centrum informatiebeveiliging en privacybescherming

Grip op Secure Software Development (SSD) Beveiligingseisen voor (web)applicaties

20 Juli 2020 [v3.0]

Titel	Grip op Secure Software Development (SSD) - Beveiligingseisen voor (web)applicaties -
Opdrachtgever	Centrum Informatiebeveiliging en Privacybescherming – Ad Reuijl
Status	Versie 3.0 Becommentarieerde praktijk
Auteurs	CIP: Marcel Koers SIG: Rob van der Veer
Reviewers	Arjen Schijf (gem. Amsterdam), Cor Rosielle (UWV), Erica Welling (Dictu/CGI), Jan Blaakmeer (SVB), Jeroen de Meijer (Wigo4it), Martijn Korse (Bitnesswise), Michael Kuipers (Centric), Rob Roukens (UWV), Robert Mast (UWV), Ronald Paans (Noordbeek), Sanne Kuijpers (CapGemini), Sebastiaan Paans (Noordbeek), Sven van der Horst (SVB), Sylvan Rigal (SIG), Tim van Loon (Scourse), Erik Poll (RU), Jaap de Vries (UWV), Michiel Versluis (Ministerie van Defensie).
Bijdrage(n) van	Carolien Glasbergen
Datum	20-7-2020
Filenaam	20200720 SSD normen v3.0.docx

Considerans

CIP-producten steunen op kennis van professionals uit verschillende organisaties actief in het CIP-netwerk, zowel uit de overheid als de markt.

Opmerkingen en aanvullingen kun je melden op www.cip-overheid.nl/contact

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 worden gegeven aan beveiliging: bij nieuwe IT-projecten en bij veranderingen aan 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, 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', 'softwaremaker' 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. Het ISO25010-gebaseerde referentiemodel is beschikbaar gesteld door Software Improvement Group voor het structureren van de normen.

Amsterdam, 20 juli 2020.

Inhoudsopgave

1 Inleiding

Dit document beschrijft voor organisaties de belangrijkste beveiligingseisen die van toepassing zijn bij de ontwikkeling en aanschaf van applicaties. Samen met het document "Grip op SSD – de methode", waarin de aanpak "hoe grip erop te krijgen" is beschreven, wordt met de eisen de opdrachtgever een oplossing geboden om tot veilige software te komen. De eisen beperken zich daarvoor 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.

Verwijzingen naar internetpagina's zijn klikbaar in PDF versies van dit document. Voor afgedrukte versies kan in plaats van klikken worden gezocht op de zoektermen bij de link.

1.1 Scope: web- en backend-applicaties

Wanneer dit document spreekt over een applicatie gaat het om een applicatie die bereikbaar is via een webbrowser of via een andere cliënt (bijvoorbeeld een mobiele of desktop applicatie). Kenmerkend is HTTP als communicatie-protocol en de versleutelde variant HTTPS. Applicaties kunnen ook opengesteld worden via een vooraf afgesproken interface (API). Voor mobiele applicaties zijn aparte SSD-mobile normen beschikbaar. Deze publicatie is samen met andere CIP publicaties (zoals de SSD-Methode) te vinden op www.cip-overheid.nl

Per eis is weergegeven voor wat voor soort software deze toepasselijk is. Veel van de eisen zijn van toepassing voor software in het algemeen.

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 bewaakt 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 exacte maatregelen zijn daarom op zichzelf geen harde vereiste. Wel moeten 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 softwaremaker: een 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.

Het uitgangspunt is, dat de hostingpartij zorgt voor een omgeving die "*secure bij default*" is. Dat betekent dat de installatie van *operating system*, *services*, *security software* en/of *appliances*, etc., plaatsvindt volgens de functionele en beveiligingsinstructies van de producenten van die hard- en software. De hostingprovider zorgt er eveneens voor dat patches in de omgeving worden geïnstalleerd. Om te waarborgen dat de applicatie naar behoren functioneert en daarbij zo veilig mogelijk is, legt de softwaremaker in de *configuratiebeschrijving* uit wat nodig is om de applicatie goed en veilig te laten functioneren. De softwaremaker beschrijft welke poorten, protocollen, connecties, diensten, authorisaties etc., door de omgeving ondersteund moeten worden. Ook legt de softwaremaker uit hoe de applicatie gehardend moet worden, zonder dat de functionaliteit van de toepassing in gevaar komt.

2 Uitleg van de opzet van de beveiligingseisen

2.1 Gebruikte template

De SSD-normen zijn 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					
Criterium (wie en wat)	Wat (xxxxxx) <werkwoord> xxxxx trefwoorden xxxxx				
Doelstelling (waarom)	De reden waarom de norm gehanteerd wordt.				
Risico	Het risico dat de aanleiding vormt om de norm te hanteren.				
Referentie	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	
	indicatoren
/01	trefwoord
/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.

Op meerdere plekken zijn details/indicatoren van een norm vervallen vergeleken met Grip op SSD 2.0. In dat geval is er gekozen om de nummering intact te houden ten behoeve van traceerbaarheid naar een eerdere versie. Als bijvoorbeeld indicatoren zijn komen te vervallen, ontstaat er een 'gat' in de nummering.

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

- **NCSC**: 'ICT Beveiligingsrichtlijnen voor applicaties', deel 1 en 2, NCSC, 2015;
- **NIST**: Special Publication SP800-53 'Recommended Security Controls for Federal Information Systems', NIST;
- **ISO27002:** ISO/IEC 27002 'Code voor informatiebeveiliging', 2013. Voor relevante referenties kan ook gebruik worden gemaakt van de 'Baseline Informatiebeveiliging Overheid' (BIO).

Naast deze verwijzingen is de ASVS standaard van OWASP een zeer compleet overzicht van beveiligingseisen. Zie: OWASP ASVS.

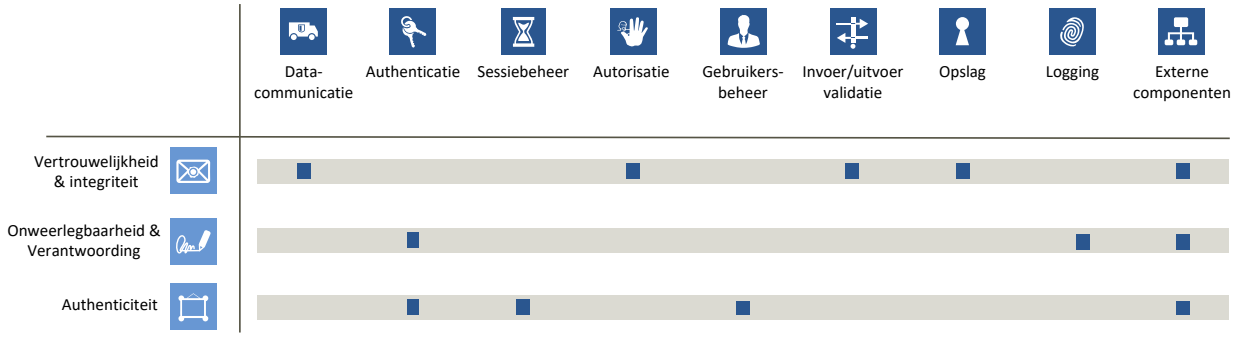
3 Structuur beveiligingseisen

ISO/IEC 25010:2011 is dé standaard voor softwarekwaliteit en definieert beveiliging op een technologie-onafhankelijke manier. Het is de basis van het SIG beveiligingsmodel[1] volgens welke de SSD beveiligingseisen zijn gestructureerd. Dit model harmoniseert bestaande standaarden in een uniform en logisch overzicht van verantwoordelijkheden, zodat duidelijk is wat geregeld moet worden bij het maken van afspraken, bij implementatie en bij toetsing.

In ISO/IEC 25010 bestaat beveiliging uit vijf kenmerken:

- **Vertrouwelijkheid**: gegevens zijn alleen toegankelijk voor geautoriseerden.
- **Integriteit**: aanpassing van computerprogramma's of gegevens alleen voor geautoriseerden.
- **Onweerlegbaarheid**: er kan worden bewezen dat acties of gebeurtenissen hebben plaatsgevonden.
- **Verantwoording**: acties van een entiteit kunnen uniek worden getraceerd.
- **Authenticiteit**: de identiteit van een onderwerp of bron kan worden aangetoond als degene die wordt geclaimd.

De vijf kenmerken zijn door middel van onderstaand model elk gerelateerd aan een aantal verantwoordelijkheden voor software-ontwikkeling en -beheer.



Figuur 1: SIG ISO25010 beveiligingsmodel met als rijen de kenmerken en in de kolommen de bijbehorende toetsbare verantwoordelijkheden.

Merk op dat 'beschikbaarheid' (availability) in twee delen is gesplitst in ISO/IEC 25010: 1) een deel dat valt onder de softwarekwaliteit 'betrouwbaarheid' (door maatregelen voornamelijk buiten de software: anti-ddos, dubbele uitvoering, etc.), en 2) een deel dat valt onder 'Integriteit' (zoals verificatie van invoer en uitvoer om 'denial of service' te voorkomen).

Hoe de verantwoordelijkheden een rol spelen in veilige software

Een applicatie zorgt dat functies en gegevens alleen toegankelijk zijn voor diegenen die daarvoor goedkeuring hebben op de manier dat de applicatie bedoeld is. Voordat het systeem aan de vragen van gebruikers voldoet, voert het eerst **Toegangscontrole** uit.

[1] Oorspronkelijke wetenschappelijke publicatie: https://zenodo.org/record/3592336 - .XgH_kNZKjUI, "A Practical Model For Rating Software Security".

Dit bestaat uit:

- **Authenticatie**: zekerheid dat een identificatie deugt en
- **Autorisatie**: controle per actie dat die geoorloofd is voor die specifieke gebruiker.
- **Sessiebeheer:** sessies voorkomen dat een gebruiker zich voor elke actie opnieuw moet identificeren. Sessies vertegenwoordigen de identiteit van de gebruiker en dit moet dan ook deugdelijk verlopen.

In elk systeem vinden logische stappen plaats van gegevens die in- en uitvloeien. In die logische verwerking hoort het systeem alle invoer en uitvoer te controleren: dit is het domein van **Invoer- en uitvoer validatie**. Om uiteindelijk deugdelijke werking te kunnen aantonen (tijdens en achteraf), is **Logging** nodig**.**

Voor en na verwerking van gegevens worden die gegevens getransporteerd en opgeslagen. De communicatie tussen gebruiker en systeem en met andere systemen hoort beschermd te zijn zodat er niet met de communicatie geknoeid kan worden. Het systeem forceert daarvoor veilige **Datacommunicatie**. Ook (tijdelijk) bewaarde gegevens horen weerstand te bieden tegen onderschepping of wijziging met veilige **Dataopslag**.

Rondom het systeem bestaat een 'operatieschil', grofweg te scheiden in techniek (**Infrastructuur**) en proces (**Gebruikersbeheer**, **Externe componenten** etc.).

Tot slot is er een aantal **architectuurprincipes** die bijdragen aan een veilige applicatie.

De structuur van de verantwoordelijkheden helpt om te bepalen op welk moment een bepaalde eis van toepassing is; afhankelijk van het type ontwikkelwerk of testactiviteit. Op deze manier hoeven alleen de van toepassing zijnde SSD normen per moment worden meegenomen. Hiervoor zijn per SSD-eis zogenaamde ***triggers*** gespecificeerd: een omschrijving van de situatie waarin de eis van toepassing is.

4 Beveiligingseisen voor de (web)applicatie

4.1 Datacommunicatie


Om getransporteerde gegevens te beschermen, moeten deze worden beveiligd met een voldoende sterke beveiligingsmethode.

4.1.1 *SSD-4: Veilige communicatie*

Voor: alle software met datacommunicatie waarop kan worden ingebroken.
Trigger: versturen van te beschermen gegevens over een (evt. gevirtualiseerd) netwerk, d.w.z. logisch geïsoleerd (VLAN) of getunneld (bijv. over VPN).

SSD-4 Veilige communicatie					
Criterium (wie en wat)	De applicatie past versleuteling toe op de communicatie van gegevens die passend is bij het classificatieniveau van de gegevens, zowel over interne als externe netwerken en controleert hierop. Van te beschermen gegevens worden alleen de noodzakelijke gecommuniceerd.				
Doelstelling (waarom)	Door versleuteling of door weglating van gecommuniceerde gegevens worden deze afgeschermd van ongeautoriseerde toegang.				
Risico	Ongeautoriseerde inzage of wijziging van gegevens via het communicatiekanaal, wat indirect kan leiden tot toegang tot overige gegevens en systeemfuncties, als toegang-verlenende gegevens worden ingezien.				
Referentie	NCSC	NIST	ISO27002		
	B.04 U/PW.05 U/WA.05	SC-8 SC-9 SC-13	10.1.1 10.1.2 18.1.5		

Toelichting
Versleuteling van communicatie beschermt vertrouwelijkheid en de integriteit van de gegevens die worden getransporteerd.

Conformiteitsindicatoren
/01 versleuteling

SSD-4 Veilige communicatie	
	indicatoren
/01	Versleuteling
/01.01	Er worden alleen protocollen en cryptografische technieken gebruikt die als veilig worden bestempeld volgens industrieel geaccepteerde standaarden[2].
/01.02	De applicatie of platform waarop de applicatie draait zorgt voor versleuteling van communicatie tussen applicatieserver en webserver en tussen applicatie en database. De webserver forceert versleuteling tussen webserver en cliënt.

[2] Zie bijvoorbeeld OWASP cheatsheet Transport Layer Protection.

SSD-4 Veilige communicatie	
/01.05	De softwaremaker stelt de software beschikbaar met bijbehorende configuratierichtlijnen.
/01.06	De hostingprovider draagt zorg voor de configuratie conform de configuratierichtlijnen van de opdrachtgever.

Toelichting

/01.01 Van de gebruikte versies zijn geen zwakheden (tenzij deze aantoonbaar geen bedreiging vormen) bekend binnen het vakgebied. Gangbaar hiervoor zijn de eisen van het NCSC – die periodiek worden bijgewerkt. Het NCSC publiceert ICT-beveiligingsrichtlijnen voor Transport Layer Security (TLS), zie https://www.ncsc.nl/onderwerpen/verbindingsbeveiliging. Ten tijde van schrijven is dit *"ICT-beveiligingsrichtlijnen voor-TLS-v2.0"*

/01.01 TLS is het meest gebruikelijk voor versleuteling van (web)applicaties, maar dit is niet per se noodzakelijk. Andere gebruikelijke technologieën zijn een SSH-tunnel of IPSec, maar deze worden algemeen beschouwd als moeilijker om te beheren.

/01.02 Het is goed gebruik om de interne datacommunicatie ook te beveiligen aangezien het 'interne netwerk' ook beveiligingsrisico's kent.

/01.05 Configuratierichtlijnen van de softwaremaker dienen gevolgd te worden op basis van pas-toe-of-leg-uit. Daarnaast kunnen algemeen geaccepteerde richtlijnen gebruikt worden zoals bijvoorbeeld onderhouden door het Center of Internet Security.

