queries, et cetera, bestaat er een (grote) kans dat een kwaadwillende de (web)applicatie compromitteert. Een gebrek aan invoervalidatie kan tot XSS-, commando- en SQL-injectie-kwetsbaarheden leiden.

Als de (web)applicatie de mogelijkheid biedt om bestanden in te lezen bestaat daarmee de mogelijkheid dat kwaadwillenden via malafide invoer willekeurige (web)applicatiecode kunnen uitvoeren op de server. Hiermee is het bijvoorbeeld mogelijk om ongeautoriseerd de database op een server te benaderen.

Als een (web)applicatie wel invoervalidatie en filtering uitvoert, blijkt deze filtering vaak niet voldoende effectief om alle mogelijke aanvallen op de (web)applicatie te blokkeren. Dit is voornamelijk het geval op het moment dat de (web)applicatie gebruik maakt van blacklisting¹⁴ om mogelijk gevaarlijke strings uit de invoer te verwijderen.

De belangrijkste vuistregel voor invoer in een (web)applicatie is dat de applicatie geen enkele invoer mag vertrouwen en daarom moet valideren. Alle invoer moet daarom voor verwerking door de (web)applicatie worden gevalideerd op juistheid, volledigheid en geldigheid. Daarbij dient de invoer minimaal gevalideerd te worden op waarden die buiten het geldige bereik vallen (grenswaarden), ongeldige tekens, ontbrekende of onvolledige gegevens, gegevens die niet aan het juiste formaat voldoen en inconsistentie van gegevens ten opzichte van andere gegevens binnen de invoer dan wel in andere gegevensbestanden. Invoervalidatie is dé voorwaarde voor betrouwbare gegevensverwerking en ongeldige invoer wordt door de (web)applicatie geweigerd.

Er geldt een aantal uitgangspunten met betrekking tot invoer bij het ontwikkelen van (web)applicaties, deze zijn:

- De client (applicatie)¹⁵ is niet te vertrouwen en dus de invoer die hier vandaan komt ook niet.
- De invoer wordt voor valideren eerst genormaliseerd (SSD-19).
- De invoer die niet aan één of meerdere controles voldoet wordt verwijderd of geweigerd.

In het ontwikkelproces van de (web)applicatie zal de software expliciet op een correcte invulling van deze uitgangspunten onderzocht moeten worden. Dit vraagt om uitgebreide testen of gerichte codereviews.

¹⁴ <http://en.wikipedia.org/wiki/Blacklist>

¹⁵ http://nl.wikipedia.org/wiki/Client_%28applicatie%29

Conformiteitsindicatoren

/01 valideren

Valideren van de inhoud zorgt ervoor dat alleen geldige gegevens verwerkt worden. Validatie vindt zowel op protocol-niveau (meestal HTTP) (SSD-18) als op applicatie-niveau plaats. Het doel is te voorkomen dat de software op applicatie-niveau misbruikt wordt of faalt door de door de gebruiker aangeleverde gegevens.

SSD-22 Invoervalidatie	
	<i>indicatoren</i>
<u>/01</u>	<u>valideren</u>
/01.01	Foute, ongeldige of verboden invoer wordt geweigerd. De (web)applicatie voert deze controle van de invoer uit aan de serverzijde en vertrouwt niet op de maatregelen aan de client-zijde.
/01.02	De (web)applicatie valideert alle invoer die de gebruiker aan de (web)applicatie verstrekt.
/01.03	Voor elke controle die de (web)applicatie uitvoert aan de client -zijde, is er een equivalent aanwezig aan de server-zijde.

Toelichting

- /01 De richtlijnen voor invoerbehandeling zijn van toepassing voor *alle* invoer die van buiten de (web)applicatie komt. Dus niet alleen (eind)gebruikers, maar ook externe systemen en applicaties.
- /01.01 Valideer de inhoud van een HTTP-request op basis van verwerkbare invoer (whitelist)
Valideer de invoer op malafide sleutelwoorden, tekens en patronen (blacklist).

3.24. SSD-23: File includes

SSD-23 Remote File Includes					
<i>Criterium (wie en wat)</i>	De (web)applicatie voorkomt de mogelijkheid van <u>dynamische file includes</u> .				
<i>Doelstelling (waarom)</i>	Voorkom (on)opzettelijke manipulatie van de (web)applicatie, waardoor de vertrouwelijkheid, integriteit en/of beschikbaarheid van de (web)applicatie aangetast worden.				
<i>Risico</i>	Inzage, wijziging, verlies, of misbruik van gegevens door bijvoorbeeld manipulatie van de (web)applicatielogica.				
Referentie	NCSC	NIST	ISO27002		
	B3-7				

Toelichting

Remote File Inclusion (RFI) is een kwetsbaarheid die zich voordoet op webservern die gebruik maken van dynamische file includes in script- en programmeertalen. Wanneer een dergelijke website kwetsbaar is voor RFI, kan een kwaadwillende zijn eigen code door de server laten uitvoeren. RFI is mogelijk op het moment dat een pagina op een webserver de volgende kenmerken heeft:

- De pagina is geschreven in PHP.
- De pagina maakt gebruik van andere PHP-scripts via een include (of een andere vergelijkbare functie).
- Gebruikersinvoer bepaalt de naam van de scripts waarvan de pagina gebruik maakt.
- PHP staat URL includes toe (allow_url_include = 'On').

Of een aanval succesvol is, hangt ook af van de configuratie van de webserver. Als de PHP-optie allow_url_include bijvoorbeeld is ingesteld op de waarde 'Off', zal PHP het importeren van een PHP-script vanaf een externe locatie niet toestaan en zal het moeilijker zijn om deze RFI-kwetsbaarheid uit te buiten. Wel is het in dat geval nog steeds mogelijk om willekeurige lokale bestanden op de server (bijvoorbeeld /etc/passwd) te laten importeren door het script. Ook als de webserver zelf geen verbinding met internet kan opzetten, is het nog steeds mogelijk een RFI-kwetsbaarheid uit te buiten. Hiervoor kan een kwaadwillende bijvoorbeeld gebruik maken van een base64 data include.

Conformiteitsindicatoren

/01 dynamische file includes

SSD-23 Remote File Includes	
	<i>indicatoren</i>
/01	<u>dynamische file includes</u>
/01.01	De (web)applicatie maakt geen gebruik van dynamische file includes, tenzij dit vanaf een vertrouwde locatie plaatsvindt.
/01.02	De (web)applicatie beperkt (op de server) de keuzemogelijkheid voor het uploaden van bestanden, bijvoorbeeld via whitelisting ¹⁶ .
/01.03	In de configuratiehandleiding van de softwareleverancier zijn eventueel benodigde maatregelen beschreven die includes moeten voorkomen.
/01.04	De server is zo geconfigureerd door de hostingpartij dat file includes niet mogelijk zijn.

