|  |
| --- |
| Standaard Informatiebeveiliging in Software OntwikkelingTemplate |

|  |  |
| --- | --- |
| Auteur(s): |  |
| Versie: |  |
| Datum: | Kies of typ een datum |
| Kenmerk: |  |
|  |  |
|  |  |

**Documentinformatie**

Dit document maakt onderdeel uit van een complete set (beleidspiramide) met formeel vastgestelde documenten op strategisch, tactisch en operationeel niveau. Dit document heeft betrekking op de laag Standaard in de beleidspiramide.



**Versiebeheer**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Auteur** | **Verwerking** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Distributielijst**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Ontvanger** | **Doel** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Vaststelling**

|  |  |  |  |
| --- | --- | --- | --- |
| **Versie** | **Datum** | **Vastgesteld door** | **Vastgesteld op** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Samenhang met andere documenten**

|  |  |  |  |
| --- | --- | --- | --- |
| **Naam** | **Bovenliggend** | **Gelijk niveau** | **Onderliggend** |
| [INFORMATIEBEVEILIGINGSBELEID] | x |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Verwijzingen naar SURFaudit Toetsingskader en ISO27001**

|  |  |
| --- | --- |
| **Kader** | **Verwijzing (tags)** |
| SURFaudit Toetsingskader | SD.01, SD.02 |
| ISO27001 |  |

**Creative Commons**

Dit template is een product van het SURF Security Expertise Centrum en beschikbaar onder de licentie Creative Commons Naamsvermelding 4.0 Internationaal. <https://creativecommons.org/licenses/by/4.0/deed.nl>

Inhoudsopgave

|  |
| --- |
| [Samenvatting 4](#_Toc202352220)[1 Inleiding 5](#_Toc202352221)[1.1 Doelstelling 5](#_Toc202352222)[1.2 Reikwijdte en toepassingsgebied 5](#_Toc202352223)[1.3 Rollen en verantwoordelijkheden 5](#_Toc202352224)[1.4 Definities en termen 6](#_Toc202352225)[2 Standaard 7](#_Toc202352226)[2.1 Principes 7](#_Toc202352227)[2.2 Ontwikkelproces 7](#_Toc202352228)[2.3 Veilig programmeren en coderen 8](#_Toc202352229)[2.4 Beveiligingstesten 8](#_Toc202352230)[2.5 Toegangsbeheer 9](#_Toc202352231)[2.6 Open source en andere componenten 9](#_Toc202352232)[2.7 Bescherming van persoonsgegevens en privacy 9](#_Toc202352233)[2.8 Scheiding van omgevingen 10](#_Toc202352234)[2.9 Uitbesteding en leveranciers 10](#_Toc202352235)[2.10 Risicobeoordeling en -beheer 10](#_Toc202352236)[2.11 Logging en monitoring 11](#_Toc202352237)[2.12 Bewustwording en training 11](#_Toc202352238)[3 Rapportage en borging 12](#_Toc202352239)[3.1 Rapportage 12](#_Toc202352240)[3.2 Borging 12](#_Toc202352241)[4 Vaststelling 13](#_Toc202352242)[Bijlage 1 Rapportage Toepassing Standaard Informatiebeveiliging Softwareontwikkeling 14](#_Toc202352243)[1.1 Algemene informatie 14](#_Toc202352244)[1.2 Toepassing van de standaard 14](#_Toc202352245)[1.3 Bevindingen en risico’s 14](#_Toc202352246)[1.4 Goedkeuring 14](#_Toc202352247) |

Samenvatting

Hier komt een samenvatting, op zichzelf leesbaar, van de inhoud van dit document in max. 1 A4.

# Inleiding

Deze standaard maakt onderdeel uit van het informatiebeveiligingsbeleid van onze instelling en beschrijft de formele principes en beveiligingsvereisten voor softwareontwikkeling. Het document is bedoeld om ervoor te zorgen dat software die door of voor de instelling wordt ontwikkeld, voldoet aan actuele normen voor informatiebeveiliging en privacybescherming.

## Doelstelling

Het doel van deze standaard is het vastleggen van beveiligingsprincipes en vereisten die tijdens het volledige ontwikkelproces van software moeten worden toegepast. De standaard ondersteunt ontwikkelteams en opdrachtgevers bij het structureel borgen van veilige softwareontwikkeling, gebaseerd op erkende normen en best practices.

