2.12 Sicherheit, Datenschutz und Least Privilege

Automatisierungen verbinden häufig mehrere Systeme und verarbeiten dabei Benutzer-, Kunden- oder Unternehmensdaten.

Deshalb müssen Zugriffe, Datenübertragungen und Berechtigungen besonders geschützt werden.

Wichtige Grundprinzipien:

- nur notwendige Daten verarbeiten
- nur notwendige Rechte vergeben
- Zugangsdaten sicher speichern
- Änderungen protokollieren
- kritische Aktionen kontrollieren
- Datenschutzvorgaben beachten
- Zugänge regelmäßig überprüfen

> Eine Automatisierung sollte nur auf die Daten und Funktionen zugreifen, die sie tatsächlich benötigt.

Lernziele

Nach dieser Seite solltest du erklären können:

- was Least Privilege bedeutet
- wie Rollen und Scopes Zugriffe begrenzen
- warum Secrets geschützt werden müssen
- welche Daten eine Automatisierung verarbeiten darf
- was Datenminimierung bedeutet
- wie Test- und Produktivsysteme getrennt werden
- welche Sicherheitsrisiken APIs und Webhooks besitzen
- wie sichere Automatisierungen aufgebaut werden

Least Privilege

Least Privilege bedeutet:

> Benutzer, Anwendungen und Workflows erhalten nur die Rechte, die sie für ihre Aufgabe benötigen.

Beispiel:

Ein Workflow soll Slack-Nachrichten senden.

Benötigte Berechtigung:

```
messages.send
```

Nicht benötigte Berechtigungen:

```
users.delete
billing.manage
administrators.write
```

Je weniger Rechte ein Zugang besitzt, desto geringer ist der mögliche Schaden bei:

- Fehlkonfiguration
- Programmierfehler
- gestohlenem Token
- kompromittiertem Benutzerkonto
- versehentlicher Ausführung

Rollenbasierte Zugriffskontrolle

RBAC bedeutet:

> Role-Based Access Control

Berechtigungen werden über Rollen vergeben.

Beispiele:

Rolle	Mögliche Rechte
Mitarbeiter	eigene Daten lesen
IT-Support	Benutzerkonten bearbeiten
Projektleitung	Projekte verwalten
Buchhaltung	Finanzdaten verwenden
Administrator	Systeme konfigurieren
Automatisierungsdienst	bestimmte API-Funktionen ausführen

Vorteile:

- einheitliche Rechtevergabe
- einfacheres Onboarding
- schnelleres Offboarding
- bessere Kontrolle
- geringere Fehlkonfiguration

Scopes

Scopes begrenzen die Rechte eines API-Tokens.

Beispiele:

Scope	Berechtigung
`users.read`	Benutzer lesen
`users.write`	Benutzer erstellen oder ändern
`projects.read`	Projekte lesen
`messages.send`	Nachrichten senden

Ein Workflow sollte nur die benötigten Scopes erhalten.

Beispiel:

Ein Workflow liest Benutzerinformationen.

Ausreichend:

```
users.read
```

Nicht erforderlich:

```
users.delete
```

Service Account

Ein Service Account ist ein technisches Benutzerkonto für Anwendungen oder automatisierte Prozesse.

Typische Verwendung:

- API-Zugriffe

- Hintergrundprozesse
- Datenübertragungen
- automatisierte Berichte
- Systemintegrationen

Ein Service Account sollte:

- einen eindeutigen Namen besitzen
- nur für einen bestimmten Zweck verwendet werden
- keine interaktive Anmeldung benötigen
- nur notwendige Rechte erhalten
- regelmäßig geprüft werden
- deaktiviert werden, wenn er nicht mehr benötigt wird

Beispiel:

```
svc-n8n-onboarding
```

Besser als:

```
admin2
```

Keine persönlichen Administratorkonten verwenden

Ein Workflow sollte möglichst nicht dauerhaft mit dem persönlichen Administratorkonto eines Mitarbeiters arbeiten.