/01.06 De opdrachtgever is verantwoordelijk voor het stellen van de juiste richtlijnen aan derden zoals de hostingprovider. Volgend uit /01.05 moeten in deze richtlijnen de richtlijnen van de softwaremaker worden meegenomen.

/02 passend bij het classificatieniveau

SD-4 Veilige communicatie	
	indicatoren
/02	passend bij het classificatieniveau
/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 De classificatie van gegevens is bepaald door de gegevenseigenaar door bijvoorbeeld het uitvoeren van een BIA. De classificatie geeft duidelijkheid over de mate waarin gegevens beschermd moeten worden; in dit geval door versleuteling. Als er geen BIA of GEB is uitgevoerd, of waar het een gegeven betreft voor het laten werken van de techniek een risicoanalyse, dan geldt per default dat versleuteling conform de baseline plaatsvindt.

/02.02 Niet voor ieder gegeven zal in de praktijk bekend zijn wat de classificatie is, zeker als het een technisch gegeven betreft. Er dient dan per default een veilige classificatie gekozen te worden of een classificatie te worden bepaald door de gegevenseigenaar.

/03 controleert

De applicatie dient waar mogelijk te controleren of de vereiste maatregelen voor communicatieversleuteling daadwerkelijk worden toegepast.

SD-4 Veilige communicatie	
	indicatoren
/03	Controleert
/03.01	De applicatie zorgt waar mogelijk voor verificatie dat het certificaat is ondertekend door een vertrouwde Certificate Authority.
/03.02	De applicatie verifieert waar mogelijk dat het certificaat een valide geldigheidsduur heeft.
/03.03	De applicatie verifieert waar mogelijk dat een certificaat nog geldig is en niet is ingetrokken.
/03.04	De versleutelde communicatie wordt zo geconfigureerd, dat er geen terugval naar (niet of onvoldoende) versleutelde communicatie kan zijn.

Toelichting

Certificaat-verificatie voor webapplicaties wordt aan de eindgebruikerskant door de browser gedaan en valt buiten de controle van de applicatie. Verificatie met applicatiecomponenten en achterliggende systemen valt wel onder de controle en wordt geboden door de meeste bibliotheken en raamwerken (bv. OpenSSL).

/03.02 Certificaten waarvan einddatum is verstreken of begin datum nog niet is aangebroken gedurende de validatie worden afgewezen.

/03.03 Een certificaat kan na afgifte worden ingetrokken. Deze komt dan op een revocation lijst te staan en mag niet worden gebruikt ook al is de geldigheidsduur nog niet verstreken. Revocation kan bijvoorbeeld worden bepaald via een gepubliceerde Certificate Revocation List (CRL) of door gebruik te maken van OCSP.

/04 noodzakelijke

SSD-4 Veilige communicatie	
	indicatoren
/04	Noodzakelijke
/04.01	Communicatie van te beschermen gegevens wordt tot een minimum beperkt om de impact van misbruik te verkleinen.

Handreiking

- NCSC beveiligingsrichtlijnen TLS
- OWASP cheat sheets:
  - Key Management Cheat Sheet
  - Pinning
  - TLS Cipher String
  - Transport Layer Protection
  - Transaction Authorization Cheat Sheet
- Voorbeeld overzicht crypto libraries "cryptobook" op website Svetlin Nakov
- OWASP Web Security Testing Guide (WSTG):
  - Testing for weak cryptography

4.2 Opslag


Toegang tot een gegevensopslag moet worden voorkomen of beperkt, waarbij de aanname is dat gegevensopslag ook fysiek wordt beschermd. Gevoelige gegevens moeten worden beschermd tegen ongeoorloofde toegang door deze te versleutelen. Als toegang tot de oorspronkelijke gegevens niet vereist is (zoals wachtwoorden), moet leesbaarheid onmogelijk worden gemaakt door een hash correct toe te passen op de gegevens. In dat geval wordt een mathematisch derivaat (de hash) vergeleken in plaats van het origineel. Zie ook:

- Toegangsbeheer (want het consistent afdwingen van toegangscontrole is een voorwaarde voor sterke autorisatie).
- Sessiebeheer (want gebruikersrechten worden typisch binnen een sessie toegepast en daarmee afhankelijk van de sessie-maatregelen).
- Gebruikersbeheer (want systemen moeten bij toepassing van autorisaties aannemen dat gebruikersrechten juist geadministreerd zijn).
- Invoer/uitvoer validatie (voor bescherming tegen injectie-aanvallen op opslag)

4.2.1 *SSD-2: Veilige gegevensopslag*

Voor: alle software.
Trigger: opslaan van te beschermen gegevens.

SSD-2 Veilige gegevensopslag					
Criterium (wie en wat)	Te beschermen gegevens worden veilig opgeslagen in databases of bestanden, waarbij zeer gevoelige gegevens worden versleuteld. Opslag vindt alleen plaats als noodzakelijk.				
Doelstelling (waarom)	Toegang tot opgeslagen gegevens door onbevoegden wordt verhinderd voor het geval toegang wordt verschaft tot de database of het bestandssysteem.				
Risico	Opgeslagen gegevens worden ongeautoriseerd ingezien, aangepast of verwijderd door ontoereikende versleuteling.				
Referentie	**NCSC**	**NIST**	**ISO27002**		
	B.04 U/WA.05 U/PW.03		9.4.5 10.1.1 12.1.4 12.4.2&3 14.1.1&2 18.1.3&4		

Conformiteitsindicatoren

/01 te beschermen gegevens

SSD-2 Veilige gegevensopslag	
	indicatoren
/01	te beschermen gegevens
/01.01	De opdrachtgever specificeert de classificatie van gegevens.
/01.02	Indien van een gegeven niet de classificatie van de vertrouwelijkheid is vastgesteld, wordt het gegeven (per default) veilig opgeslagen.

Toelichting

/01.01 Er is een BIA of GEB uitgevoerd. De classificatie geeft duidelijkheid over welke gegevens veilig opgeslagen dienen te worden en welke versleuteld.

/02 veilig opgeslagen

Gevoelige gegevens worden afgeschermd tegen ongeautoriseerde kennisname en/of manipulatie door technieken. De bijbehorende normatiek is terug te vinden in andere beveiligingseisen:

- Toegangsregeling tot de opslag is georganiseerd conform SSD-5 (Authenticatie), SSD-8 (Autorisatie) en SSD-7 (Gebruikersrechtenbeheer).
- Benadering van de database of het opslagmechanisme is beveiligd tegen SQL-/Commando injectie (SSD-19, SSD-21 en SSD-22).
- Applicatielogging wordt toegepast conform SSD-30 op toegang tot de opslag die informatiewaarde heeft.
- Gegevens in cookies worden beschermd conform SSD-33.

/03 versleuteld

SSD-2 Veilige gegevensopslag	
	indicatoren
/03	Versleuteld
/03.01	Voorkom dat wachtwoorden in leesbare vorm worden opgeslagen door gebruik van hashing in combinatie met salts en minimaal 10.000 rounds of hashing.
/03.02	Gegevens worden door de applicatie of database deugdelijk versleuteld opgeslagen, naar op dat moment gangbare cryptografie (bijvoorbeeld AES-GCM 256 voor opslag), tenzij door de gegevens-eigenaar is gedocumenteerd dat dit niet noodzakelijk is (*secure by default).*

Toelichting

/03.02 Afhankelijk van gevoeligheid en technische haalbaarheid kan database versleuteling bijvoorbeeld op database,-, tabel,-, kolom- of regelniveau zijn. Belangrijkste afweging is dat een doorbraak op één plek niet kan leiden tot toegang tot alle gegevens. Een risicoanalyse kan hier de noodzakelijke helderheid bieden.

/03.02 Versleuteling die uitsluitend plaatsvindt op schijfniveau of databaseniveau geeft beperkte garanties als die opslag gedeeld wordt met andere applicaties of gebruikers.

Handreiking

- OWASP cheat sheets:
  - Cryptographic Storage
  - Database Security
  - Password Storage
- OWASP Web Security Testing Guide (WSTG):
  - Testing for weak cryptography

/04 noodzakelijk

SSD-2 Veilige gegevensopslag	
	indicatoren
/04	Noodzakelijk
/04.01	Te beschermen gegevens worden alleen opgeslagen als dat nodig is voor het doel en voor de kortst mogelijke tijd, zijnde de kortste periode tussen de periode voor het vervullen van de toepassing en de door wet- of regelgeving verplichte periode.

Toelichting

/04.01 Maatregelen hiervoor omvatten ook het forceren van anti-caching en voorkomen van opslag van gevoelige/persoonlijke data aan de gebruikerskant (bijv. localstorage, IndexedDB), zowel als het schonen van legitieme opslag aan de gebruikerskant na beëindigen van een sessie.

4.3 Authenticatie

Authenticatie moet worden afgedwongen voor alle systeemfuncties (waaronder ook toegang tot bronnen) en voor alle gebruik van systeemfuncties. De authenticatiemethode moet intrinsiek sterk zijn en de implementatie van de methode voldoende sterk[3]. In geval van mislukte authenticatie, mag het systeem geen functies uitvoeren of informatie vrijgeven. Geauthenticeerde gebruikers/systemen worden geïdentificeerd en geregistreerd op een manier die identificatie uniek maakt. Deze identificatie mag niet kwetsbaar zijn voor identiteits-manipulatie (spoofing).
Zie ook:

- Datacommunicatie (want toegangscontrole vereist minstens een beveiligde overdracht van "toegangsbewijs"/login)
- Logging (want logging vertrouwt op sterkte van de toegangscontrole).
- Sessiebeheer (want een sessie leunt op sterke toegangscontrole)
- Autorisatie (want die past de vastgestelde rechten toe)

4.3.1 *SSD-5: Authenticatie van gebruikers en systemen*

Voor: alle software waar toegangscontrole voor nodig is.
Trigger: authenticatie-opzet: toegangsverlening door controle van identiteit.

[3] Wet- en regelgeving, organisatiebeleid en de applicatie-eigenaar of *product owner* bepalen wat als voldoende sterk wordt beschouwd

SSD-5 Authenticatie van gebruikers en systemen					
Criterium (wie en wat)	Applicaties stellen de identiteit van gebruikers en systemen vast op basis van een mechanisme voor identificatie en authenticatie, waarbij de authenticatiegegevens in een centrale authenticatievoorziening worden beheerd.				
Doelstelling (waarom)	Door het vaststellen en controleren van identiteit, voorkomen van onbevoegde toegang tot applicaties en het kunnen herleiden van handelingen.				
Risico	Doordat de identiteit van een gebruiker of systeem onvoldoende wordt vastgesteld wordt ongewenst toegang verkregen en kunnen acties onvoldoende worden getraceerd.				
Referentie	NCSC	NIST	ISO27002		
	B.02 U/TV.01 U/PW.01 U/PW.05	IA-1 IA-2	9.4.2		

Toelichting

Identificatie en authenticatie heeft tot doel alle handelingen te kunnen herleiden tot natuurlijke personen of systemen.
De wijze waarop de identiteit van een gebruiker moet worden vastgesteld, vind plaats op basis van het identiteits- en toegangsvoorzieningsbeleid van de opdrachtgever. Het identiteits- en toegangsvoorzieningsbeleid geeft regels en voorschriften voor de organisatorische en technische inrichting van de toegang tot ICT-voorzieningen.

In eerdere versies van de normen kwam Indicator "/01 gebruikers of systemen" voor. Aangezien dit een duplicaat was met SSD-7, is deze indicator hier verwijderd.

Conformiteitsindicatoren

/02 mechanisme voor identificatie en authenticatie

SSD-5 Authenticatie van gebruikers en systemen	
	indicatoren
/02	mechanisme voor identificatie en authenticatie
/02.01	Indien de opdrachtgever eisen heeft gesteld aan de identificatie- en authenticatievoorziening en/of eisen aan credentials, dan worden deze opgevolgd door de applicatie.
/02.02	De configuratie van de identificatie- en authenticatievoorziening waarborgt dat de geauthentiseerde persoon/systeem, inderdaad de geïdentificeerde persoon/systeem is.
/02.03	Het inlogmechanisme, ook tijdens het herstellen van het wachtwoord, laat niet toe dat bepaald kan worden of een gebruikersnaam geldig is of niet.
/02.04	Het inlogmechanisme is robuust tegen herhaaldelijke, geautomatiseerde of verdachte pogingen om wachtwoorden te raden (brute-forcing of password spraying, hergebruik van gelekte wachtwoorden).

Toelichting

/02.01 Het hoe en wanneer van authenticatie is gebaseerd op het toegangsvoorzieningsbeleid. Deze hoort gebaseerd te zijn 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.

/02.01 Algemene aanbevelingen rond authenticatie zijn:

- Zorg dat het systeem niet toegankelijk is op basis van 'anonieme' accounts zoals een groepsaccount. Accounts horen uniek per gebruiker/persoon/systeem te zijn, omdat gebruik anders ook niet herleidbaar is. Bovendien is elke extra account een extra mogelijkheid voor aanvallers om misbruik te maken.
- Kies afhankelijk van een risico-analyse voor adequaat sterke authenticatiemechanismen. Sterke mechanismen kenmerken zich door het gebruik van meerdere factoren (dimensies van bewijs van identiteit) voor authenticatie (via twee verschillende kanalen). Kies voor een sterk mechanisme als geen risico-analyse is gedaan (secure by default).
- Beperk het aantal groepen waartoe een gebruiker behoort (groepslidmaatschappen) tot het noodzakelijke ("need to know").
- Gebruik geen hard gecodeerde wachtwoorden.
- Geef nieuwe gebruikers geen standaard wachtwoorden.
- Het wachtwoordbeleid wordt door de centrale authenticatievoorziening technisch afgedwongen. In het wachtwoordbeleid worden de minimale wachtwoordlengte, toegestane tekens, complexiteit van het wachtwoord en maatregelen zoals account/ip lock-outs of eventueel tarpitting vastgelegd, evenals het aantal hashes wat onthouden wordt om te controleren of een wachtwoord wordt hergebruikt. De implementatie van account/ip lock-outs dient te voorkomen dat gebruikers het inloggen belet kan worden.
- Sla wachtwoorden veilig op: zie SSD-2.
- Verwijder of deactiveer ongebruikte accounts, standaard aanwezige accounts en accounts van vertrekkende gebruikers.
- Hernoem 'bekende' accounts die niet verwijderd kunnen worden (zoals 'administrator').
- Vraag om herauthenticatie bij zeer kritische functionaliteiten (bijvoorbeeld een banktransactie) op basis van een risico-analyse.

