Informatieclassificatie Vlaamse overheid (Vo-ICR)

Vo Informatieclassificatie - Minimale maatregelen DevOps

Minimale maatregelen

Team Informatieveiligheid| Digitaal Vlaanderen

Dit is een document voor publiek gebruik

AGENTSCHAP
DIGITAAL VLAANDEREN
HAVENLAAN 88 BUS 60, 1000 BRUSSEL

INHOUD VAN DIT DOCUMENT

Situering van het document

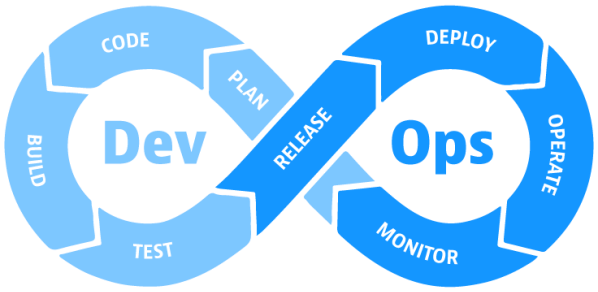
Dit document maakt deel uit van de begeleidende documentatie in context van het generieke Informatieclassificatieraamwerk (ICR) van de Vlaamse overheid (Vo) op initiatief van het Stuurorgaan Vlaams Informatie- en ICT Beleid.

Doel van het document

Dit document beschrijft de kwaliteitseisen van de minimale maatregelen die moeten afgedwongen worden in het kader van DevOps. Deze kwaliteitseisen worden verbonden aan de informatieklassen zoals deze zijn beschreven in het ICR van de Vo.

Werkprincipe van het document

Aangezien DevOps een manier van werken is en geen nauw afgelijnde technologie, is een **gestructureerde aanpak** voor het beschrijven van de Minimale Maatregelen van groot belang. Om die reden zijn de Minimale Maatregelen voor DevOps uitgewerkt in twee delen, dit volgens de typisch weergave van het DevOps-proces (zie Figuur 1).



Figuur 1: De typische weergave van het DevOps-proces (bron)

De Minimale Maatregelen zijn opgedeeld in een *Dev* en een *Ops*-component, met daarin de specifieke fases die erin thuishoren:

- **Dev**: Plan, Code, Build, en Test
- **Ops**: Release, Deploy, Operate, en Monitor

Per fase worden dan de specifieke onderwerpen die relevant zijn qua beveiliging besproken.

De focus van dit document ligt niet op de maturiteit op vlak van de DevOps-praktijk, maar enkel op de specifieke security-aspecten rond DevOps.

In het laatste deel wordt bijkomende aanvullende informatie ter beschikking gesteld samen met de overeenkomende minimale maatregelen.

De organisatie van de maatregelen in het kader van informatieclassificatie wordt beschreven in het document ‘Vo informatieclassificatie – Organisatie - Informatieclassificatieraamwerk’.

Verspreiding van het document

Dit document is voornamelijk voor intern gebruik en bevat informatie die mag gedeeld worden met personen verbonden aan de Vo en zijn diensten leveranciers.

Het document mag gedeeld worden met derden onder de richtlijnen openbaarheid van bestuur van de Vlaamse regering.

Vrijwaring

Dit document geeft de huidige status en daaraan gerelateerde informatie binnen het project weer. Gelieve de auteur of de bevoegde project begeleiders te contacteren om zich ervan te vergewissen dat u de laatst gevalideerde versie van dit document in handen heeft.

Eigenaar

Voorzitter werkgroep informatieveiligheid | Digitaal Vlaanderen
security@vlaanderen.be

Classificatie

Dit document valt onder de vertrouwelijkheidsklasse 1 (Publiek) en mag toegankelijk zijn voor iedereen.
Dit document valt onder integriteitsklasse 2. De organisatie heeft **geen directe hinder** indien de informatie gewijzigd wordt door onbevoegden en de **dienstverlening blijft gegarandeerd**. Maar bijsturing van de criteria en/of maatregelen is aangewezen.

