Ignorieren bis es knallt?

Security-Analysen

aus Entwickler- und Management-Perspektive

CQSE

Ann-Sophie Kracker
Nils Göde

Software-**Audits**

Kontinuierliche **Qualitäts- und Testkontrolle**

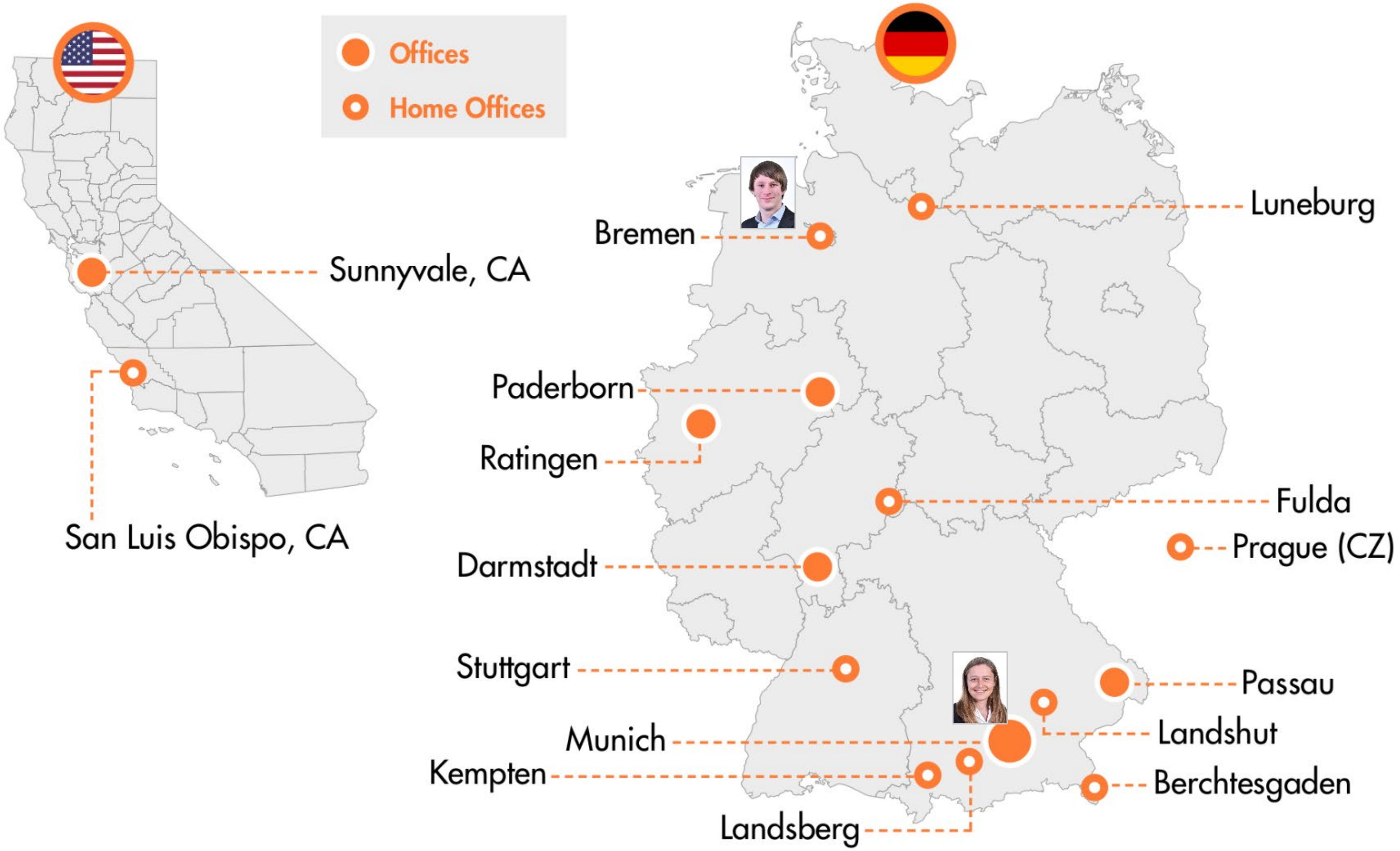
Teamscale

CQSE

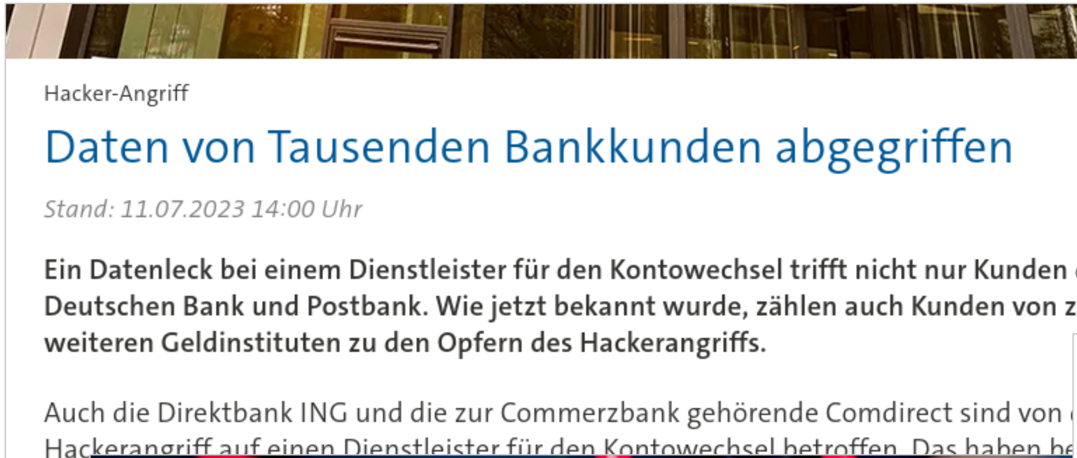
20+ **Promotionen** in Software Engineering

Eigene **Forschung**

Enger Kontakt zu **Universitäten**


● Offices
○ Home Offices
Sunnyvale, CA
San Luis Obispo, CA
Bremen
Luneburg
Paderborn
Ratingen
Fulda
Prague (CZ)
Darmstadt
Stuttgart
Passau
Landshut
Munich
Kempten
Berchtesgaden
Landsberg

Security



Hacker-Angriff

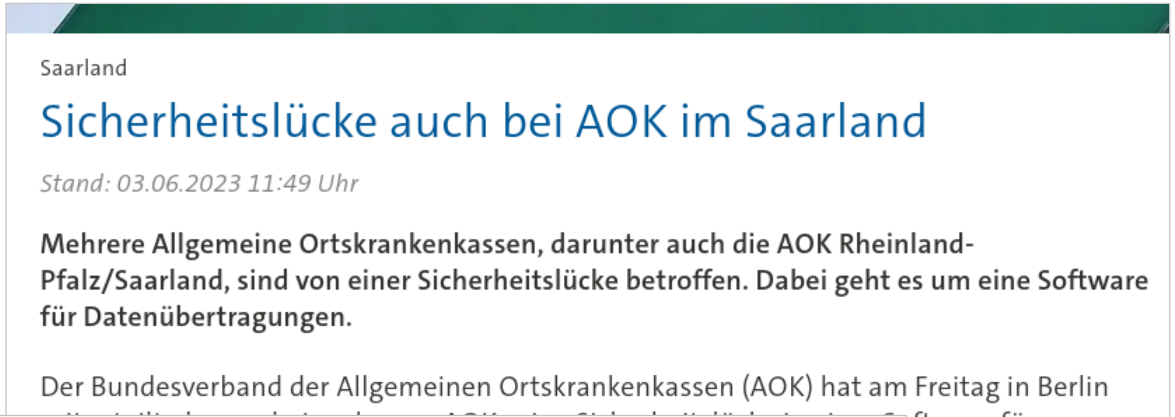
Daten von Tausenden Bankkunden abgegriffen

Stand: 11.07.2023 14:00 Uhr