/02.03 De procedure voor wachtwoordwijziging biedt voldoende zekerheid dat de aanvraag wordt gedaan door de huidige gebruiker (de duur en sterkte van authenticatie hiervoor is minstens zo sterk als bij normaal aanmelden). Wachtwoordherstel dient te verlopen via een eerder geverifieerd kanaal (typisch e-mail adres) en via een eenmalig (OTP) en kortlevend token (bijv. 10 minuten) van een onraadbare willekeur. Het mechanisme voor tijdelijke wachtwoord-generatie (functioneel een eenmalige toegangstoken) verzekert dat deze niet hergebruikt mag worden (een permanent wachtwoord wordt).

/02.03 Het inlogmechanisme is zodanig ingericht dat er geen merkbare verschillen zijn tussen geldige en ongeldige gebruikersnamen. Het afleiden van geldigheid van gebruikersnamen maakt het mogelijk om te bepalen of een individu een gebruiker is (voor sommige diensten is dat gegeven vertrouwelijk) en het maakt vervolgens het raden van wachtwoorden eenvoudiger. Afleiden kan doordat de applicatie zegt "Wachtwoord onjuist, probeer opnieuw". Een andere manier van afleiden is de responstijd te meten van een mislukte inlogpoging – die kan namelijk anders zijn met een geldige gebruikersnaam dan met een ongeldige gebruikersnaam. Het wachtwoordherstel-mechanisme geeft soms op soortgelijke manier informatie prijs.

/02.04 Een mogelijk manier om password brute-forcing tegen te gaan is een (toenemende) vertraging tussen inlogpogingen.

/03 centrale authenticatievoorziening

Het centraliseren van authenticatie heeft als voordeel dat de toegangscontrole consistent kan worden afgedwongen. Ook heeft deze voordelen voor traceerbaarheid (centrale logging) en code-onderhoud (eenvoud van het aanroepen van een toegangsfunctie). Zo is traceerbaar welke gebruiker welke acties heeft uitgevoerd.

SSD-5 Authenticatie van gebruikers en systemen	
	indicatoren
/03	centrale authenticatie- en autorisatievoorziening
/03.01	Indien wet- of regelgeving dat voorschrijft wordt gebruik gemaakt van een federatieve voorziening(zoals bijvoorbeeld DigiD).
/03.02	Er wordt voor de authenticatie van interne- of externe medewerkers gebruik gemaakt van een bedrijfsbrede authenticatievoorziening, indien deze beschikbaar is gesteld voor de applicatie.
/03.03	Het beheren en onderhouden van identiteiten en autorisaties wordt gelogd.

Toelichting

/03.02 Dit betekent: er is geen specifieke faciliteit voor identiteit- en toegangsbeheer in de applicatie ingebouwd. Wel worden functies voor toegangsbeheer in de code expliciet aangeroepen. De applicatie vertrouwt op centraal belegde functionaliteiten op dit gebied. In geval van gedistribueerde toegang/samenhang van meerdere organisaties wordt er bij voorkeur gebruik gemaakt van een federatieve authenticatievoorziening (het delegeren van identiteitscontrole binnen een vertrouwd verband).

/03.02 Consolideer authenticatie- en autorisatiegegevens zoveel mogelijk in dezelfde dataverzameling. De keuze voor een centrale authenticatievoorziening voorkomt repliceren of synchroniseren van authenticatiemechanismen. Ook kan een centrale opslag analyse/audits vergemakkelijken.

/03.03 Het ondersteunde systeem logt de volgens SSD-13 aangewezen transacties. Zie verder SSD-30.

Handreiking

- OWASP cheat sheets:
  - Authentication
  - Choosing and using security questions
  - Credentials stuffing prevention
  - Forgot password
  - Multifactor authentication
  - SAML security
- OWASP Web Security Testing Guide (WSTG):
  - Authentication testing
  - Identity management testing

4.3.2 *Vervallen: SSD6*

Vaststellen identiteit interne gebruiker samengegaan met SSD-5.

4.4 Autorisatie

Veilige autorisatie vindt plaats binnen het systeem, zodat de gebruiker toegangscontrole niet kan omzeilen. Autorisatie moet plaatsvinden voor elke systeemfunctie en bij elke poging om toegang te krijgen tot een systeemfunctie of bron. Als de autorisatie mislukt, moet het systeem deze gebeurtenis registreren en de gebruiker alleen informeren dat de autorisatie is mislukt. Gebruikers krijgen bij autorisatie de minst mogelijke privileges/gebruiksrechten.
Zie ook:

- Authenticatie (want het consistent afdwingen van toegangscontrole is een voorwaarde voor sterke autorisatie)
- Logging (want logging vertrouwt op sterkte van zowel de toegangscontrole als de toewijzing van gebruiksrechten. Het registreren van gefaalde toegangspogingen is een taak van de logger/faciliteit)
- Sessiebeheer (want gebruikersrechten worden typisch binnen een sessie toegepast en zijn daarmee afhankelijk van de sessie-maatregelen)
- Gebruikersbeheer (want systemen moeten voor autoriseren aannemen dat gebruikersrechten juist geadministreerd zijn)

4.4.1 *SSD-8: Autoriseer toegang*

Voor: alle software met beperkte gebruikersrechten.
Trigger: gebruikersrechten-opzet en verlenen van toegang tot functionaliteit en bronnen zoals webpagina's en API's.

SSD-8 Autoriseer toegang					
Criterium (wie en wat)	De applicatie dwingt de door de opdrachtgever voorgeschreven beperkende set van rechten en privileges af met alleen de voor de gebruiker en systemen noodzakelijke toegang.				
Doelstelling (waarom)	Door toegang op functies en gegevens te controleren en te beperken wordt ongewenste toegang tegengegaan.				
Risico	Een gebruiker of systeem verkrijgt meer toegang dan nodig of bevoegd en maakt daar bedoeld of onbedoeld misbruik van. Dit misbruik kan ook plaatsvinden doordat een kwaadwillende het account van die gebruiker heeft gecompromitteerd (misbruik/kapen van andermans toegang).				
Referentie	NCSC	NIST	ISO27002		
	B.02 U/TV.01 U/WA.02 U/WA.09	AC-5	9.2.1 9.2.3 9.4.1		

Toelichting
Risico's van misbruik kunnen aanzienlijk worden verminderd door rechten op een systeem of applicatie te beperken. Principes die in het beleid gehanteerd kunnen worden, zijn bijvoorbeeld gebaseerd op

'standaard geen toegang', 'least privilege', 'need-to-know' of functiescheiding ('segregation 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 is daarom rekening gehouden met het principe van least privilege.

Conformiteitsindicatoren

/01 rechten en privileges

SSD-8 Autoriseer toegang	
	indicatoren
/01	rechten en privileges
/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 applicatie zorgt via de toekenning van autorisaties dat niet meer dan de toegewezen rechten beschikbaar komen voor gebruikers en systemen.
/01.04	Applicaties draaien op de onderliggende servers met "least privileges". Door de hostingpartij is het relevante deel van rechteninperking geïmplementeerd, zoals die door de softwaremaker zijn meegegeven in de configuratiebeschrijving.

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.

/01.03 Cross-Site Request Forgery (CSRF) is een kwetsbaarheid waarin een aanvaller een actie forceert vanaf een ander domein. In feite is dat een autorisatieprobleem, want de sessie en het verzoek zelf zijn geldig, alleen komen deze niet van de gebruiker zelf. Er is onvoldoende zekerheid over de oorsprong. Meestal wordt dit opgelost in de vorm van extra sessiegeheimen die de gebruiker moet meesturen bij verzoeken (csrf-tokens). Zie ook SSD-33 over secure headers, waaronder CSP en SameSite.
Nota bene: in het geval van een XSS/Cross-Site Scripting lek kunnen maatregelen tegen CSRF omzeild worden.

Handreiking

- OWASP cheat sheets:
  - Access control
  - Cross-Site Request Forgery prevention
  - SAML security
  - Transaction authorization
- OWASP Web Security Testing Guide (WSTG):
  - Authorization testing
  - Client-side testing
  - Configuration and deployment management, m.n. File permission
  - Identity management testing, m.n. Role Definitions

4.5 Gebruikersbeheer


Adequaat gebruikersbeheer is essentieel voor effectieve toegangscontrole en omvat beheer van rechten, veilige blokkering en verwijdering van gebruikers en veilig beheer van gebruikersgegevens.
Zie ook:

- Authenticatie
- Autorisatie

4.5.1 *SSD-7: Gebruikersrechtenbeheer*

Voor: alle software met gebruikersrechten.
Trigger: opzet gebruikersrechten.

SSD-7 Gebruikersrechtenbeheer					
Criterium (wie en wat)	De rechten die gebruikers hebben binnen een applicatie (inclusief beheerders) zijn zo ingericht dat autorisaties kunnen worden toegewezen aan organisatorische functies en scheiding van niet verenigbare autorisaties mogelijk is.				
Doelstelling (waarom)	Het effectief en veilig inrichten van de toegangsrechten van gebruikers tot functionaliteit en gegevens.				
Risico	Misbruik van een applicatie vindt plaats doordat een gebruiker effectief meer rechten, of combinaties van rechten heeft dan gewenst.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.02 B.02	AC-5	9.2.3.1 9.2.6 9.4.4 12.1.4		

Conformiteitsindicatoren

/01 autorisaties

SSD-7 Gebruikersrechtenbeheer	
	indicatoren
/01	autorisaties
/01.01	De rechten voor toegang tot gegevens en functies in de applicatie zijn op een beheersbare wijze geordend, gebruik makend van autorisatiegroepen.
/01.02	Bij het definiëren van autorisaties is extra aandacht voor accounts met hoge privileges, zoals functioneel beheeraccounts en platformaccounts.
/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.
/01.04	Als een gebruiker bevoegd is om bepaalde aanvragen goed te keuren, dan mag hij niet zijn eigen aanvragen goedkeuren. Dit moet een andere daartoe bevoegde collega doen.

Toelichting

/01.01 De autorisatiegroepen zijn zodanig opgezet dat zij eenvoudig en beheersbaar te koppelen zijn met taken en organisatorische functies. Hierbij kan gebruik worden gemaakt van indeling in rollen (bekend als RBAC) of attributen (bekend als ABAC).
Toegangsbeheer betreft alle activiteiten die systemen moeten uitvoeren om de autorisaties voor applicaties in te regelen en af te dwingen.
Het toekennen van toegang is gebaseerd op classificatie van de vertrouwelijkheid en risicoanalyse. Daarbij is rekening gehouden met situaties waarin toegang wordt verschaft van buiten het interne vertrouwde netwerk.

/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 kan bij applicaties worden gedacht aan de volgende typen platformaccounts:

- Accounts voor het opzetten van verbindingen tussen de applicatie en (applicaties 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.
- Platformaccounts voor toegang tot het bestandssysteem. Vaak gebruikt een applicatie voor toegang tot het bestandssysteem het account waaronder de applicatie 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 de noodzakelijke voor het uitvoeren van de nodige taken en verantwoordelijkheden. Zie ook SSD-8.

/01.03 De toegewezen autorisaties worden periodiek gecontroleerd. Dit proces is afgestemd met de opdrachtgever.

Indicator /02 uit eerdere versies is hier verwijderd in verband met duplicatie.

/03 scheiding van niet te verenigen autorisaties

SSD-7 Gebruikersrechtenbeheer	
	indicatoren
/03	scheiding van niet verenigbare autorisaties
/03.01	Op basis van taken, verantwoordelijkheden en bevoegdheden zijn de verenigbare 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 is er een uitleg beschikbaar over de ondersteuning van de functiescheiding.
/03.04	Er bestaat een proces voor het definiëren en onderhouden van de autorisaties.

Toelichting

/03.01 Voorkomen wordt 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. Een voorbeeld hiervan is de scheiding van het bepalen van de hoogte van vergoedingen en het uitvoeren van betalingen.

4.6 Sessiebeheer

Na authenticatie moet een sessiemanager-component vanuit de server acties van de gebruiker op een veilige manier volgen. Hiervoor zijn bewezen raamwerken en bibliotheken beschikbaar. Gewoonlijk gebeurt dit met behulp van een beveiligd sessietoken. Een sessietoken moet op een veilige manier worden gecreëerd en verlopen, en het mag geen misleidbare of gevoelige informatie bevatten. Wanneer sessiebeheer onveilig is in webapplicaties, kunnen er kwetsbaarheden optreden, zoals session hi-jacking, cross-site request forgery (CSRF) en clickjacking.

4.6.1 *Vervallen: SSD-10:*

SSD-10 Concurrent Session Control is vervallen.

4.6.2 *Vervallen: SSD-12A*

SSD-12A Session lock Vervalt als eis (staat geheel los van de applicatie).

4.6.3 *SSD-12B: Sessie-beëindiging*

Voor: alle software met sessies.
Trigger: opzet sessie-faciliteit.

SSD-12 Sessie-beëindiging					
Criterium (wie en wat)	De applicatie beëindigt een sessie na een vooringestelde periode van inactiviteit van de gebruiker via automatische sessie-beëindiging,.				
Doelstelling (waarom)	Beperken van de sessieduur zodat het risico op misbruik van de sessie wordt geminimaliseerd.				
Risico	Onbevoegde toegang door toegangsverschaffing tot een te lang open (applicatie)sessie.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.08	AC-12	10.1.2		

Toelichting

Als een sessie met een applicatie blijft openstaan, kan hiervan misbruik worden gemaakt. Sessie-beëindiging zorgt ervoor dat de sessie wordt beëindigd na een door de opdrachtgever voorgeschreven tijdsinterval van inactiviteit.

Conformiteitsindicatoren

/01 vooringestelde periode

SSD-12 Sessie-beëindiging	
	indicatoren
/01	vooringestelde periode
/01.01	Als default wordt 15 minuten aangehouden, tenzij de functionaliteit anders noodzakelijk maakt.

/02 automatische sessie-beëindiging

SSD-12 Sessie-beëindiging	
	indicatoren
/02	automatische sessie-beëindiging
/02.01	De applicatie activeert automatisch session termination na een door de opdrachtgever voorgeschreven tijdsinterval van inactiviteit (soft time-out) en voor hoog-risico applicaties (of applicatie-onderdelen) een maximale sessieduur ongeacht gebruikersactiviteit (hard time-out).
/02.02	Session termination komt overeen met het uitloggen van de gebruiker en de applicatie vernietigt daarbij dienovereenkomstig de sessie.

Toelichting

/02.01 Voor de soft time-out kan als norm gehanteerd worden:

- Voor hoog-risico applicaties /onderdelen/ rollen: 2-5 minuten
- Overig: 15-30 minuten

/02.01 De hard time-out is vooral bedoeld om de periode te beperken dat een aanvaller misbruik kan maken, als deze eenmaal toegang heeft gekregen tot een sessie.

/02.01 Hard time-out kan indien gewenst worden ingericht voor specifieke onderdelen van een applicatie, waardoor een gebruiker na een hard time-out normaal gesproken kan doorwerken en alleen opnieuw hoeft in te loggen op high-risk delen.

/02.01 Bij de keuzes rondom sessie-beëindiging is het van belang een goede afweging te maken tussen beveiliging en het belasten van gebruikers met opnieuw inloggen.

