LEANOFY

EXECUTIVE WHITEPAPER · AUSGABE Q2 2026

API-Sicherheit unter DORA & NIS2

Wie regulierte Unternehmen logische API-Schwachstellen datensouverän, kontinuierlich und agil im eigenen Netz absichern – ohne ihre Release-Geschwindigkeit zu opfern.

TECHNOLOGIE
sectestx · Native Desktop-App

EXPERTISE
LEANOFY · Testmanagement & Security

SECTESTX_SCANNER

```
> initializing engine …
> loading openapi.yaml …
> checking BOLA / BFLA vulns …
> ALERT: logic flaw at /api/v1/zaehler/{id}
> DATA ON-PREMISE: OK  ·  0 bytes left the network
> generating bruno collection … _
```

100 % Datensouveränität

Betreiben Sie sectestx vollständig offline, mit lokalen KI-Modellen (Llama-3 via Ollama) oder über Ihre eigenen Enterprise-API-Keys. Keine API-Strukturen und keine Kunden-Payloads verlassen jemals Ihr geschütztes Netzwerk.

AIR-GAPPED
ON-PREMISE
BYOK

KONTAKT
whitepaper@leanofy.de

COMPLIANCE
DORA / NIS2 Ready

ÜBERBLICK

Executive Summary

Die Software-Entwicklung in Banken, Versicherungen, Energieversorgern und KRITIS-Betreibern steht unter doppeltem Druck: Der Markt erzwingt immer kürzere Release-Zyklen – mehrere Deployments pro Woche sind Normalität – während DORA und NIS2 einen lückenlosen, kontinuierlichen und revisionssicheren Nachweis der API-Sicherheit verlangen. Es entsteht ein scheinbar unauflösbarer Widerspruch: **Geschwindigkeit gegen Compliance.**

Agile Time-to-Market

Entwicklungsteams deployen mehrmals pro Woche. Nahezu jeder geschäftskritische Prozess wird über APIs orchestriert – sie sind damit der zentrale Angriffsvektor moderner Anwendungen.

Strikte Regulatorik

DORA und NIS2 fordern lückenlose, risikobasierte und kontinuierliche Sicherheitsnachweise – bei persönlicher Haftung der Geschäftsleitung im Falle von Verstößen.

Warum klassische Sicherheitsprüfungen hier an Grenzen stoßen

- **DAST- und SAST-Scanner** übersehen logische Berechtigungsfehler prinzipbedingt – sie prüfen Syntax und Standardmuster, nicht die Geschäftslogik.
- **Cloud-basierte SaaS-Scanner** erzwingen den Abfluss sensibler API-Strukturen und Kunden-Payloads in externe Clouds – ein Verstoß gegen DSGVO und Bankgeheimnis.
- **Manuelle Penetrationstests** sind teuer, binden knappe Security-Experten und liefern stets nur eine Momentaufnahme zum Stichtag.

DER ANSATZ DIESES WHITEPAPERS

„Regulierte Unternehmen lösen diesen Widerspruch durch einen datensouveränen Self-Service-Sicherheitsansatz: kontinuierliche, logische API-Tests, fest verankert im Entwicklungsprozess – nicht erst am Stichtag."

Die folgenden Kapitel zeigen, wie **sectestx** – begleitet von den Expertinnen und Experten von **LEANOFY** – diesen Widerspruch auflöst und API-Security von der Release-Bremse zum Compliance-Turbo macht.

KONTAKT
whitepaper@leanofy.de

COMPLIANCE
DORA / NIS2 Ready

KAPITEL 01 · REGULATORIK

Der regulatorische Rahmen: DORA & NIS2 im Überblick

IKT-Sicherheit ist im regulierten Umfeld längst keine reine Fachbereichsaufgabe mehr. Die jüngsten EU-Regulierungen heben das Risikomanagement direkt auf die Ebene von Vorstand und Geschäftsführung – mit konkreten, prüfbaren und einklagbaren Pflichten.