Ein Datenleck bei einem Dienstleister für den Kontowechsel trifft nicht nur Kunden der Deutschen Bank und Postbank. Wie jetzt bekannt wurde, zählen auch Kunden von zwei weiteren Geldinstituten zu den Opfern des Hackerangriffs.

Auch die Direktbank ING und die zur Commerzbank gehörende Comdirect sind von
Hackerangriff auf einen Dienstleister für den Kontowechsel betroffen. Das haben be
Häu
Con

Bere
unb



Saarland

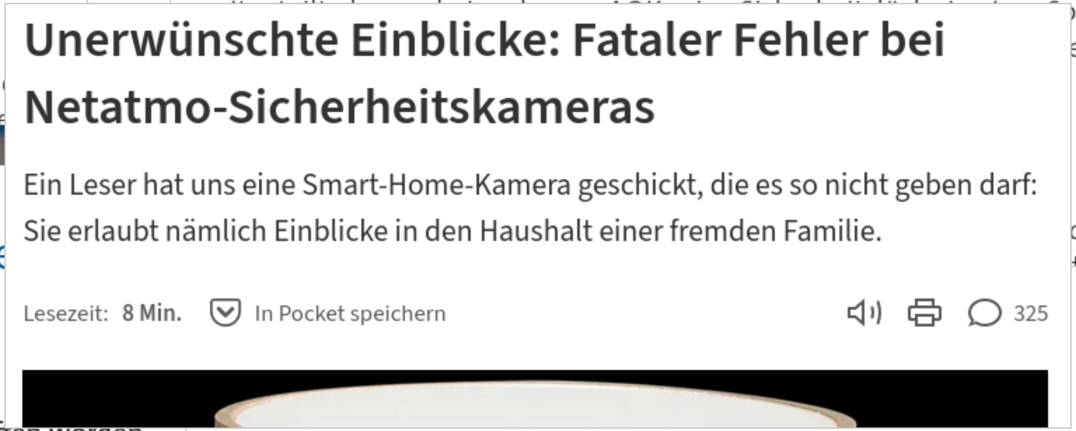
Sicherheitslücke auch bei AOK im Saarland

Stand: 03.06.2023 11:49 Uhr

Mehrere Allgemeine Ortskrankenkassen, darunter auch die AOK Rheinland-Pfalz/Saarland, sind von einer Sicherheitslücke betroffen. Dabei geht es um eine Software für Datenübertragungen.