Handreiking

- OWASP cheat sheets: Session management
- OWASP Web Security Testing Guide (WSTG): Session management testing

4.6.4 *SSD-14: Borgen van Sessie Authenticiteit*

Voor: alle software met sessies.
Trigger: opzet sessie-faciliteit.

SSD-14 Borgen van Sessie Authenticiteit	
Criterium (wie en wat)	De applicatie hanteert voor sessie-identifiers onvoorspelbare tekenreeksen en bij het uitloggen van de gebruiker wordt de sessie actief beëindigd.
Doelstelling (waarom)	Voorkom het raden van sessie-identifiers en beëindig sessies naar vastgestelde criteria om ongewenste toegang tot die sessie te voorkomen.

SSD-14 Borgen van Sessie Authenticiteit					
Risico	Door een sessie-identifier te raden of te kopiëren wordt ongewenst toegang verschaft tot de applicatie.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.05	SC-23			

Toelichting

Een sessie (via https) tussen de applicatie en de gebruiker krijgt een unieke sessie-identifier. Na het uitloggen van de gebruiker dient de sessie actief te worden beëindigd door de applicatie om te voorkomen dat iemand de nog openstaande sessie kan voortzetten.
In dit kader worden ook eisen gesteld aan de sessie-identifier, onder andere dat deze onvoorspelbaar is. Hiertoe wordt een tekenreeks gebruikt met voldoende[4] lengte en willekeurigheid (~entropie binnen de generatie van de identifier).

Conformiteitsindicatoren

/01 onvoorspelbare tekenreeksen

SSD-14 Borgen van Sessie Authenticiteit	
	indicatoren
/01	onvoorspelbare tekenreeksen
/01.01	Een sessie-identifier is voldoende[5] sterk, namelijk een lange willekeurige reeks tekens met voldoende entropie om deze onvoorspelbaar te maken.
/01.02	De applicatie genereert steeds een nieuwe onvoorspelbare sessie-identifier bij het inloggen, bij opnieuw inloggen en bij verandering van toegangsniveau van een gebruiker.

Toelichting

/01.01 Een entropie van minimaal 128 bits wordt op het moment van het schrijven beschouwd als goed genoeg om weerstand te bieden tegen aanvallen.

/01.01 De sessie-identifier dient geen gevoelige (bijv. persoonlijke) gegevens te bevatten of andere invoer.

/01.02 In de praktijk is de standaard functionaliteit voor het genereren van een sessie-identifier in een security-specifiek raamwerk voldoende.

/02 actief beëindigd

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

[4] Wet- en regelgeving, organisatiebeleid en de applicatie-eigenaar of *product owner* bepalen wat de minimale entropie is en/of hoe lang de gemiddelde tijd moet bedragen om de sleutel met een brute force attack te achterhalen
[5] Wet- en regelgeving, organisatiebeleid en de applicatie-eigenaar of *product owner* bepalen wat de minimale entropie is en/of hoe lang de gemiddelde tijd moet bedragen om de sleutel met een brute force attack te achterhalen

/03 geheim

SSD-14 Borgen van Sessie Authenticiteit	
	indicatoren
/03	Geheim
/03.01	Houd de sessie-identifier geheim tijdens zijn levensduur.

Toelichting

/03.01 Het is niet toegestaan dat de sessie-identifier wordt opgenomen in de URL, omdat URLs vaak terecht komen in logs en eventueel http referrer headers. Sessiegeheimen kunnen wel door cookies worden meegestuurd (mits juist gebruikt, zie SSD-33) of in de header/body van verzoeken, mits TLS is afgedwongen.

/03.01 Het is een good practice om session-renewal (typisch met refresh tokens) toe te passen waarbij de sessietoken steeds (elke 2-5 minuten) ververst wordt. Hiermee worden de mogelijkheden voor session hyjacking drastisch beperkt. Toepassing hiervan is een afweging tegen de bijkomende performance-impact, wat kan leiden tot een toepassing hiervan bij kritische functies en rollen.

/04 basic authentication

SSD-14 Borgen van Sessie Authenticiteit	
	indicatoren
/04	basic authentication
/04.01	Maak geen gebruik van basic authentication in combinatie met een web browser

Toelichting

/04.01 Als bij *basic authentication* dezelfde site nogmaals wordt bezocht, voordat de browser afgesloten is geweest, dan logt de browser automatisch opnieuw in, ongeacht de verstreken termijn. Bovendien worden telkens de login-gegevens verstuurd waardoor versleuteling kritiek is.

Nota bene dat versleutelde communicatie bij basic authentication kritiek is aangezien telkens de inloggevens worden gecommuniceerd.

Handreiking

- OWASP cheat sheets:
  - Cross-site request forgery prevention
  - Session management
- OWASP Web Security Testing Guide (WSTG): Session management testing

4.7 Logging

Om onweerlegbaarheid en verantwoording mogelijk te maken, moet het systeem kunnen aantonen dat een gebruiker actief een actie heeft goedgekeurd en uitgevoerd. Dit bewijs moet op een veilige manier worden bewaard en op een manier die het ophalen en analyseren vergemakkelijkt.

Logging dient specifieke traceerbare informatie te bevatten met betrekking tot de identiteit van de gebruiker, zoals bijvoorbeeld locatie of herkomst.
Zie ook:

- Authenticatie (want sterke toegangscontrole vereist unieke identificatie)
- Autorisatie (want het consistent afdwingen van toegangscontrole is een voorwaarde voor sterke autorisatie).
- Sessiebeheer (want gebruikersrechten worden typisch binnen een sessie toegepast en daarmee afhankelijk van de sessie-maatregelen).

4.7.1 *SSD-30: Applicatie logging*

Voor: alle software.
Trigger: opzet logging-faciliteit of functie die logging vereist.

SSD-30 Applicatie logging					
Criterium (wie en wat)	In de applicatieomgeving zijn signaleringsfuncties (registratie en detectie) actief en efficiënt, effectief en beveiligd ingericht.				
Doelstelling (waarom)	Maak het mogelijk om achteraf een reconstructie te maken van handelingen en andere gebeurtenissen, detecteer onregelmatigheden en mogelijke aanleidingen van beveiligingsincidenten en signaleer actief die gebeurtenissen en dreigingen waarvan de applicatie-eigenaar of product owner heeft bepaald dat die inzichtelijk gemaakt moeten worden				
Risico	Door onvoldoende logging kan misbruik van de applicatie niet worden gedetecteerd en niet worden herleid tot de oorzaak en de veroorzaker. Logging is daarmee een voorwaarde voor het kunnen bieden van onweerlegbaarheid.				
Referentie	NCSC	NIST	ISO27002		
	U/PW.08 U/C.06	IR-4 IR-5 IR-6 IR-7	9.4.4.2 12.4.1 12.4.2 12.4.3		

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 standaard loggingsfunctionaliteit kennen (bijv. syslog), wordt vooraf een beperkte, maar wel representatieve selectie van de te loggen systeem-gegevens gemaakt, met het oog op werkbaar beheer.

Houd rekening met de volgende aspecten:

Afschermen en integriteit van loggingsgegevens:
De opgeslagen loggingsinformatie is vaak 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. Typische maatregel is het extern wegschrijven van logs, het digitaal tekenen/hashen van logbestanden. Naast validatie van integriteit, wordt die integriteit ook actief beschermd, door bijvoorbeeld alleen leestoegang toe te laten op besturingssysteem/bestandssysteem niveau.

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.
- Gebruik log-aggregatie tooling om te voorkomen dat het herkennen van verdachte patronen in de logging afhangt van de kunde van de operationeel beheerder.

Bewaartermijnen van registraties:
Er wordt bepaald hoe lang logging online en offline beschikbaar moet en mag zijn. Dit is een risico-gebaseerde afweging tussen functionele eis, operationele beheerwens en wet/regelgeving. 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, is 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:
Centraliseren van loggegevens is van belang om versnippering tegen te gaan, die analyse en signalering bemoeilijkt. Verschillende type logs, met verschillende bronnen, kunnen bij elkaar geveegd worden door tooling zoals *Splunk*.

Synchroniseren van systeemklokken:
Om gebeurtenissen uit verschillende componenten te correleren worden de timestamps van deze gebeurtenissen gebruikt. Standaard wordt daarvoor Network Time Protocol (NTP) gebruikt.

Alternatieven voor beschikbaarheid van registraties (logbestanden):
Bepaal op voorhand welke actie een systeem/service moet nemen op het moment dat een centraal 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 en acties achteraf mogelijk niet traceerbaar zijn.
- 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.
- De component acuut laten stoppen met functioneren.

Vanuit het oogpunt van beveiliging en beschikbaarheid heeft het de voorkeur om – zodra een centraal logging mechanisme uitvalt - componenten eerst lokaal gebeurtenissen te laten opslaan (alternatief 2) om vervolgens de component te laten stoppen met functioneren zodra deze opslag vol is (alternatief 3).

Conformiteitsindicatoren

/01 registratie en detectie

SSD-30 Applicatie logging	
	indicatoren
/01	registratie en detectie
/01.01	De te registreren acties worden centraal opgeslagen.
/01.02	Er is bepaald welke gebeurtenissen en/of beheeractiviteiten aan de applicatie vastgelegd moeten worden. Van de signaleerbare acties wordt alleen de noodzakelijke informatie gelogd, conform AVG/GDPR/privacy richtlijnen.
/01.03	In de applicatie-infrastructuur zijn detectiesystemen actief voor het detecteren van aanvallen.
/01.04	Onderliggende servers zijn door de hostingpartij zodanig geconfigureerd dat security-gerelateerde events worden vastgelegd, conform de configuratiebeschrijving van de softwaremaker.

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 tekortkomingen hebben opgeleverd.

/01.01 Ruimteproblemen bij opslag dienen te worden voorkomen, bijvoorbeeld door detectie van ruimtetekort.

/01.02 Er is 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 applicatie,
- succesvolle en geweigerde toegangspogingen (zie SSD-9),
- (on)geoorloofde activiteiten door gebruikers.

/01.02 Gebruiken van filters is een best practice ten behoeve van leesbaarheid en efficiency.

/01.03 In de ontwerp- dan wel configuratiedocumentatie is vastgelegd waar en hoe *Intrusion Detection Systems* worden ingezet. De specifieke gevallen/(ab)use cases worden typisch door opdrachtgever bepaald.

/02 efficiënt en effectief

In het applicatiedomein 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	
	indicatoren
/02	efficiënt en effectief
/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. In de ontwerp- dan wel configuratiedocumentatie is vastgelegd hoe het synchroniseren van de systeemklokken is geconfigureerd.

/03 beveiligd

SSD-30 Applicatie logging	
	indicatoren
/03	beveiligd
/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 "write-only" en daarmee beveiligd tegen achteraf wijzigen/verwijderen en tegen inzage door onbevoegden.

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 Ontwerpdocumentatie, configuratie-instellingen en autorisatieprofielen geven aan hoe logfiles beschermd zijn tegen achteraf of ongeautoriseerd wijzigen/verwijderen. Voor de inrichting van het logging en monitoring mechanisme zijn SSD-19 (Invoer normalisatie), SSD-22 (Invoervalidatie) en SSD-2 (Veilige gegevensopslag) toegepast. Vermijden van gevoelige gegevens in logbestanden is uiteraard sterk preventief tegen ongewenste inzage.

Handreiking

- OWASP cheat sheets:
  - Logging
  - Session management, specifiek Logging Session life cycle

4.7.2 *SSD-13: Onweerlegbaarheid*

Voor: alle software.
Trigger: opzet logging-faciliteit.

SSD-13 Onweerlegbaarheid					
Criterium (wie en wat)	De applicatie ondersteunt de onweerlegbaarheid voor daartoe aangewezen transacties via cryptografische technieken.				
Doelstelling (waarom)	Het onweerlegbaar vastleggen van transacties, waarvan duidelijk moet zijn dat die door een specifieke gebruiker daadwerkelijk zijn uitgevoerd.				
Risico	Indien een aangewezen transactie niet onweerlegbaar aan een persoon kan worden gekoppeld, kan die later betrokkenheid bij de transactie ontkennen.				
Referentie	NCSC	NIST	ISO27002		
	U/TV.01 U/WA.05	AU-10			

Toelichting

Voor bepaalde transacties wordt onweerlegbaar 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	
	indicatoren
/01	aangewezen transacties
/01.01	Tijdens de risicoanalyse wordt door de opdrachtgever bepaald voor welke transacties onweerlegbaarheid nodig is. Denk hierbij aan het verwerken van financiële, persoonsgerelateerde en andere vertrouwelijke gegevens.
/01.02	Tijdens de ontwerpfase worden de aangewezen transacties nader gedefinieerd en wordt bepaald op welke wijze de onweerlegbaarheid wordt geïmplementeerd.
/01.03	De applicatie dwingt de onweerlegbaarheid af voor de aangewezen transacties.

Toelichting

/01.01 De organisatie bepaalt de sterkte van de binding tussen de persoon die de transactie uitvoert en de informatie. Deze 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 controle-rekeningen (checksums ofwel hashing).

/01.03 Onweerlegbaarheid kan worden afgedwongen door het gebruik van verschillende technieken of mechanismen, onder meer door (logisch) scheiden van de draaiende omgeving/infrastructuur van de applicatie en de logging.

/02 cryptografische technieken

SSD-13 Onweerlegbaarheid	
	indicatoren
/02	cryptografische technieken
/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 applicatie gebruikt voor de aangewezen transacties cryptografische techniek die aan de eisen van de opdrachtgever voldoet.
/02.04	De softwaremaker bevestigt de wijze van sleutelbeheer in de configuratiebeschrijving.
/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 zowel binnen de applicatie als voor de uitwisseling tussen applicaties, 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 transactie plaatsvond,
- In sommige gevallen de eindstaat van een transactie.

Omdat moderne applicaties vaak verschillende systeem- en componentgrenzen overschrijden, is het voor het (anoniem) traceren van gebruikers nodig gebruik te maken van unieke kenmerken die aan een sessie gekoppeld zijn (bijv. UUID/correlation-ids). Van belang is dat een consistente set van attributen wordt gelogd die over een keten of verschillende services dezelfde vorm en betekenis heeft. Zo is een origin-IP-adres in de meeste gevallen onvoldoende omdat sessies verschillende netwerken/subnets kunnen overstijgen doordat tussenschakels technische sprongen over het netwerk maken (bijv. firewalls, proxies).

4.7.3 *SSD-9: Registreren van inlogpogingen*

Voor: alle software met een inlogproces.
Trigger: authenticatie of autorisatie.

SSD-9 Registreren van (un)successvolle login-pogingen	
Criterium (wie en wat)	De applicatie registreert gelukte en mislukte login-pogingen.
Doelstelling (waarom)	Log inlogpogingen om aanvallen te detecteren en misbruik te kunnen herleiden tot de oorzaak en de veroorzaker.
Risico	Onbevoegde toegang wordt niet gesignaleerd en/of kan niet worden herleid tot de veroorzaker doordat login-pogingen niet zijn gelogd.