01 DORA Digital Operational Resilience Act

DORA harmonisiert die Anforderungen an die digitale Betriebsstabilität im gesamten europäischen Finanzsektor. Seit Januar 2025 müssen Banken, Versicherungen, Wertpapierfirmen und Zahlungsdienstleister ein umfassendes Framework zur Bewältigung von IKT-Risiken nachweisen.

- **Kapitel IV (Art. 24–27):** Alle geschäftskritischen IKT-Systeme und -Schnittstellen müssen einem systematischen, risikobasierten Testprogramm unterzogen werden.
- **Die Herausforderung:** Bei agilen Release-Zyklen ist ein jährlicher Sicherheitsnachweis bereits veraltet, sobald der nächste Commit in Produktion geht. DORA fordert eine „angemessene Häufigkeit“ – kontinuierliche Tests sind damit der einzige rechtssichere Weg.

02 NIS2 Netz- und Informationssicherheit

Die NIS2-Richtlinie betrifft eine deutlich größere Zahl an Unternehmen in ganz Europa – vom Energieversorger über die Logistik bis zum produzierenden Gewerbe ab 50 Mitarbeitenden.

- **Supply-Chain-Sicherheit & Schwachstellenmanagement:** Unternehmen müssen die Sicherheit ihrer gesamten Software-Lieferkette überwachen und bekannte Schwachstellen unverzüglich und nachweisbar schließen.
- **Haftung der Geschäftsleitung:** NIS2 etabliert eine direkte, persönliche Verantwortung von Vorständen und Geschäftsführern. Unwissenheit schützt ausdrücklich nicht vor Haftung.

DIE OPERATIVE KONSEQUENZ

Unternehmen müssen bei **jedem Release** nachweisen können, dass ihre geschäftskritischen APIs frei von logischen Lücken sind. Sicherheitstests müssen damit zum integralen, automatisierten Bestandteil jeder Deployment-Pipeline werden.

KAPITEL 02 · MARKTANALYSE

Das SaaS- & Scanner-Dilemma

Wer im Markt nach automatisierten Werkzeugen zur API-Prüfung sucht, stößt fast ausschließlich auf cloud-native SaaS-Plattformen. Für regulierte Branchen im DACH-Raum führt das in ein doppeltes Dilemma – ein rechtliches und ein technologisches.

A Das Datenschutz- und Souveränitätsdilemma

Um eine API wirksam auf logische Lücken zu prüfen, benötigt ein Scanner die genaue Struktur der Schnittstelle – OpenAPI-Spezifikationen, URL-Pfade und reale Test-Payloads wie Kunden-IDs, Verträge oder Messdaten.

- **Cloud-Zwang:** Herkömmliche SaaS-Plattformen erzwingen den Transfer dieser sensiblen Metadaten – oft samt echter Kunden-Payloads – in ihre eigene Infrastruktur, meist auf US-Hyperscalern.
- **Der Compliance-Bruch:** Für Banken, Versicherungen und KRITIS-Betreiber ist dieser Datenabfluss ein Ausschlusskriterium. Er verletzt das Bankgeheimnis, verstoßt gegen die DSGVO und widerspricht internen Datenschutzrichtlinien.

B Die technologische Sackgasse klassischer Scanner

Viele Unternehmen vertrauen auf vorhandene AppSec-Werkzeuge – statische Code-Analyse (SAST) und dynamische Scanner (DAST). Bei der API-Sicherheit erzeugt das jedoch eine gefährliche **Scheinsicherheit**.

SAST · Statische Code-Analyse

SAST liest Quellcode auf syntaktischer Ebene. Berechtigungsprüfungen – „Darf Benutzer A den Zähler von Benutzer B abrufen?“ – entstehen aber erst zur Laufzeit in Datenbank-abfragen und Tokens. Für diese logische Ebene ist SAST blind.