Der Bundesverband der Allgemeinen Ortskrankenkassen (AOK) hat am Freitag in Berlin
oftware für
en betroffen,

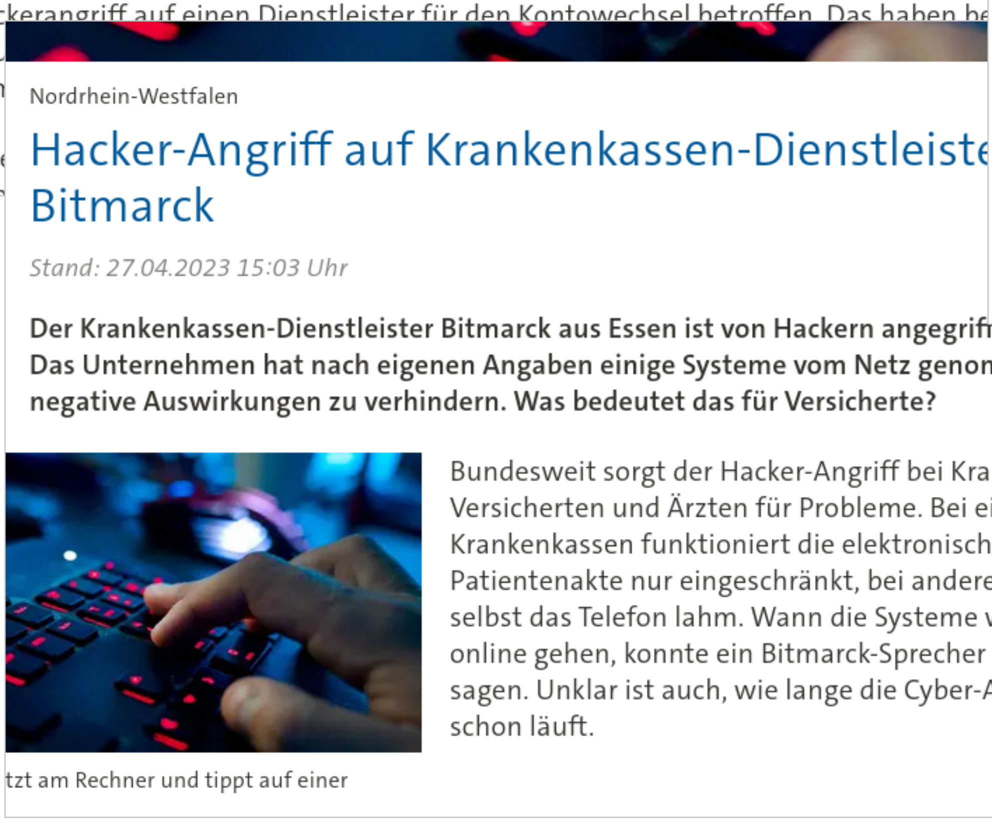
die zum
tur für Arbeit



Unerwünschte Einblicke: Fataler Fehler bei Netatmo-Sicherheitskameras

Ein Leser hat uns eine Smart-Home-Kamera geschickt, die es so nicht geben darf: Sie erlaubt nämlich Einblicke in den Haushalt einer fremden Familie.

Lesezeit: **8 Min.** In Pocket speichern 325



Nordrhein-Westfalen

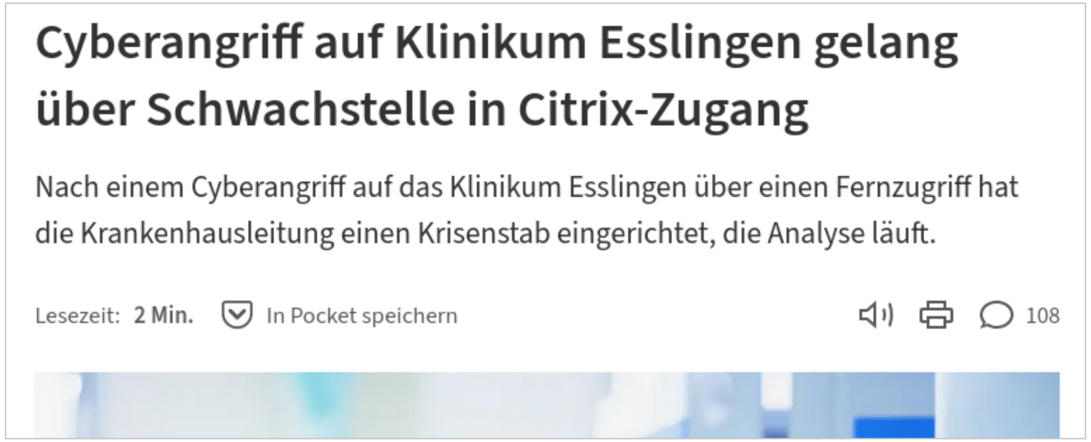
Hacker-Angriff auf Krankenkassen-Dienstleiste Bitmarck

