Security & Privacy by Design in the SDLC:

Why, When, How?

PRESENTER

Cat Easdon

Senior Privacy Engineer & Team Captain

whoami

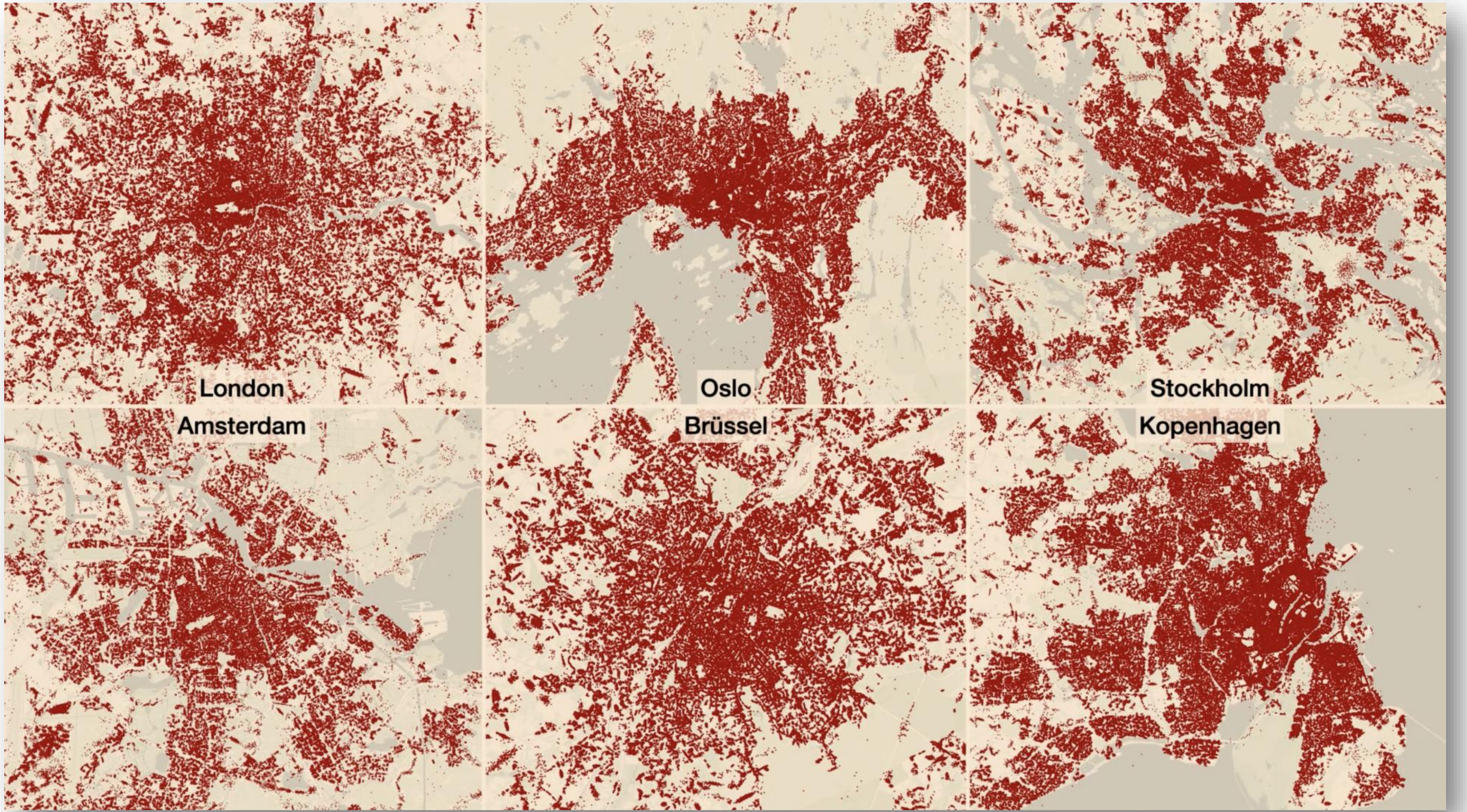
- Lead Dynatrace's Privacy Engineering team in close collaboration with our Product Security teams
- Outside of work: research and tech policy, trail running, hiking, skiing…
- Previously: hacking CPUs at TU Graz

Outline

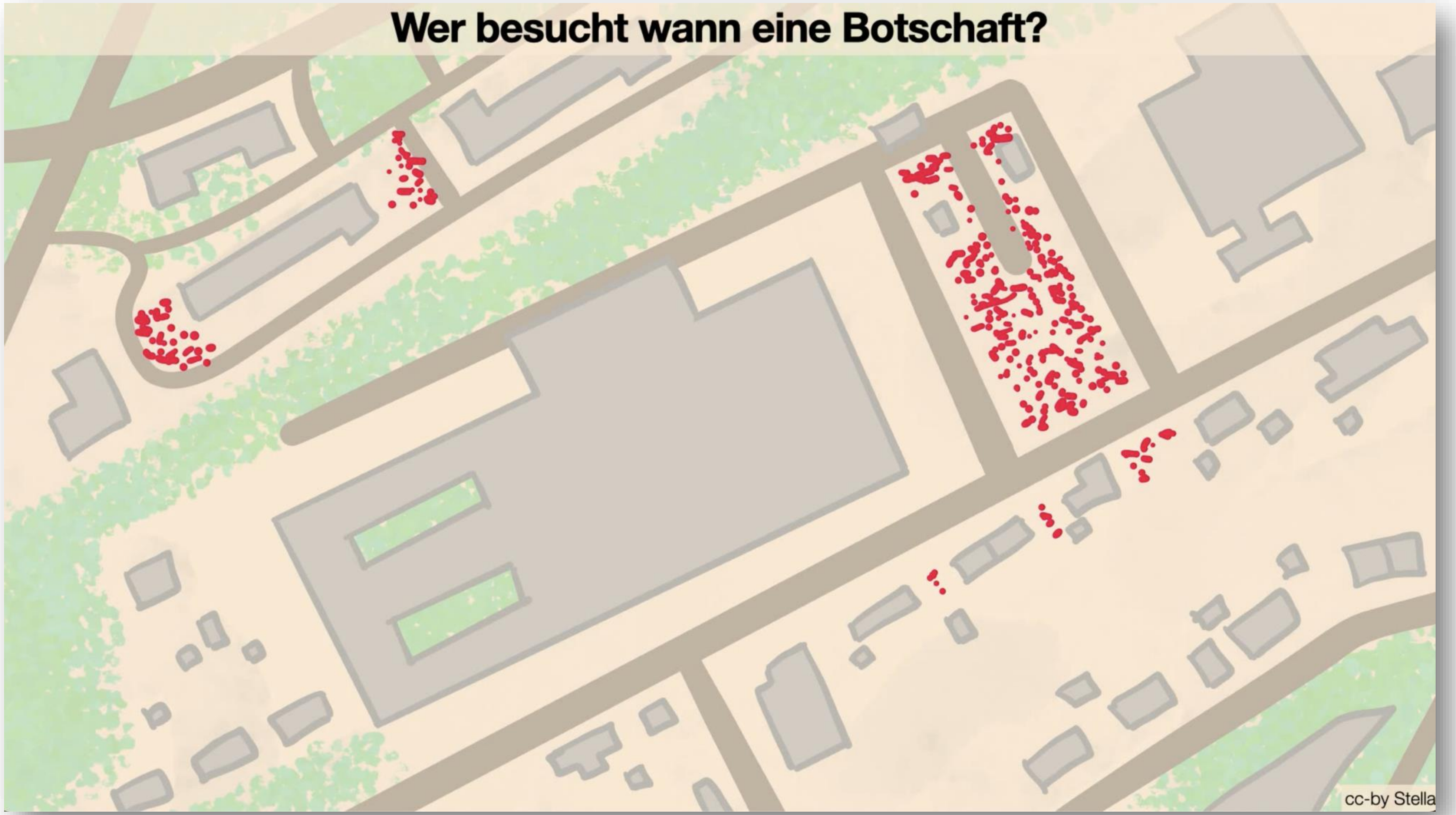
- **Why?**
    - And what does ‘by design’ really mean?
- **When?**
- **How?**
    - Two case studies: technical privacy controls, scaling vulnerability management
- **Q&A**

Why?