Nachteile:

- Workflow fällt beim Mitarbeiteraustritt aus
- Aktionen sind schwerer zuzuordnen
- Rechte sind häufig zu umfangreich
- Passwortänderungen können Integrationen unterbrechen
- private und technische Zugriffe werden vermischt

Besser:

> Für die Integration wird ein eigener technischer Zugang mit begrenzten Rechten verwendet.

Secrets

Secrets sind vertrauliche Zugangsdaten.

Beispiele:

- Passwörter
- API-Keys
- Bearer-Tokens
- Refresh Tokens
- Client Secrets
- private Schlüssel
- Datenbankkennwörter
- Webhook-Geheimnisse

Secrets dürfen nicht offen gespeichert werden in:

- Wiki-Seiten
- Workflow-Beschreibungen
- Code
- Screenshots
- E-Mails
- öffentlichen Git-Repositories
- unverschlüsselten Textdateien

Sichere Speicherung von Secrets

Geeignete Speicherorte:

- n8n-Credential-Verwaltung
- Umgebungsvariablen
- Secret-Management-System
- verschlüsselte Konfigurationsdatei
- geschützter Passwortmanager

Unsicher:

```
const token = "abc123";
```

Sicherer:

```
API_TOKEN wird aus einem geschützten Credential geladen
```

> Workflow-Logik und Zugangsdaten müssen getrennt gespeichert werden.

Secret Rotation

Secret Rotation bedeutet:

> Ein Passwort, Token oder Schlüssel wird durch einen neuen ersetzt.

Rotation ist sinnvoll:

- regelmäßig
- nach einem Sicherheitsvorfall
- bei Verdacht auf Veröffentlichung
- nach einem Mitarbeiterwechsel
- nach Abschluss eines Projekts
- wenn ein Dienst nicht mehr benötigt wird

Ablauf:

1. neues Secret erzeugen
2. Integration aktualisieren
3. Funktion prüfen
4. altes Secret widerrufen
5. Änderung dokumentieren

Datenminimierung

Datenminimierung bedeutet:

> Es werden nur die Daten verarbeitet, die für den Zweck notwendig sind.

Beispiel:

Ein Workflow soll einen Slack-Zugang vorbereiten.

Benötigte Daten:

- Name
- dienstliche E-Mail-Adresse
- Abteilung

Nicht notwendig:

- private Adresse
- Gehalt
- Bankverbindung
- Gesundheitsdaten
- private Telefonnummer

Beispiel:

Quellsystem:

```
{
 "name": "Max Mustermann",
 "email": "max.mustermann@example.com",
 "department": "IT",
 "salary": 50000,
 "privateAddress": "Musterstraße 1"
}
```

Übertragung an Slack:

```
{
 "name": "Max Mustermann",
 "email": "max.mustermann@example.com",
 "department": "IT"
}
```

Zweckbindung

Daten dürfen nur für den festgelegten Zweck verarbeitet werden.

Beispiel:

Mitarbeiterdaten werden für das technische Onboarding verwendet.

Sie dürfen nicht automatisch für andere Auswertungen verwendet werden, wenn dafür keine Freigabe oder rechtliche Grundlage besteht.

> Daten sollten nicht nur gesammelt werden, weil sie verfügbar sind.

Personenbezogene Daten

Personenbezogene Daten sind Informationen, die sich auf eine bestimmte oder bestimmbare Person beziehen.

Beispiele:

- Name
- E-Mail-Adresse
- Mitarbeiter-ID
- IP-Adresse
- Benutzerkonto
- Telefonnummer
- Standortdaten
- Zugriffsprotokolle

Besonders sensible Daten benötigen einen erhöhten Schutz.

Datenschutz bei Logs

Logs können personenbezogene oder vertrauliche Daten enthalten.

Deshalb sollte geprüft werden:

- Welche Daten werden protokolliert?
- Wer darf die Logs lesen?
- Wie lange werden Logs gespeichert?
- Werden Tokens oder Passwörter maskiert?
- Sind vollständige Nutzdaten wirklich notwendig?