Historiek

Versie	Datum	Auteurs	Opmerking
2.2	27/06/2024	Alexander Dekker, Fabrizio Noviello, Evert Van Hirtum, Kristel Van Aken	Eerste versie goedgekeurd door stuurorgaan

Bronnen en verwijzingen

De inhoud van dit document werd samengesteld op basis van workshops en onderzoek.

Documentverwijzingen

- Vo informatieclassificatie – Organisatie Informatieveiligheid (PDF)
- *Vo Informatieclassificatie – Organisatie Informatieclassificatieraamwerk* (PDF)
- Adviezen Vlaamse Toezichtcommissie
- Vo Informatieclassificatie – Minimale maatregelen – (PDF)
  - *Vo Informatieclassificatie - Minimale maatregelen – Cryptografie*
  - *Vo Informatieclassificatie - Minimale maatregelen – IAM*
  - *Vo Informatieclassificatie - Minimale maatregelen – netwerken*
- Vo Informatieclassificatie – Overzicht baseline maatregelen (XLS)
- Cloud Security Alliance (CSA) Cloud Controls Matrix (CCM)
- NIST Special Publication 800-218 Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities
- NIST Special Publication 800 NIST SP 800-204D Strategies for the Integration of Software Supply Chain Security in DevSecOps CI/CD Pipelines

Inhoudsopgave

1. MINIMALE MAATREGELEN

1.1 Minimale maatregelen – Dev

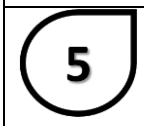
1.1.1 Minimale algemene maatregelen

Vertrouwelijkheid / Integriteit / Beschikbaarheid

IC klasse	Minimale maatregelen
1 2	Klasse 1 en Klasse 2 kennen dezelfde maatregelen: **PLAN** 〉 **Training en bewustwording** 〉〉 Elk teamlid dient minstens eenmalig een training te volgen in DevOps beveiligingspraktijken. 〉〉 Implementeer een proces voor continu leren en verbeteren (continuous improvement). 〉 **Beveiligingsbeleid en richtlijnen** 〉〉 Integreer “secure by design” en “privacy by design” vanaf het de ontwerpfase van de elke toepassing. 〉〉 Evalueer de beveiligingsmaatregelen van derde partijen en leveranciers. 〉 **Risicobeheer** 〉〉 Implementeer een proces voor regelmatige risicoanalyses en evaluaties waarbij de outputs uit de Operate-fase de input vormen voor risicobeheer. Consulteer ook de Minimale Maatregelen rond Risicobeheer. **CODE** 〉 **Veilige codeerstandaarden en ontwikkelomgeving** 〉〉 Gebruik beveiligingsgerichte ontwikkelingsframeworks en bibliotheken. 〉〉 Gebruik veilige coderingspraktijken, zoals omschreven in de Minimale Maatregelen voor ontwikkeling en gebruik van toepassingen. 〉〉 Gebruik security best practices voor IDE's en development omgevingen. 〉 **Beveiliging van repositories** 〉〉 Beheer repositories in een geautoriseerde versiebeheer-oplossing die gecontroleerd wordt. 〉〉 Verleen enkel toegang tot accounts beheerd door de organisatie zoals beschreven in de Minimale Maatregelen voor Identity en access management (IAM) en de Minimale maatregelen voor Privileged Access Management (PAM). 〉〉 Implementeer toegang tot broncode-repositories volgens het principe van least privilege. 〉 **Code review** 〉〉 Code moet door minstens één goedkeurder of reviewer (die niet de auteur is van de code) worden beoordeeld alvorens definitief te worden opgenomen in de codebase.