## Reikwijdte en toepassingsgebied

Deze standaard is van toepassing op alle software die door medewerkers of studenten van onze instelling wordt ontwikkeld, software die specifiek in opdracht van onze instelling voor intern gebruik wordt gerealiseerd, en software die binnen de IT-omgeving van onze instelling wordt beheerd. Dit geldt ongeacht of de software open source, closed source of hybride is. De standaard heeft betrekking op de volledige softwarelevenscyclus: van ontwerp en ontwikkeling tot testen, oplevering en onderhoud.

## Rollen en verantwoordelijkheden

|  |  |
| --- | --- |
| Rol | Verantwoordelijkheden |
| Product Owner | Zorgt dat beveiligingseisen onderdeel zijn van de projectscope en acceptatiecriteria. |
| Ontwikkelaar | Past veilige ontwikkelpraktijken toe en volgt richtlijnen en standaarden. |
| QA/Test Engineer | Test beveiligingsaspecten en rapporteert afwijkingen. |
| CISO / Security Officer | Toetst op naleving van deze standaard en ondersteunt teams bij implementatie. |
| Opdrachtgever / Manager | Stelt kaders vast, beoordeelt risico's en faciliteert compliance. |
| Inkoop / contractbeheer | Ziet toe op opname van beveiligingseisen bij externe opdrachtverlening. |

## Definities en termen

|  |  |
| --- | --- |
| Begrip | Definitie |
| SDLC | Software Development Lifecycle – het volledige ontwikkelproces van ontwerp tot onderhoud. |
| Secure coding | Ontwikkelpraktijken gericht op het voorkomen van kwetsbaarheden in softwarecode. |
| SAST / DAST | Statische en dynamische analysetechnieken om software op kwetsbaarheden te testen. |
| OWASP | Open Web Application Security Project – bron van best practices en beveiligingsstandaarden. |
| CVE / CVSS | Common Vulnerabilities & Exposures (CVE), Common Vulnerability Scoring System (CVSS). Internationaal systeem voor het melden en scoren van kwetsbaarheden in software. |
| DPIA | Data Protection Impact Assessment – beoordeling van privacyrisico’s bij gegevensverwerking. |
| MFA | Multi-Factor Authenticatie – toegangsbeveiliging met meerdere verificatiemechanismen. |

# Standaard

Dit hoofdstuk beschrijft de principes en vereisten die gelden voor softwareontwikkeling binnen de instelling. Doel is om structureel informatiebeveiliging en privacy te waarborgen gedurende het gehele ontwikkelproces, ongeacht ontwikkelmethode of gebruikte technologie.

## Principes

* **Security en Privacy by design**

Beveiliging en privacy worden vanaf het begin integraal meegenomen in het ontwerp van de software. Persoonsgegevens worden alleen verwerkt als dat noodzakelijk is, met passende bescherming, en beveiligingsmaatregelen zijn structureel ingebed in alle fasen van het ontwikkelproces.

* **Security en Privacy by default**

Software wordt standaard veilig en privacyvriendelijk ingericht: met veilige standaardinstellingen, minimale toegangsrechten en dataminimalisatie.

* **Least privilege**

Toegang is beperkt tot personen die uit hoofde van hun functie erbij moeten en beperkt tot wat noodzakelijk is, dit geldt zowel voor de ontwikkelomgeving als voor de code.

* **Verantwoording**

Wijzigingen in code en configuratie zijn traceerbaar en verantwoord.

* **Toetsbaarheid**

Beveiliging is aantoonbaar getest en geborgd in alle fasen van ontwikkeling.

* **Beveilig de ontwikkelomgeving**

Zorg voor een veilige ontwikkelomgeving door het minimaliseren van het aanvalsoppervlak en het implementeren van passende beveiligingsmaatregelen.

* **Beveilig de build- en deploymentpipeline**

Bescherm de integriteit van het build- en deploymentproces tegen ongeautoriseerde wijzigingen en toegang.

* **Bescherm de coderepository**

Implementeer toegangscontrole en monitoring op coderepositories om ongeautoriseerde toegang en wijzigingen te voorkomen.

* **Produceer schone en onderhoudbare code**

Schrijf duidelijke, begrijpelijke en goed gedocumenteerde code om fouten te verminderen en onderhoud te vergemakkelijken.