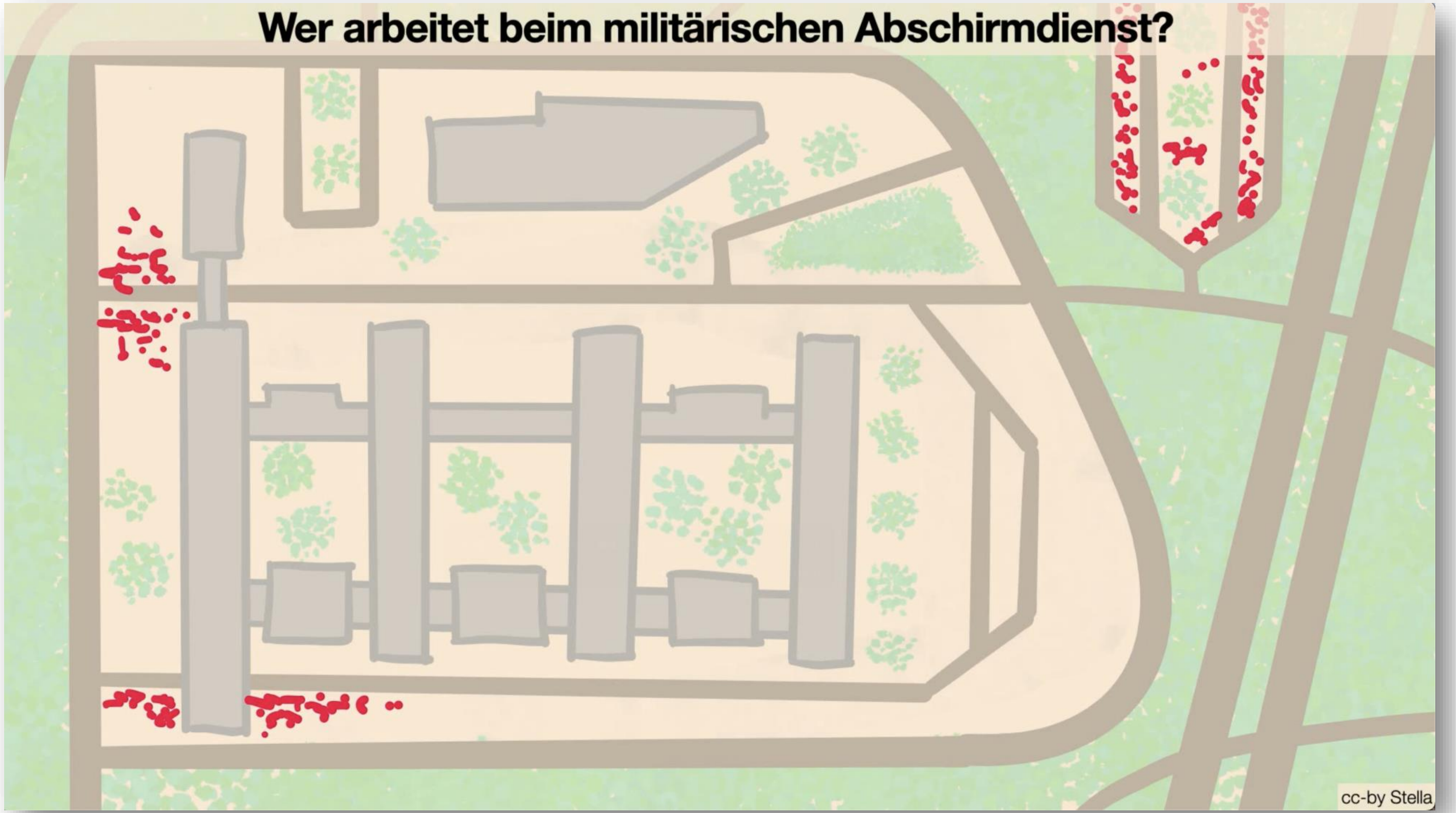
Why care about security and privacy by design?



Why care about security and privacy by design?



Why care about security and privacy by design?



Why care about security and privacy by design?

Volkswagen 2024 data breach (coverage: 38C3, Der Spiegel)

Insecurity by design

- Spring API endpoint to retrieve heap dumps publicly exposed
  - Contained credentials for AWS account and IDP's client ID and client secret
  - Leaked data for 800 000 electric cars; precise location data for 460 000 of them
- Compromised credentials not revoked after responsible disclosure

Why care about security and privacy by design?

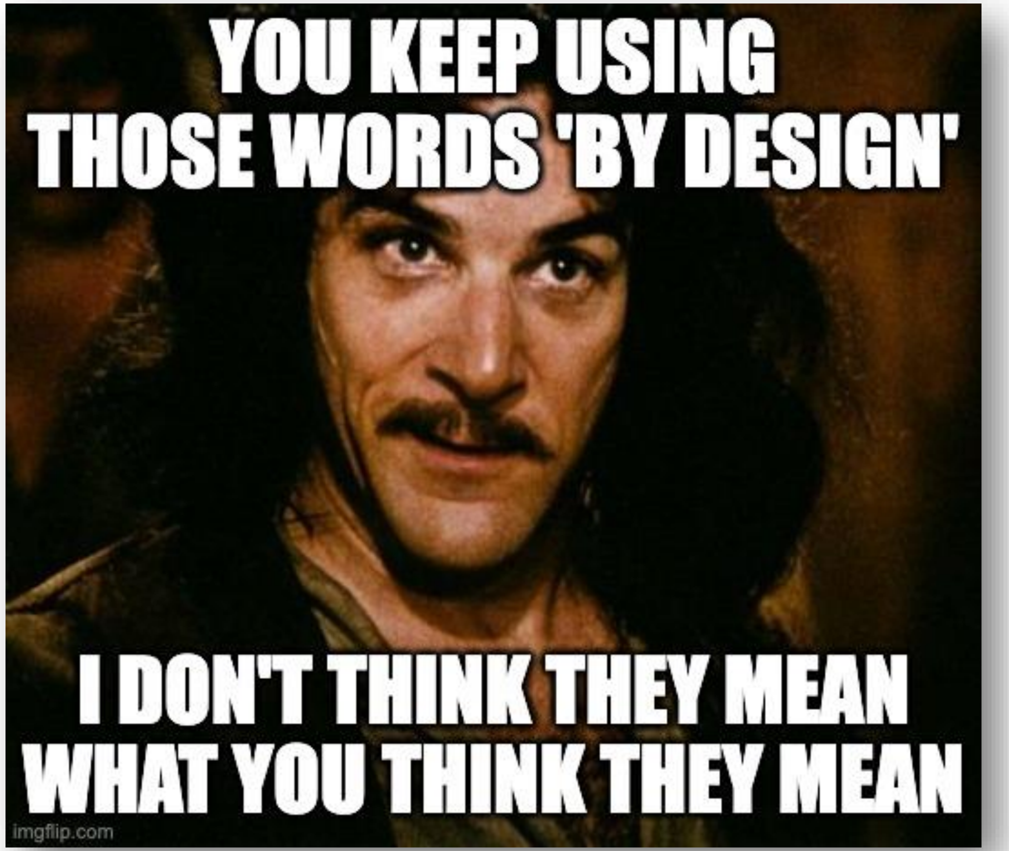
Volkswagen 2024 data breach (coverage: 38C3, Der Spiegel)

Privacy violated by design - but it’s ok, it’s not ‘sensitive’ data (!!)

- No pseudonymization
- Geo-coordinates not truncated (10cm precision)
- EU eCall Regulation (2015): mandates microphones, GPS, and internet/cellular comms in cars

What?

What does ‘by design’ really mean?



What does 'by design' really mean?

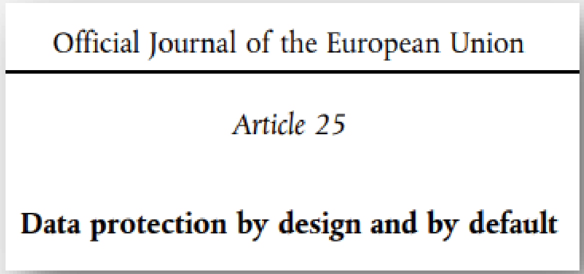
1

Voluntary pledge to demonstrate progress in seven areas (US CISA)

SHIFTING THE BALANCE OF CYBERSECURITY RISK: PRINCIPLES AND APPROACHES FOR SECURE BY DESIGN SOFTWARE

2

Security is a core business requirement, **principles** implemented during design, **secure defaults** (US CISA)