Stand: 27.04.2023 15:03 Uhr

Der Krankenkassen-Dienstleister Bitmarck aus Essen ist von Hackern angegriffen worden. Das Unternehmen hat nach eigenen Angaben einige Systeme vom Netz genommen, um negative Auswirkungen zu verhindern. Was bedeutet das für Versicherte?

Bundesweit sorgt der Hacker-Angriff bei Krankenkassen, Versicherten und Ärzten für Probleme. Bei einigen Krankenkassen funktioniert die elektronische Patientenakte nur eingeschränkt, bei anderen liegt selbst das Telefon lahm. Wann die Systeme wieder online gehen, konnte ein Bitmarck-Sprecher noch nicht sagen. Unklar ist auch, wie lange die Cyber-Attacke schon läuft.

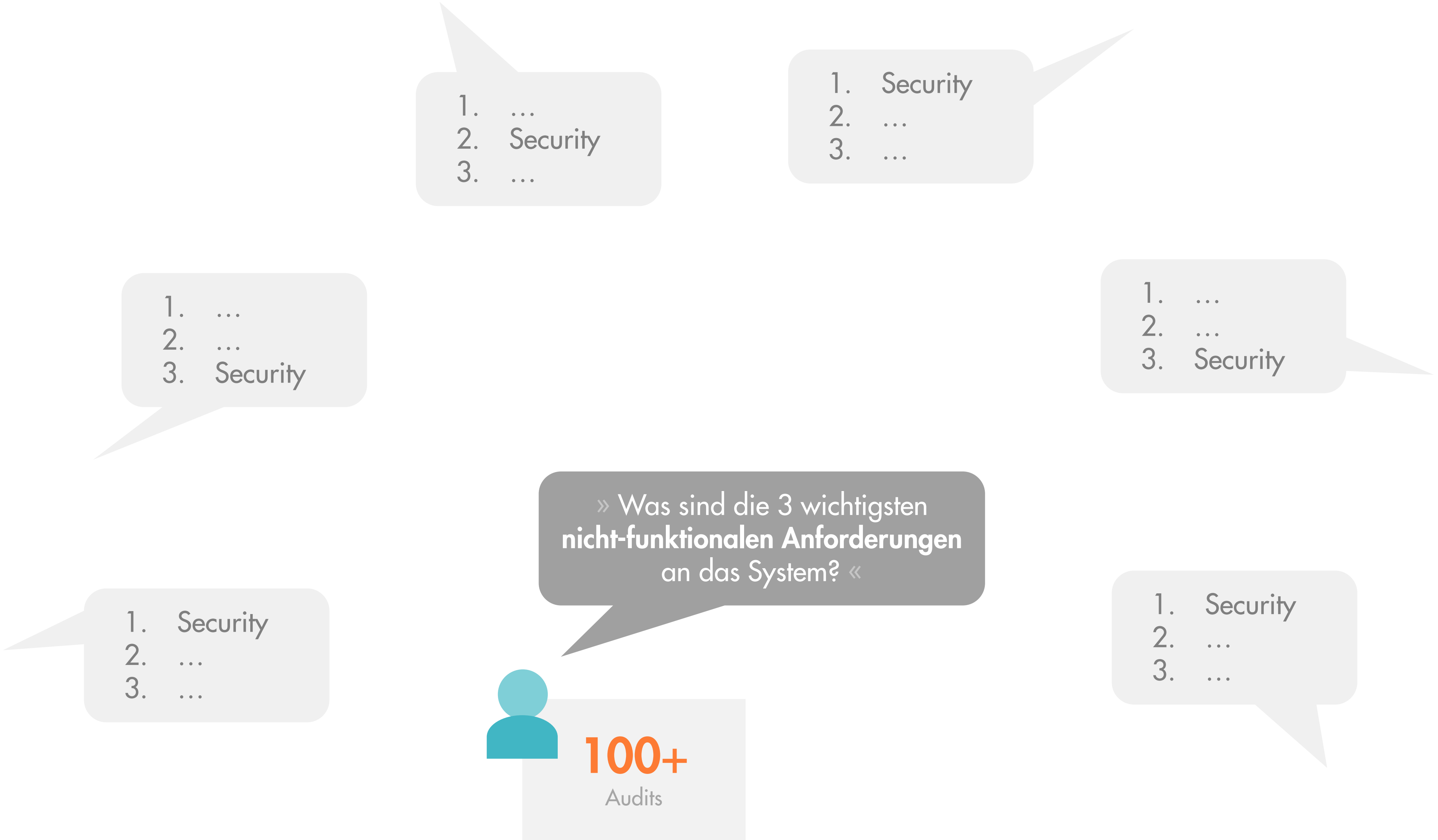
tzt am Rechner und tippt auf einer



Cyberangriff auf Klinikum Esslingen gelang über Schwachstelle in Citrix-Zugang

Nach einem Cyberangriff auf das Klinikum Esslingen über einen Fernzugriff hat die Krankenhausleitung einen Krisenstab eingerichtet, die Analyse läuft.