¹⁶ <http://en.wikipedia.org/wiki/Whitelist>

Toelichting

- /01.01 De vertrouwddheid van een locatie moet formeel zijn vastgesteld.
- /01.01 Wanneer kwaadwillenden via malafide invoer willekeurige bestanden kunnen verwerken in de (web)applicatie, bestaat de mogelijkheid dat willekeurige webapplicatie-code wordt uitgevoerd op de server. Hiermee is het bijvoorbeeld mogelijk om ongeautoriseerd de database op een server te benaderen.

3.25. SSD-24: Beperking van te versturen HTTP-headers

SSD-24 Beperking van te versturen HTTP-headers					
<i>Criterium (wie en wat)</i>	De webserver stuurt bij een antwoord aan een gebruiker alleen die informatie in de <u>HTTP-headers</u> mee die van belang is voor het functioneren van HTTP.				
<i>Doelstelling (waarom)</i>	Voorkom inzage, wijziging of verlies van gegevens door manipulatie van de logica van de webserver of webapplicatie, door zo min mogelijk overbodige informatie op te nemen in een antwoord aan gebruikers.				
<i>Risico</i>	De werking van de webserver of webapplicatie wordt gemanipuleerd, waardoor deze onder controle komt van een aanvaller.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-8				

Toelichting

De webserver ondersteunt het HTTP-protocol. HTTP kent methoden, headers en foutinformatie, die mogelijk misbruikt kunnen worden. Daarom is het gebruik hiervan beperkt tot het minimum dat noodzakelijk is voor de goede werking van de ontsloten webapplicaties. Een HTTP-header bevat informatie die de gebruiker kan identificeren. Als de webserver antwoord geeft aan de gebruiker en daarbij de HTTP-header kopieert, wordt soms privacygevoelige informatie meegestuurd met het antwoord. Het lekken van informatie moet zoveel mogelijk worden voorkomen. Door middel van HTTP-headers kan onnodig informatie worden vrijgegeven. Het gebruik van de volledige HTTP-header dient daarom waar mogelijk te worden beperkt.

Conformiteitsindicatoren

/01 HTTP-headers

SSD-24 Beperking van te versturen HTTP-headers	
	<i>indicatoren</i>
<u>/01</u>	<u>HTTP-headers</u>
/01.01	De softwareleverancier legt in de ontwerpdocumentatie vast welke HTTP-headers worden gebruikt door de webapplicatie en hoe HTTP-headers uit de antwoorden worden verwijderd.
/01.02	De softwareleverancier legt in de configuratiedocumentatie vast welke HTTP-headers worden gebruikt door de webserver en hoe HTTP-headers uit de antwoorden worden verwijderd.
/01.03	De softwareleverancier onderbouwt en beschrijft eventuele noodzakelijke afwijkingen van de standaard configuratie op de webserver, die nodig zijn voor het goed functioneren van de webapplicatie.
/01.04	De webserver verstuurt alleen HTTP-headers die voor het functioneren van HTTP van belang zijn.
/01.05	De webserver verwijdert alle niet noodzakelijke informatie uit de HTTP-header, alvorens deze voor het antwoord te gebruiken.

Toelichting

/01.01 In de ontwerp- casu quo configuratiedocumentatie is vastgelegd:

- welke HTTP-methoden voor het functioneren van HTTP van belang zijn.
- welke HTTP-headers voor het functioneren van HTTP van belang zijn.

- welke HTTP-requests methoden (GET, POST, et cetera) voor de ondersteunde webapplicaties benodigd zijn.
- welke informatie in de HTTP-headers voor het functioneren van belang zijn.
- welke standaard foutmelding(en) worden getoond/verstuurd.
- op welke wijze bovenstaande is gerealiseerd, denk hierbij aan de configuratie van de webserver en, indien van toepassing, de application level firewall.

Eventuele afwijkingen van bovenstaande die noodzakelijk zijn, omdat de webapplicatie anders niet kan functioneren, zijn onderbouwd.

- /01.02 Hoe bij beantwoording delen van informatie uit een HTTP-header kunnen worden verwijderd, is afhankelijk van het gebruikte type webserver. In de regel zal dit via instructies in de configuratie van de webserver gerealiseerd worden. Deze worden bij implementatie ingevuld en bij een audit gecontroleerd.
- /01.05 Het versturen van vertrouwelijke informatie is mogelijk via de http-methoden GET en POST. Alle overige HTTP-headers kan de applicatie in de regel zonder gevolgen uit een HTTP-response verwijderen. Er is hierdoor geen noodzaak vertrouwelijke informatie via de http-header te versturen.

3.26. SSD-25: Beperken van te tonen HTTP-header informatie

SSD-25 Beperken van te tonen HTTP-header informatie					
<i>Criterium (wie en wat)</i>	De webapplicatie toont alleen noodzakelijke <u>informatie in HTTP-headers</u> die van belang is voor het functioneren van HTTP.				
<i>Doelstelling (waarom)</i>	Voorkom dat overbodige technische informatie in een antwoord leidt tot informatie die gebruikt kan worden voor manipulatie van de logica van de webserver of webapplicatie.				
<i>Risico</i>	Gedetailleerde informatie over de technische inrichting van de webserver wordt gebruikt, waardoor deze onder controle komt van een aanvaller.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-9				

Toelichting

HTTP-headers kunnen veel informatie bevatten over de webapplicatie en de software waarvan de webapplicatie gebruik maakt. Eén van de bekendste HTTP-headers die informatie vrijgeeft, is de 'Server'-header. In veel gevallen zal de webserver via deze header informatie geven over het type webserver waar de pagina van afkomstig is. Als een webserver antwoord geeft aan een gebruiker, staat er soms te veel informatie in de HTTP-header. Overbodige technische informatie, zoals het type webserver of een versienummer, kan worden misbruikt door een kwaadwillende.

Het is voor een client niet van belang om te weten welk type webserver antwoord heeft gegeven op het HTTP-request. In dit kader kan bijvoorbeeld de 'Server'-header uit het antwoord worden verwijderd of worden vervangen door een nietszeggende inhoud.

Conformiteitsindicatoren

/01 informatie in HTTP-headers

SSD-25 Beperken van te tonen HTTP-header informatie	
	<i>indicatoren</i>
<u>/01</u>	<u>informatie in HTTP-headers</u>
/01.01	De softwareleverancier legt in de ontwerpdocumentatie vast welke overbodige informatie uit antwoorden wordt verwijderd of tot een minimum wordt beperkt door de webserver.
/01.02	De softwareleverancier onderbouwt en beschrijft eventuele noodzakelijke afwijkingen van de standaard configuratie op de webserver, die nodig zijn voor het goed functioneren van de webapplicatie.
/01.03	De webserver verwijdert de 'Server'-header uit het antwoord of vult hiervoor een nietszeggende inhoud in.