Official Journal of the European Union

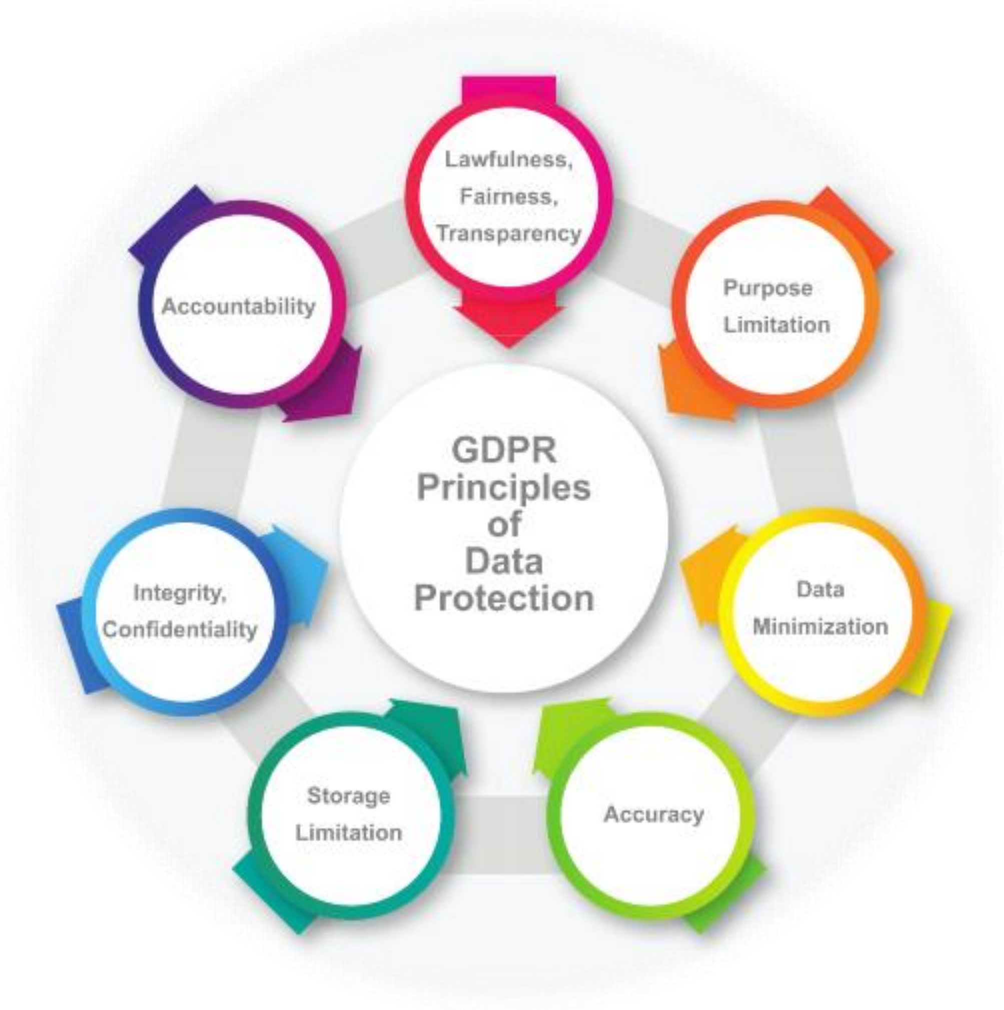
Article 25

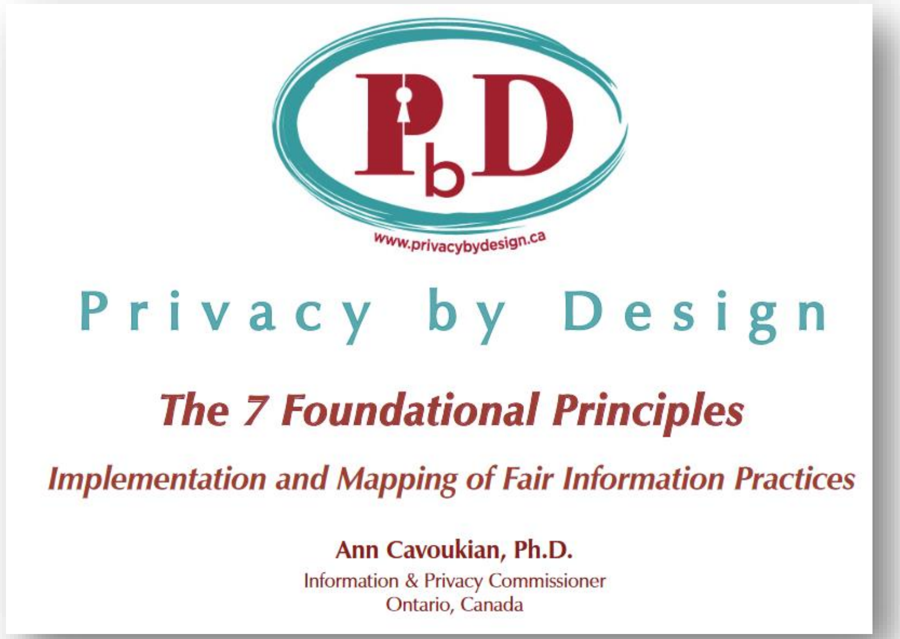
Data protection by design and by default

3

Principles implemented during design and data processing, **private defaults** (GDPR)

What does ‘by design’ really mean?


GDPR Principles of Data Protection
Lawfulness, Fairness, Transparency
Purpose Limitation
Data Minimization
Accuracy
Storage Limitation
Integrity, Confidentiality
Accountability


www.privacybydesign.ca

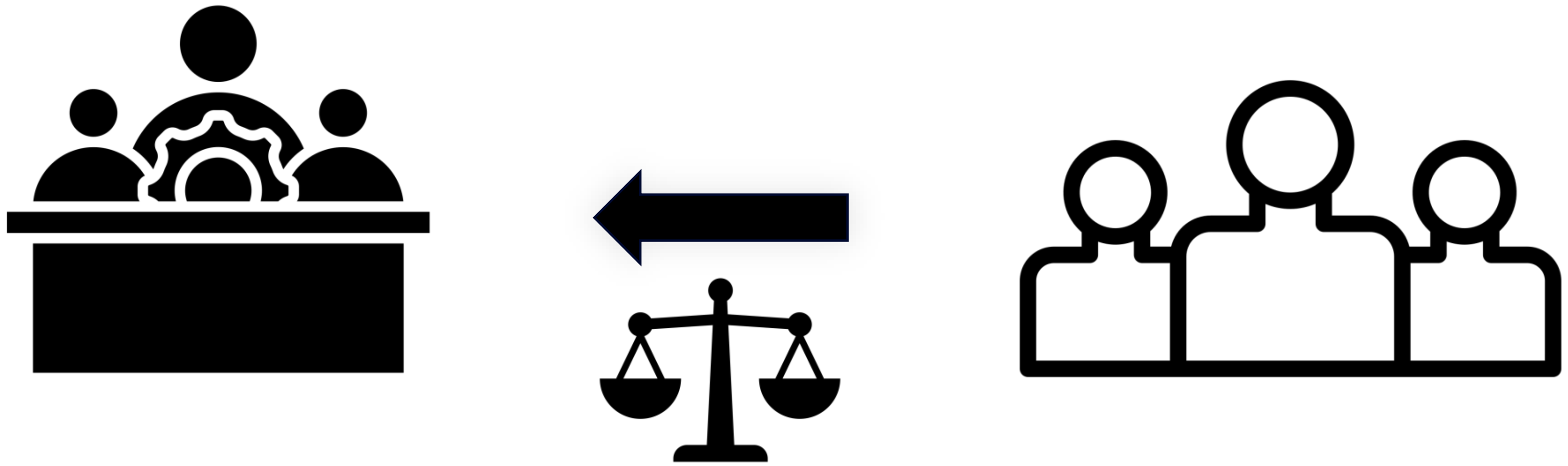
Privacy by Design

The 7 Foundational Principles

Implementation and Mapping of Fair Information Practices

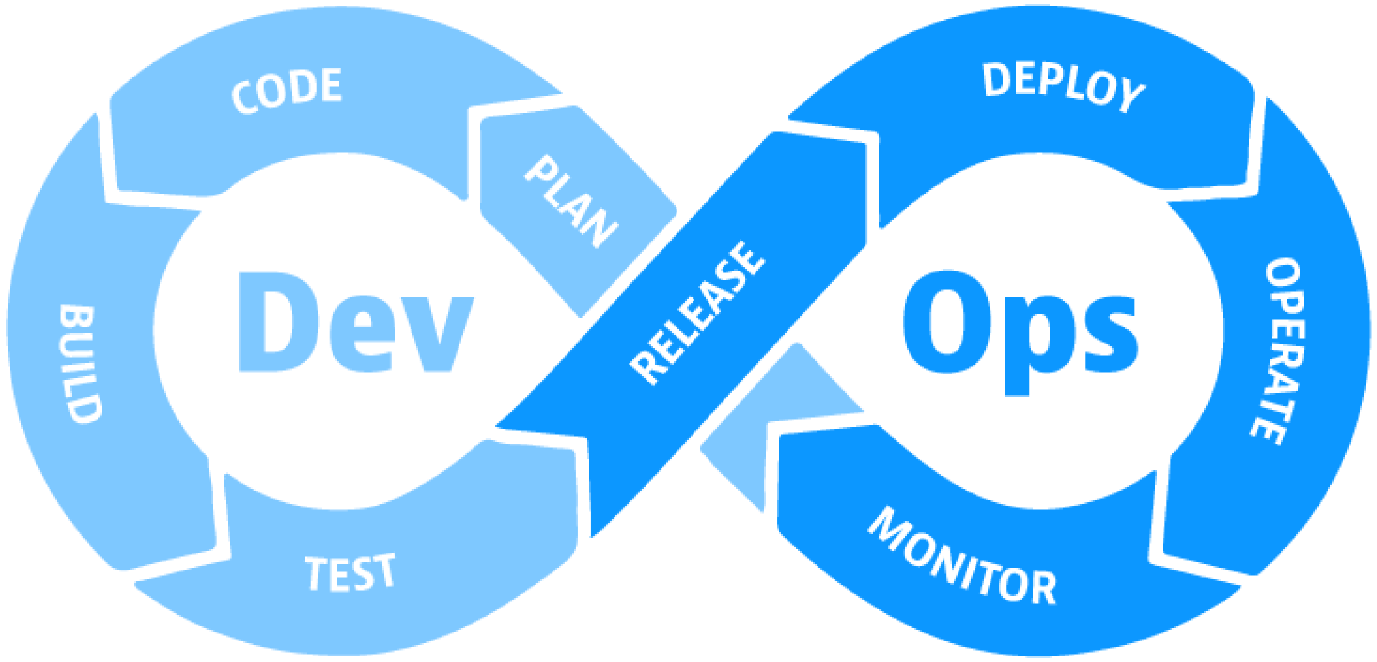
Ann Cavoukian, Ph.D.
Information & Privacy Commissioner
Ontario, Canada

Shifting risk from customers -> company



When?