* **Beveiliging is ieders verantwoordelijkheid**

Bevorder een cultuur waarin alle teamleden zich verantwoordelijk voelen voor beveiliging en actief bijdragen aan een veilig ontwikkelproces.

* **Test beveiliging continu**

Integreer beveiligingstests in het ontwikkelproces en voer regelmatig tests uit om nieuwe kwetsbaarheden te identificeren en te verhelpen.

## Ontwikkelproces

Beveiliging moet worden geïntegreerd in elke fase van de softwareontwikkelcyclus (SDLC). Dat vraagt om duidelijke scheiding van omgevingen, controlemechanismen en borging in project- en releaseprocessen.

**Vereisten**

* Beveiligingseisen worden vastgesteld tijdens de specificatie- en ontwerpfase van elk project.
* Acceptatiecriteria bevatten beveiligingstests of -audits als expliciet onderdeel.
* Een verantwoordelijke uit het ontwikkel- of projectteam (zoals benoemd in tabel 1.3) borgt de toepassing van deze standaard per softwareontwikkeling. De CISO/ISO houdt toezicht op naleving op organisatieniveau.

## Veilig programmeren en coderen

Het veilig programmeren en coderen is van belang om kwetsbaarheden in software te voorkomen. Ontwikkelaars dienen zich te houden aan erkende richtlijnen en standaarden voor ‘secure coding’.

**Vereisten**

* Volg de OWASP Secure Coding Practices[[1]](#footnote-2) en de OWASP Top 10[[2]](#footnote-3) om veelvoorkomende beveiligingsrisico's te vermijden.
* Kwetsbaarheden in de broncode worden aantoonbaar vroegtijdig opgespoord met statische analysetools (SAST).
* Implementeer inputvalidatie en output-encoding om injectieaanvallen te voorkomen.
* Toegang van applicaties tot beveiligde systemen wordt verleend conform onze ‘Standaard Identiteit- en Toegangsbeheer’.
* Gevoelige informatie, zoals wachtwoorden, connection strings en API-sleutels, komt niet voor in broncode en repositories
* Wachtwoorden en API-sleutels verschillen per omgeving en worden dus niet hergebruikt.
* Pas het principe van 'least privilege' toe bij het toekennen van rechten binnen de applicatie.

## Beveiligingstesten

Beveiligingstests worden uitgevoerd om de effectiviteit van beveiligingsmaatregelen te verifiëren en kwetsbaarheden op te sporen.

**Vereisten**

* Beveiligingstests worden uitgevoerd met geautomatiseerde testtools die consistentie en volledigheid ondersteunen.
* Voer regelmatig dynamische applicatiebeveiligingstests (DAST) uit om runtime-kwetsbaarheden te identificeren.
* Bij grote updates of nieuwe releases worden aantoonbaar penetratietests uitgevoerd, met prioriteit voor systemen die kritieke processen ondersteunen of als kroonjuwelen zijn aangemerkt.
* Het softwareteam is verantwoordelijk voor het registreren, classificeren en opvolgen van gevonden kwetsbaarheden.
* Kwetsbaarheden worden beoordeeld op impact en prioriteit en binnen redelijke termijn verholpen.
* Testresultaten zijn vastgelegd en worden aantoonbaar opgevolgd.

## Toegangsbeheer

Toegang tot ontwikkelomgevingen en versiebeheersystemen is beheerst en volgens onze ‘Standaard Identiteit- en Toegangsbeheer’ ingericht.

**Vereisten**

* Implementeer multi-factor authenticatie (MFA) voor toegang tot ontwikkeltools en repositories volgens de ‘Richtlijn MFA’.
* Gebruik role-based access control (RBAC) om toegang te beperken tot noodzakelijke functies.
* Zorg voor logging en monitoring van toegang en wijzigingen in versiebeheersystemen volgens de ‘Standaard Logging en Monitoring’.
* Toegangsrechten worden periodiek geëvalueerd en waar nodig aangepast op basis van rollen en functies.

## Open source en andere componenten

Het gebruik van open source componenten brengt specifieke beveiligingsrisico's met zich mee die actief beheerd moeten worden.