	〉 **Scheiding van omgevingen** 〉 Ontwikkelingswerkzaamheden dienen enkel worden uitgevoerd in de ontwikkelingsomgeving. Noodwijzigingen kunnen uitgevoerd worden in de acceptatie- of productieomgevingen: consulteer hiervoor de Minimale maatregelen Wijzigingsbeheer 〉 **Secrets management** 〉 Zorg ervoor dat gevoelige informatie veilig wordt opgeslagen op een geschikte locatie. Consulteer hiervoor de Minimale maatregelen Cryptografie. 〉 **Secret scanning** 〉 Implementeer secret scanning om te voorkomen dat gevoelige gegevens worden blootgesteld. **BUILD** 〉 **Software Bill of Materials (SBOM) en dependency management** 〉 Integreer SBOM[1]-generatie in CI/CD. 〉 Doe aan dependency scanning en tracking op basis van de SBOM-input risico's. 〉 **Static Application Security Testing (SAST)** 〉 Integreer SAST in de CI/CD-pipeline. 〉 Configureer de scanconfiguraties zodat deze zijn afgestemd op het project. 〉 Automatiseer het proces voor het toevoegen van een issue aan een issue tracker. 〉 **Code signing en verificatie** 〉 Implementeer code signing om de integriteit en authenticiteit van de code te verifiëren. 〉 Stel beleidsregels op voor toegangscontrole van encryptiesleutels. **TEST** 〉 **Penetration testing** 〉 Laat white/grey box penetration tests uitvoeren op de ontwikkelde applicatie, consulteer hiervoor de Minimale maatregelen Veiligheidstesten.
3 4	Klasse 3 en Klasse 4 kennen dezelfde maatregelen: Alle maatregelen van Klasse 1 + Klasse 2 + **TEST** 〉 **Penetration testing** 〉 Laat white/grey box penetration tests uitvoeren zoals beschreven in de Minimale maatregelen Veiligheidstesten. 〉 **Dynamic Application Security Testing (DAST)** 〉 Integreer Dynamic Application Security Testing (DAST)-tools in het CI/CD-proces.

[1] Een Software Bill of Materials (SBOM) is een gestructureerd, vaak automatisch gegenereerd, overzicht van alle compontent en dependencies van een softwaretraject.

	〉 Definieer testscenario's om alle functionaliteiten van de applicatie te dekken, inclusief randgevallen en potentiële kwetsbaarheden. 〉 Implementeer een proces om de resultaten van DAST-scans te prioriteren en te classificeren op basis van de ernst van de kwetsbaarheden.
	Alle maatregelen van Klasse 1 + Klasse 2 + Klasse 3 + Klasse 4 + **TEST** 〉 **Penetration testing** 〉 Laat black/grey/white box penetration tests uitvoeren zoals beschreven in de Minimale maatregelen Veiligheidstesten.

1.1.2 Minimale specifieke (GDPR) maatregelen

Er zijn geen specifieke GDPR-maatregelen rond DevOps.

1.1.3 Minimale specifieke (NISII) maatregelen

In afwachting van de goedkeuringen omtrent NISII is er in dit document alvast de nodige ruimte voorzien voor toekomstige minimale specifieke NISII-maatregelen.

1.1.4 Minimale specifieke (KSZ) maatregelen

De Minimale Normen van de Kruispuntbank Sociale Zekerheid vermelden geen specifieke normen rond DevOps. Er zijn wel rond het ontwikkelen en het gebruik van toepassingen, welke geraadpleegd kunnen worden in het document van het Informatieclassificatieraamwerk Minimale maatregelen – ontwikkeling en gebruik van toepassingen.