Lesezeit: **2 Min.** In Pocket speichern 108



» Was sind die 3 wichtigsten **nicht-funktionalen Anforderungen** an das System? «

1. …
2. Security
3. …

1. Security
2. …
3. …

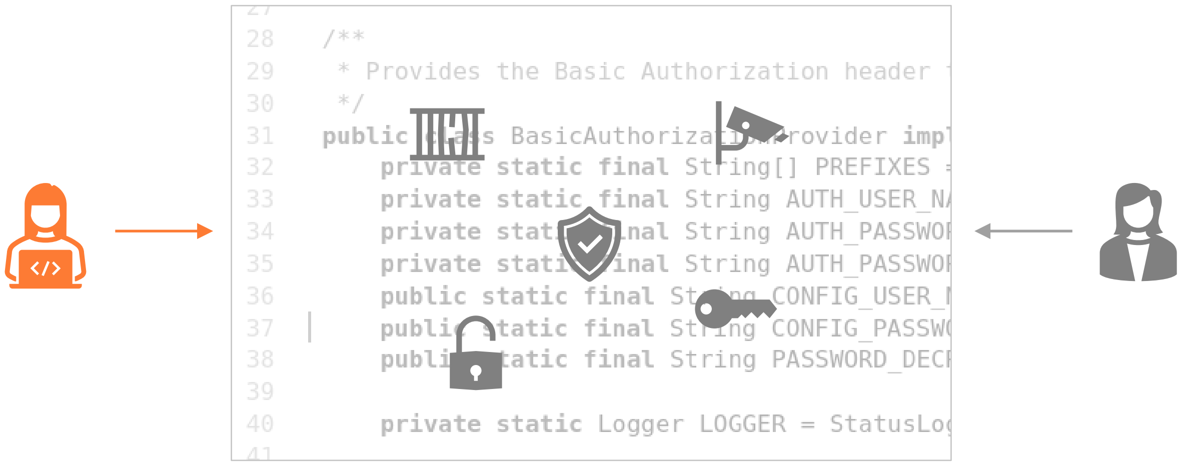
1. …
2. …
3. Security

1. …
2. …
3. Security

1. Security
2. …
3. …

1. Security
2. …
3. …

100+
Audits



```
/**
 * Provides the Basic Authorization header
 */
public class BasicAuthorizationProvider impl
    private static final String[] PREFIXES =
    private static final String AUTH_USER_N
    private static final String AUTH_PASSWOR
    private static final String AUTH_PASSWOR
    public static final String CONFIG_USER_
    public static final String CONFIG_PASSW
    public static final String PASSWORD_DEC

    private static Logger LOGGER = StatusLo
```

Fortify

Teamscale

C++

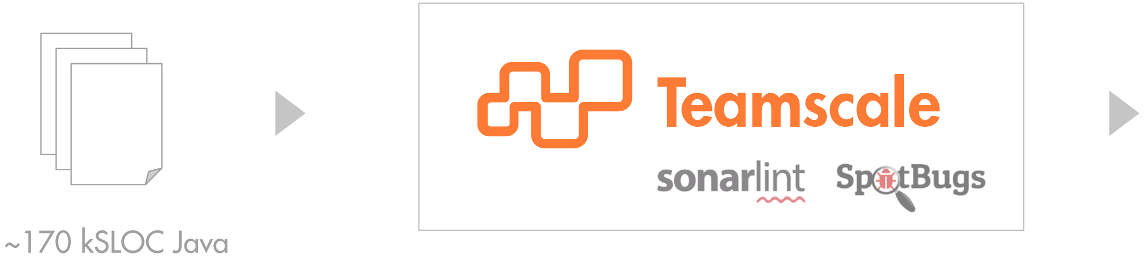
. . .