Controls in the SDLC



Secure Development Lifecycle: Dynatrace Edition

Design & Architecture
- Threat modeling
- Evaluation of external libraries & services

Development
- Code reviews
- Secure coding standards
- Security scans in the IDE
- Secret scanning

CI/CD
- Static code analysis (SAST)
- Software composition analysis (SCA)
- Code signing
- Automated dependency updates
- Container image scans
- Infrastructure-as-code scans
- Malware scans

Pre-Production
- Penetration tests
- Web-app scans (DAST)
- Security assessments

Production
- Intrusion detection & prevention
- 24/7 incident response
- Vulnerability scans
- Security posture management (*SPM)
- Runtime application protection

Protected by Dynatrace Application Security

Agile, bi-weekly sprints

Governance: security policies & KPIs, vulnerability & risk management program

Training & enablement: security champions program, security trainings and security onboarding program

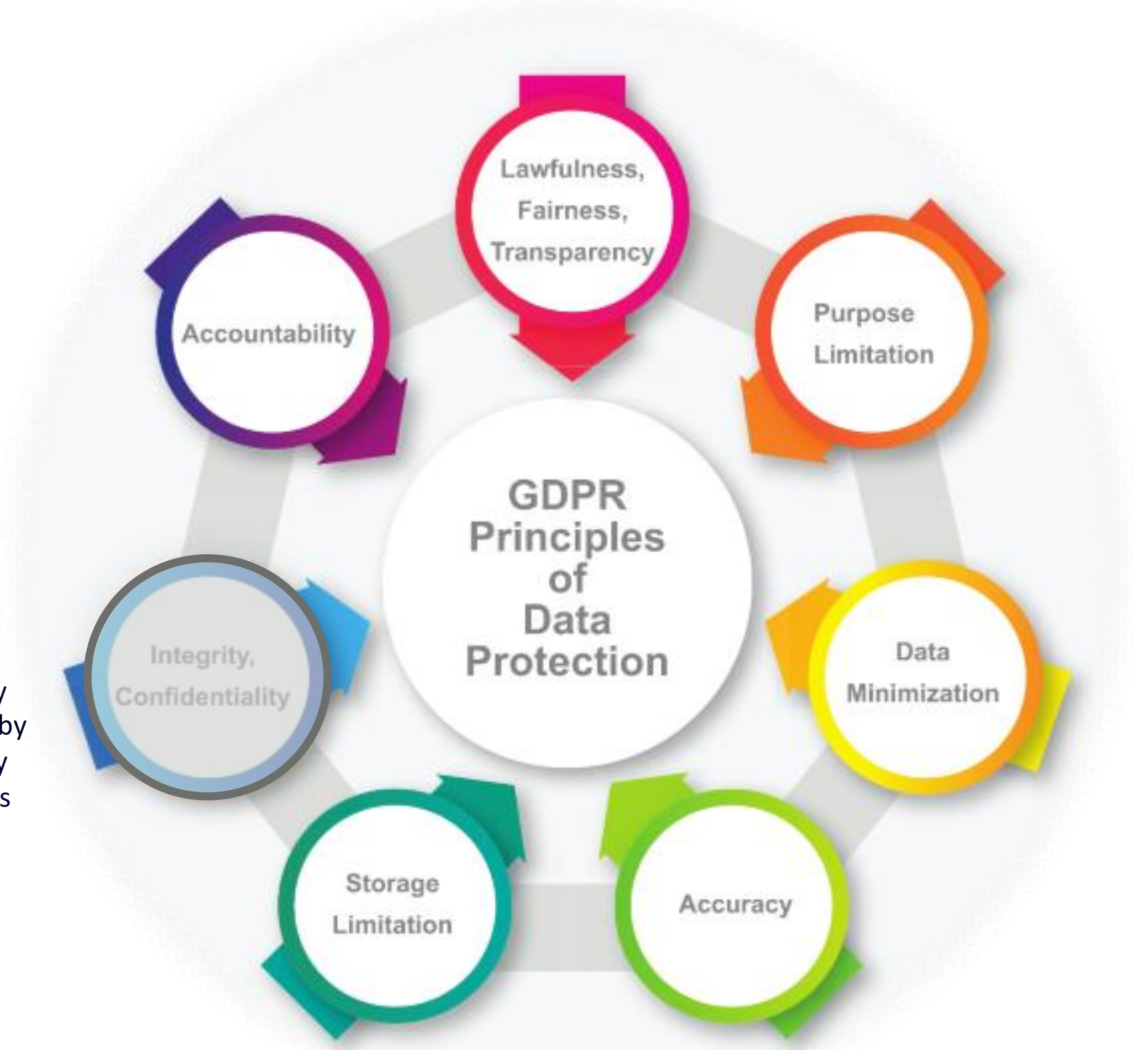
External assessments: compliance audits, annual penetration tests, annual red-team assessments, public bug bounty program

Continuous monitoring & improvement

How?

Case Study 1: Privacy Controls

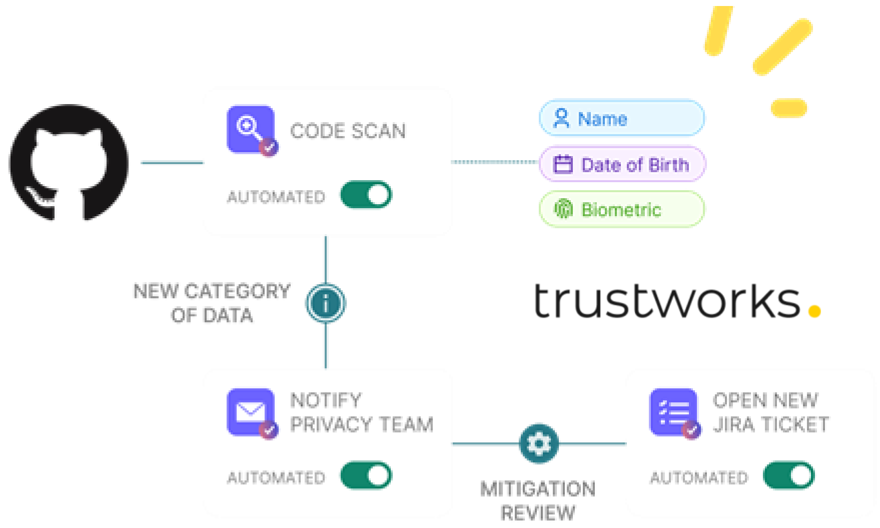
(Ideal) Scope of Privacy Controls

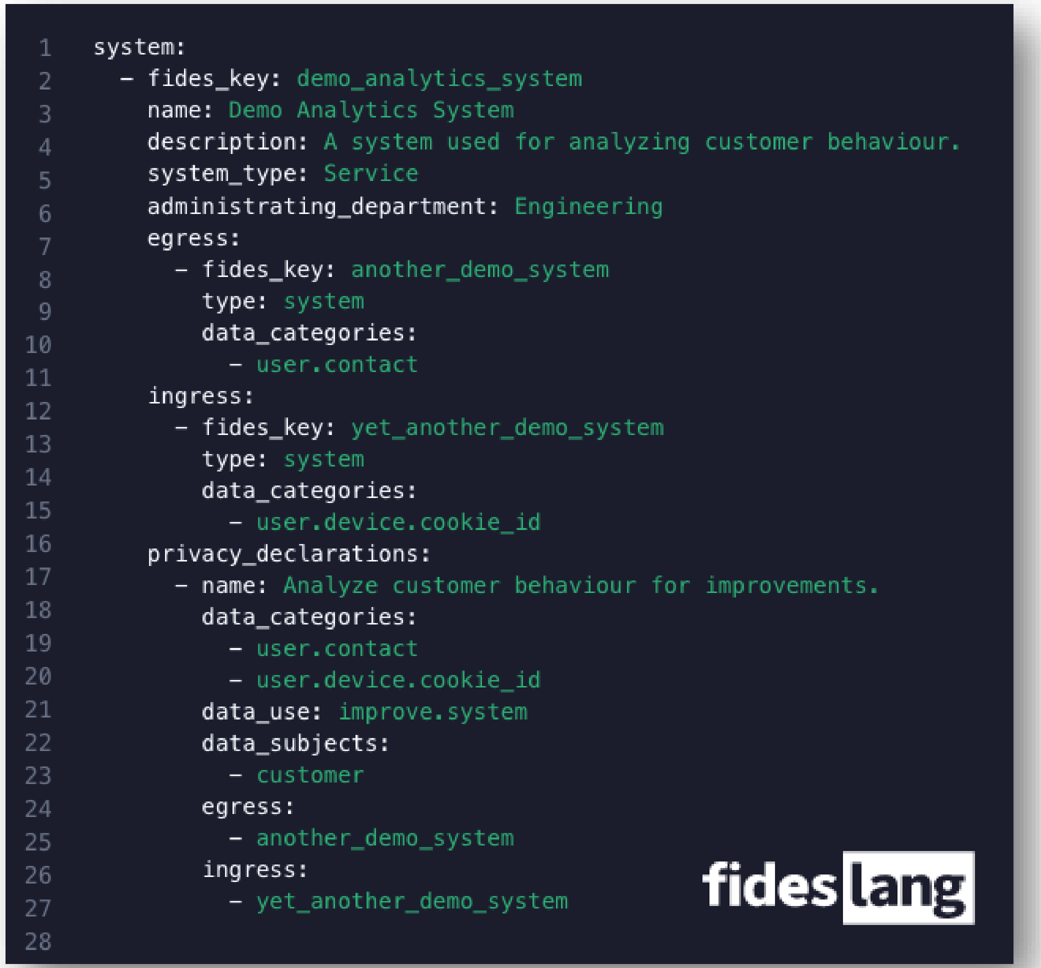

Lawfulness, Fairness, Transparency
Purpose Limitation
Data Minimization
Accuracy
Storage Limitation
Integrity, Confidentiality
Accountability
GDPR Principles of Data Protection