1.2 Minimale maatregelen – Ops

1.2.1 Minimale algemene maatregelen

Vertrouwelijkheid / Integriteit / Beschikbaarheid

IC klasse	Minimale maatregelen
1 2	Klasse 1 en Klasse 2 kennen dezelfde maatregelen: **RELEASE** 〉 **CI/CD Pipelines** 〉 Gebruik beveiligde configuratiebeheer-oplossingen om (semi)geautomatiseerd releases uit te voeren. 〉 **Role Based Access Contol (RBAC)** 〉 Implementeer RBAC (Role-Based Access Control) om toegang tot productieomgevingen en gevoelige bronnen te beperken. 〉 **Rollback procedures** 〉 Implementeer rollback-mechanismen en documenteer deze, en automatiseer deze waar mogelijk. **DEPLOY** 〉 **Infrastructure as Code (IaC)** 〉 Gebruik IaC tools om infrastructuur op te zetten en te beheren. 〉 Zorg ervoor dat IaC-configuratiebestanden veilig opgeslagen worden en de toegang hiertoe zoveel mogelijk wordt beperkt. Consulteer hiervoor de Minimale maatregelen Cryptografische maatregelen en de Minimale maatregelen Asset en configuratie beheer. 〉 Definieer standaard IaC-configuraties en templates voor resources. **OPERATE** 〉 **Patch en vulnerability management** 〉 Implementeer een proactief beveiligingspatch-beheerproces. 〉 Automatiseer patch management taken. 〉 Voer wekelijks automatische kwetsbaarheidsscans uit. Consulteer hiervoor de Minimale maatregelen ICT-systemen. 〉 **Gepriviligeerd Toegangsbeheer** 〉 Beperk geprivilegieerde toegang tot de productieomgeving zodat alleen geautoriseerde personen toegang hebben. Consulteer hiervoor de Minimale maatregelen Privileged Access Management (PAM). 〉 Implementeer Just-in-Time (JIT) access waar technisch mogelijk op productieomgevingen om beveiligingsrisico's te minimaliseren. 〉 **Back-up en Disaster Recovery** 〉 Implementeer back-up- en disaster recovery (DR)-oplossingen om gegevensverlies te minimaliseren en bedrijfscontinuïteit te waarborgen. 〉 Test minstens jaarlijks back-up- en disaster recovery-procedures om hun effectiviteit en betrouwbaarheid te verifiëren.

	〉 **Continuous Improvement (Continue verbetering)** 〉〉 Implementeer een proces voor Continuous Improvement. 〉 **Sleutelbeheer** 〉〉 Consulteer de Minimale Maatregelen rond Sleutelbeheer voor het veilig beheren van encryptiesleutels op cloudomgevingen. 〉〉 Zorg ervoor dat er steeds een *soft delete*-functie geactiveerd is wanneer sleutels verwijderd worden. **MONITOR** 〉 **Voortdurend monitoren** 〉〉 Implementeer continue monitoring voor applicaties en infrastructuur. 〉〉 Implementeer *health checks* om problemen vroegtijdig op te sporen. 〉〉 Implementeer processen rond Security Information and Event Management (SIEM) tools conform de Minimale Maatregelen Veiligheidslogging en monitoring (SIEM). 〉 **Incident response** 〉〉 Ontwikkel en onderhoud een Incident Response Plan (IRP).
3 4	Klasse 3 en Klasse 4 kennen dezelfde maatregelen Alle maatregelen van Klasse 1 / Klasse 2 + **OPERATE** 〉 **Scheiding van omgevingen** 〉〉 Gegevens in de productieomgeving mogen enkel hergebruikt worden in niet-productieomgevingen wanneer deze omgevingen op dezelfde manier beveiligd zijn als de productieomgeving, of wanneer data-anonimisering is toegepast. Zie ook de Minimale Maatregelen rond Release en deployment beheer.
5	Alle maatregelen van Klasse 1 / Klasse 2 + Klasse 3 / Klasse 4 + **OPERATE** 〉 **Scheiding van omgevingen** 〉〉 Gegevens in de productieomgeving mogen niet hergebruikt worden in de test- of ontwikkelingsomgeving, zie ook de Minimale Maatregelen rond Release en deployment beheer.

1.2.2 Minimale specifieke (GDPR-)maatregelen

Er zijn geen specifieke GDPR-maatregelen rond DevOps.

1.2.3 Minimale specifieke (NISII) maatregelen

In afwachting van de goedkeuringen omtrent NISII is er in dit document alvast de nodige ruimte voorzien voor toekomstige minimale specifieke NISII maatregelen.

1.2.4 Minimale specifieke (KSZ) maatregelen

De Minimale Normen van de Kruispuntbank Sociale Zekerheid vermelden geen specifieke normen rond DevOps. Er zijn wel rond het ontwikkelen en het gebruik van toepassingen, welke geraadpleegd

kunnen worden in het document van het Informatieclassificatieraamwerk Minimale maatregelen – ontwikkeling en gebruik van toepassingen.