Toelichting

/01.01 In de ontwerp- casu quo configuratiedocumentatie is vastgelegd:

- welke HTTP-methoden voor het functioneren van HTTP van belang zijn.
- welke HTTP-headers voor het functioneren van HTTP van belang zijn.
- welke HTTP-requests methoden (GET, POST, et cetera) voor de ondersteunde webapplicaties benodigd zijn.
- welke informatie in de HTTP-headers voor het functioneren van belang zijn.
- welke standaard foutmelding(en) worden getoond/verstuurd.
- op welke wijze bovenstaande is gerealiseerd, denk hierbij aan de configuratie van de webserver en, indien van toepassing, de application level firewall.

Eventuele afwijkingen van bovenstaande die noodzakelijk zijn, omdat de webapplicatie anders niet kan functioneren, zijn onderbouwd.

/01.02 In de regel zal dit via instructies in de configuratie van de webserver gerealiseerd worden. Deze worden bij implementatie ingevuld en bij een audit gecontroleerd.

3.27. SSD-26: HTTP-methoden

SSD-26 HTTP-methoden					
<i>Criterium (wie en wat)</i>	De webserver gebruikt alleen de <u>HTTP-functionaliteiten</u> die nodig zijn voor het functioneren van de webapplicatie.				
<i>Doelstelling (waarom)</i>	Voorkom het gebruik van niet noodzakelijke methoden, die gebruikt kunnen worden voor de manipulatie van de logica van de webserver of webapplicatie.				
<i>Risico</i>	De werking van de webserver of webapplicatie wordt gemanipuleerd, waardoor deze onder controle komt van een aanvaller.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-12				

Toelichting

De webserver ondersteunt het HTTP-protocol. HTTP kent methoden, headers en foutinformatie, die mogelijk misbruikt kunnen worden. Daarom is het gebruik hiervan beperkt tot het minimum dat noodzakelijk is voor de goede werking van de ontsloten webapplicaties. HTTP 1.1 en 2.0 ondersteunen verschillende functionaliteiten. In de praktijk gebruikt een webapplicatie alleen de functies GET en POST. Voor veel scripts en objecten geldt zelfs dat alleen de GET nodig is. De overige functionaliteiten zijn vrijwel nooit nodig binnen traditionele webapplicaties en vormen een extra beveiligingsrisico.

Conformiteitsindicatoren

/01 HTTP-functionaliteiten

SSD-26 HTTP-methoden	
	<i>indicatoren</i>
<u>/01</u>	<u>HTTP-functionaliteiten</u>
/01.01	Op de webserver worden, indien mogelijk, alleen de GET en POST geactiveerd. De softwareleverancier onderbouwt en beschrijft eventuele noodzakelijke methoden anders dan GET en POST en legt dit vast in de ontwerpdocumentatie.
/01.02	De softwareleverancier legt in de configuratiedocumentatie vast welke HTTP-methoden worden gebruikt door de webserver.
/01.03	De hostingpartij waarborgt dat alleen de door de webapplicaties benodigde HTTP-requests methoden op de webserver zijn toegestaan en de overige niet noodzakelijke HTTP-requests methoden zijn gedeactiveerd.

Toelichting

/01.01 In de ontwerp- casu quo configuratiedocumentatie is vastgelegd:

- welke HTTP-methoden voor het functioneren van HTTP van belang zijn.
- welke HTTP-headers voor het functioneren van HTTP van belang zijn.
- welke HTTP-requests methoden (GET, POST, et cetera) voor de ondersteunde webapplicaties benodigd zijn.
- welke informatie in de HTTP-headers voor het functioneren van belang zijn.
- welke standaard foutmelding(en) worden getoond/verstuurd.
- op welke wijze bovenstaande is gerealiseerd, denk hierbij aan de configuratie van de webserver en, indien van toepassing, de application level firewall.

Eventuele afwijkingen van bovenstaande die noodzakelijk zijn, omdat de webapplicatie anders niet kan functioneren, zijn onderbouwd.

- /01.01 Methoden anders dan GET en POST zijn vrijwel nooit nodig binnen traditionele web-applicaties en vormen alleen een extra beveiligingsrisico (misbruik). Voor 'Web 2.0' zijn vaak wel aanvullende methoden nodig.
- /01.03 Het is in alle gevallen aan te raden om niet benodigde HTTP-methoden via configuratie van de webserver of via de application level firewall te blokkeren.

3.28. SSD-27: Error handling

SSD-27 Error handling					
<i>Criterium (wie en wat)</i>	De webserver neemt in een foutmelding aan de client geen <u>inhoudelijke foutinformatie</u> op die misbruikt kan worden.				
<i>Doelstelling (waarom)</i>	Voorkom dat technische informatie over de webapplicatie naar buiten lekt, waardoor manipulatie van de logica van de webserver of webapplicatie mogelijk wordt.				
<i>Risico</i>	De foutinformatie wordt gebruikt om de werking van de webserver of webapplicatie te manipuleren, waardoor deze onder controle komt van een aanvaller.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-10	SI-11	12.2.1 12.2.2 12.2.3 12.2.4		

Toelichting

De webserver ondersteunt het HTTP-protocol. HTTP kent methoden, headers en foutinformatie, die mogelijk misbruikt kunnen worden. Daarom is het gebruik hiervan beperkt tot het minimum dat noodzakelijk is voor de goede werking van de ontsloten webapplicaties. Sommige webapplicaties leveren bij het optreden van een foutsituatie allerlei informatie aan over de achtergrond(en) van de fout. Een uitgebreide foutmelding kan een kwaadwillende helpen om meer inzicht te krijgen in de programmalogica van een webapplicatie. Een foutmelding vertelt vaak iets over de gebruikte database, het uitgevoerde SQL-verzoek of het aangeroepen bestand. Al deze informatie draagt bij aan kennisvorming van de kwaadwillende over de infrastructuur.

Op het moment dat zich een probleem voordoet binnen een webapplicatie zal de webserver bijvoorbeeld veelal een statuscode '500 Internal Server Error' terugsturen. Dit wijst op een exceptie. Hierbij is het mogelijk dat de webserver gevoelige informatie over de webapplicatie openbaart, zoals databasenames, gebruikersnamen, bestandsnamen, interne IP-adressen etc. Om het lekken van technische informatie te voorkomen zou bijvoorbeeld een application level firewall een dergelijke statuscode kunnen detecteren. De firewall kan het gedetailleerde antwoord van de webserver negeren en een standaard foutmelding terugsturen naar de client. Dit kan bijvoorbeeld zijn 'Er heeft zich een onbekende fout voorgedaan'. Ook webserver zelf bieden functionaliteit om standaard meldingen te laten genereren aan de hand van specifieke statuscodes.