Already covered by security controls

Mapping Personal Data Flows

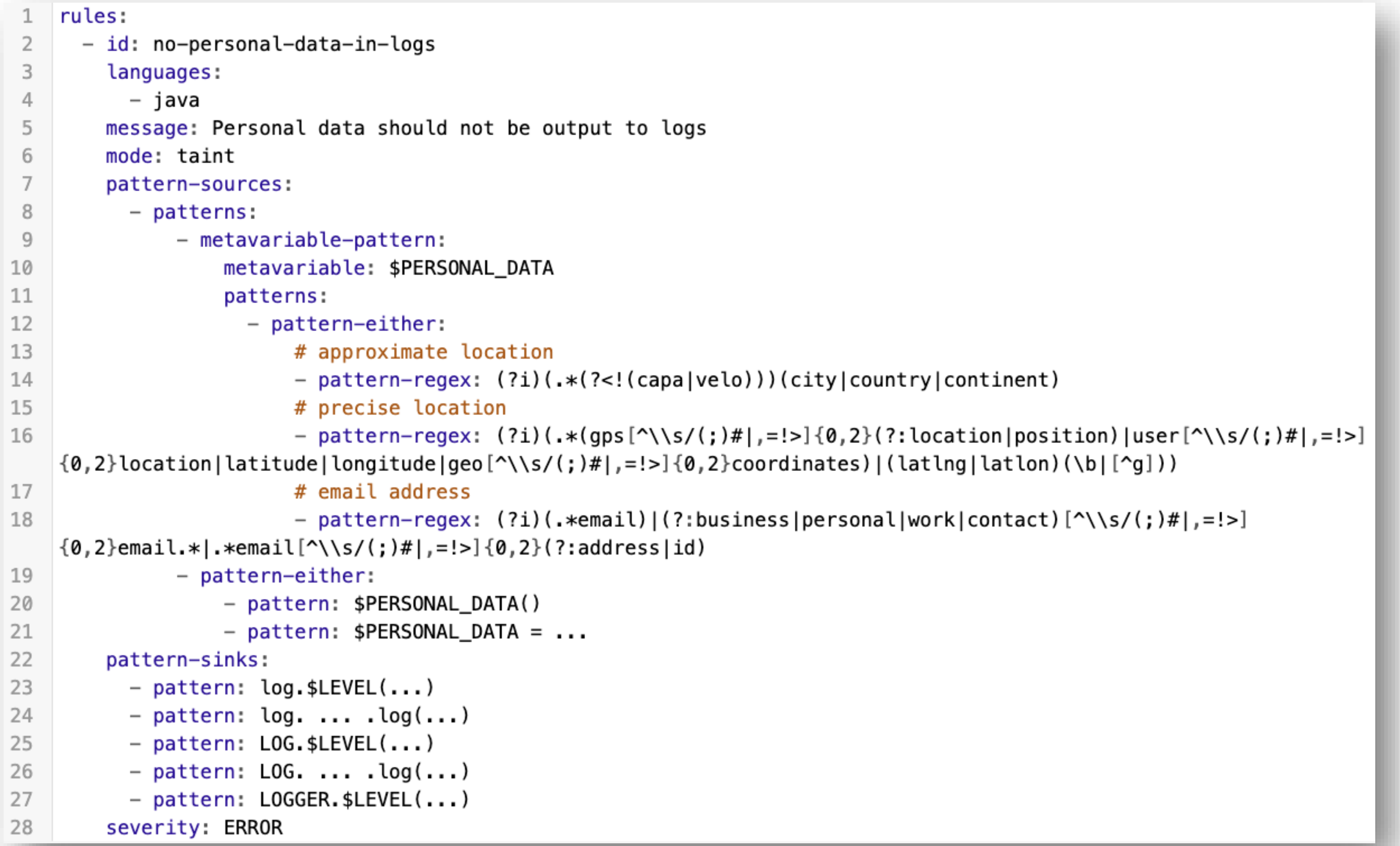
- Code-level annotations
- Per-system manifests
- Scanning
  - Code-level: Privado, Trustworks
  - Database and data warehouse level: assorted vendors ('data discovery')




```yaml
system:
  - fides_key: demo_analytics_system
    name: Demo Analytics System
    description: A system used for analyzing customer behaviour.
    system_type: Service
    administrating_department: Engineering
    egress:
      - fides_key: another_demo_system
        type: system
        data_categories:
          - user.contact
    ingress:
      - fides_key: yet_another_demo_system
        type: system
        data_categories:
          - user.device.cookie_id
    privacy_declarations:
      - name: Analyze customer behaviour for improvements.
        data_categories:
          - user.contact
          - user.device.cookie_id
        data_use: improve.system
        data_subjects:
          - customer
        egress:
          - another_demo_system
        ingress:
          - yet_another_demo_system

```

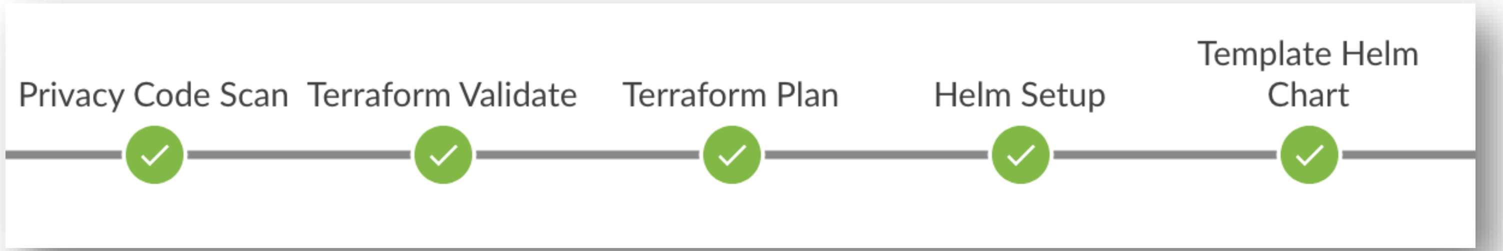
Privacy Code Scanning with Semgrep

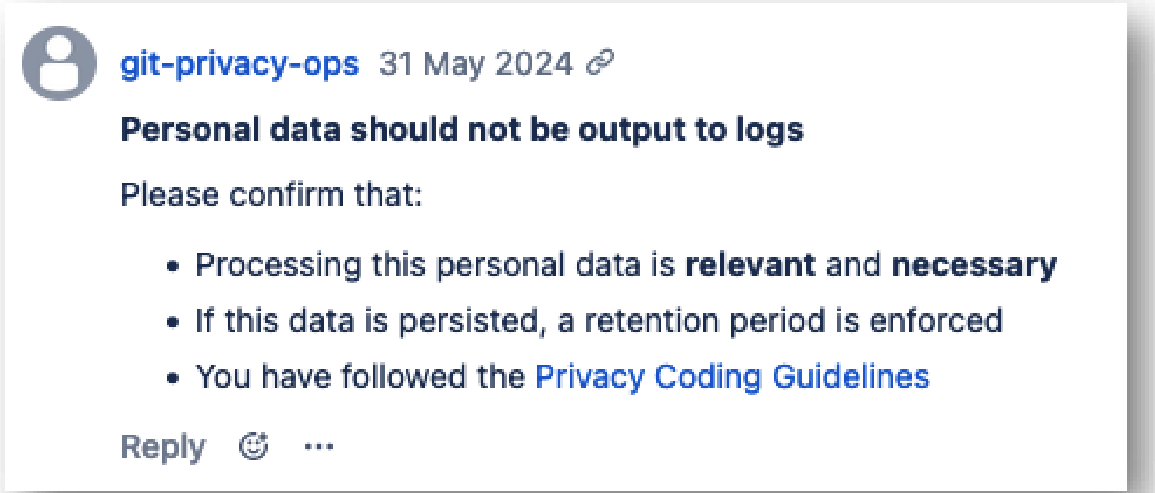


```yaml
rules:
  - id: no-personal-data-in-logs
    languages:
      - java
    message: Personal data should not be output to logs
    mode: taint
    pattern-sources:
      - patterns:
          - metavariable-pattern:
              metavariable: $PERSONAL_DATA
              patterns:
                - pattern-either:
                    # approximate location
                    - pattern-regex: (?i)(.*(?<!(capa|velo)))(city|country|continent)
                    # precise location
                    - pattern-regex: (?i)(.*(gps[^\\s/(;)#|,=!>]{0,2}(?:location|position)|user[^\\s/(;)#|,=!>]{0,2}location|latitude|longitude|geo[^\\s/(;)#|,=!>]{0,2}coordinates)|(latlng|latlon)(\b|[^g]))
                    # email address
                    - pattern-regex: (?i)(.*email)|(?:business|personal|work|contact)[^\\s/(;)#|,=!>]{0,2}email.*|.*email[^\\s/(;)#|,=!>]{0,2}(?:address|id)
          - pattern-either:
              - pattern: $PERSONAL_DATA()
              - pattern: $PERSONAL_DATA = ...
    pattern-sinks:
      - pattern: log.$LEVEL(...)
      - pattern: log. ... .log(...)
      - pattern: LOG.$LEVEL(...)
      - pattern: LOG. ... .log(...)
      - pattern: LOGGER.$LEVEL(...)
    severity: ERROR
```

Regexes adapted from Privado OSS

Privacy Code Scanning with Semgrep




git-privacy-ops 31 May 2024

Personal data should not be output to logs

Please confirm that:

- Processing this personal data is **relevant** and **necessary**
- If this data is persisted, a retention period is enforced
- You have followed the Privacy Coding Guidelines

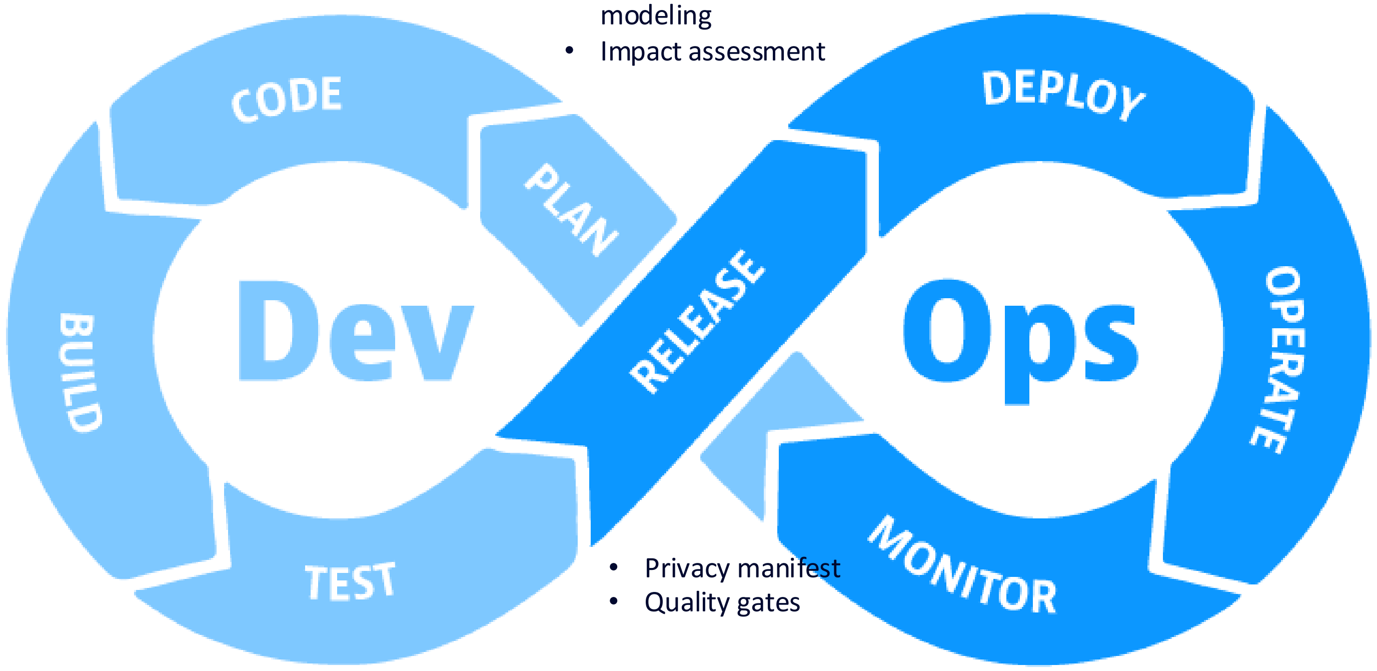
A Look Into The Future…

- Technical privacy reviews
- Coding guidelines
- Docs and enablement
- Developer platform integrations

- Privacy threat modeling
- Impact assessment

- IaC policies for cross-border infra

- Static code analysis
- Code- and system-level personal data annotations



- Privacy ops (privacy rights fulfillment)
- Incident / data breach response
- Disaster recovery

- Privacy manifest
- Quality gates

- Automated testing of privacy-relevant features
- Privacy penetration testing
- Red teaming

- Data catalog
- Automated personal data flow mapping
- PII detection
- Data loss prevention
- Abuse detection

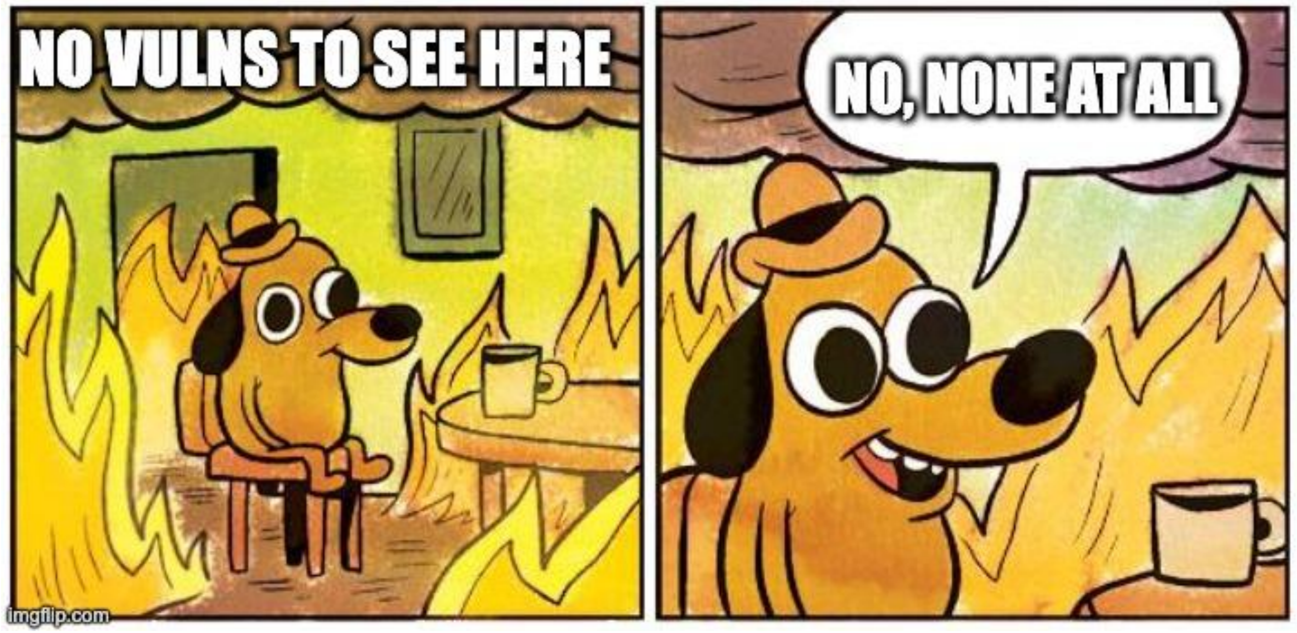
How?

Case Study 2: Scaling Vulnerability Management

Scaling Vulnerability Management

Maturity level 0

- Vulnerabilities get discovered by accident ☺
- Nobody feels responsible for them



Scaling Vulnerability Management

Maturity level 1

- Somebody feels responsible for manually finding and handling vulnerabilities
- …and is drowning in work trying to handle them
- The teams owning vulnerable codebases are rarely cooperative

Scaling Vulnerability Management

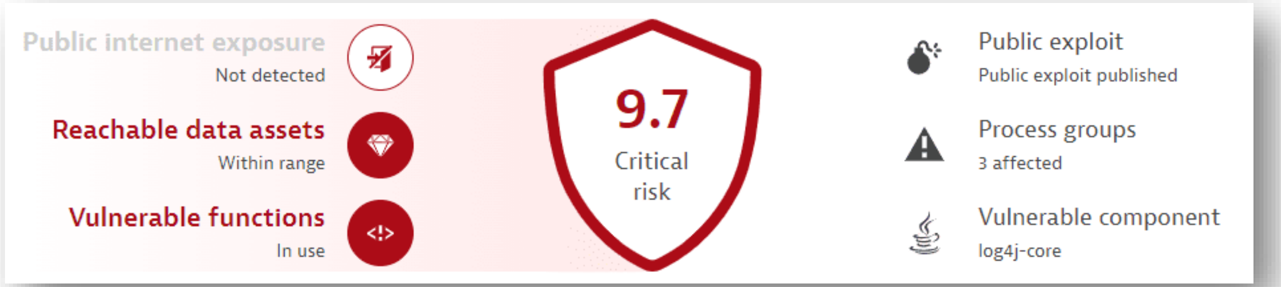
Maturity level 2

- Automated scanning will solve all our problems 💯 ✨ 🪄
- Owning teams have been ~~coerced into~~ *gently reminded of the importance of* prompt vuln handling

Scaling Vulnerability Management

Maturity level 3

- Basic scan results are extended with context-specific insights to reduce false positives and speed up triage and prioritization