2. AANVULLENDE INFORMATIE OVER DE MAATREGELEN

2.1 Dev

PLAN

- **Training en bewustwording**
  - Training en bewustwording zijn essentieel voor de goede werking van een DevOps-team, te meer omdat de betrokken technologieën snel evolueren. Er zijn tal van gratis of betaalde trainingen beschikbaar zoals bijvoorbeeld de documentatie van Atlassian – Devops-101.
- **Beveiligingsbeleid en richtlijnen**
  - "Security by design" en "privacy by design" zijn begrippen die beide onderstrepen dat beveiliging en privacy niet slechts achteraf geïmplementeerd dient te worden, maar steeds vanaf het begin ingebouwd hoort te zijn in het ontwerp en de ontwikkeling van software en systemen. De beveiligings- en privacymaatregelen dienen niet apart worden toegevoegd nadat de software is gebouwd, maar wordt opgenomen in het gehele ontwikkelingsproces, vanaf de eerste planningsfasen tot aan de implementatie en het onderhoud.
- **Risicobeheer**
  - Implementeer een proces voor regelmatige risicoanalyses en evaluaties waarbij de outputs uit de Operate-fase de input vormen voor risicobeheer. Consulteer hiervoor zeker ook de extra informatie rond Risicobeheer uit het Informatieclassificatieraamwerk.

CODE

- **Veilige codeerstandaarden en ontwikkelomgeving**
  - Om te voldoen aan de Minimale Maatregelen kan men gebruik maken van beveiligingsgerichte ontwikkelingsframeworks en bibliotheken (bv. OWASP Top 10) om het risico op kwetsbaarheden te verminderen en de ontwikkeling van veilige code te vergemakkelijken.
- **Beveiliging van repositories**
  - Het is belangrijk om alle code op te slaan op een door de organisatie goedgekeurde locatie opdat deze veilig beheerd en gecontroleerd kan worden. Alleen daartoe geautoriseerd personeel mag toegang hebben, steeds volgens het *least privilege* principe zodat de code veilig gehouden wordt en problemen voorkomen kunnen worden.
- **Scheiding van omgevingen**
  - Een strikte scheiding van omgevingen houdt in dat er tussen verschillende omgevingen, zoals ontwikkeling, acceptatie en productie, een gecontroleerde ontwikkelingsworkflow is. Ontwikkelingswerkzaamheden vinden als eerste plaats in de ontwikkelingsomgeving, waar ontwikkelaars de ruimte hebben om te kunnen experimenteren, testen en herhalen zonder de operationele omgevingen te verstoren. Dit biedt een gecontroleerde en veilige ruimte waarin nieuwe functies kunnen worden ontwikkeld en bugs kunnen worden opgelost zonder risico's voor de productieomgeving. Noodwijzigingen, die bijvoorbeeld nodig zijn om urgente beveiligingslekken te verhelpen of ernstige fouten te corrigeren, kunnen echter worden uitgevoerd in de acceptatie- of zelfs de productieomgevingen, mits strikte procedures en goedkeuringsprocessen worden gevolgd om de impact op operationele systemen te minimaliseren en de integriteit van de productieomgeving te garanderen.

- **Secrets management**
  - Gevoelige informatie zoals wachtwoorden, API keys, certificaten en andere authenticatie tokens dienen veilig te worden opgeslagen op een geschikte locatie zoals een secret management tool (vb. HashiCorp Vault, AWS Secrets Manager, Azure Key Vault, …), en nooit rechtstreeks in code- of configuratiebestanden in het repository zelf. Het accidenteel publiceren van zulke credentials op een openbaar repository is namelijk een vaak voorkomende oorzaak van cyberaanvallen.
- **Code review**
  - Code review omvat het uitvoeren van een zorgvuldige controle om de codekwaliteit te garanderen, en tekortkomingen te identificeren en te mitigeren. Typisch wordt deze controle uitgevoerd door een ander teamlid dan de auteur van de code. Op die manier verbetert dit proces ook de samenwerking en kennisdeling binnen het ontwikkelingsteam. Door de andere blik van een beoordelaar wordt het risico op over het hoofd geziene fouten of ongeschikte code verminderd wat de algehele betrouwbaarheid van de software ten goede komt.
- **Secret scanning**
  - Secret scanning is een techniek om automatisch op te sporen of er gevoelige informatie zoals inloggegevens of API-sleutels per ongeluk toegevoegd werden aan de codebase van een applicatie.