Conformiteitsindicatoren

/01 inhoudelijke foutinformatie

SSD-27 Error handling	
	<i>indicatoren</i>
/01	<u>inhoudelijke foutinformatie</u>
/01.01	Bij het optreden van een fout wordt de informatie in een HTTP-response tot een minimum beperkt. Een eventuele foutmelding zegt wel <i>dat</i> er iets is fout gegaan, maar niet <i>hoe</i> het is fout gegaan.
/01.02	De softwareleverancier legt in de configuratiedocumentatie vast hoe de webserver en/of de application-level firewall moet worden geconfigureerd om alleen standaard foutmeldingen te tonen en versturen.

SSD-27 Error handling	
	<i>indicatoren</i>
/01.03	De webserver toont bij een fout geen gedetailleerde informatie aan de client, maar alleen een standaard foutmelding.
/01.04	De webserver en/of application firewall verwijdert alle niet noodzakelijke foutinformatie uit de HTTP-response.

3.29. SSD-28: Commentaar(regels)

SSD-28 Commentaar(regels)					
<i>Criterium (wie en wat)</i>	De aan de gebruiker aangeboden scripts / code bevat geen <u>commentaar</u> , zodat die niet kunnen worden misbruikt.				
<i>Doelstelling (waarom)</i>	Voorkom dat technische informatie over de webapplicatie naar buiten lekt, waardoor manipulatie van de logica van de webserver of webapplicatie mogelijk wordt.				
<i>Risico</i>	Kennis nemen van de technologieën van de webapplicatie, om deze vervolgens te gebruiken om de webapplicatie aan te vallen.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-11				

Toelichting

Commentaar(regels) in scripts of code kunnen ongewild informatie vrijgeven. Webapplicaties bevatten HTML-code of 'client-side scripts' (zoals javascript) vaak commentaar. Commentaar(regels) zijn niet altijd problematisch. In sommige gevallen bevat commentaar echter 'een geheugensteuntje' voor programmeurs gedurende de ontwikkel- en testfase en vergeten zij deze informatie te verwijderen zodra een webapplicatie in productie gaat. Het commentaar kan echter ook informatie bevatten die gebruikt kan worden om zwakheden op te sporen. Denk hierbij aan scripting code die een aanvalleur gebruikt in XSS-aanvallen, informatie over gebruikte technologieën op de server en uitgebreide foutmeldingen. Als een webapplicatie onvoldoende controles uitvoert op de uitvoer die het terugstuurt naar de eindgebruiker, kan het gebeuren dat er zich onbedoelde of ongewenste inhoud in de uitvoer bevindt.

Conformiteitsindicatoren

/01 commentaar

De webapplicatie geeft geen informatie over de interne werking of configuratie van de webapplicatie zelf of een van de systemen waarmee de webapplicatie samenwerkt.

SSD-28 Commentaar(regels))	
	<i>indicatoren</i>
<u>/01</u>	<u>commentaar</u>
/01.01	De applicatieontwikkelaar heeft alle commentaarregels uit de scripts (code) verwijderd.
/01.02	Ook andere elementen dan commentaarregels, zoals META, PARAM, OBJECT en INPUT, bevatten commentaar die onnodig informatie vrijgegeven, zoals gebruikte software, versies of infrastructurele componenten, dienen uit de voor de gebruiker zichtbare code te worden verwijderd.

Toelichting

- /01 Tijdens deployment van een webapplicatie kan alle code die naar clients wordt gestuurd ontdaan worden van commentaar. Als alternatief zijn application-level firewalls in staat om commentaar(regels) uit HTML- en scriptcode te verwijderen en zodoende 'gefilterde' antwoorden terug te geven aan de cliënt. Indien hiervoor gekozen wordt dienen hierover in de configuratiedocumentatie configuratie-eisen te zijn opgenomen.
- /01.01 Soms vereisen gebruiksvoorwaarden dat bij het gebruik van code, bepaalde gegevens verstrekt worden bij het gebruik van code. Het gaat dan vaak om het beschermen van auteursrecht of het verstrekken van de contactgegevens van de auteur. Het ge-

bruik van commentaar is onder deze omstandigheden acceptabel, tenzij het commentaar de eisen uit de gebruiksvoorwaarden overstijgt.

3.30. SSD-29: Directory listing

SSD-29 Directory listing					
<i>Criterium (wie en wat)</i>	De aan de gebruiker getoonde informatie bevat geen <u>directory listings</u> , zodat die niet kunnen worden misbruikt.				
<i>Doelstelling (waarom)</i>	Compliance met dit criterium zorgt dat een buitenstaander ongeautoriseerd kennis kan nemen van wat in de directories van de (web)server is opgeslagen.				
<i>Risico</i>	Kennis nemen van de inhoud van directories biedt de mogelijkheid van het compromitteren van vertrouwelijke informatie of van het kennis nemen van de technologieën van de webapplicatie. Deze kennis kan vervolgens gebruikt worden om de webapplicatie aan te vallen.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B3-13				

Toelichting

Via een zogenaamde 'directory listing' kan een gebruiker via internet de inhoud van een directory bekijken. Het opvragen van een 'directory listing' via internet komt overeen met het lokaal uitvoeren van een dir-commando onder Windows of een ls-commando onder UNIX/Linux. Zodra een webserver de mogelijkheid biedt om 'directory listings' uit te voeren, bestaat de mogelijkheid dat een kwaadwillende de inhoud van 'vertrouwelijke' directories raadpleegt (zoals de '/etc/'-directory onder UNIX/Linux-systemen).

Conformiteitsindicatoren

/01 directory listings

Voorschrift voor de configuratie van de webserver. Dit voorschrift zal in ieder geval enkele basale instellingen beschrijven om onnodig lekken van informatie tegen te gaan.

SSD-29 Directory listing	
	<i>indicatoren</i>
<u>/01</u>	<u>directory listings</u>
/01.01	De toegang tot de inhoud van bestanden in directories moet altijd verlopen via de webapplicatie.
/01.02	De softwareleverancier maakt geen gebruik van directory listings, tenzij er bewust voor deze functionaliteit is gekozen. De keuze is in de ontwerpdocumentatie onderbouwd en vastgelegd.
/01.03	De softwareleverancier legt in de configuratiehandleiding vast welke directory listings niet uitgeschakeld moeten worden.
/01.04	De hostingpartij schakelt, met uitzondering van de in de configuratiehandleiding vastgelegde uitzonderingen, de directory listings uit.