DAST · Dynamische Prüfung

DAST-Scanner senden Standard-Angriffsmuster und prüfen auf Server-Fehler. Ein logischer Angriff ist aber eine syntaktisch einwandfreie Anfrage – die API antwortet mit 200 OK, und der Scanner meldet fälschlich eine „grüne“ Pipeline.

ZWISCHENFAZIT

Regulierte Unternehmen brauchen ein Werkzeug, das die **Geschäftslogik** ihrer APIs versteht – und das **vollständig im eigenen Netz** arbeitet. Genau diese Lücke schließt ein datensouveräner, semantischer Testansatz.

KAPITEL 03 · GRUNDLAGEN

Die OWASP API Security Top 10

Bevor wir konkrete Angriffe betrachten, lohnt ein Blick auf den Bezugsrahmen, an dem sich Auditoren, Ausschreibungen und Sicherheitsteams weltweit orientieren – und der erklärt, warum die gefährlichsten API-Risiken gerade von automatisierten Werkzeugen so zuverlässig übersehen werden.

Was ist OWASP – und warum eine eigene API-Liste?

Das **Open Worldwide Application Security Project (OWASP)** ist eine herstellerunabhängige Non-Profit-Stiftung und seit über zwei Jahrzehnten die maßgebliche Instanz für Anwendungssicherheit. Ihre bekannteste Veröffentlichung – die OWASP Top 10 – gilt branchenübergreifend als De-facto-Standard und wird in Prüfrahmenwerken und Compliance-Vorgaben referenziert.

Weil Programmierschnittstellen eine eigene Angriffsfläche mit eigenen Schwachstellenmustern besitzen, pflegt OWASP eine gesonderte Liste: die **OWASP API Security Top 10** (aktuelle Ausgabe 2023). Sie benennt die zehn kritischsten Risiken speziell für APIs – von fehlerhafter Autorisierung über unsichere Bestandsführung bis zu ungebremstem Ressourcenverbrauch.

DER ENTSCHEIDENDE UNTERSCHIED

Klassische Web-Schwachstellen wie SQL-Injection sind **technische Fehler** – sie hinterlassen Spuren in Fehlermeldungen. Die gefährlichsten API-Risiken sind dagegen Fehler in der **Geschäftslogik**: Die API funktioniert technisch einwandfrei und antwortet mit 200 OK – sie stellt nur die entscheidende Frage nicht: „Darf dieser Benutzer das wirklich?“ DAST-Scanner suchen Fehler und finden keine, SAST-Scanner sehen keine Laufzeit-Berechtigung. Die Lücke bleibt unsichtbar – bis ein Angreifer sie nutzt.

Die drei Risiken im Fokus dieses Whitepapers

API1:2023	**Broken Object Level Authorization (BOLA)** Zugriff auf fremde Objekte über manipulierte ID-Parameter – das häufigste und folgenschwerste API-Risiko.
API3:2023	**Broken Object Property Level Authorization · Mass Assignment** Einschleusen nicht vorgesehener Objekt-Eigenschaften, z. B. administrativer Felder, in reguläre Anfragen.
API5:2023	**Broken Function Level Authorization (BFLA)** Ausführen administrativer Funktionen durch Benutzer ohne die dafür erforderliche Berechtigung.

Die folgenden Seiten zeigen diese drei Risiken anhand konkreter Praxis-Szenarien aus einer Energieversorger-API – und wie sectestx sie vollautomatisch aufdeckt.

KAPITEL 03 · PRAXIS-SZENARIO 1

BOLA – Broken Object Level Authorization

BOLA ist die unangefochtene Nummer 1 der API-Risiken (OWASP API1:2023). Sie entsteht, wenn eine API den Zugriff auf Objekte über ID-Parameter erlaubt, im Backend aber nicht prüft, ob der angemeldete Benutzer tatsächlich Eigentümer des Objekts ist.