SSD-9 Registreren van (un)successvolle login-pogingen					
Referentie	NCSC	NIST	ISO27002		
	U/WA.09	AC-7	12.4.1		

Toelichting

Door tijdig te acteren op zowel gelukte als 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 login-pogingen en het resultaat vast te leggen, te monitoren en hierop te reageren kan schade worden voorkomen of worden beperkt.

Conformiteitsindicatoren

/01 gelukte en mislukte login-pogingen

SSD-9 Registreren van (un)successvolle login-pogingen	
	indicatoren
/01	login-pogingen
/01.01	Alleen daar waar geen gebruik gemaakt kan worden van centrale authenticatievoorzieningen, zoals beschreven in SSD 5, vindt authenticatie en logging door de applicatie plaats.
/01.02	De applicatie en de server leggen login-handelingen vast in logbestanden.
/01.03	De logbestanden worden voor real time monitoring beschikbaar gesteld 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 login-pogingen om dreigingen adequaat aan te pakken, tenzij anders overeengekomen.

Toelichting

/01.01 Om te voorkomen dat de applicatie logging moet implementeren wordt bij voorkeur gebruik gemaakt van centrale authenticatievoorzieningen.

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

Handreiking

- OWASP cheat sheets: Logging

4.8 Invoer/uitvoer validatie

Er zijn veel verschillende manieren waarop misbruik van een applicatie kan plaatsvinden via invoer van buiten de applicatie. Ook de uitvoer van een applicatie kan worden gemanipuleerd of misbruikt. Om dit tegen te gaan zijn verscheidene beschermingsmechanismen nodig.
Zie ook:

- Logging (want logging is een vorm van uitvoer die geschoond moet worden uitgevoerd).
- Sessiebeheer (want een systeem verwacht dat acties binnen een geldige sessie plaatsvinden).

4.8.1 ***Vervallen: SSD-18***

Vervallen vanwege duplicatie SSD-18 en SSD-22

4.8.2 ***SSD-19: Invoer-normalisatie***

Voor: alle software.
Trigger: verwerken van invoer buiten het systeem (bijvoorbeeld van gebruiker of database).

SSD-19 Invoer-normalisatie					
Criterium (wie en wat)	De applicatie voorkomt manipulatie door alle ontvangen invoer te normaliseren alvorens die te valideren. De richtlijnen voor invoerbehandeling zijn van toepassing voor alle invoer die van buiten de applicatie komt. Dus niet alleen (eind)gebruikers, maar ook externe systemen en applicaties.				
Doelstelling (waarom)	Bewerk invoer voor zodat deze een algemeen herkenbare vorm heeft, waardoor validatie kan plaatsvinden.				
Risico	Door onvoldoende normalisatie faalt invoervalidatie, waardoor misbruik kan plaatsvinden.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.03	SC-2			

Toelichting
De applicatie ontvangt invoer van de gebruiker en van andere applicaties. Deze invoer kan verschillende vormen hebben. De 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 applicatie veilig verwerkt kan worden. In sommige gevallen kan normalisatie verborgen logica blootleggen (verschillende typen "injecties").
Via normalisatie voorkomt de applicatie dat malafide verzoeken abusievelijk de filters voor validatie kunnen passeren.

Conformiteitsindicatoren
/01 normaliseren

SSD-19 Invoer-normalisatie	
	indicatoren
/01	normaliseren
/01.01	De applicatie zorgt dat invoer in een vorm komt die zodanig is gestandaardiseerd dat deze herkend en gevalideerd kan worden.

Toelichting
/01.01 Maak (minimaal) gebruik van de daarvoor bestaande standaardservices.
/01.01 Normalisatie vindt zodanig plaats dat de integriteit van gegevens gewaarborgd is. Als dat niet mogelijk is, overweeg weigering van gegevens als alternatief, of het vragen van correctietoestemming aan de invoerder.

/01.01 Klassieke normalisatie kan bijvoorbeeld vervanging zijn van een bepaald teken (bijvoorbeeld een "\" naar "/", of het verifiëren van de juiste encoding van de karakters.
/01.01 Normalisatie verzorgt bijvoorbeeld:

- Escaping: het converteren van "risicovolle tekens" naar een serie veilige tekens;
- 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 omzetten van backslashes '\' naar forward slashes '/'.;

Handreiking

- OWASP cheat sheets:
  - Bean validation
  - Command injection
  - Deserialization
  - DOM-based XSS prevention
  - Injection prevention (Java)
  - Input validation
- OWASP wiki Canonicalization
- OWASP Web Security Testing Guide (WSTG): Input validation testing

4.8.3 *SSD-20: Uitvoer-schoning*

Voor: alle software waarvan uitvoer terecht kan komen in een webbrowser of een andere component die opdrachten in invoer kan verwerken.
Trigger: versturen van uitvoer naar een ander systeem of naar de gebruikersinterface.

SSD-20 Uitvoer-schoning (output sanitization)					
Criterium (wie en wat)	De applicatie beperkt de uitvoer tot waarden die (veilig) verwerkt kunnen worden door deze te normaliseren naar de juiste context.				
Doelstelling (waarom)	Schoon uitvoer om manipulatie en daarmee misbruik te voorkomen.				
Risico	Door manipulatie van de uitvoer van de applicatie wordt misbruik gemaakt van de applicatie of een ander systeem waar deze uitvoer wordt gebruikt.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.04				

Toelichting

Vrijwel elke webpagina bevat informatie die afkomstig is uit een bron die het risico heeft op manipulatie, bijvoorbeeld een database, een externe bron, of gebruikersinvoer.
Als een applicatie gegevens aanbiedt aan een eindgebruiker of aan een ander systeem, zorgt de applicatie ervoor dat voor de ontvanger duidelijk is dat het gaat om data en niet om instructies. Als die gegevens tekens bevatten, waaraan door de ontvanger een bijzondere betekenis kan worden toegekend, worden van die bijzondere betekenis ontdaan. Meestal gebeurt dit door: quoting, escaping of omschrijvingen (zoals HTML-entities).

Uitvoerschoning voorkomt dat de applicatie ongewenste opdrachten geeft aan de client, waarvoor XSS (*Cross-site* scripting) een sprekend voorbeeld is. Andere voorbeelden zijn instructies aan achterliggende systemen, zoals: LDAP-injectie, commando-injectie en SQL-injectie.

Het is zeer belangrijk te definiëren welk systeem of welke schakel in een systeemketen de uitvoer-schoning uitvoert. Als schoning "te vroeg" wordt uitgevoerd krijgt een later systeem in een keten daar mogelijk last van. Het heeft de voorkeur schoning aan de serverkant te doen in geval dat precieze eindgebruikers (nog) onbekend of onvertrouwd zijn. Voorbeeld is een publieke API waar onbekende apps op kunnen aansluiten.

Conformiteitsindicatoren

/01 normaliseren

SSD-20 Uitvoer-schoning	
	indicatoren
/01	normaliseren
/01.01	Alle uitvoer wordt naar een veilig formaat geconverteerd.
/01.02	In de configuratiebeschrijving van de softwaremaker is beschreven welk deel van de controles op het verwijderen van ongewenste invoer door de web-/applicatieserver moet worden uitgevoerd.
/01.03	Door de hostingpartij is de configuratie uitgevoerd conform de configuratiebeschrijving.

Toelichting

/01.01 Het schonen van webpagina-inhoud houdt in dat de applicatie mogelijk 'gevaarlijke' tekens codeert. Hoe de applicatie deze informatie moet coderen is afhankelijk van de plek in de pagina waar deze 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: &gt
- HTML-attribuut gecodeerd: &#x3E
- javascript gecodeerd: \x3E
- CSS gecodeerd: \3E
- URL gecodeerd: %3E

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

Handreiking

- OWASP cheat sheets:
  - AJAX security
  - Bean validation
  - Deserialization
  - Command injection
  - DOM-based XSS prevention
  - Injection prevention (Java)
  - Input validation
- OWASP wiki

  - Canonicalization
- OWASP top 10 proactive controls
  - Encode and escape data
- OWASP Web Security Testing Guide (WSTG):
  - Client-side testing
  - Encoded injection
  - Input validation testing

4.8.4 *SSD-21: Beperkte commando/query-toegang*

Voor: alle software die commando's/queries stuurt naar achterliggende onderdelen.
Trigger: commando of query sturen naar achterliggend onderdeel.

SSD-21 Beperkte commando/query-toegang					
Criterium (wie en wat)	De applicatie legt beperkingen aan queries en commando's op daar waar met achterliggende systemen wordt gecommuniceerd en deze communicatie wordt alleen ingericht indien strikt noodzakelijk.				
Doelstelling (waarom)	De mogelijkheden van gemanipuleerde commando's en queries beperken door de opzet van hoe deze worden uitgevoerd.				
Risico	Via manipulatie van commando's en queries (bijvoorbeeld SQL-injectie) wordt onbevoegde toegang verschaft tot gegevens of functies van onder- en achterliggende systemen (bijvoorbeeld database of besturingssysteem).				
Referentie	NCSC	NIST	ISO27002		
	U/WA.07				

Conformiteitsindicatoren

/01 beperkingen aan queries en commando's

SSD-21 Beperkte commando/query-toegang	
	indicatoren
/01	beperkingen aan queries en commando's
/01.01	De applicatie stelt commando- en queryteksten op zo'n manier samen dat data niet misbruikt kan worden om de opdracht te manipuleren. Maak daarvoor indien mogelijk gebruik van geparametriseerde aanroepen van de achterliggende systemen. Indien niet mogelijk, wordt invoer gevalideerd en wordt bij het samenstellen van de opdrachten gebruik gemaakt van vaste tekstelementen in de broncode waar mogelijk.

Toelichting

/01.01 Geparametriseerde queries voor databases worden ook wel *prepared statements* genoemd. In plaats van dat een query wordt samengesteld door strings (bijvoorbeeld een SELECT-statement) te plakken aan een variabele (bijvoorbeeld de inhoud van de WHERE-clause), wordt een geparametriseerde query klaargezet in de database en wordt de variabele als parameter meegegeven. Op geen enkele manier kan de inhoud van die variabele dan nog invloed uitoefenen op de query zelf.

/01.01 Naast databases kennen andere achterliggende systemen ook geparametriseerde vormen van toegang, zoals een API of een library. Het is bijvoorbeeld beter om de *mkdir* functie van een besturingssyteembibliotheek aan te roepen om een folder aan te maken, dan het commando

"mkdir"+variabele naar het besturingsysteem te sturen, want de inhoud van de variabele kan dan bijvoorbeeld nog extra commando's uitvoeren.

/01.01 Een speciale vorm van commando/query manipulatie is de *zogenaamde directory traversal*: invoer die gebruikt wordt voor toegang tot bestanden wordt gemanipuleerd zodat toegang wordt gekregen tot directories waartoe men niet bevoegd is.

/01.01 Met de zogenaamde ReDos aanval wordt een speciale reguliere expressie verwerkt in invoer. Als de applicatie deze invoer dan ook gaat verwerken met behulp van een component die reguliere expressies gebruikt (bijvoorbeeld om te zien een tekst voorkomt in een andere tekst), dan kan de speciale invoer ertoe leiden dat de applicatie hangt. Zie https://www.owasp.org/index.php/Regular_expression_Denial_of_Service_-_ReDoS

/01.01 Zie verder SSD-22 voor invoervalidatie als alternatieve of aanvullende maatregel tegen misbruik via commands/query toegang.

/02 strikt noodzakelijk

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

SSD-21 Beperkte commando/query-toegang	
	indicatoren
/02	Strikt noodzakelijk
/02.01	Van elke applicatie is bekend welke functionaliteit van backend-systemen en databases 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 veilige varianten van File Transfer Protocol (FTPS getunneld in HTTPS en SFTP getunneld in SSH) en veilige protocollen voor (beheer-)interactie zoals SSH.

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

Handreiking

Invoervalidatie is geïntegreerd met HTTP validatie in SSD-22

- OWASP cheat sheets:
  - Command injection
  - Injection prevention (Java)
  - Input validation
  - LDAP injection prevention
  - Query parameterization
  - SQL injection prevention
- OWASP Web Security Testing Guide (WSTG): Input validation testing

4.8.5 *SSD-22: Invoer-validatie*

Voor: alle software.

Trigger: verwerken van externe invoer (bijvoorbeeld van gebruiker, service of database).

SSD-22 Invoer-validatie					
Criterium (wie en wat)	De applicatie controleert invoer (bijvoorbeeld een HTTP-request) door deze te valideren alvorens die te gebruiken.				
Doelstelling (waarom)	Voorkom misbruik door invoer te controleren op manipulatie.				
Risico	Door manipulatie van de invoer wordt de applicatie misbruikt.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.03 U/PW.02	SC-2			

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, 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.

Korte toelichting over de terminologie:

- **Normalisatie** (zie SSD-19): Invoer omzetten naar een normaalvorm of conventie, zodat goed kan worden gevalideerd en eventueel geschoond.
- **Validatie** (zie SSD-18)**:** Controle van invoer die leidt tot een beslissing. Over het algemeen is verwerpen/rejection de voorkeursactie. Alternatief kan invoer geschoond worden.

Als validatie leidt tot een schoningsactie zijn de volgende termen relevant:

- **Opschoning/ sanitization/encoding:** "Schoning", het actief aanpassen van karakters.
- Na opschoning valideert het systeem de invoer mogelijk opnieuw om de juiste verwerking te kunnen garanderen.

Context

Alle invoer wordt door de applicatie 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 applicatie geweigerd.

De applicatie ontvangt invoer van de gebruiker en van andere applicaties. Hierbij is een belangrijke vuistregel dat de applicatie geen enkele invoer mag vertrouwen die onder controle staat van een gebruiker. Dit is een bewuste afweging. De applicatie is daarvoor 'defensief' geprogrammeerd, waarbij alle via bijvoorbeeld HTTP-requests of uit databases ontvangen inhoud eerst wordt gevalideerd, alvorens die wordt gebruikt.
De verdeling van controles op de specifieke kenmerken is afhankelijk van de webserver en de applicatie(s) waarmee deze samenwerkt.

Conformiteitsindicatoren

/01 valideren

SSD-22 Invoer-validatie	
	indicatoren
/01	Valideren
/01.01	Foute, ongeldige of verboden invoer wordt geweigerd of onschadelijk gemaakt. 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.
/01.04	Indien van toepassing is in de configuratiebeschrijving bij de applicatie voor de webserver beschreven welke ongewenste invoer door de webserver moet worden geweigerd.
/01.05	Eventuele gespecificeerde configuratie van webserver met betrekking tot invoer (zie boven) is toegepast.

Toelichting

/01.01 Als **alternatief** van weigering kan input geschoond worden van ongewenste inhoud (sanitization). Als het moeilijk is om op basis van 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. Duidelijk mag zijn dat dit complexe materie is, waar beter gebruik kan worden gemaakt van bestaande bibliotheken en raamwerken.