BUILD

- **Software Bill of Materials (SBOM) en dependency management**
  - Automatische SBOM-generatie in de CI/CD heeft als doel om een volledige en up-to-date inventaris van de software zijn afhankelijkheden op te bouwen, bijvoorbeeld door middel van tools of standaarden zoals SPDX of CycloneDX. Als in een latere fase zou blijken dat een dependency van de software een kwetsbaarheid bevat, dan kan dit vastgesteld worden en kan de dependency zo snel mogelijk gepatched worden.
  - Dependency scanning en tracking op basis van de SBOM-input om risico's sneller te identificeren: voer regelmatige scans van alle externe bibliotheken en afhankelijkheden uit om te controleren op bekende kwetsbaarheden en om ervoor te zorgen dat alleen veilige versies worden gebruikt.
- **SAST**
  - Static Application Security Testing (SAST) is een methode waarbij de broncode van software wordt geanalyseerd door een specifiek daarop voorziene tool om mogelijke beveiligingskwetsbaarheden op te sporen, zonder dat de code effectief uitgevoerd worden. Tools zoals SonarQube kunnen hiervoor worden ingezet, maar er zijn ook tal van gratis en open source alternatieven zoals hier gedocumenteerd door OWASP.
- **Code signing en verificatie**
  - Voor het tekenen en verifiëren van code kan er gebruik worden gemaakt van NuGet bij .NET applicaties of van NPM-packages bij JavaScript om de integriteit en authenticiteit van de code te waarborgen.

TEST

- **DAST**
  - Dynamic Application Security Testing (DAST) is een methode om de beveiliging van applicaties te testen door deze tijdens runtime te analyseren. In tegenstelling tot SAST,

waarbij broncode statisch wordt onderzocht, simuleert DAST aanvallen op de applicatie zoals een kwaadwillende hacker dat zou doen om zo kwetsbaarheden op te sporen. Hierdoor kunnen beveiligingslekken worden geïdentificeerd die tijdens het gebruik van de applicatie kunnen optreden. Er zijn tal van DAST-tools beschikbaar, zowel commercieel (Burp Suite, Netsparker, Veracode) of gratis en open source, zoals Zed Attack Proxy (ZAP).

- **Penetration testing**
  - Een penetratietest (ook wel *pentest* genoemd) is een gesimuleerde cyberaanval op een computersysteem, netwerk of webapplicatie om beveiligingslekken te identificeren en te exploiteren die uitgevoerd wordt door een cybersecurityspecialist. Er zijn verschillende types van pentests:
    - White box: de pentester ontvangt alle documentatie en heeft volledige toegang tot de broncode en netwerkopzet van de omgeving.
    - Grey box: de pentester heeft slechts een beperkte kennis van de omgeving.
    - Black box: de pentester heeft geen enkele voorkennis, net zoals een echte externe aanvaller.

2.2 Ops

RELEASE

- **CI/CD Pipelines**
  - Tools zoals Azure DevOps, Github Actions, Bitbucket Pipelines, Ansible, Puppet, of Chef kunnen gebruikt worden om (semi)geautomatiseerd releases uit te voeren.
- **Role-Based Access Contol (RBAC)**
  - Role-Based Access Control (RBAC), of rol-gebaseerde toegangscontrole, is een methode voor het beheren van toegang tot een omgeving op basis van de rollen die toegewezen worden aan gebruikers.
- **Rollback procedures**
  - Geautomatiseerde rollback-mechanismen helpen om snel terug te kunnen schakelen naar een vorige versie van de software in geval van onverwachte problemen tijdens de release.