Unsicher:

```
Passwort: GeheimesPasswort123
```

Sicherer:

```
Passwort: ***
```

Unsicher:

```
Authorization: Bearer vollständiges-token
```

Sicherer:

```
Authorization: Bearer ***
```

Datenaufbewahrung

Daten und Logs sollten nicht unbegrenzt gespeichert werden.

Mögliche Regeln:

- erfolgreiche Testausführungen nach kurzer Zeit löschen
- Fehlerlogs für einen festgelegten Zeitraum speichern
- personenbezogene Daten nach Zweckfortfall entfernen
- alte Workflow-Daten regelmäßig prüfen
- Aufbewahrungsfristen dokumentieren

> Daten sollten nur so lange gespeichert werden, wie sie benötigt werden.

Transportverschlüsselung

API- und Webhook-Daten sollten verschlüsselt übertragen werden.

Verwendet wird normalerweise:

```
HTTPS
```

HTTPS schützt vor:

- Mitlesen
- Manipulation
- unbemerkter Veränderung
- Übertragung von Zugangsdaten im Klartext

Unsicher:

```
http://api.example.com
```

Sicherer:

```
https://api.example.com
```

Zertifikate

Bei HTTPS prüft der Client das TLS-Zertifikat des Servers.

Zu prüfen sind:

- Zertifikat gültig
- Hostname stimmt
- Zertifikat nicht abgelaufen
- vertrauenswürdige Zertifizierungsstelle
- vollständige Zertifikatskette

Zertifikatsprüfungen sollten nicht dauerhaft deaktiviert werden.

Webhook-Sicherheit

Ein Webhook-Endpoint kann öffentlich erreichbar sein.

Mögliche Schutzmaßnahmen:

- HTTPS
- geheime Webhook-URL
- API-Key
- Bearer-Token
- Signaturprüfung
- Zeitstempel
- Event-ID
- Rate Limit
- IP-Filter
- Datenvalidierung

Ablauf:

```
Webhook empfangen
    |
    v
Signatur prüfen
    |
    v
Zeitstempel prüfen
    |
    v
Event-ID prüfen
```

```
    |
    v
Daten validieren
    |
    v
Workflow ausführen
```

Eingabevalidierung

Daten aus APIs und Webhooks dürfen nicht automatisch als vertrauenswürdig behandelt werden.

Zu prüfen sind:

- Pflichtfelder
- Datentypen
- erlaubte Werte
- Textlänge
- Datum
- E-Mail-Format
- IDs
- Dateitypen
- Dateigröße

Beispiel:

Erlaubte Abteilungen:

```
IT
Vertrieb
Buchhaltung
```

Unbekannter Wert:

```
Administrator
```

Dieser Wert sollte nicht automatisch Administratorrechte auslösen.

Test- und Produktivsysteme

Test- und Produktivsysteme sollten getrennt werden.

Testsystem	Produktivsystem
Testdaten	echte Unternehmensdaten
Testkonten	reale Benutzer
ungefährliche Aktionen	betriebliche Auswirkungen
Entwicklung und Prüfung	täglicher Betrieb

Mögliche Trennung:

- eigene Credentials
- eigene API-Endpunkte
- getrennte Datenbanken
- getrennte Slack-Kanäle
- getrennte Projekte
- getrennte Workflows

Produktivänderungen absichern

Besonders kritische Aktionen:

- Benutzer löschen
- Administratorrechte vergeben
- Kundendaten verändern
- Zahlungen auslösen
- Dateien endgültig löschen
- große Datenmengen ändern

Mögliche Schutzmaßnahmen:

- manuelle Freigabe
- Vier-Augen-Prinzip
- Testlauf
- Vorschau
- Backup
- Rollback-Möglichkeit
- zusätzliche Bestätigung

Vier-Augen-Prinzip

Beim Vier-Augen-Prinzip prüft eine zweite Person eine kritische Aktion.