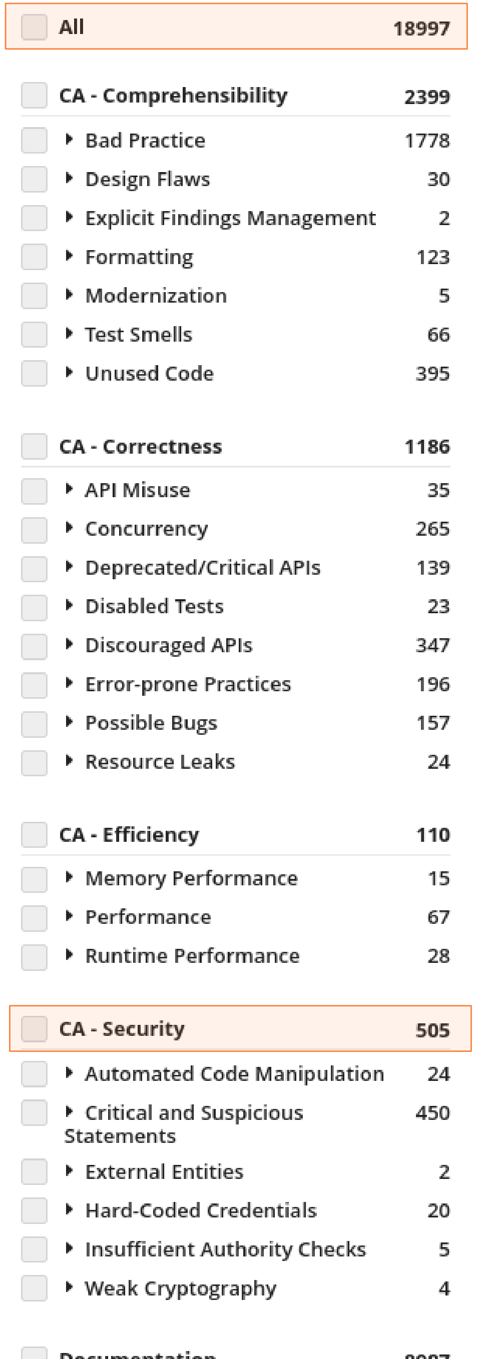
klocwork

sonarlint

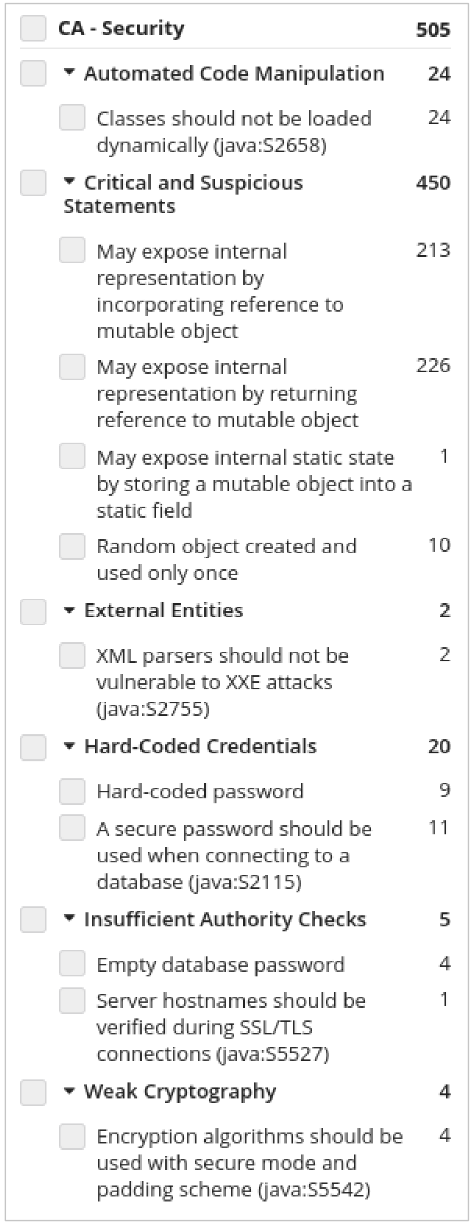
BLACKDUCK

SpotBugs

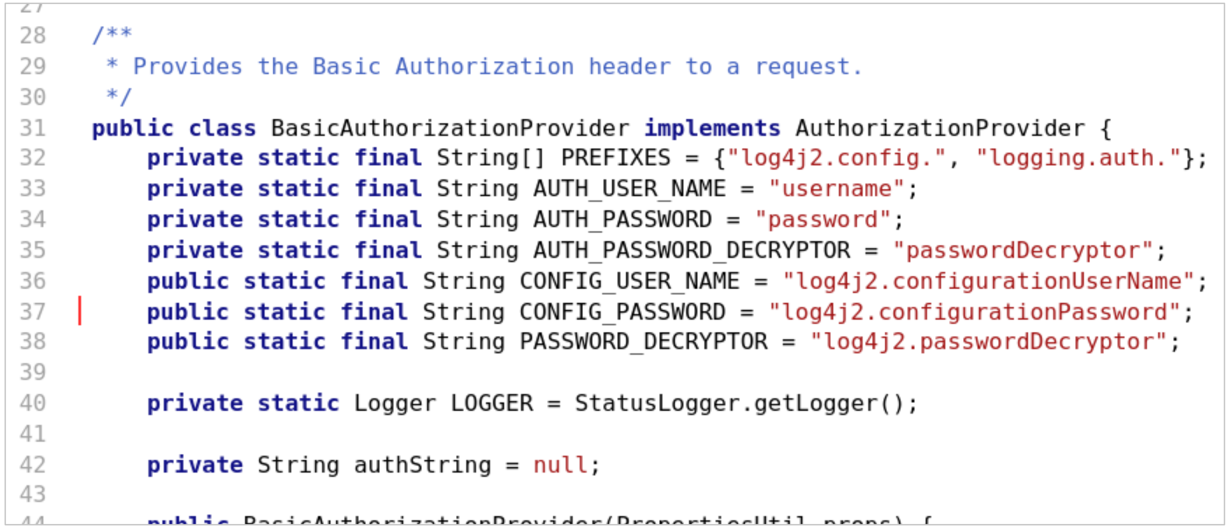




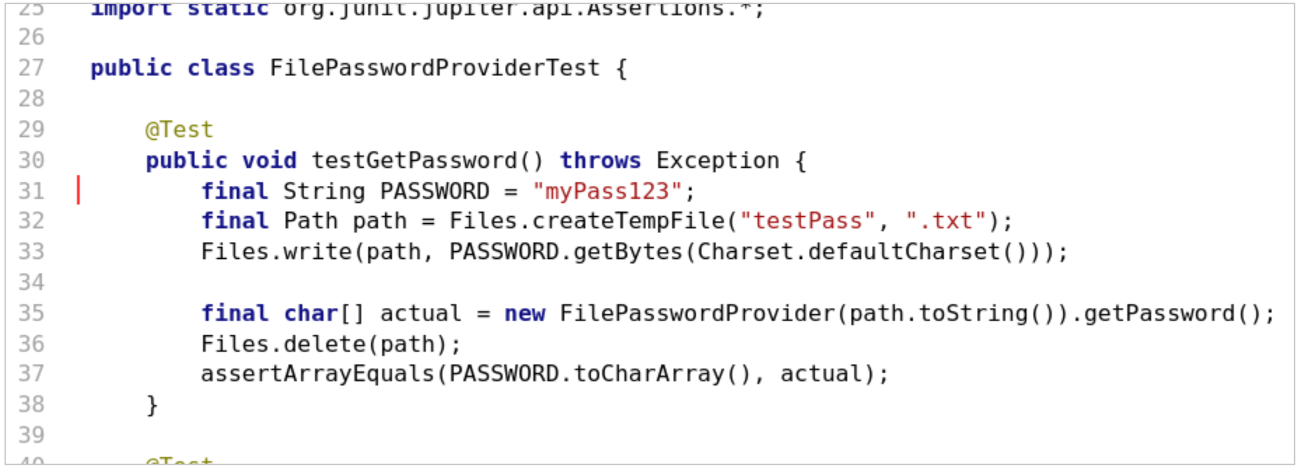
□ All	18997
□ CA - Comprehensibility	2399
□ ▸ Bad Practice	1778
□ ▸ Design Flaws	30
□ ▸ Explicit Findings Management	2
□ ▸ Formatting	123
□ ▸ Modernization	5
□ ▸ Test Smells	66
□ ▸ Unused Code	395
□ CA - Correctness	1186
□ ▸ API Misuse	35
□ ▸ Concurrency	265
□ ▸ Deprecated/Critical APIs	139
□ ▸ Disabled Tests	23
□ ▸ Discouraged APIs	347
□ ▸ Error-prone Practices	196
□ ▸ Possible Bugs	157
□ ▸ Resource Leaks	24
□ CA - Efficiency	110
□ ▸ Memory Performance	15
□ ▸ Performance	67
□ ▸ Runtime Performance	28
□ CA - Security	505
□ ▸ Automated Code Manipulation	24
□ ▸ Critical and Suspicious Statements	450
□ ▸ External Entities	2
□ ▸ Hard-Coded Credentials	20
□ ▸ Insufficient Authority Checks	5
□ ▸ Weak Cryptography	4
□ Documentation	8087