DEPLOY

- **Infrastructure as Code (IaC)**
  - Je kan tools voor Infrastrcture as Code gebruiken zoals Terraform, AWS CloudFormation, Azure Resource Manager, of Azure Bicep templates om het deployen van infrastructuurcomponenten te automatiseren en de weerbaarheid te verhogen.
  - Zorg ervoor dat IaC-configuratiebestanden zoals de Terraform state-files veilig opgeslagen worden en enkel toegankelijk zijn volgens *least privilege*, aangezien deze gevoelige informatie kunnen bevatten.

OPERATE

- **Patch en vulnerability management**
  - Het regelmatig uitvoeren van geautomatiseerde kwetsbaarheidsscans op zowel containers als virtuele machines is essentieel om de kwetsbaarheden op te sporen. Daarnaast is een uitgewerkt proces voor het patchen van deze kwetsbaarheden van groot belang. Er zijn tal van tools op de markt voor vulnerability scanning, zoals OpenVAS, Nessus, Qualys, Docker Bench for Security, Clair en Anchore Engine.

- **Gepriviligeerd Toegangsbeheer**
  - Geprivilegieerd toegangsbeheer omvat het beheren en beperken van toegang tot gevoelige systemen en gegevens binnen organisaties, met behulp van tools zoals CyberArk Privileged Access Security, BeyondTrust Privileged Access Management en Thycotic Secret Server. Deze platforms hebben functies zoals wachtwoordbeheer en rotatie, sessiebeheer, en toegangscontrole om zo de risico's van ongeautoriseerde toegang tot accounts te verminderen. Digitaal Vlaanderen biedt hierrond ook de PAMaaS-dienst aan.
- **Back-up en Disaster Recovery**
  - Back-up- en disaster recovery (DR)-oplossingen kunnen ingezet worden om gegevensverlies en bedrijfscontinuïteit te waarborgen, zoals onder meer Azure Backup, AWS Backup, Dell NetWorker, en Veeam Backup & Replication. Daarnaast is het in cloudomgevingen belangrijker om oplossing zoals *versioning* of *object lock* voor object storage-oplossingen te gebruiken, en immutable backups (backups die niet verwijderd kunnen worden) om het risico op permanent dataverlies tegen te gaan.
- **Continuous Improvement (Continue verbetering)**
  - Implementeer een proces voor Continuous Improvement door doorlopend feedback te verzamelen, post-incident reviews (PIR) uit te voeren en *lessons learned* te implementeren.
- **Sleutelbeheer**
  - Wanneer een encryptiesleutel definitief is verwijderd is bijgevolg ook alle data die ermee geëncrypteerd werd permanent ontoegankelijk. Om die reden zorg je er best voor dat steeds de *soft delete*-functie geactiveerd is wanneer sleutels verwijderd worden: hierdoor wordt een sleutel eerst gemarkeerd als verwijderd, maar wordt deze pas definitief verwijderd na een verplichte wachtperiode. Zo kan de verwijdering nog ongedaan gemaakt worden wanneer deze niet terecht bleek.
- **Scheiding van omgevingen**
  - Soms kan het delen van bepaalde onderdelen tussen verschillende niet-productieomgevingen handig zijn, bijvoorbeeld door een gedeelde databankserver of application gateway te gebruiken. De productieomgeving moet echter steeds gescheiden zijn, en het hergebruiken van data uit productie in andere omgevingen mag alleen als die omgevingen net zo goed beveiligd zijn als de productieomgeving, of als de gegevens geanonimiseerd zijn. Vooral bij gevoelige gegevens, zoals persoonsgegevens, moeten men extra voorzichtig zijn om te voorkomen dat ze per ongeluk naar ontwikkel- of testdatabases worden gekopieerd.