Ein angemeldeter Benutzer mit gültigem Token (Bearer token_A) ruft gezielt die Zählerdaten eines fremden Benutzers (ID 2002) ab – er ändert dafür lediglich eine Zahl in einer ansonsten völlig regulären Anfrage:

REQUEST · ANGRIFF

```
GET /api/v1/zaehler/2002 HTTP/1.1
Host: api.versorger.de
Authorization: Bearer eyJhbGciOiJIUzI1Ni… [Token Benutzer A]
Accept: application/json
```

Da die API zwar das Token verifiziert, aber keine logische Berechtigungsprüfung durchführt, liefert sie die fremden Daten bereitwillig aus:

RESPONSE · VERWUNDBARE API

```
HTTP/1.1 200 OK
Content-Type: application/json
{
  "zaehler_id": 2002,
  "eigentuemer": "Dr. Max Mustermann",
  "adresse": "Goethestraße 12, 80337 München",
  "aktueller_stand": "14582.4 kWh"
}
```

⚠ Das Dilemma

Die Antwort liefert **200 OK** und perfekt valides JSON. Für einen klassischen DAST-Scanner ist das ein erfolgreicher Request – er meldet **kein Problem**, obwohl gerade fremde Kundendaten abgeflossen sind.

✓ Wie sectestx es löst

sectestx erkennt /zaehler/{id} aus der OpenAPI-Spec als sensiblen Endpunkt und generiert eine Test-Collection mit zwei Konten. Der Runner sendet die ID von User B mit dem Token von User A – erwartet wird **403/404**. Kommt **200 OK**, schlägt der Test rot an.

KAPITEL 03 · PRAXIS-SZENARIO 2 & 3

Mass Assignment & BFLA

2 Mass Assignment OWASP API3:2023

Mass Assignment entsteht, wenn Benutzereingaben ohne explizite Filterung direkt in interne Datenbank-Objekte geschrieben werden. Der Angreifer schmuggelt einen administrativen Parameter in einen ansonsten harmlosen Profil-Update:

REQUEST · REGULÄR

```
PUT /api/v1/users/profile HTTP/1.1
Content-Type: application/json
{
  "display_name": "Max Mustermann",
  "phone": "+491701234567"
}
```

REQUEST · MANIPULIERT

```
PUT /api/v1/users/profile HTTP/1.1
Content-Type: application/json
{
  "display_name": "Max Mustermann",
  "phone": "+491701234567",
  "is_admin": true,
  "role": "superuser"
}
```

⚠ Die Auswirkung

Überschreibt das Backend das User-Objekt ungefiltert, hat sich der Angreifer mit einem **einzigen Request** administrative Rechte erschlichen – eine klassische Privilege Escalation.

✓ Wie sectestx es löst

Die semantische KI – der **SlotFiller** – liest OpenAPI-Spec und Datenmodell, erkennt sensible Schlüsselwörter wie admin, role oder privileges und injiziert sie gezielt in die Request-Bodies.

3 Broken Function Level Authorization OWASP API5:2023

APIs trennen reguläre und administrative Funktionen oft nur über den Pfad – /api/users/profile gegenüber /api/admin/users/delete. Versagt die Berechtigungsprüfung auf Funktionsebene, kann ein nicht privilegierter Benutzer administrative Befehle ausführen, indem er schlicht den Pfad errät oder manipuliert.

⚠ Das Dilemma

Herkömmliche Scanner raten URLs blind oder prüfen nur, ob ein Endpunkt existiert. Sie führen keine Funktionsaufrufe mit **unterschiedlichen Privilegienstufen** durch.

✓ Wie sectestx es löst

sectestx klassifiziert Endpunkte nach Kritikalität (**Admin / User / Public**) und führt systematische Berechtigungs-Kreuztests durch – administrative Funktionen müssen für normale Benutzer absolut gesperrt bleiben.

KAPITEL 04 · DIE LÖSUNG

Die drei Säulen von sectestx

Die Auflösung des Widerspruchs zwischen agiler Entwicklung und kompromissloser Compliance erfordert ein Umdenken: Sicherheitstests müssen als automatisierter Self-Service – **Shift-Left** – direkt im eigenen Netzwerk stattfinden. sectestx ruht dabei auf drei tragenden Säulen.