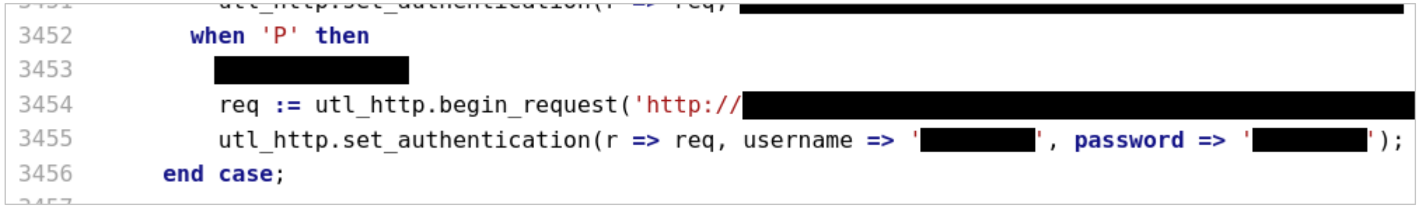
□ CA - Security	505
□ ▾ Automated Code Manipulation	24
□ Classes should not be loaded dynamically (java:S2658)	24
□ ▾ Critical and Suspicious Statements	450
□ May expose internal representation by incorporating reference to mutable object	213
□ May expose internal representation by returning reference to mutable object	226
□ May expose internal static state by storing a mutable object into a static field	1
□ Random object created and used only once	10
□ ▾ External Entities	2
□ XML parsers should not be vulnerable to XXE attacks (java:S2755)	2
□ ▾ Hard-Coded Credentials	20
□ Hard-coded password	9
□ A secure password should be used when connecting to a database (java:S2115)	11
□ ▾ Insufficient Authority Checks	5
□ Empty database password	4
□ Server hostnames should be verified during SSL/TLS connections (java:S5527)	1
□ ▾ Weak Cryptography	4
□ Encryption algorithms should be used with secure mode and padding scheme (java:S5542)	4