Toelichting

- /01.01 De webapplicatie bepaalt absolute paden voor bestanden die de gebruiker rechtstreeks mag benaderen (bijvoorbeeld afbeeldingen) en fungeert als medium voor bestanden die de gebruiker niet rechtstreeks mag benaderen (bijvoorbeeld gegevensbestanden).
- /01.02 Het is niet mogelijk om de inhoud van het filesysteem van de server op te vragen. De webserver ondersteunt standaard geen directory listings.
- /01.03 Staat directory listing aan, dan kan de websitebezoeker de inhoud van bepaalde mappen zien. In de regel zal dit via instructies in de configuratie van de webserver

gerealiseerd worden. Deze worden bij implementatie ingevuld en bij een audit gecontroleerd.

3.31. SSD-30: Applicatie logging

SSD-30 Applicatie logging					
<i>Criterium (wie en wat)</i>	In de (web)applicatieomgeving zijn signaleringsfuncties (<u>registratie en detectie</u>) actief en efficiënt, effectief en beveiligd ingericht.				
<i>Doelstelling (waarom)</i>	Het maakt mogelijk eventuele schendingen van functionele en beveiligingseisen te kunnen detecteren en achteraf de juistheid van de uitgevoerde acties, zowel op strategisch als operationeel niveau te kunnen vaststellen.				
<i>Risico</i>	Tekortkomingen en zwakheden in de geleverde producten/diensten kunnen niet gesignaleerd worden en herstelacties kunnen niet tijdig worden genomen.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
	B7-1	IR-4	6.1.6		
	B7-2	IR-5	6.2.2		
	B7-4	IR-6	6.2.3		
	B7-5	IR-7	10.10.6		
	B7-6		13.1.1		
	B7-7		13.1.2		
	B7-8		13.2.1		
			13.2.2		
			14.1.3		

Toelichting

Logging is een proces voor het registreren van activiteiten en gebeurtenissen in systemen om achteraf de rechtmatigheid van de resourcebenaderingen, alsmede vroegtijdige ongeautoriseerde toegangspogingen van systemen en netwerken te kunnen signaleren. Omdat systemen uitgebreide loggingsfunctionaliteit kennen moet vooraf een beperkte, maar wel representatieve selectie van de te loggen systeem-gegevens worden gemaakt, om de controlewerkzaamheden zo doelmatig mogelijk te laten verlopen. Hierbij dient men met een aantal organisatorische en technische aspecten rekening te houden.

Het afschermen van loggingsgegevens:

De opgeslagen loggingsinformatie is zeer interessant voor kwaadwillenden aangezien ze hiermee veel kunnen leren over de opbouw van de infrastructuur en ze eventuele sporen van misbruik kunnen wissen. Daarom is het van belang veel aandacht te besteden aan de beveiliging van de loggingsgegevens, zodat onbevoegden hiertoe geen toegang hebben en hierin geen wijzigingen kunnen aanbrengen. Een andere mogelijke beveiligingsmaatregel in dit kader kan ook zijn om logbestanden digitaal te ondertekenen.

Integriteit van registraties:

Om te voorkomen dat kwaadwillende sporen uitwissen moeten logfiles zo zijn ingesteld dat deze achteraf niet kunnen worden aangepast. Deze beveiligingseis is essentieel bij reconstructievraagstukken in relatie tot opgetreden issues/incidenten.

Aandachtspunten log-informatie:

- Bepaal welke gebeurtenissen worden gelogd en onderhoud deze regels.
- Onderhoud kennis van correlaties die op misbruik duiden.
- Ter verbetering van de leesbaarheid van de logging is het aan te raden filters op de logging te plaatsen.
- Voorkom dat het herkennen van verdachte patronen in de logging afhangt van de kunde van de operationeel beheerder.

Bewaartermijnen van registraties:

Er moet worden bepaald hoe lang logging online en offline beschikbaar moet en mag zijn. Online-beschikbaarheid van logging kan essentieel zijn voor het efficiënt verhelpen van beveiligingsincidenten. De duur van offline-beschikbaarheid kan beperkt worden door wet- en regelgeving. Voordat wordt besloten om gebeurtenissen in een omgeving te loggen, moet zijn

vastgesteld hoe lang en op welke manier logging beschikbaar moet blijven. Dit bepaalt welke media nodig zijn en hoeveel capaciteit je voor de logging moet reserveren. Het systeem, waarmee gegevens opgeslagen en behandeld worden, dient dusdanig te zijn dat de gegevens duidelijk geïdentificeerd kunnen worden gedurende hun wettelijke of reglementaire bewaartermijn. De gegevens dienen op een passende wijze vernietigd te kunnen worden na afloop van die termijn voor zover ze niet meer nodig zijn voor de organisatie.

In sommige gevallen is de bewaartermijn voor informatie en het type informatie dat bewaard moet worden geregeld in de nationale wetgeving of voorschriften. Deze beveiligingseis is tevens essentieel bij reconstructievraagstukken in relatie tot opgetreden issues/incidenten.

Centraliseren van loggingsgegevens:

Vaak worden verschillende logging-mechanismen naast elkaar gebruikt. Zo ondersteunt het ene systeem alleen logging op basis van SYSLOG, maakt een ander systeem alleen lokaal logbestanden aan en stelt weer een ander systeem alleen informatie beschikbaar via SNMP. Al deze verschillende logging-mechanismen zorgen ervoor dat logging versnipperd raakt en een organisatie het overzicht over alle gebeurtenissen gemakkelijk kwijtraakt. Om aanvallen efficiënt te kunnen detecteren is het van belang deze logging op één centraal punt weer bijeen te brengen. Beperk het aantal logging-mechanismen zoveel mogelijk.

Door de logging op een centraal punt bijeen te brengen en deze intelligent te combineren en te filteren ontstaat een heldere blik op alle informatie vanuit de verschillende componenten uit de infrastructuur.

In een centrale log-database komt de loginformatie uit verschillende onderdelen samen. Denk hierbij aan de volgende typen informatie: logging op het niveau van netwerk-, platform- en (web)applicatiebeveiliging, logging op het niveau van identiteit- en autorisatiebeheer en logging op het niveau van vertrouwelijkheid en onweerlegbaarheid.

Synchroniseren van systeemklokken:

Om gebeurtenissen uit verschillende componenten te correleren worden de timestamps van deze gebeurtenissen gebruikt. Deze timestamps zijn afhankelijk van de juiste instelling van de tijd op de desbetreffende componenten. Met behulp van het Network Time Protocol (NTP) kan worden bereikt dat de tijd op alle servers en andere componenten overeen komt.