01

100 % Datensouveränität

Geliefert als native Desktop-App für Ihre Infrastruktur. Sie entscheiden, wie KI eingebunden wird – über eigene Enterprise-Keys, private lokale LLMs oder vollständig offline. Keine sensiblen Daten verlassen Ihr Netz.

02

Semantische Tiefe statt Oberflächen-Scan

Heuristiken und die semantische KI (SlotFiller) verstehen die Geschäftslogik Ihrer API. sectestx erkennt Identifier wie zaehler_id und baut gezielt Angriffsszenarien für BOLA, Mass Assignment und BFLA.

03

Tests-as-Code & Zero Lock-in

sectestx generiert lesbare, textbasierte Bruno-Collections. Sie werden wie Quellcode in Git versioniert und laufen über den Open-Source-Bruno-CLI in jeder CI/CD-Pipeline – auch völlig unabhängig von sectestx.

Shift-Left im Agile Testing Quadrant 4

Sicherheitstests gehören im agilen Modell in Quadrant 4 – technologieorientierte Tests, die das Produkt bewerten. In der Praxis werden sie jedoch meist ans Ende des Release-Zyklus geschoben, was zu späten, teuren Entdeckungen führt. sectestx verlagert diese Routineprüfungen nach vorn: QA- und Entwicklungsteams spüren logische Schwachstellen **kontinuierlich und frühzeitig** auf – auch ohne tiefes Security-Expertenwissen.

DAS ERGEBNIS

Routineprüfungen laufen vollautomatisiert. Wertvolle manuelle Penetrationstests werden gezielt für die komplexesten Architekturszenarien frei – statt sie an Standardprüfungen zu verbrauchen.

KAPITEL 04 · ARCHITEKTUR

Datensouveränität & Tests-as-Code

sectestx wurde speziell für die strengen Anforderungen regulierter Branchen entwickelt. Sie entscheiden flexibel, wie KI eingebunden wird – in drei klar abgegrenzten Betriebsmodi, passend zu Ihren Compliance-Richtlinien.

MODUS 01

Bring Your Own Key

Binden Sie leistungsstarke Cloud-LLMs über Ihre eigenen Enterprise-API-Keys ein. Die Verarbeitung erfolgt unter Ihren Verträgen – ohne Modelltraining mit Ihren Daten.

MODUS 02

Lokale KI · On-Premise

Private lokale Sprachmodelle (z. B. Llama-3 via Ollama / vLLM) laufen direkt in Ihrer Infrastruktur. Es fließen keinerlei Daten nach außen.

MODUS 03

Air-Gapped · Deterministisch

Vollständig offline und rein regelbasiert auf Basis präziser Heuristiken – für Hochsicherheitsumgebungen ohne jede externe Verbindung.

Tests-as-Code & Zero Vendor Lock-in

sectestx generiert standardisierte, textbasierte **Bruno-Collections** (.bru-Dateien). Diese werden wie normaler Programmcode im Git-Repository versioniert und im Team geteilt – ideal für agiles Branching und Code-Review.

- **Eigenständig lauffähig:** Die Collections laufen über den Open-Source-Bruno-CLI-Runner in jeder Pipeline – GitLab, GitHub Actions, Azure DevOps.
- **100 % Ihr Eigentum:** Einmal generiert, gehören die Tests vollständig Ihnen. Es gibt keinen Tool-Zwang, um sie auszuführen.
- **Ohne Infrastruktur-Hürden:** Keine Docker-Umgebung, keine WSL, keine Linux-Kenntnisse nötig – Installation und Start als native Desktop-App.

DER PRAKTISCHE EFFEKT

Ihre Sicherheitstests werden zu einem normalen, versionierten Teil des Repositorys – nachvollziehbar, wiederholbar und unabhängig von jedem Anbieter. Was einmal generiert wurde, läuft auch in fünf Jahren noch.