**Vereisten**

* Van alle gebruikte open source componenten en hun versies is een actueel overzicht.
* Software composition analysis (SCA) tools worden gebruikt om kwetsbaarheden in afhankelijkheden te detecteren.
* Softwarecomponenten zijn aantoonbaar up-to-date en niet verouderd of zonder actief onderhoud.
* Het gebruik van verouderde of niet-onderhouden libraries wordt vermeden.
* Licentievoorwaarden van open source en andere componenten worden gecontroleerd op compliance-risico's.
* Nieuwe open source en andere componenten worden voor gebruik beoordeeld en goedgekeurd.

## Bescherming van persoonsgegevens en privacy

Bij de ontwikkeling van software die persoonsgegevens verwerkt, moeten privacyaspecten vanaf het begin worden meegenomen.

**Vereisten**

* Bij nieuwe verwerkingen of significante wijzigingen wordt een Data Protection Impact Assessment (DPIA) uitgevoerd.
* Privacy by design en privacy by default principes zijn geborgd in de softwarearchitectuur.
* Verzameling van persoonsgegevens alleen op basis van geldende grondslag en is beperkt tot wat noodzakelijk is voor het doel van de verwerking.
* Passende beveiligingsmaatregelen zijn ingericht om de vertrouwelijkheid en integriteit van persoonsgegevens te waarborgen.
* Verwerkingsactiviteiten en bewaartermijnen zijn vastgelegd in de betreffende registers conform de AVG.

## Scheiding van omgevingen

Het gescheiden houden van ontwikkel-, test-, acceptatie- en productieomgevingen is belangrijk voor het voorkomen van fouten, ongeautoriseerde toegang en incidenten in productie.

**Vereisten**

* Ontwikkel-, test-, acceptatie- en productieomgevingen zijn fysiek en logisch van elkaar gescheiden.
* Productiegegevens worden niet gebruikt in ontwikkel-, test- of acceptatieomgevingen, tenzij geanonimiseerd.
* Voor toegang tot productieomgevingen worden aparte, specifiek geautoriseerde (beheer)accounts gebruikt die niet worden ingezet voor ontwikkel-, test- of acceptatieomgevingen, en vice versa.
* Alleen geautoriseerde gebruikers hebben toegang.
* Veranderingen in productieomgevingen verlopen uitsluitend via gecontroleerde deploymentprocedures.

## Uitbesteding en leveranciers

Bij het uitbesteden van softwareontwikkeling is het van belang om beveiligingseisen expliciet op te nemen in contracten en toezicht te houden op de naleving ervan.

**Vereisten**

* Beveiligingseisen zijn opgenomen in aanbestedingsdocumenten en contracten.
* Beveiligingsmaatregelen van leveranciers worden beoordeeld incl. bewijs van compliance, zoals ISO 27001-certificering.
* Duidelijke afspraken over eigendom van code, toegang tot repositories en overdracht van kennis zijn geborgd.
* De mogelijkheden tot periodieke audits bij leveranciers worden uitgevoerd om naleving van beveiligingseisen te verifiëren. Bij structurele tekortkomingen zonder verbeterpotentieel kan de samenwerking worden herzien of beëindigd.
* Exitstrategieën om continuïteit en beveiliging te waarborgen bij beëindiging van de samenwerking zijn geïmplementeerd.

## Risicobeoordeling en -beheer

Het toepassen van beveiliging in softwareontwikkeling vereist een risicogebaseerde aanpak. Voor alle softwareprojecten moet worden vastgesteld welke risico’s relevant zijn voor de toepassing, de gegevensverwerking en de beoogde gebruikersgroep.

**Vereisten**

* Beveiligingseisen zijn gebaseerd op een voorafgaande risicobeoordeling.
* Bij wijzigingen aan bestaande software moet de impact op informatiebeveiliging en bescherming van persoonsgegevens worden beoordeeld.
* Risicoanalyses worden uitgevoerd conform de methode die binnen onze instelling is vastgesteld (Beleid Risicobeheer en onderliggende documenten).
* Bevindingen uit de risicobeoordeling en de mitigerende maatregelen zijn aantoonbaar vastgelegd en afgestemd met betrokken stakeholders.

## Logging en monitoring

Om beveiligingsincidenten tijdig te kunnen detecteren en analyseren, moet software voorzien zijn van adequate logging en monitoring op beveiligingsrelevante gebeurtenissen.