Alternatieven voor beschikbaarheid van registraties(logbestanden):

Het gebruik van centrale logging-mechanismen brengt een belangrijk vraagstuk met zich mee: wat doen we op het moment dat een centrale logging-mechanisme uitvalt? Op het moment dat een component zijn logging niet meer kwijt kan, bestaat de kans dat deze logging verloren gaat. Dit zou kunnen betekenen dat componenten aanvallen van kwaadwillenden niet meer registreren, of dat transacties niet meer onweerlegbaar zijn. Bepaal daarom op voorhand welke actie een component moet nemen op het moment dat een centrale logging-mechanisme niet meer beschikbaar is. Er bestaan op dit gebied grofweg de volgende mogelijke acties:

- De component normaal laten functioneren terwijl deze de logging niet kan opslaan. Dit betekent dat logging verloren gaat.
- De component normaal laten functioneren en de logging lokaal laten opslaan. Veel componenten beschikken over een lokaal mechanisme om logging tijdelijk op te slaan. Op het moment dat het centrale logging-mechanisme weer beschikbaar komt, sluist de component de verzamelde logging alsnog door. Dit voorkomt dat de component niet meer beschikbaar is en voorkomt tevens dat logging verloren gaat. Dit is echter wel een tijdelijke oplossing. Op het moment dat de lokale opslag vol loopt, moet opnieuw besloten worden wat de component hierna doet (blijven functioneren - zie bovenstaande optie - of stoppen met functioneren - zie volgende optie).
- De component acuut laten stoppen met functioneren. Dit betekent dat gebruikers hoogst waarschijnlijk niet meer kunnen doorwerken. Dit voorkomt wel dat aanvallen op de component onopgemerkt blijven doordat de component ze niet meer logt.

Vanuit het oogpunt van beveiliging en beschikbaarheid heeft het de voorkeur om – zodra een centrale logging mechanisme uitvalt - componenten eerst lokaal gebeurtenissen te laten opslaan om vervolgens de component te laten stoppen met functioneren zodra deze opslag vol is. Bij de selectie van een nieuw beveiligingscomponent is het daarom zaak goed te evalueren of deze voldoet aan de eisen op het gebied van logging en tijdelijke opslag van logging.

Conformiteitsindicatoren

/01 registratie en detectie

De registratiefunctie houdt verband met het vastleggen van menselijke- en systeemgerichte acties en informatie over gebeurtenissen voor controle- en analysedoeleinden.

De detectiefunctie houdt verband met het (selectief) waarnemen van signalen voor het opsporen en blootleggen van ongewenste gebeurtenissen in de webapplicatie omgeving.

SSD-30 Applicatie logging	
	<i>indicatoren</i>
<u>/01</u>	<u>registratie en detectie</u>
/01.01	De te registreren acties zijn gecentraliseerd.
/01.02	Er is bepaald welke gebeurtenissen en/of beheeractiviteiten aan de webapplicatie vastgelegd moeten worden.
/01.03	In de webapplicatie-infrastructuur zijn detectiesystemen actief voor het detecteren van aanvallen.

Toelichting

/01.01 De inrichting is gebaseerd op een vastgesteld inrichtingsdocument / ontwerp waarin is vastgelegd welke uitgangspunten gelden voor logging.

/01.01 Er is een plan beschikbaar met daarin activiteiten die worden uitgevoerd (wie, wat en wanneer) indien log records op kwaadwillend misbruik duiden, geïmplementeerde maatregelen niet aan de gestelde eisen en/of verwachtingen hebben voldaan of te kortkomingen hebben opgeleverd.

/01.02 Er regelgeving over vast te leggen gebeurtenissen dan wel handelingen. De regels hierover worden onderhouden. Voorbeelden van vast te leggen gegevens zijn:

- verdachte gebeurtenissen en wijzigingen aan de webapplicatie,
- succesvolle en geweigerde toegangspogingen,
- (on)geoorloofde activiteiten door functionarissen.

/01.02 Eventueel worden ten behoeve van leesbaarheid en efficiency filters gebruikt.

/01.03 In de ontwerp- casu quo configuratiedocumentatie is vastgelegd waar en hoe IDS'en worden ingezet.

/01.03 De zakelijke behoeften, overwegingen en maatregelen. Rapportage van de risicoanalyse waarop de beslissing is gebaseerd.

/02 efficiënt en effectief

In het webapplicatiedomein worden vaak verschillende loggingsmechanismen (registraties)naast elkaar gebruikt. Om de loggingsinformatie niet omslachtig, met beperkte inspanning en doeltreffend te kunnen analyseren is het van belang deze te centraliseren.

SSD-30 Applicatie logging	
	<i>indicatoren</i>
<u>/02</u>	<u>efficiënt en effectief</u>
/02.01	De systemen zijn zodanig geconfigureerd dat interne systeemklokken automatisch gesynchroniseerd worden.

Toelichting

/02.01 Systemen gebruiken interne systeemklokken voor "timestamps" bij het vastleggen van loggegevens. De ontwerp- casu quo configuratiedocumentatie is vastgelegd hoe het synchroniseren van de systeemklokken is geconfigureerd.

/03 beveiligd

Beveiligde inrichting is gerelateerd aan maatregelen met betrekking tot juiste en tijdige werking en beschikbaarheid van de registratie en detectiefunctie, beveiliging van logbestanden tegen manipulaties, alternatieve paden bij uitval, veiligstellen van log-bestanden. Juiste en tijdige werking houdt verband met bijtijds activeren van de registratiefunctie. Hiervoor is het van belang dat systeemklokken van systemen gesynchroniseerd zijn.

SSD-30 Applicatie logging	
	<i>indicatoren</i>
<u>/03</u>	<u>beveiligd</u>
/03.01	Er is vooraf bepaald wat te doen bij het uitvallen van loggingmechanismen (alternatieve paden).
/03.02	De (online of offline) bewaartermijn voor logging is vastgesteld en komt tot uitdrukking in de configuratie-instellingen van de systemen.
/03.03	De loggegevens zijn beveiligd tegen achteraf wijzigen/verwijderen.

Toelichting

- /03.01 Er is een procedurebeschrijving van het logging mechanisme en bewijsvoering dat het mechanisme van alternatieve actie bij uitvallen loggingsmechanismen ook daadwerkelijk werkt.
- /03.01 Er wordt aangegeven welke actie een component moet nemen op het moment dat het centrale loggingsmechanisme niet meer beschikbaar is.
- /03.02 Er zijn bewaartermijnen vastgesteld voor de loginformatie. Dit zal moeten blijken uit de configuratie-instellingen.
- /03.03 Ontwerpdokumentatie, configuratie-instellingen en autorisatieprofielen geven aan hoe logfiles beschermd zijn tegen achteraf of ongeautoriseerd wijzigen/verwijderen.