KONTAKT
whitepaper@leanofy.de

COMPLIANCE
DORA / NIS2 Ready

KAPITEL 05 · ZUSAMMENARBEIT

Das LEANOFY Service- & Kooperationsmodell

Technologie löst Probleme erst, wenn sie nahtlos in die Teamkultur integriert ist. Deshalb liefert LEANOFY nicht nur ein Werkzeug, sondern begleitet Ihr Team mit erfahrenen Testmanagern und Security-Engineers – flexibel, in vier klar definierten Phasen.

PHASE 01 **Vorbereitung & Setup** › PHASE 02 **Test-Generierung** › PHASE 03 **Analyse & Verifizierung** › PHASE 04 **CI/CD & Handover**

PHASE 01 **Vorbereitung & Setup.** Sichere lokale Einrichtung der nativen Desktop-App, Konfiguration des KI-Modells (BYOK, lokal oder offline) passend zu Ihren Compliance-Vorgaben sowie Einlesen und Strukturieren Ihrer OpenAPI-Spezifikationen.

PHASE 02 **Automatisierte Test-Generierung.** Der SlotFiller analysiert die Business-Logik, identifiziert kritische Endpunkte und generiert einsatzbereite Bruno-Collections für BOLA, Mass Assignment und BFLA.

PHASE 03 **Analyse & Verifizierung.** Gemeinsames Review der Findings mit unseren Experten, False-Positive-Tuning zur Minimierung von Rauschen und Generierung revisionssicherer PDF- und HTML-Reports für CISOs und Auditoren.

PHASE 04 **CI/CD-Integration & Handover.** Nahtlose Integration der Tests in Ihre Pipeline, Team-Training für den eigenständigen Betrieb und flexibler Experten-Support über die Einführungsphase hinaus.

AUDIT-READY REPORTING

Jeder Testlauf mündet in einen revisionssicheren Report mit OWASP-Mapping und Compliance-Verweisen – der lückenlose Nachweis der DORA- und NIS2-Konformität für Ihre Auditoren.

FAZIT · NÄCHSTER SCHRITT

Agilität und API-Sicherheit gehören zusammen

Regulierte Unternehmen müssen sich nicht länger zwischen Release-Geschwindigkeit und Compliance entscheiden. Mit sectestx werden kontinuierliche, logische API-Sicherheitstests zum festen Bestandteil des agilen Entwicklungsprozesses – datensouverän im eigenen Netz, automatisiert und revisionssicher. So wird API-Security von der Release-Bremse zum echten Compliance-Turbo.

Für Entwicklungsleiter

Den Deploy-Button mit gutem Gewissen drücken – ohne Angst vor Datenpannen durch logische Lücken in geschäftskritischen APIs.

Für CISOs

API-Sicherheit kontinuierlich und datensouverän nachweisen – mit revisionssicheren Reports für jedes DORA- und NIS2-Audit.

Für QA & DevOps

Sicherheitstests als Code im Git-Repository, automatisiert in der Pipeline, mit sofortigem Feedback zu jeder Code-Änderung.

Ihr Weg zum Pilotprojekt

SCHRITT 01

Erstgespräch

30 Minuten, unverbindlich: Wir klären Scope, Compliance-Anforderungen und Ihr gewünschtes KI-Betriebsmodell.

SCHRITT 02

Pilot an Ihrer API

sectestx wird lokal eingerichtet und generiert die ersten Test-Collections für eine Ihrer geschäftskritischen APIs.

SCHRITT 03

Auswertung & Roadmap

Gemeinsames Review der Findings und ein konkreter Fahrplan für die CI/CD-Integration.

Schließen Sie die Lücke zwischen Agilität und API-Sicherheit.

Vereinbaren Sie ein unverbindliches Erstgespräch – absolut datensouverän, ohne Cloud-Zwang.

Erstgespräch vereinbaren → sectestx.leanofy.de

LEANOFY GMBH · INFO@LEANOFY.DE · SECTESTX.LEANOFY.DE