Public internet exposure
Not detected

Reachable data assets
Within range

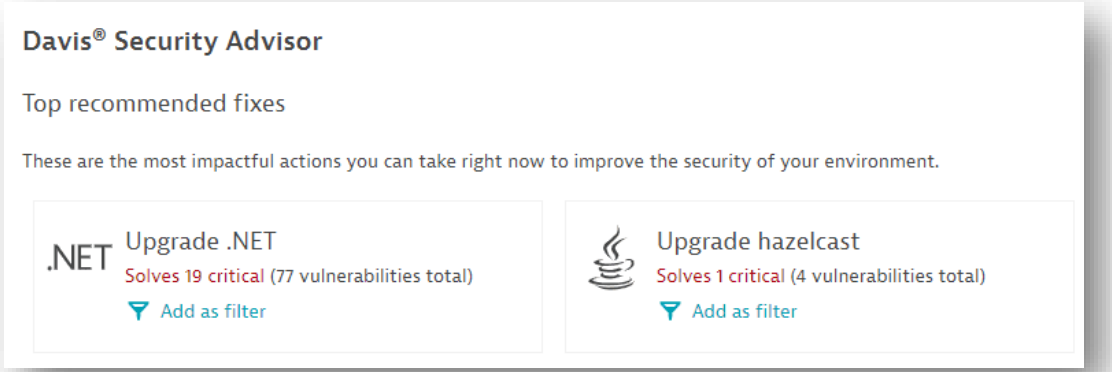
Vulnerable functions
In use

9.7
Critical risk

Public exploit
Public exploit published

Process groups
3 affected

Vulnerable component
log4j-core



Davis® Security Advisor

Top recommended fixes

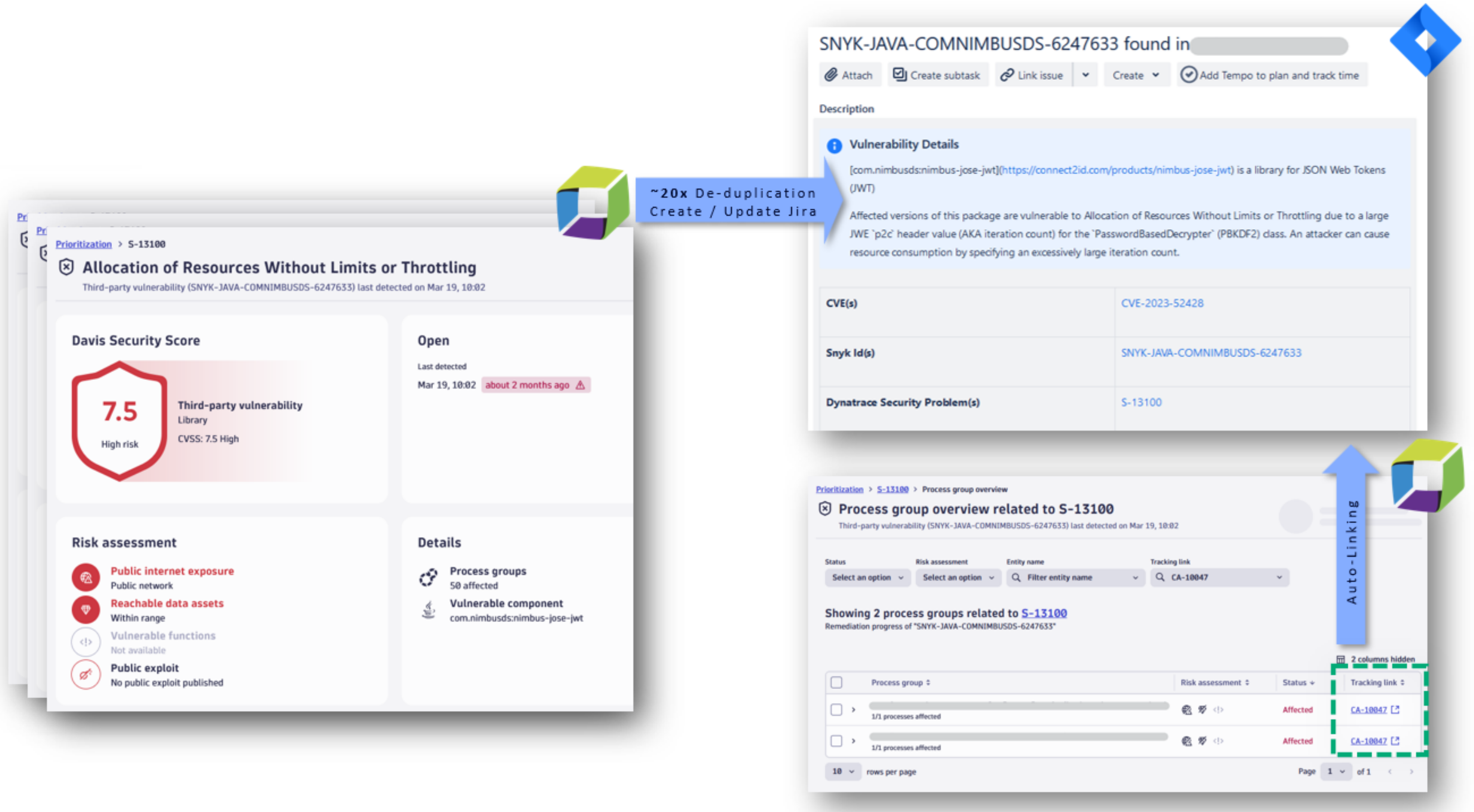
These are the most impactful actions you can take right now to improve the security of your environment.

.NET Upgrade .NET Solves 19 critical (77 vulnerabilities total) Add as filter	Upgrade hazelcast Solves 1 critical (4 vulnerabilities total) Add as filter

Scaling Vulnerability Management

Maturity level 3+

- Context-specific deduplication and automated triage



Prioritization > S-13100

Allocation of Resources Without Limits or Throttling

Third-party vulnerability (SNYK-JAVA-COMNIMBUSDS-6247633) last detected on Mar 19, 10:02

Davis Security Score

7.5 High risk

Third-party vulnerability
Library
CVSS: 7.5 High

Open

Last detected
Mar 19, 10:02 about 2 months ago

Risk assessment

- Public internet exposure – Public network
- Reachable data assets – Within range
- Vulnerable functions – Not available
- Public exploit – No public exploit published

Details

- Process groups – 50 affected
- Vulnerable component – com.nimbusds:nimbus-jose-jwt

SNYK-JAVA-COMNIMBUSDS-6247633 found in [redacted]

Attach | Create subtask | Link issue | Create | Add Tempo to plan and track time

Description

Vulnerability Details

[com.nimbusds:nimbus-jose-jwt](https://connect2id.com/products/nimbus-jose-jwt) is a library for JSON Web Tokens (JWT)

Affected versions of this package are vulnerable to Allocation of Resources Without Limits or Throttling due to a large JWE `p2c` header value (AKA iteration count) for the `PasswordBasedDecrypter` (PBKDF2) class. An attacker can cause resource consumption by specifying an excessively large iteration count.

CVE(s)	CVE-2023-52428
Snyk Id(s)	SNYK-JAVA-COMNIMBUSDS-6247633
Dynatrace Security Problem(s)	S-13100

Prioritization > S-13100 > Process group overview

Process group overview related to S-13100

Third-party vulnerability (SNYK-JAVA-COMNIMBUSDS-6247633) last detected on Mar 19, 10:02

Status: Select an option | Risk assessment: Select an option | Entity name: Filter entity name | Tracking link: CA-10047

Showing 2 process groups related to S-13100

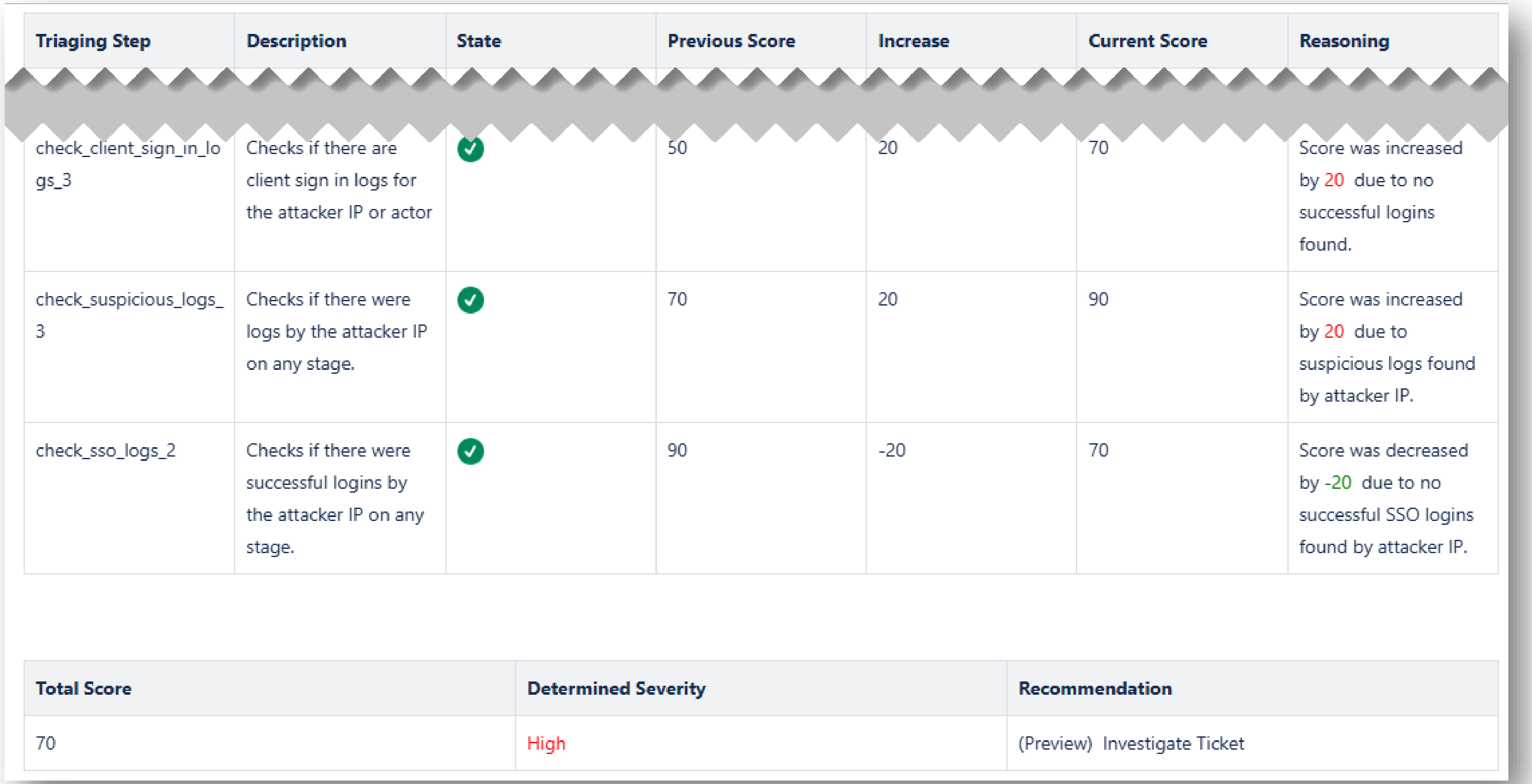
Remediation progress of "SNYK-JAVA-COMNIMBUSDS-6247633"