3.32. SSD-31: Standaard stack

SSD-31 Standaard stack					
<i>Criterium (wie en wat)</i>	De (web-)applicatie(-omgeving) maakt gebruik van <u>systeemcomponenten en services</u> die onderdeel zijn van een formeel gespecificeerde stack.				
<i>Doelstelling (waarom)</i>	Compliance met dit criterium zorgt ervoor dat integrale beveiliging mogelijk is om adequate bescherming te bieden tegen de actuele bedreigingen.				
<i>Risico</i>	Het gebruik van onbekende componenten of services kan leiden tot nieuwe en onbekende zwakheden en risico's.				
<i>Referentie</i>	NCSC	NIST	ISO27002		
			6.1.4 10.3.2 15.1.1		

Toelichting

Stack is een manier om de verschillende lagen van componenten en services te beschrijven die worden gebruikt bij een softwarematige oplossing. Een met succes gehanteerde standaard stack wordt toegepast, omdat de componenten en services zich hebben bewezen in de praktijk op elkaar aan te sluiten en waarbij zwakheden bekend zijn en daardoor gedicht zijn. Stacks waarmee geen ervaring is kunnen nieuwe en onbekende zwakheden bevatten. Zwakheden kunnen zich in technische vorm voordoen, maar kunnen zich ook voordoen doordat bijvoorbeeld geen end-of-life strategie is bepaald, waardoor zich in de toekomst bedrijfscontinuïteitsrisico's kunnen voordoen. Het is daarom van belang dat ook de beheerafspraken met de hostingpartij onderdeel uitmaken van een keuze voor een stack. In de praktijk wordt de standaard stack, in afstemming met de opdrachtgever, door de hostingpartij vastgelegd en beheerd.

De stack bestaat uit hardwarematige en softwarematige componenten, inclusief (koppelingen met) beveiligingsvoorzieningen, zoals identiteit- en accessvoorzieningen.

Om zwakheden in oudere versies te voorkomen gelden op een stack toegestane onderhoudsniveaus voor de verschillende componenten in de stack.

Conformiteitsindicatoren

/01 systeemcomponenten en services

SSD-31 Standaard stack	
	<i>indicatoren</i>
/01	<u>systeemcomponenten en services</u>
/01.01	Voor het vastleggen en het beheren van de standaard stack is één partij aangewezen.
/01.02	De softwareleverancier maakt uitsluitend gebruik van systeemcomponenten en services die zijn opgenomen in de standaard stack.
/01.03	De softwareleverancier onderbouwt en documenteert afwijkingen op de stack en laat dergelijke afwijkingen goedkeuren door de hostingpartij.
/01.04	De hostingpartij maakt gebruik van de laatste beveiligingsmaatregelen, beleid en procedures voor de systeemcomponenten en services die in gebruik zijn.

Toelichting

/01.01 In de praktijk wordt de standaard stack, in afstemming met de opdrachtgever, door de hostingpartij vastgelegd en beheerd en is deze gepubliceerd.

- /01.01 De partij houdt rekening met de inbreng van alle betrokken partijen, zodat de stack een stabiele en veilige omgeving vormt. De standaard stack kan daartoe standaard zijn per standaard applicatieomgeving, waardoor bij de hostingpartij meer dan één standaard stack bestaat.
- /01.02 De softwareleverancier mag voor de ontwikkeling van een (web)applicatie alleen gebruik maken van systeemcomponenten, zoals platformen en middleware, die formeel zijn goedgekeurd en adequaat worden ondersteund.
- /01.04 Verouderde beveiligingsstandaarden bieden namelijk verminderde bescherming tegen bedreigingen.

Bijlage: De SIVA-methode voor het opstellen van beveiligingseisen

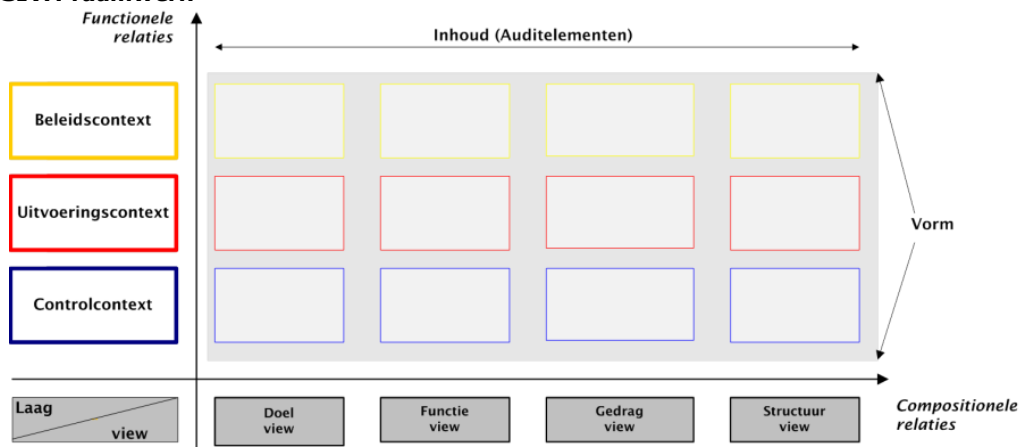
Het raamwerk

Deze tweede versie van de SSD-normen is gebaseerd op de SIVA-methode [Tewarie, 2014]. De SIVA-methode hanteert een raamwerk dat is onderverdeeld in domeinen, met daarbij een separaat algemeen gedeelte dat beleidsaspecten en beheersingsaspecten bevat. Dit raamwerk bevat specifieke lagen en kolommen om een verband tussen de beveiligingsmaatregelen voor de applicatie-omgeving weer te geven.

Het voordeel van het hanteren van deze methode is dat duidelijk wordt aangegeven wie wat binnen een norm moet doen, terwijl de SIVA-methode tevens goed laat zien wat de context is van de normen.

De normen richten zich hier op het object "software" (de applicatie) en stellen dus bijvoorbeeld geen eisen aan het voortbrengingsproces van een applicatie of het beleid.

Het SIVA-raamwerk



Het SIVA-raamwerk bestaat uit vier componenten, te weten *Structuur*, *Inhoud*, *Vorm* en *Analysevolgorde*. Deze componenten zijn hulpmiddelen en worden als volgt omschreven:

- Structuur
De omgeving, in dit geval de applicatie-omgeving, is verdeeld in een aantal domeinen. Dit bevordert de volledigheid, relevantie, duidelijkheid en samenhang van de aspecten die worden onderzocht.
- Inhoud
Vanuit verschillende invalshoeken worden per domein basiselementen geïdentificeerd.
- Vorm
Per element worden de beveiligingseisen geformuleerd door middel van een formuleringvoorschrift (template).
- Analysevolgorde
Een iteratief analyseproces van de bij structuur genoemde lagen.

Analysevolgorde gaat over het proces om te komen tot de normen en is hier niet relevant, omdat we zoveel mogelijk uitgaan van bestaande normenkaders. Dit geldt ook voor de structuur. De *structuur* is opgebouwd uit drie onderkende contexten: beleids-, uitvoerings- en control-context. Omdat de SSD-normen zich concentreren op de eisen ten aanzien van de implementatie van applicaties en dus op de uitvoeringscontext worden geen eisen gesteld vanuit de beleidscontext en de control-context. Bij het opstellen van de SSD-normen zijn wel de *Inhoud* en de *Vorm* als hulpmiddel gehanteerd.