Beispiele:

- Administratorrolle vergeben

- größeren Benutzerimport starten
- Produktivdaten löschen
- API-Scopes erweitern
- neue SaaS-Integration aktivieren

> Kritische Änderungen sollten nicht immer vollständig automatisch ausgeführt werden.

Onboarding und Offboarding

Beim Onboarding:

- nur notwendige Zugänge anlegen
- passende Rollen vergeben
- MFA aktivieren
- Lizenzen zuweisen
- Rechte dokumentieren

Beim Offboarding:

- Konten deaktivieren
- aktive Sitzungen beenden
- Tokens widerrufen
- Gruppen entfernen
- Daten übertragen
- Lizenzen freigeben
- technische Zugänge prüfen

Ein unvollständiges Offboarding kann ein erhebliches Sicherheitsrisiko darstellen.

Berechtigungen regelmäßig prüfen

Berechtigungen verändern sich im Laufe der Zeit.

Mögliche Ursachen:

- Mitarbeiter wechselt Abteilung
- Projekt endet
- Aufgabe ändert sich
- Integration wird ersetzt
- temporäre Rechte werden vergessen

Deshalb sollten regelmäßig geprüft werden:

- Benutzerrollen
- Gruppenmitgliedschaften
- API-Scopes
- Service Accounts
- aktive Tokens
- Administratorrechte
- ungenutzte Konten

Audit-Logs

Audit-Logs dokumentieren sicherheitsrelevante Aktionen.

Beispiele:

- Benutzer angelegt
- Rolle geändert
- Token erzeugt
- MFA deaktiviert
- Datei freigegeben
- Administratorrecht vergeben
- Workflow geändert
- Credential aktualisiert

Audit-Logs helfen bei:

- Fehlersuche
- Sicherheitsanalyse
- Nachvollziehbarkeit
- internen Kontrollen
- Datenschutzprüfungen

Schatten-IT

Schatten-IT bezeichnet nicht genehmigte Anwendungen oder Dienste.

Beispiel:

Ein Mitarbeiter verbindet ohne Freigabe einen externen Onlinedienst mit Unternehmensdaten.

Risiken:

- unbekannter Speicherort
- fehlende Sicherheitsprüfung
- unkontrollierte API-Tokens
- fehlendes Offboarding

- Datenschutzverletzung
- zusätzliche Kosten

Neue SaaS- und API-Verbindungen sollten deshalb freigegeben und dokumentiert werden.

Typische Sicherheitsfehler

- Token besitzt Administratorrechte
- API-Key steht direkt im Workflow
- Testworkflow verwendet Produktivdaten
- Webhook besitzt keine Authentifizierung
- Logs enthalten Passwörter
- ausgeschiedener Mitarbeiter besitzt noch Zugänge
- Integration verwendet persönliches Konto
- nicht benötigte Daten werden übertragen
- alte Tokens bleiben aktiv
- Fehlerpfade geben vertrauliche Daten aus

Praxisbeispiel: Sicheres Onboarding

Möglicher Ablauf:

1. Webhook-Signatur prüfen
2. Event-ID auf Duplikat prüfen
3. Pflichtfelder validieren
4. nur benötigte Mitarbeiterdaten übernehmen
5. Service Account mit begrenzten Scopes verwenden
6. Benutzerkonto ohne Administratorrechte anlegen
7. Gruppen anhand der Abteilung zuweisen
8. MFA-Einrichtung anfordern
9. Ergebnis ohne Secrets protokollieren
10. bei kritischem Fehler IT informieren

Sicherheitsprüfung vor Veröffentlichung

Vor der Aktivierung eines Workflows sollte geprüft werden:

- Welche Daten werden verarbeitet?
- Werden nur notwendige Daten verwendet?
- Welche Systeme sind verbunden?
- Welche Rechte besitzen die Credentials?
- Sind Secrets geschützt gespeichert?
- Wird HTTPS verwendet?
- Sind Webhooks abgesichert?