2 columns hidden

Process group	Risk assessment	Status	Tracking link
[redacted] 1/1 processes affected		Affected	CA-10047
[redacted] 1/1 processes affected		Affected	CA-10047

10 rows per page   Page 1 of 1

Scaling Vulnerability Management



Triaging Step	Description	State	Previous Score	Increase	Current Score	Reasoning
check_client_sign_in_logs_3	Checks if there are client sign in logs for the attacker IP or actor	✓	50	20	70	Score was increased by 20 due to no successful logins found.
check_suspicious_logs_3	Checks if there were logs by the attacker IP on any stage.	✓	70	20	90	Score was increased by 20 due to suspicious logs found by attacker IP.
check_sso_logs_2	Checks if there were successful logins by the attacker IP on any stage.	✓	90	-20	70	Score was decreased by -20 due to no successful SSO logins found by attacker IP.

Total Score	Determined Severity	Recommendation
70	High	(Preview) Investigate Ticket

Scaling Vulnerability Management

But wait a second…

Finding vulnerabilities is only part of the story!

What have we missed so far?

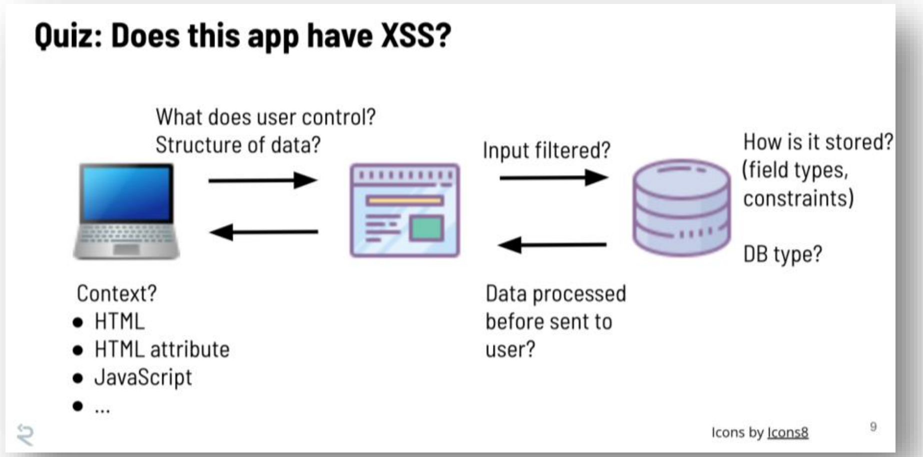
Scaling Vulnerability Management

Prevent

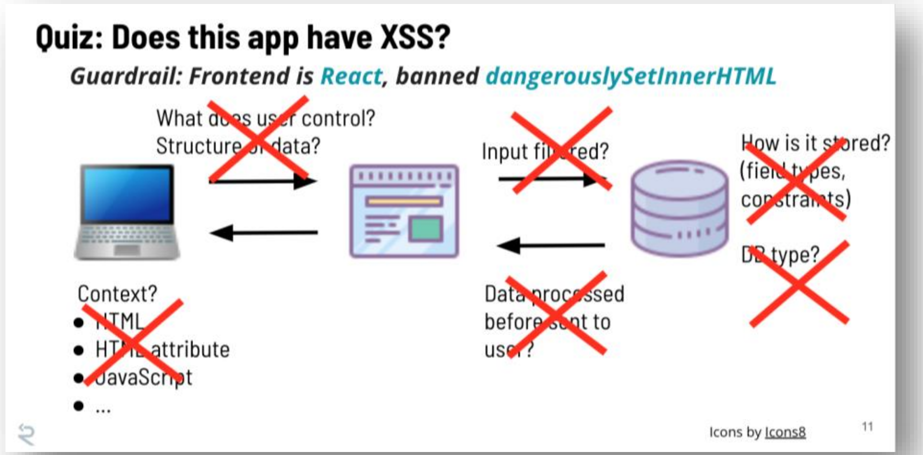
- Understand what you have where (software + data catalog)
- Secure defaults, a.k.a. paved roads, golden paths, guardrails -> prevent entire vuln classes
- Automated patching
- Security advocate / champion programs

Remediate

- Leadership buy-in
- Establish guidance for each vuln class, upskill devs
- Automate remediation monitoring



Quiz: Does this app have XSS?

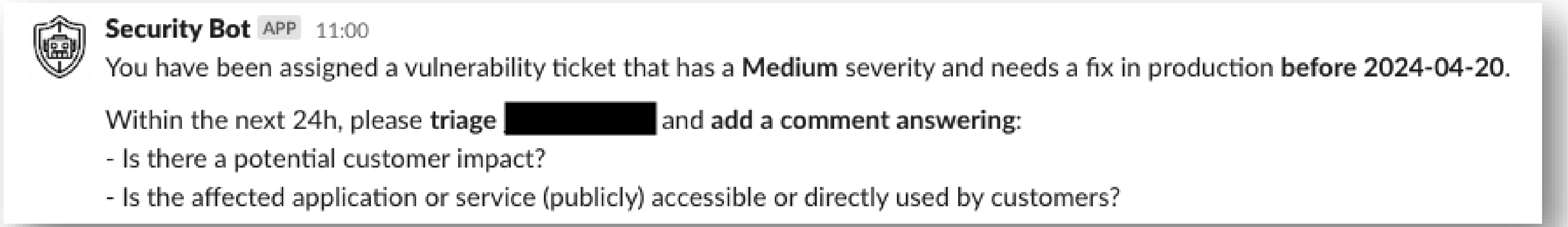


Quiz: Does this app have XSS?

Guardrail: Frontend is React, banned dangerouslySetInnerHTML

How to Kill Bug Classes with Secure Guardrails, Clint Gibler & Colleen Dai

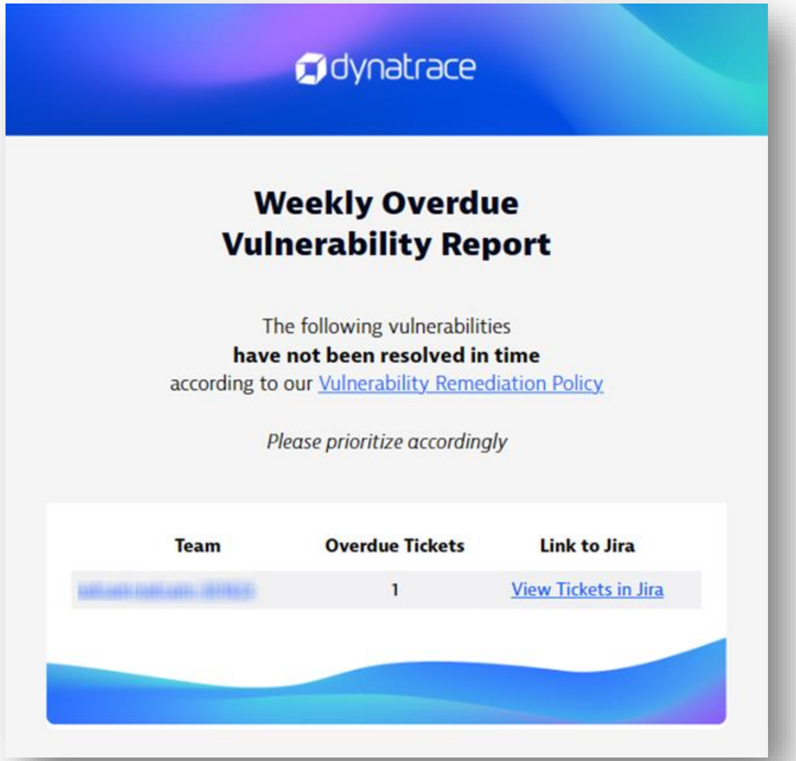
Scaling Vulnerability Management



Security Bot APP 11:00

You have been assigned a vulnerability ticket that has a **Medium** severity and needs a fix in production **before 2024-04-20**.

Within the next 24h, please **triage** [redacted] and **add a comment answering**:
- Is there a potential customer impact?
- Is the affected application or service (publicly) accessible or directly used by customers?



dynatrace

Weekly Overdue Vulnerability Report

The following vulnerabilities
have not been resolved in time
according to our Vulnerability Remediation Policy

Please prioritize accordingly

Team	Overdue Tickets	Link to Jira
[redacted]	1	View Tickets in Jira

Thanks for listening!

Copyright Notice

dynatrace

CLOUD DONE RIGHT