Inhoud

De component inhoud wordt in de SIVA-methode bereikt door middel van vier invalshoeken: doel, functie, gedrag en structuur (DFGS). Vanuit elke DFGS-invalshoek wordt een specifieke verzameling basiselementen (objecten) geïdentificeerd. De invalshoeken houden het volgende in:

- **doel** – het waarom-aspect
De bestaansreden van een organisatie.
Voorbeelden: organisatie, visie, doelstellingen, wetten en beleid, stakeholders en middelen
- **functie** – het wat-aspect
De organisatorische - en technologische elementen die de intenties van de organisatie moeten realiseren.
Voorbeelden: organisatorische - en technische functies, processen, taken en taakvereisten
- **gedrag** – het hoe-aspect (gedragsaspect)
De menselijke en technische resources en eigenschappen van de technische resources die de organisatorische - en technische functies moeten vormgeven.
Voorbeelden: actor, object, interactie, toestand, eigenschap en historie
- **structuur** – het hoe-aspect (vormaspect)
De manier waarop een organisatorische - en personele structuur is vormgegeven.
Voorbeelden: business-organisatiestructuur, business- architectuur, IT-architectuur en business-IT-alignment

De relaties tussen de objecten vanuit de DFGS-invalshoeken kunnen als volgt worden gelezen: de elementen uit de *doel-invalshoek* reguleren en/of worden bereikt door elementen uit de *functie-invalshoek*. De elementen uit de *functie-invalshoek* gebruiken of realiseren de elementen uit de *gedrag-invalshoek* die op hun beurt worden vormgegeven door elementen uit de *structuur-invalshoek*.

Voor de SSD-normen is de volgende checklist gebruikt om te bepalen of de normen de aandachtsgedebieden (basiselementen) voor (web)applicaties afdekken en daarmee de risicogebieden afdekken.

Uitvoeringscontext		
<i>(Web)applicatie</i>		
Invalshoeken	Basiselementen	Geïdentificeerde elementen
doel	beleid	- <i>Operationeel beleid</i>
	middelen	- <i>(Web)applicatiemiddelen</i>
functie	proces	- <i>(Web)applicatierechten</i>
gedrag	object	- <i>(Web)applicatie</i>
	protocol (invoer)	- <i>(Web)applicatie-invoer</i>
	protocol (uitvoer)	- <i>(Web)applicatie-uitvoer</i>
	koppeling	- <i>Koppeling (Web)applicatie front-end</i>
	verbindingstijd	- <i>(Web)applicatiesessie</i>
structuur	architectuur	- <i>(Web)applicatiearchitectuur (scheiding)</i>

Kijkende naar de beveiligingseisen in hoofdstuk 3 blijkt dat de normen de te identificeren elementen afdekken. Dat betekent overigens niet dat de risicogebieden compleet zijn afgedekt. De SSD-normen zijn opgezet om grip te krijgen op de veiligheid van applicaties en niet om een complete lijst op te leveren, zoals met de volledig doorgevoerde SIVA-methode wordt beoogd.

Vorm

De *Vorm*-component van het SIVA-raamwerk geeft een formule (syntax) weer voor de normen:

Predicaat (object-1 , object-2 , Object-3)
Actieype (Wie , Wat , Waarom)

In deze formule komen vier elementen voor. Het eerste element is de handeling (actietype). Het tweede en derde element zijn de objecten welke de handeling uitvoeren (actor, wie) respectievelijk ondergaan (wat). Het vierde element vertegenwoordigt het resultaat of doel van de handeling. De onderstaande tabel verduidelijkt deze elementen.

<i>Wie</i>	<i>Betrokken Actor</i>
<i>Wat</i>	<i>Hierbij worden zaken uitgedrukt:</i> • <i>die gedaan moeten worden om doelen te bereiken/te realiseren/te controleren/te bewaken en verantwoording te kunnen afleggen,</i> • <i>wat iemand moet doen of</i> • <i>wat een technische functie/apparaat doet.</i>
<i>Actietype</i>	<i>Specifieke werkwoorden gerelateerd aan het wat-aspect en aan een bepaalde laag.</i>

Gebruikte template

De elementen "wat" en "waarom" zijn separaat vermeld. In de uitdrukking van de normen worden trefwoorden gebruikt die als indicatoren dienst doen. Per indicator worden indicatoren benoemd. De indicatoren geven inzicht in hoe aan de beveiligingseis kan worden voldaan. De trefwoorden in de formulering van de beveiligingseisen zorgen ervoor dat er slechts relevante criteria per beveiligingseis worden benoemd.

Bij de uitwerking van de normen is gebruik gemaakt van een template, waarbij het element "wie" veelal achterwege is gelaten. Dit element komt wel terug in de indicatoren van de beveiligingseisen, zodat duidelijk wordt wie in de keten functioneel beheer, applicatiebeheer en technisch beheer verantwoordelijk is voor de realisatie voor dat deel van de norm. Deze keten is in hoofdstuk 2.2 uitgelegd.

Het gebruikte template voor de normen is:

SSD-nr Onderwerp van de norm					
<i>Criterium (wie en wat)</i>	Wat (xxxxxx) <werkwoord> xxxxx <u>trefwoorden</u> xxxxx				
<i>Doelstelling (waarom)</i>	De reden waarom de norm gehanteerd wordt.				
<i>Risico</i>	Het risico dat de aanleiding vormt om de norm te hanteren.				
<i>Referentie</i>	Bron 1	Bron 2	...		

Ieder trefwoord vormt een indicator, waaraan voldaan moet worden. Om die reden is ieder trefwoord uitgewerkt. Het gebruikte template voor trefwoorden is:

SSD-nr Onderwerp van de norm	
	<i>indicatoren</i>
/01	<u>trefwoord</u>
/01.01	indicator 1.1
/01.01	indicator 1.2
...	...

De trefwoorden (/01, /02, etc) en de invalshoeken zijn genummerd (/01.01, /01.02, etc), zodat in de toelichting hieraan gerefereerd kan worden.

Bij de referenties wordt, voor zover relevant, aangegeven waar in de volgende standaarden en richtlijnen additionele informatie is te vinden:

- **NCSC:**
'ICT Beveiligingsrichtlijnen voor webapplicaties', deel 1 en 2, NCSC, januari 2012;
- **NIST:**
Special Publication SP800-53 'Recommended Security Controls for Federal Information Systems', NIST;
- **ISO27002:**
NEN-ISO/IEC 27002 'Code voor informatiebeveiliging', 2005. Voor de relevante referenties kan ook gebruik worden gemaakt van de 'Baseline Informatiebeveiliging Rijksdienst' (BIR).