**Vereisten**

* Applicaties loggen toegangspogingen, systeemfouten, wijzigingen in code of configuratie, en andere beveiligingsrelevante gebeurtenissen, zoals pogingen om beveiligingsinstellingen te verzwakken.
* Logs bevatten voldoende detail om misbruik of aanvallen te reconstrueren.
* Logging voldoet aan bewaartermijnen, integriteits- en toegangsvereisten zoals gesteld in onze Standaard Logging en Monitoring en de daaronder liggende richtlijn.
* Monitoring is ingericht op detectie van afwijkend gedrag of indicatoren van compromitteren (sabotage/beschadiging).

## Bewustwording en training

Ontwikkelaars en andere betrokkenen moeten beschikken over actuele kennis van beveiligingsprincipes en -technieken. Het vergroten van kennis, bewustzijn en dienovereenkomstig juist gedrag/handelen is een noodzakelijke voorwaarde voor het leveren van veilige software.

**Vereisten**

* Ontwikkelaars worden periodiek geïnformeerd over actuele beveiligingsthema’s (bijv. via trainingen of workshops, nieuwsbrieven vanuit Security / Privacy).
* Nieuwe teamleden krijgen instructie over de standaard en bijbehorende richtlijnen.
* Rollen met bijzondere verantwoordelijkheden (security champion, test lead, leden van security kernteam, CSIRT) volgen aanvullende verdiepingstrainingen.

# Rapportage en borging

Het doel van rapportage en borging is om inzicht te bieden in de toepassing van deze standaard en te waarborgen dat softwareontwikkelprojecten voldoen aan de gestelde beveiligingseisen en wettelijke verplichtingen.

## Rapportage

Softwareontwikkelteams rapporteren periodiek over de toepassing van deze standaard aan de CISO of informatiebeveiligingscoördinator. De frequentie wordt afgestemd op de omvang en impact van het project of product.

De rapportages omvatten ten minste:

* Het aantal gedetecteerde kwetsbaarheden en de afhandeling daarvan.
* Uitgevoerde beveiligingstesten en bevindingen (zoals SAST/DAST/pentest).
* De status van opvolging van beveiligingsadviezen of bevindingen.
* Bevindingen uit code reviews en dependency-scans.
* Eventuele afwijkingen van de standaard met motivatie en goedkeuring.

## Borging

Om te waarborgen dat deze standaard daadwerkelijk wordt nageleefd, zijn verschillende beheersmaatregelen en verantwoordelijke rollen vastgesteld. Deze borging is zowel projectmatig als structureel georganiseerd.

* De verantwoordelijke opdrachtgever of product owner ziet erop toe dat binnen het project aan deze standaard wordt voldaan.
* De CISO of security officer voert periodieke audits of reviews uit op geselecteerde ontwikkelprojecten.
* Interne en, indien van toepassing, externe toetsing wordt uitgevoerd op naleving van deze standaard.
* Ernstige tekortkomingen in naleving leiden tot verplichte correctieve maatregelen, waarvan de opvolging wordt bewaakt.
* Structurele tekortkomingen of incidenten worden geëscaleerd naar het management of externe toezichthouders.
* Auditbevindingen worden gebruikt voor de doorontwikkeling van richtlijnen, procedures en ondersteunende documenten.

# Vaststelling

Deze standaard is aldus vastgesteld.

[Plaats], [Datum].

[NAAM]

[FUNCTIE].

[Na (her)vaststelling, ook de tabel op p.2 bijwerken]

1. Rapportage Toepassing Standaard Informatiebeveiliging Softwareontwikkeling
	1. Algemene informatie

Naam project / ontwikkeling: …..

Software / Product / Release: ….

Verantwoordelijke contactpersoon: ….

Rapportageperiode: ….

* 1. Toepassing van de standaard

Principes geborgd:

……

Vereisten opgevolgd:

* Ontwikkelproces
* Veilig programmeren en coderen
* Beveiligingstesten
* Toegangsbeheer
* Open source en andere componenten
* Bescherming van persoonsgegevens en privacy
* Scheiding van omgevingen
* Uitbesteding en leveranciers
* Risicobeoordeling en -beheer
* Logging en monitoring
* Bewustwording en training
	1. Bevindingen en risico’s

……

* 1. Goedkeuring

Naam verantwoordelijke: ….

Datum: ….

Akkoord door CISO: ….

1. <https://devguide.owasp.org> [↑](#footnote-ref-2)
2. <https://owasp.org/www-project-top-ten/> [↑](#footnote-ref-3)