/01.01 Invoer kan verwijzingen bevatten naar bepaalde functies of gegevens. Door manipulatie van die verwijzing (bijvoorbeeld het veranderen van een argument in een URL) kan ongewenst toegang worden verkregen: zogenaamde Insecure Direct Object Reference. Dit kan worden opgelost door de toegang tot die gegevens/functies te controleren (zie SSD-8: Autorisatie).

/01.04 Op het gebied van **webserver configuratie** kan filtering helpen in het verminderen van de hoeveelheid data die het systeem uiteindelijk bereikt na het filteren of trechteren ("throttling") van verzoeken, bijvoorbeeld met indringingsdetectie (IDS).

Handreiking

- OWASP cheat sheets:
  - Bean validation
  - Command injection
  - Deserialization
  - Input validation
  - Mass assignment
- OWASP Web Security Testing Guide (WSTG): Input validation testing

4.8.6 *SSD-23: Beperkte file includes*

Voor: alle software die bestanden met broncode ophaalt op basis van invoer.
Trigger: toegang verschaffen tot broncodebestanden op basis van invoer.

SSD-23 Beperkte file Includes					
Criterium (wie en wat)	De applicatie voorkomt de mogelijkheid van dynamische file includes.				
Doelstelling (waarom)	Voorkom dat via een intern of extern bestand ongewenste code wordt uitgevoerd.				
Risico	Een aanvaller kan code laten uitvoeren door te verwijzen naar een bestand, waardoor de applicatie wordt misbruikt.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.03				

Toelichting

Meest bekend als Remote/Local File Inclusion (RFI/LFI): een kwetsbaarheid die zich voordoet op applicaties die gebruik maken van dynamische file includes in script- en programmeertalen (bijvoorbeeld PHP of JSP). Wanneer een dergelijke applicatie kwetsbaar is, kan een kwaadwillende ongewenste code door de server laten uitvoeren, bijvoorbeeld als een pagina op een webserver een verwijzing naar een bestand direct importeert en uitvoert. In feite is dit een gefaalde invoervalidatie (zie SSD norm 18).

Conformiteitsindicatoren

/01 dynamische file includes

SSD-23 Beperkte file Includes	
	indicatoren
/01	dynamische file includes
/01.01	De applicatie maakt geen gebruik van dynamische file includes, tenzij de oorsprong en geldigheid gegarandeerd en gecontroleerd kan worden.
/01.02	De (web)applicatie beperkt (op de server) de keuzemogelijkheid voor het uploaden van bestanden, bijvoorbeeld via whitelisting.
/01.03	In de server- configuratiebeschrijving van de softwaremaker zijn eventueel benodigde maatregelen beschreven die file include misbruik moeten voorkomen.
/01.04	De server is geconfigureerd door de hostingpartij, zoals gespecificeerd in de configuratiebeschrijving met betrekking tot file includes.

Handreiking

- OWASP cheat sheets:
  - Input validation
  - PHP configuration
- OWASP Web Security Testing Guide (WSTG): Input validation testing

4.8.7 *SSD-24: Beperking van te versturen HTTP-headers*

Voor: webapplicaties en http-backends
Trigger: bij configureren van headers in webserver

SSD-24 Beperking van te versturen HTTP-headers					
Criterium (wie en wat)	De webserver stuurt bij een antwoord aan een gebruiker alleen die informatie in de HTTP-headers mee die van belang is voor het functioneren van HTTP.				
Doelstelling (waarom)	Door zo min mogelijk technische informatie op te nemen in HTTP headers worden aanvallers minimaal geïnformeerd.				
Risico	Een aanvaller gebruikt technische informatie in HTTP response headers om een manier te vinden voor misbruik van de applicatie.				
Referentie	NCSC	NIST	ISO27002		
	U/PW.02 U/WA.06		14.1.3		

Toelichting

De webserver ondersteunt het HTTP-protocol. HTTP kent methoden, headers en foutinformatie (per statuscodes), die mogelijk misbruikt kunnen worden. Hiermee kan namelijk onnodig informatie worden vrijgegeven. Het gebruik dient daarom waar mogelijk te worden beperkt.

HTTP headers kunnen informatie bevatten over de gebruiker en over de applicatie. 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 HTTP-headers

SSD-24 Beperking van te versturen HTTP-headers	
	indicatoren
/01	HTTP-headers
/01.01	De softwaremaker legt in de ontwerpdocumentatie vast welke HTTP-headers worden gebruikt door de applicatie en hoe HTTP-headers uit de antwoorden worden verwijderd.
/01.02	De softwaremaker 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 softwaremaker onderbouwt en beschrijft eventuele noodzakelijke afwijkingen van de standaard configuratie op de webserver, die nodig zijn voor het goed functioneren van de applicatie.
/01.04	De webserver verstuurt alleen HTTP-headers die voor het functioneren van de applicatie van belang zijn. Specifieke beveiligingsheaders zijn uiteraard wel wenselijk waar deze geëist worden voor veilige werking van het systeem (bv. CSP, X-Frame, HSTS).

SSD-24 Beperking van te versturen HTTP-headers	
	indicatoren
/01.05	De webserver verwijdert alle niet noodzakelijke informatie uit de HTTP-header (bijvoorbeeld de Server header), alvorens deze voor het antwoord te gebruiken.

Toelichting

/01.01 In de ontwerp- danwel configuratiedocumentatie is vastgelegd:

- welke HTTP-headers voor het functioneren van de applicatie 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/WAF.

Eventuele afwijkingen van bovenstaande die noodzakelijk zijn, omdat de applicatie 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.

Handreiking

Voorbeelden hoe te testen:

- OWASP Web Security Testing Guide (WSTG):
  - Configuration and deployment management
  - Information gathering

4.8.8 *Vervallen: SSD-25*

Beperken van te tonen HTTP-header: opgenomen in SSD-24

4.8.9 *SSD-27: Discrete foutmeldingen*

Voor: alle software.
Trigger: samenstelling foutmelding.

SSD-27 Discrete foutmeldingen					
Criterium (wie en wat)	De applicatie neemt in een foutmelding geen inhoudelijke foutinformatie op die misbruikt kan worden.				
Doelstelling (waarom)	Door zo min mogelijk technische informatie op te nemen in foutmeldingen worden aanvallers minimaal geïnformeerd. Daarnaast dienen vertrouwelijke gegevens niet terecht te komen in foutmeldingen.				
Risico	Een aanvaller gebruikt technische informatie in foutmeldingen om een manier te vinden voor misbruik van de applicatie, of vertrouwelijke gegevens te bemachtigen via een foutmelding.				
Referentie	NCSC	NIST	ISO27002		
	U/PW.02 U/WA.06	SI-11			

Toelichting

Foutmeldingen kunnen afkomstig zijn rechtstreeks uit de applicatie en van onderliggende of tussenliggende systemen (bijvoorbeeld een webserver).
Een uitgebreide foutmelding kan een kwaadwillende helpen om meer inzicht te krijgen in de programmalogica van een applicatie. 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 applicatie, 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 applicatie openbaart, zoals databasenamen, 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 webservers zelf bieden functionaliteit om standaard meldingen te laten genereren aan de hand van specifieke statuscodes.
Buiten technische informatie kan ook andere vertrouwelijk informatie lekken via foutmeldingen, zoals persoonsnamen of wachtwoorden.

Conformiteitsindicatoren

/01 inhoudelijke foutinformatie

SSD-27 Discrete foutmeldingen	
	indicatoren
/01	inhoudelijke foutinformatie
/01.01	Bij het optreden van een fout wordt de informatie tot een minimum beperkt. Een eventuele foutmelding zegt wel *dat* er iets is fout gegaan, maar niet *hoe* het is fout gegaan.
/01.02	De softwaremaker legt eventuele speciale instructies in de configuratiedocumentatie vast hoe de webserver en/of de application-level firewall moet worden geconfigureerd om alleen standaard foutmeldingen te tonen en versturen.
/01.03	De webserver toont bij een fout geen gedetailleerde informatie aan de client, maar alleen een standaard foutmelding.

Handreiking

- OWASP cheat sheets: Error handling
- OWASP Web Security Testing Guide (WSTG): Configuration and deployment management

4.8.10 *SSD-28: Discreet commentaar*

Voor: webapplicaties.
Trigger: versturen van code (typisch HTML).

SSD-28 Discreet commentaar					
Criterium (wie en wat)	De aan de gebruiker aangeboden scripts / code bevat geen commentaar dat tot misbruik kan leiden.				
Doelstelling (waarom)	Door zo min mogelijk technische informatie op te nemen in commentaar worden aanvallers minimaal geïnformeerd.				
Risico	Een aanvaller gebruikt technische informatie in commentaar om een manier te vinden voor misbruik van de applicatie.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.06				

Toelichting

Commentaar(regels) in scripts of code kunnen ongewild informatie vrijgeven. Webapplicaties serveren webtechnologieën zoals HTML-opmaak en client-side scripts (JavaScript) - vaak met 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 applicatie in productie gaat. Het commentaar kan echter ook informatie bevatten die gebruikt kan worden om zwakheden op te sporen. Denk hierbij aan informatie over gebruikte technologieën.

Conformiteitsindicatoren

/01 commentaar

SSD-28 Discreet commentaar	
	indicatoren
/01	commentaar
/01.01	De applicatieontwikkelaar heeft bewuste afweging gemaakt of deze technisch-georiënteerde commentaarregels uit de scripts (code) worden verwijderd.
/01.02	Dit hoort een bewuste afweging te zijn voor andere elementen dan commentaarregels, zoals META, PARAM, OBJECT en INPUT. Deze bevatten onnodig technische verwerkingsinformatie, zoals gebruikte software, versies of infrastructurele componenten.

Toelichting

/01 Tijdens deployment van een applicatie kan alle code die naar clients wordt gestuurd ontdaan worden van commentaar of code zelf worden ge-"minified". Als alternatief zijn application-level firewalls in staat om commentaar(regels) uit HTML- en scriptcode te verwijderen en zodoende 'gefilterde' data te sturen naar de client. 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. Het gaat dan vaak om het beschermen van auteursrecht of het verstrekken van de contactgegevens van de auteur. Het gebruik van commentaar is onder deze omstandigheden acceptabel, tenzij het commentaar de eisen uit de gebruiksvoorwaarden overstijgt.

Handreiking

Voorbeelden van hoe te testen:

- OWASP Web Security Testing Guide (WSTG): Information gathering

4.8.11 *SSD-32: Bescherming tegen (XXE) XML externe entiteit injectie*

Voor: alle software die XML gebruikt.
Trigger: verwerking van XML.

XML is naast JSON een veel gebruikt formaat om gegevens te structureren en in te lezen. Bij XML worden naar HMTL conventie de entiteiten (gegevens) weergegeven tussen een openings- en een sluitingstag.
Met de codes wordt de structuur en betekenis bepaald. Voor het uitlezen van de gegevens uit de XML-invoer wordt typisch gebruik gemaakt van componenten voor de interpreatie (*parsers*).

Deze eis is in opzet gelijk aan SSDm-19.

SSD-32: Bescherming tegen XML externe entiteit injectie					
Criterium (wie en wat)	De applicatie beperkt de mogelijkheid tot manipulatie door alle externe XML invoer te beschermen tegen entiteit injectie.				
Doelstelling (waarom)	Voorkomen van misbruik van de applicatie via de mogelijkheid voor verwijzen naar external entities in XML.				
Risico	Een aanvaller misbruikt de applicatie via XML injectie.				
Referentie	NCSC	NIST	ISO27002		

Conformiteitsindicatoren

/01 Externe XML invoer

SSD-32: XML externe entiteit injectie	
	indicatoren
/01	externe XML invoer
/01.01	Externe XML invoer (het ontvangen van externe/onvertrouwde XML bronnen) is expliciet niet toegestaan of wordt expliciet gevalideerd.
/01.02	Foute, ongeldige of verboden invoer wordt geweigerd.

Toelichting

/01.01 Via XXE zijn diverse soorten aanvallen mogelijk, waaronder inzage in lokale bestanden en ook het onbeschikbaar maken van systemen. De entiteiten in XML kunnen namelijk eindeloos in elkaar worden genest wat kan leiden tot een Denial of Service van de parser (vergelijkbaar met de klassieke billion laughs aanval). Dit leidt tot extra netwerkverkeer voor het steeds weer benaderen van de XML-bron.

/01.02 De invoer wordt verwijderd, dit betekent dus dat deze niet wordt geschoond en daarna alsnog wordt ingevoerd.

/02 Entiteit injectie

SSD-32: XML externe entiteit injectie	
	indicatoren
/02	Entiteit injectie

SSD-32: XML externe entiteit injectie	
	indicatoren
/02.01	Als externe XML invoer toch vereist is, wordt bij het aanroepen van de parser voor externe XML-bronnen wordt door de applicatie de entity resolver uitgezet. Zodoende zijn dan het parsen van de namespace en documentdefinities uitgeschakeld.

Toelichting voor mobiele apps
(Toevoeging zodat deze eis dan gelijk is aan de eis in SSDm) Android biedt voor XML voor drie soorten XML parsers: XMLPullParser, DOM en SAX. iOS biedt twee parsers: NSXMLParser en libxml2.

Handreiking

- OWASP wiki vulnerabilities: XML External Entity (XXE) processing
- OWASP cheat sheets:
  - Input validation
  - XML external entity injection
  - XML security
- OWASP Web Security Testing Guide (WSTG): Input validation testing

4.9 Externe componenten


Moderne applicaties maken steeds meer gebruik van externe componenten (open source of derde partijen). Dit biedt voordelen vanuit oogpunt van efficiëntie en ook beveiliging – want met name voor volwassen componenten is typisch veel aandacht besteed aan beveiliging. Aan de andere kant biedt dit risico's want er kunnen kwetsbaarheden aanwezig zijn in deze componenten en de kans dat die kwetsbaarheden worden ontdekt is groter vanwege de bredere verspreiding. Verstandige keuze van externe componenten en goed beheer is daarom van belang.

4.9.1 *SSD-3: Veilige externe componenten*

Voor: alle software met externe componenten.
Trigger: kiezen van externe componenten en regelmatig ter controle.

SSD-3 Veilige externe componenten					
Criterium (wie en wat)	Applicaties maken gebruik van veilige en actief onderhouden externe componenten.				
Doelstelling (waarom)	Zorgen dat de applicatie niet kwetsbaar wordt door kwetsbaarheden in externe componenten, of door logica die door kwaadwillenden in die componenten is geplaatst.				
Risico	Er wordt misbruik gemaakt door middel van kwaadaardige logica of kwetsbaarheden in externe componenten.				
Referentie	NCSC	NIST	ISO27002		
			12.6.1 14.2.5		

Toelichting

Externe componenten zijn van derde partijen (bijvoorbeeld open source) en worden tijdens de ontwikkeling (statisch) verwerkt in de applicatie, of tijdens uitvoering (dynamisch: zogenaamde mobile code – zoals Javacript, en DLLs.