- Werden Eingabedaten validiert?
- Können doppelte Ausführungen entstehen?
- Enthalten Logs vertrauliche Daten?
- Existiert eine Fehlerbehandlung?
- Sind Test- und Produktivsystem getrennt?
- Gibt es einen verantwortlichen Ansprechpartner?

Systematische Fehlersuche

Bei einem Sicherheits- oder Berechtigungsproblem:

1. Welches Konto oder Token wird verwendet?
2. Welche Rolle besitzt der Zugang?
3. Welche Scopes wurden vergeben?
4. Ist der Zugang noch erforderlich?
5. Ist das Secret gültig?
6. Wurde das Secret möglicherweise veröffentlicht?
7. Wird HTTPS verwendet?
8. Welche Daten werden übertragen?
9. Enthalten Logs vertrauliche Informationen?
10. Ist der Webhook authentifiziert?
11. Wurden Test- und Produktivzugänge verwechselt?
12. Welche Aktion zeigt das Audit-Log?
13. Muss ein Token sofort widerrufen werden?
14. Müssen betroffene Personen informiert werden?

Wichtige Begriffe

Begriff	Bedeutung
Least Privilege	nur notwendige Rechte vergeben
RBAC	rollenbasierte Zugriffskontrolle
Scope	begrenzte API-Berechtigung
Service Account	technisches Benutzerkonto
Secret	vertrauliche Zugangsinformation
Rotation	Secret durch neues Secret ersetzen
Datenminimierung	nur notwendige Daten verarbeiten
Zweckbindung	Daten nur für festgelegten Zweck verwenden
Audit-Log	Protokoll sicherheitsrelevanter Aktionen
Vier-Augen-Prinzip	zweite Person prüft kritische Aktion
Schatten-IT	nicht freigegebene Software oder Dienste

Begriff	Bedeutung
TLS	verschlüsselte Netzwerkübertragung

Gesamtmerksatz

> Sichere Automatisierungen verwenden begrenzte Rollen und Scopes, geschützte Secrets, verschlüsselte Verbindungen und geprüfte Eingangsdaten. Sie verarbeiten nur notwendige Informationen, protokollieren keine Geheimnisse und trennen Test- von Produktivsystemen.

Kontrollfragen

Was bedeutet Least Privilege?

Benutzer, Anwendungen und Workflows erhalten nur die Rechte, die sie tatsächlich benötigen.

Was ist RBAC?

Berechtigungen werden anhand festgelegter Rollen vergeben.

Was ist ein Service Account?

Ein technisches Benutzerkonto für Anwendungen und automatisierte Prozesse.

Was ist ein Secret?

Eine vertrauliche Zugangsinformation wie Passwort, API-Key oder Token.

Was bedeutet Datenminimierung?

Es werden nur die für den Zweck notwendigen Daten verarbeitet.

Warum sollten persönliche Administratorkonten nicht für Workflows verwendet werden?

Weil Rechte häufig zu umfangreich sind und die Integration vom einzelnen Mitarbeiter abhängig wird.

Warum müssen Test- und Produktivsysteme getrennt werden?

Damit Tests keine unbeabsichtigten Änderungen an echten Unternehmensdaten verursachen.

Was ist ein Audit-Log?

Ein Protokoll sicherheitsrelevanter Änderungen und Zugriffe.

Was bedeutet Secret Rotation?

Ein Schlüssel, Token oder Passwort wird durch ein neues Secret ersetzt.

Warum sollte ein Webhook eine Signaturprüfung besitzen?

Damit Herkunft und Unveränderheit der Nachricht überprüft werden können.

Quellen

- BSI – IT-Grundschutz
- OWASP – Authorization Cheat Sheet
- OWASP – Secrets Management Cheat Sheet
- OWASP – Logging Cheat Sheet
- NIST – Least Privilege
- n8n-Dokumentation – Credentials