MONITOR

- **Voortdurend monitoren**
  - Tools zoals Prometheus, Grafana, of cloud native monitoringdiensten zoals Azure Monitor en Amazon CloudWatch zorgen voor een continue monitoring van applicaties en infrastructuur.
- **Incident response**
  - Wanneer er een beveiligingsincident optreedt is het belangrijk om zo snel mogelijk op te treden. Hiervoor wordt doorgaans een *incident detection and response* of *extended detection and response* (XDR)-oplossing gebruikt zodat de impact van cybersecurity-incidenten kan beperkt worden. Er zijn tal van oplossingen op de markt, zoals Splunk Enterprise Security, IBM QRadar, Azure Sentinel, en CrowdStrike Falcon. Deze tools bieden functionaliteiten zoals realtime monitoring, detectie van bedreigingen, forensische analyse

en incidentresponsautomatisering om snel te reageren op incidenten en de schade te beperken.

3. LINK MET ANDERE MAATREGELEN

Bij het werken met DevOps zijn er links met alle andere maatregelen uit het Informatieclassificatieraamwerk. Waar relevant werd hier reeds naar verwezen in sectie *1. Minimale maatregelen.*

Echter, bepaalde vereisten opgenomen in andere documenten met Minimale Maatregelen uit het Informatieclassificatieraamwerk zijn in de praktijk niet altijd letterlijk haalbaar wanneer er met een DevOps-werkwijze en een cloudgebaseerde opzet gewerkt wordt. Hieronder wordt toegelicht voor welke vereisten dit het geval is, en hoe er alsnog tegemoet wordt gekomen aan de noden van deze vereisten op een andere manier.

3.1 Minimale Maatregelen Asset- en Configuratiebeheer

De Minimale Maatregelen rond Asset- en Configuratiebeheer vereisen het definiëren van alle componenten van de ICT-omgeving als configuratie-items. In een snel veranderende omgeving zoals de cloud kunnen er gemakkelijk virtuele machines toegevoegd of vervangen worden, bijvoorbeeld met autoscaling groups. Daarnaast wordt er ook heel vaak gebruik gemaakt van container-gebaseerde omgevingen, zoals Kubernetes clusters.

In beide scenario's blijft het uiteraard absoluut aangewezen om vanuit Asset- en Configuratiebeheer deze ICT-componenten op te volgen. Het zou echter zinloos zijn om te proberen alle virtuele machines in een autoscaling-opzet, of containers in een Kubernetes-cluster te inventariseren en op te volgen, aangezien dit geen permanente resources zijn. Om die reden is het bij het gebruik van DevOps en cloud van groot belang om op het juiste abstractieniveau aan asset management te doen, in dit geval bijvoorbeeld op het niveau van de autoscaling group of de Kubernetes cluster, en niet op het vluchtige subniveau daarvan.

3.2 Minimale Maatregelen Wijzigingsbeheer en Minimale maatregelen Release en Deploymentbeheer

Bij een DevOps-manier van werken wordt code typisch zeer frequent gedeployed, mogelijk zelfs meerdere keren op eenzelfde dag. De Minimale Maatregelen voor Wijzigingsbeheer en de Minimale Maatregelen voor Release- en Deploymentbeheer bevatten echter een heel aantal vereisten die in de praktijk voor DevOps-teams moeilijker haalbaar zullen zijn.

Zo wordt in de Minimale Maatregelen voor Wijzigingsbeheer aangegeven dat wijzigingen aan een Change Advisory Board (CAB) moeten worden voorgelegd, en voor bepaalde informatieklassen zelfs aan de Data Protection Officer (DPO). In het geval van DevOps is zoiets praktisch uiteraard zeer lastig. Echter, de integratie, opname of zelfs automatisatie van verschillende beveiligingsprocessen zoals onder meer code review, secret scanning, SAST, DAST, etc. in de Continuous Integration en Continuous Deployment (CI/CD) pipeline zal dezelfde principes die normaal door een Change Advisory Board opgevolgd worden garanderen.

Een andere vereiste is de maatregel rond het opnemen van alle wijzigingen in een logboek. Bij DevOps wordt op een andere manier voldaan aan deze vereiste: alle codewijzingen zoals commits en merges waarbij een ontwikkelaar wijzigingen aanbrengt in de broncode van een softwareproject worden getraceerd in het versiebeheersysteem, en elke wijziging wordt normaliter voorzien van een omschrijving.