Conformiteitsindicatoren

/01 veilige externe componenten

Bij keuze en gebruik van een extern component wordt zekerheid verkregen over de veiligheid van de code.

	Indicatoren
/01	veilige en actief onderhouden externe componenten
/01.01	De applicatie maakt alleen gebruik van een extern component als daarvan de oorsprong en de veiligheid is vast te stellen of redelijkerwijs kan worden aangenomen. Dit is ingeregeld in een Lifecycle Management plan tijdens ontwikkeling en onderhoud, inclusief actualisering en patching. Meestal wordt LCM technisch gefaciliteerd met dependency management tooling.
/01.02	Vervallen
/01.03	Indien gebruik van een externe component vereist is, waarvan de oorsprong of de veiligheid niet met zekerheid kan worden vastgesteld, wordt deze alleen in een besloten virtuele omgeving, 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. Dit suggereert de laatste stabiele versie zonder bekende zwakheden, conform bedrijfsbeleid.

/01.01 Er is basis voor vertrouwen in een component, bijvoorbeeld door een securitytest of een code review en/of onderzoek naar/inzicht in de reputatie en de activiteiten van de makers.

/01.01 De componenten voldoen elk aan de door de softwaremaker gestelde eisen met betrekking tot hoeveel versies/tijd is verstreken sinds er een nieuwere stabiele versie beschikbaar is. Hoe langer een versie achterloopt, des te groter de kans is op een kwetsbaarheid.

/01.03 De softwaremaker geeft in het ontwerp aan hoe dit is gerealiseerd en eventueel bij hosting moet worden geconfigureerd, als de afhankelijkheden dynamisch worden ingeladen.

Handreiking

- Maak indien mogelijk alleen gebruik van mobile code waarvan de bron bekend en vertrouwd is. In de praktijk wil je dan gebruik kunnen maken van versleuteld datatransport (https / TLS), checksums en digitale handtekeningen. Deze technieken garanderen niet dat de code zelf veilig is, maar wel dat de code redelijkerwijs van de bron afkomstig is.
- Er zijn verschillende tools beschikbaar waarop open source componenten op kwetsbaarheden getoetst kunnen worden: onder andere de OWASP dependency checker, en ook specifieke online diensten en javascript libraries zoals retirejs en auditjs.
- Package managers zijn tools die softwaremakers ondersteunen met het beheren van externe componenten.

Meer informatie en voorbeelden hoe hierop te testen:

- OWASP cheat sheets: Third party Javascript management
- OWASP Dependency check wiki: OWASP DependencyCheck broncode

4.10 Architectuurprincipes

Bij het ontwerpen en bouwen van applicaties zijn er veel zaken op architectuurvlak om rekening mee te houden, waarvan enkele al beschreven zijn in de diverse normen. Dit hoofdstuk richt zich op algemene beveiligingsprincipes in architectuur en beschrijft enkele normen op dit vlak.
Belangrijke principes zijn:

Defense-in-Depth & Zero Trust
Bij Defense-in-Depth (DiD) worden gelaagde maatregelen toegepast door niet te vertrouwen op een enkele maatregel. Daardoor is een zwakheid of kwetsbaarheid lastiger te misbruiken. Een aanvulling op het DiD principe is Zero Trust, waarbij het uitgangspunt is dat entiteiten en gebruiker binnen het eigen netwerk niet automatisch vertrouwd wordt. In een Zero Trust netwerk worden daarom verzoeken specifiek gemonitord en geverifieerd.

Isolatie
Isolatie van functionaliteit kan op meerdere niveaus zorgen voor een beperking van propagatie van een probleem. Een typische toepassing op netwerkniveau is segregeren, het groeperen van delen met verschillende (security-)eisen.
Conceptueel kan isolatie gezien worden als horizontale en verticale doorsnedes:

- Typische verticale isolatie op niveau van architectuur zijn silo's, die bijvoorbeeld een datastroom isoleren zonder tussentijdse zijstromen: dit soort isolatie van verzoek tot databron is typisch zichtbaar als microservices.
- Horizontale isolatie (layering) is het hiërarchisch structureren van een systeem in verschillende abstractieniveaus. Deze niveaus maken het mogelijk om functionaliteit bij elkaar te groeperen en op elkaar af te stemmen. Hierdoor ontstaan bundels van een hogere onderlinge cohesie en is functionaliteit eenvoudiger te valideren en kunnen per laag verschillende beveiligingsmaatregelen worden toegepast. Zie SSD-15 en 17.

Veerkracht
"Voorbereiden op het ergste" is gebruikelijk in de fysieke wereld: nooduitgangen, autogordels, overspanningsbeveiliging. Ook in IT gaan systemen kapot, code breekt en processen worden niet correct verwerkt. De opzet van een systeem moet daarom zorgen voor veerkracht (resilience): hoe gedraagt het zich onder onverwachte omstandigheden en hoe keert het weer in een normale staat terug. Overwegingen die bijdragen aan een goede veerkracht:

- Het systeem herkent (onverwachte) fouten en handelt ernaar. Technisch gezien zijn tenminste onderscheidbaar exceptions, errors, faults, en failures. Ze worden gelogd en netjes (gracefully) afgevangen zonder dat het systeem in instabiele staat komt of gevoelige informatie lekt.
- Wanneer een systeem niet meer reageert, vervalt het in een safe-state waarin verhoogde beveiliging normen gelden: als beschikbaarheid van het systeem wegvalt, moeten integriteit en vertrouwelijkheid (extra) gecontroleerd blijven.

Minimaal aanvalsoppervlak
Met het 'aanvalsoppervlak' van een systeem worden alle services en interfaces bedoeld die intern en extern toegankelijk zijn; niet alleen van de (web)applicatie, maar ook het besturingssysteem, de netwerkservices en protocollen. Het reduceren hiervan vermindert het aantal mogelijkheden om een systeem aan te vallen (de "vectoren"), en vermindert de complexiteit van het systeem waardoor de kans lager is op beveiligingsfouten.

4.10.1 *SSD-15: Scheiding Presentatie, Applicatie en Gegevens*

Voor: alle software.
Trigger: bij ontwerp.

SSD-15 Scheiding van presentatie, applicatie en gegevens					
Criterium (wie en wat)	De architectuur van een applicatie is gebaseerd op een gelaagde structuur door de presentatie-laag, de applicatielaag en de gegevens te scheiden, zodat de lagen beschermd kunnen worden binnen de netwerkzones.				
Doelstelling (waarom)	Scheiden van onderliggende systemen zodat deze elk afdoende afgeschermd kunnen worden voor onbevoegde toegang.				
Risico	Misbruik van de applicatie door onbevoegde toegang tot een onderliggend systeem dat onvoldoende afgeschermd kon worden omdat het geen gescheiden onderdeel is.				
Referentie	**NCSC**	**NIST**	**ISO27002**		

Toelichting
Zonering is een maatregel wanneer vertrouwensgrenzen in een netwerk worden overschreden. Voor applicaties die over internet ontsloten worden, vindt dat op netwerkniveau plaats door middel van een DMZ. Door deze compartimentering of gelaagdheid ook toe te passen in de architectuur van applicaties, wordt het compromitteren bemoeilijkt. 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	
	Indicatoren
/01	gelaagde structuur
/01.01	Implementaties van presentatie, applicatie en gegevens zijn gescheiden.
/01.02	Bedrijfskritische applicatielogica, vertrouwelijke gegevens of gegevens waarvan de onweerlegbaarheid moet 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

/01.02 De DMZ wordt als niet vertrouwde omgeving gezien, waardoor deze niet geschikt is voor de opslag van vertrouwelijke gegevens.

Handreiking

- OWASP cheat sheets:
  - AJAX security
  - Insecure Direct Object Reference Prevention

4.10.2 *SSD-17: Gescheiden beheerinterface /functionaliteit*

Voor: alle software met een beheerinterface.
Trigger: bij ontwerp.

SSD-17 Gescheiden beheerinterface					
Criterium (wie en wat)	Beheeractiviteiten vinden plaats via een van de standaard gebruikersinterface gescheiden beheerinterface.				
Doelstelling (waarom)	Scheiden van beheerfuncties ter voorkomen van onbevoegde toegang daartoe.				
Risico	Doordat beheerfuncties niet zijn gescheiden van normale applicatiefuncties verschaft een aanvaller toegang tot beheerfuncties door manipulatie van normale applicatiefuncties.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.02 U/PW.05	SC-2	9.4.4		

Toelichting

Voor het beheer van een applicatie kan een beheerinterface worden aangeboden. Door het onderscheid tussen beheer- en productiefunctionaliteit wordt voorkomen dat productie en beheerhandelingen door elkaar gaan lopen.
Een additioneel voordeel van het scheiden van een beheerinterface is dat deze uitsluitend kan worden opengesteld voor het interne netwerk – indien toegang via internet niet noodzakelijk is.

Conformiteitsindicatoren

/01 gescheiden beheerinterface

SSD-17 Gescheiden beheerinterface	
	indicatoren
/01	gescheiden beheerinterface
/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 communicatie/protocollen.

SSD-17 Gescheiden beheerinterface	
	indicatoren
/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.

/01.05 Bijvoorbeeld: implementeer beheer-clients op een beperkte groep werkstations, bijvoorbeeld in een afgeschermde ruimte en eventueel via een apart beheer-netwerk.

Handreiking

- OWASP Web Security Testing Guide (WSTG): Configuration and deployment management

4.11 Infrastructuur

4.11.1 *SSD-1: Hardening van technische componenten*

Voor: alle software.
Trigger: bij inrichting infrastructuur en regelmatig ter controle.

SSD-1 Hardening					
Criterium (wie en wat)	De applicatie voldoet aan het hardeningbeleid. De software en het platform zijn daartoe geconfigureerd volgens de bijbehorende hardeningrichtlijn. Het configureren is procesmatig en procedureel ingericht.				
Doelstelling (waarom)	De geleverde dienst beveiligen tegen misbruik via zwakheden door het beperken van de functionaliteit tot hetgeen noodzakelijk is voor het correct functioneren van de applicatie.				
Risico	Door manipulatie van onnodig beschikbare technische faciliteiten/services wordt misbruik van de applicatie gemaakt.				
Referentie	NCSC	NIST	ISO27002		
	U/WA.07 U/WA.07 U/PW.01 U/PW.06-8 U/NW.06		14.2.4		

Toelichting

De meeste computer- en netwerkapparatuur en softwarepakketten bevatten meer functionaliteit dan een organisatie nodig heeft, wat tot een onnodige kwetsbaarheid kan leiden. Hardenen heeft als doel het aanvalsoppervlak te verkleinen tot het strikt noodzakelijke om aanvalsmogelijkheden te verminderen. Hardenen kan door verwijderen, uitschakelen, onbereikbaar maken of beperken van functionaliteiten en technische communicatie-openingen. Voorbeelden zijn technische services, communicatie-protocollen, software, gebruikersaccounts en systeemdiensten.
Dit vraagt ten eerste het in kaart brengen van de samenstelling hiervan en onderlinge afhankelijkheden.

Conformiteitsindicatoren

/01 hardeningsbeleid

SSD-1 Hardening	
	indicatoren
/01	hardeningsbeleid
/01.01	Er zijn voorschriften voor hardening en patching van ICT-componenten.

/02 hardeningrichtlijn

SSD-1 Hardening	
	indicatoren
/02	hardeningsrichtlijn
/02.01	De softwaremaker 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 ingericht 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.
- beschrijft en wijst naar automatiseringstappen die hardening zelf makkelijk maken (bijv. het draaien van een daemon op zoek naar ongewenste draaiende processen, poortscans, dependency-update-tooling, configuratie-standaarden/defaults van software en systeem-images).

Algemeen: het Center for Internet Security (CIS) stelt voor infrastructuur benchmarks en best practices beschikbaar die kunnen helpen om een hardeningsrichtlijn op te stellen.

/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	
	indicatoren
/03	Procesmatig en procedureel
/03.01	ICT-componenten zijn (aantoonbaar) volgens de instructies en procedures van de softwaremaker ingericht.
/03.02	ICT-componenten ondersteunen alleen die services die vanuit het ontwerp noodzakelijk zijn, andere services zijn gedeactiveerd of verwijderd.

SSD-1 Hardening	
	indicatoren
/03.03	ICT-componenten zijn voorzien van de meest recente patches, tenzij er een aanwijsbare en gedocumenteerde reden is dat dit niet zo is en de opdrachtgever hier schriftelijk mee heeft ingestemd.
/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 door de opdrachtgever.
/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 /03.05).

Toelichting

/03.01 Bij voorkeur wordt de controle door een geautomatiseerd proces uitgevoerd. In dat geval volstaat het noteren van het versienummer van de gebruikte software en eventuele parameters. Het toepassen van de vervolgstappen is daarmee herhaalbaar. Vaste of periodieke handmatige controles zijn onvermijdelijk.

/03.01 Neem in de werkinstructies het toepassen van de instructies en procedures van de softwaremaker 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 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 aantoonbaar nodig zijn volgens gebruikersvereisten of technische zekerheden. Schakel "perifere" services uit. Verwijder zo mogelijk deze services van de software-component (de-installatie).

/03.03 Sluit aan bij een notificatieservice 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 software-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, met name in geval van een klant-leveranciersrelatie.

Handreiking

Meer informatie en voorbeelden:

- OWASP cheat sheets: Docker security
  OWASP Web Security Testing Guide (WSTG):
  - Configuration and deployment management
  - Information gathering

4.11.2 *SSD-26: Beperkte HTTP-methoden*

Voor: webapplicaties en http-backends.
Trigger: bij configureren webserver.

SSD-26 Beperkte HTTP-methoden					
Criterium (wie en wat)	De webserver faciliteert alleen de HTTP-functionaliteiten die nodig zijn voor het functioneren van de applicatie.				
Doelstelling (waarom)	Voorkom het gebruik van niet noodzakelijke methoden, die anders tot misbruik kunnen leiden.				
Risico	Door misbruik van een onnodig geaccepteerd HTTP commando wordt de applicatie misbruikt.				
Referentie	NCSC	NIST	ISO27002		
	U/PW.02				

Toelichting

De webserver ondersteunt het HTTP-protocol. HTTP kent methoden, headers en foutinformatie, die mogelijk functioneel misbruikt kunnen worden. Daarom is het gebruik hiervan beperkt tot het minimum dat noodzakelijk is voor de goede werking van de ontsloten applicaties.
HTTP 1.1 en 2.0 ondersteunen verschillende functionaliteiten. In de praktijk gebruikt een applicatie vooral de functies GET en POST.
Afhankelijk van functionele eisen kunnen andere verbs/werkwoorden worden gebruikt. Deze vragen een bewuste afweging omdat ze niet allemaal "safe" zijn (staat-veranderend) en "idempotent" (bereikt dezelfde eindstaat als deze meerdere keren verwerkt wordt). Deze toegestane methoden kunnen bijvoorbeeld zijn gedefinieerd in een REST API.

Conformiteitsindicatoren

/01 HTTP-functionaliteiten

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

Toelichting

/01.01 Methoden anders dan GET en POST zijn vrijwel nooit nodig binnen traditionele applicaties en vormen alleen een extra beveiligingsrisico (misbruik).

/01.03 Het is in alle gevallen aan te raden om alleen benodigde HTTP-methoden toe te staan (whitelisting) via configuratie van de webserver of via de application level firewall (waar de TLS offloading plaatsvindt, want anders kan de inhoud niet gelezen worden).

Handreiking

- OWASP cheat sheets: REST security
- OWASP Web Security Testing Guide (WSTG):
  - Configuration and deployment management
  - Information gathering

4.11.3 *SSD-29: Voorkom directory listing*

Voor: webapplicaties en http-backends.
Trigger: configureren webserver of implementatie van toegang tot directory-structuur.

SSD-29 Voorkom directory listing					
Criterium (wie en wat)	De aan de gebruiker getoonde informatie bevat geen directory listings, zodat die niet kunnen worden misbruikt.				
Doelstelling (waarom)	Voorkom dat inzage wordt gekregen in de inhoud van directories zodat aanvallers minimaal worden geïnformeerd.				
Risico	Een aanvaller gebruikt technische informatie, verkregen door de inhoudsbeschrijving van directories te zien, voor misbruik van de applicatie, of ziet vertrouwelijke informatie daarin.				
Referentie	NCSC	NIST	ISO27002		
	U/PW.03				

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.

Conformiteitsindicatoren

/01 directory listings

SSD-29 Voorkom directory listing	
	indicatoren
/01	directory listings
/01.02	De softwaremaker 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 softwaremaker legt in de configuratiebeschrijving vast welke directory listings niet uitgeschakeld moeten worden.
/01.04	De hostingpartij schakelt, met uitzondering van de in de configuratiebeschrijving vastgelegde uitzonderingen, de directory listings uit.

Toelichting

/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.

Handreiking

- OWASP cheat sheets:
  - Insecure Direct Object Reference Prevention
  - REST security

Voorbeelden hoe te testen:

- OWASP Web Security Testing Guide (WSTG):
  - Configuration and deployment management
  - Information gathering

4.11.4 *SSD-31: Standaard stack*

Voor: alle software.
Trigger: bij inrichten infrastructuur en regelmatig ter controle.

SSD-31 Standaard stack					
Criterium (wie en wat)	De (web-)applicatie(-omgeving) maakt gebruik van systeemcomponenten en voorzieningen die onderdeel zijn van een formeel gespecificeerde stack.				
Doelstelling (waarom)	Een hoger beveiligingsniveau bereiken door standaardisatie van gebruikte technologie.				
Risico	Het gebruik van onbekende componenten of diensten kan leiden tot nieuwe en onbekende zwakheden en risico's die tot misbruik leiden.				
Referentie	NCSC	NIST	ISO27002		
			14.2.4		

Toelichting

De verschillende lagen van componenten en technische voorzieningen vormen de *stack* voor een softwarematige oplossing. Het voordeel van een standaard stack is dat componenten en voorzieningen hebben bewezen in de praktijk op elkaar aan te sluiten, waarbij zwakheden bekend zijn en geadresseerd. Stacks waarmee geen ervaring is kunnen voor nieuwe en onbekende risico's zorgen. Zwakheden kunnen zich in technische vorm voordoen, maar ook 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 voorzieningen

SSD-31 Standaard stack	
	indicatoren
/01	systeemcomponenten en voorzieningen
/01.01	Voor het vastleggen en het beheren van de standaard stack is één partij aangewezen.
/01.02	De softwaremaker maakt uitsluitend gebruik van systeemcomponenten en voorzieningen die zijn opgenomen in de standaard stack.
/01.03	De softwaremaker 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 voorzieningen 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 softwaremaker mag voor de ontwikkeling van een 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.

4.11.5 *SSD-33: Veilige HTTP response headers*

Voor: webapplicaties en http-backends.
Trigger: bij configureren webserver voor response headers en cookies.

SSD-33: Veilige HTTP response headers					
Criterium (wie en wat)	De applicatie maakt gebruik van veilige response headers.				
Doelstelling (waarom)	Maak gebruik van voorzieningen in gebruikers endpoint/app/moderne browsers om aanvallen af te slaan.				
Risico	Misbruik van de applicatie slaagt doordat de gebruikers endpoint/app/browser onvoldoende bijdraagt aan beveiliging.				
Referentie	NCSC	NIST	ISO27002		
	U/PW.03				

Toelichting

Met HTTP response headers kunnen webservers configuratie sturen naar de browser die de veiligheid van de applicatie verhoogt. Deze headers voorkomen dat gebruikers van moderne browsers, slachtoffer worden van kwetsbaarheden, die de browser eenvoudig kan voorkomen.

Omdat de headers worden ondersteund door alleen moderne browsers, kan de applicatie kwetsbaar zijn bij gebruikers met verouderde browsers.

Conformiteitsindicatoren

/01 veilige response headers

SSD-33: Veilige HTTP response headers	
	indicatoren
/01	veilige response headers
/01.01	De server responses bevatten de in de toelichting genoemde veilige response headers met de aanbevolen waarden.

Toelichting

/01.01 Het gebruik van de headers in onderstaande tabel wordt aanbevolen.

Header	Omschrijving
Strict-Transport-Security	HTTP Strict Transport Security (HSTS) is een mechanisme dat websites en -applicaties helpt beschermen tegen *downgrade attacks* en *cookie hijacking*. Het geeft webservers de mogelijkheid om webbrowsers (of andere *user agents*) uitsluitend verzoeken over veilige HTTPS verbindingen te verzenden en nooit meer via het onveilige HTTP protocol.
X-Frame-Options	X-Frame-Options response header verbetert de bescherming tegen Clickjacking. De header geeft aan of de browser de inhoud van andere webpagina's in frames, wel of niet moet weergeven (met sameorigin).
X-XSS-Protection	Deze header zet het Cross-site scripting (XSS) filter aan in de browser, mits beschikbaar en gerespecteerd in de browser.
X-Content-Type-Options	Forceert typering: voorkomt dat de browser zelf probeert het MIME-type van een file vast te stellen en de het bestand anders interpreteert dan is aangegeven in het content-type. *Nosniff* is een gebruikelijke uitbreiding om de content-type-options.
Content-Security-Policy	Een Content Security Policy (CSP) biedt gedeeltelijke bescherming tegen een groot aantal verschillende aanvallen inclusief Cross Site Scripting/injecties, met name rond verzoeken van andere domeinen en uitvoer van scripting logica.
Referrer-Policy	De Referrer-Policy HTTP header geeft aan welke referrer informatie bij het verzenden van het verzoek in de Referrer header wordt opgenomen.
Expect-CT	Expect-CT dwingt Certificate Transparency af en detecteert certificaten die ten onrechte zijn uitgegeven.

Set-Cookie	Set-Cookie biedt mogelijkheden om cookies te beschermen tegen onderschepping en cross site scripting. Aanbevolen is: • De flag 'HttpOnly' zorgt ervoor dat de cookie uitsluitend door de browser kan worden gelezen en niet door javascript, wat een externe oorsprong kan hebben (in geval van XSS). • De flag 'secure' limiteert de communicatie van cookies tot een beveiligde verbinding via HTTPS. • Het effect van 'Samesite' is dat moderne browsers geen of minder cookies versturen als de applicatie wordt aangeroepen vanaf een ander domein. Hiermee wordt de kans op CSRF (Cross-Site Request Forgery) gereduceerd. De waarde 'Strict' is de veilige standaard waarde en als een risico-analyse een lager niveau van beveiliging toelaat, dan kan 'Lax' overwogen worden.

Er zijn technische beperkingen aan deze flags door de complexe samenhang van browsers, cookies en javascript. Bijvoorbeeld de beperking van de 'httponly' flag: De 'httponly flag' probeert te beschermen tegen XSS waarbij een aanvaller met JavaScript een cookie met bijvoorbeeld een sessiegeheim kan uitlezen (om te stelen). Wanneer een aanvaller in een XSS-aanval met javascript een verzoek/XHR kan forceren, wordt ondanks een "httponly" flag de sessie-token uit het cookie alsnog meegestuurd, en in sommige gevallen komt die vervolgens terug als antwoord door de server.
JSON Web Tokens, als zelfbeschrijvende authenticatie/autorisatie berichten, zijn een modern alternatief voor sessie-identifiers in cookies en zijn zeer gangbaar in web-applicaties die stateless communiceren over REST APIs. Overige extensies kunnen worden toegepast met de kennis dat ze beperkte garantie bieden, afhankelijk of ze door browsers gerespecteerd worden, bijvoorbeeld:

- Definitie van HTTP methods in de (preflight) onderhandeling tussen server en gebruiker (Access-Control-Allow-Methods als antwoord op de browser vraag Access-Control-Request-Methods). Verwachte http verbs moeten sowieso gedefinieerd zijn in interfaces waarbij de gebruiker geen voorkennis heeft (d.w.z. REST/statelessness).

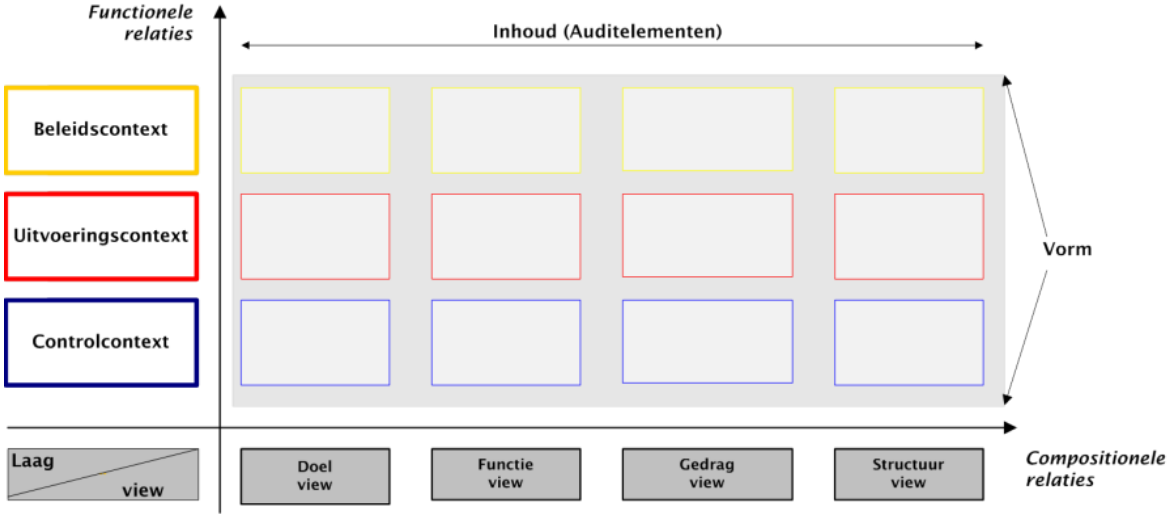
Handreiking

- OWASP wiki: HTTP verb tampering
- OWASP cheat sheets:
  - Clickjacking defense
  - Content Security Policy
  - Cross-Site Request Forgerry prevention
  - HTTP Strict Transport Security
  - Insecure Direct Object Reference Prevention
  - JSON Web Token
  - REST security
- OWASP Web Security Testing Guide (WSTG):
  - Client-side testing
  - Configuration and deployment management
  - Information gathering

Bijlage 1: De SIVA-methode voor het opstellen van beveiligingseisen

Het raamwerk

Deze derde 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 laat zien wat de context is.
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.



Figuur: 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 formuleringsvoorschrift (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 zijn:

- *doel* – het waarom-aspect
  De bestaansreden van een organisatie.
  Voorbeelden: organisatie, visie, doelstellingen, wet/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 aandachts-gebieden (basiselementen) voor applicaties afdekken en daarmee de risicogebieden afdekken.

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

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 SIVA geeft een formule (syntax) weer voor de normen:

Predicaat	(	object 1,	object 2,	object 3	)
Actietype	(	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.

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

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.

Het gebruikte template voor de normen is:

SSD-nr Onderwerp van de norm					
Criterium (wie en wat)	Wat (xxxxx) <werkwoord> xxxxx trefwoorden xxxxx				
Doelstelling (waarom)	De reden waarom de norm gehanteerd wordt.				
Risico	Het risico dat de aanleiding vormt om de norm te hanteren.				
Referentie	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	
	indicatoren
/01	trefwoord
/01.01	(conformiteits)indicator 1.1
/01.01	(conformiteits)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.

Bijlage 2: Wijzigingen ten opzichte van de versie 2

Hieronder volgt een overzicht van de belangrijkste wijzigingen die in deze versie (3.0) van Grip op SSD Beveiligingseisen zijn doorgevoerd. De oude versie (2.0) blijft nog beschikbaar op de site www.cip-overheid.nl voor organisaties die op basis daarvan nog lopende afspraken hebben.

Overzicht van de wijzigingen

- Honderden verbeteringen en verduidelijkingen in tekst:
  - Verzamelde input van de community is verwerkt.
  - Reviews door de werkgroepleden.
  - Meer duidelijke teksten. SSD-2 is bijvoorbeeld van titel veranderd naar "Veilige gegevensopslag" omdat het eigenlijk daarover ging. SSD-3 is om dezelfde reden van titel veranderd naar "Veilige externe componenten".
- Meer opgesteld als samenhangende gids:
  - De normen waren eerder willekeurig geordend. In v3.0 zijn de normen gestructureerd naar verantwoordelijkheden en ook zo benoemd, zodat het document de lezer meer aan de hand neemt hoe de verschillende verantwoordelijkheden worden ingevuld.
  - Per norm is beschreven in welke situaties de norm van toepassing is.
- Per eis zijn een aantal klikbare verwijzingen opgenomen naar praktische handreikingen/details.
- Gapanalyse is uitgevoerd met OWASP ASVS en enkele zaken zijn aangevuld.
- Up to date gebracht met onder meer nieuwe ISO27002- en NCSC-richtlijnen. Verwijzingen zijn aangebracht.
- SSD-6 (interne gebruikers) is opgegaan in SSD-5 die nu gaat over alle gebruikers.
- SSD-10 Concurrent Session Control is vervallen.
- SSD-12A Session lock Vervalt als eis (staat geheel los van de applicatie).
- SSD-11: System use notification is verwijderd. Dit is slechts in speciale gevallen een criterium - die dan typisch al onderdeel uitmaakt van de functionele eisen.
- SSD-18 is vervallen vanwege duplicatie in onder meer SSD-22.
- SSD-25 Beperken van te tonen headers is opgenomen in SSD-24 Beperken van te sturen headers.
- SSD-32 is nieuw en bevat alle normen voor het configureren van HTTP response headers. Eerder stond dat overal verspreid.
- SSD-33 (XML injectie) toegevoegd omdat deze ook in de SSD-mobiele eisen staat en in de laatste versie van de OWASP top 10 wordt genoemd omdat het een veel voorkomend issue is.