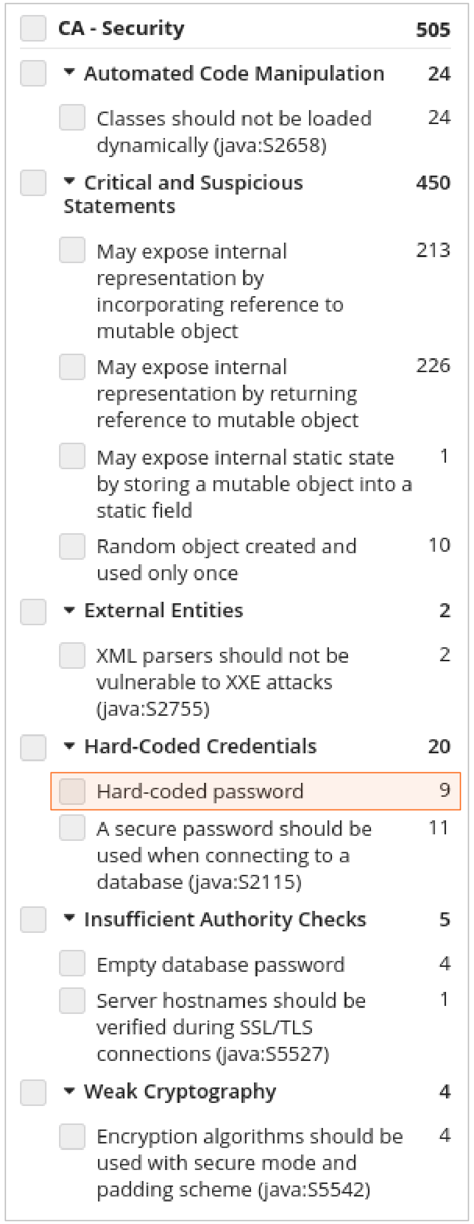
```java
/**
 * Provides the Basic Authorization header to a request.
 */
public class BasicAuthorizationProvider implements AuthorizationProvider {
    private static final String[] PREFIXES = {"log4j2.config.", "logging.auth."};
    private static final String AUTH_USER_NAME = "username";
    private static final String AUTH_PASSWORD = "password";
    private static final String AUTH_PASSWORD_DECRYPTOR = "passwordDecryptor";
    public static final String CONFIG_USER_NAME = "log4j2.configurationUserName";
    public static final String CONFIG_PASSWORD = "log4j2.configurationPassword";
    public static final String PASSWORD_DECRYPTOR = "log4j2.passwordDecryptor";

    private static Logger LOGGER = StatusLogger.getLogger();

    private String authString = null;

    public BasicAuthorizationProvider(PropertiesUtil props) {
```



```java
import static org.junit.jupiter.api.Assertions.*;

public class FilePasswordProviderTest {

    @Test
    public void testGetPassword() throws Exception {
        final String PASSWORD = "myPass123";
        final Path path = Files.createTempFile("testPass", ".txt");
        Files.write(path, PASSWORD.getBytes(Charset.defaultCharset()));

        final char[] actual = new FilePasswordProvider(path.toString()).getPassword();
        Files.delete(path);
        assertArrayEquals(PASSWORD.toCharArray(), actual);
    }

    @Test
```



```sql
        utl_http.set_authentication(r => req, [redacted]
      when 'P' then
        [redacted]
        req := utl_http.begin_request('http://[redacted]
        utl_http.set_authentication(r => req, username => '[redacted]', password => '[redacted]');
    end case;
```



□ CA - Security 505

□ ▾ Automated Code Manipulation 24

□ Classes should not be loaded dynamically (java:S2658) 24

□ ▾ Critical and Suspicious Statements 450

□ May expose internal representation by incorporating reference to mutable object 213

□ May expose internal representation by returning reference to mutable object 226

□ May expose internal static state by storing a mutable object into a static field 1

□ Random object created and used only once 10

□ ▾ External Entities 2

□ XML parsers should not be vulnerable to XXE attacks (java:S2755) 2

□ ▾ Hard-Coded Credentials 20

□ Hard-coded password 9

□ A secure password should be used when connecting to a database (java:S2115) 11

□ ▾ Insufficient Authority Checks 5

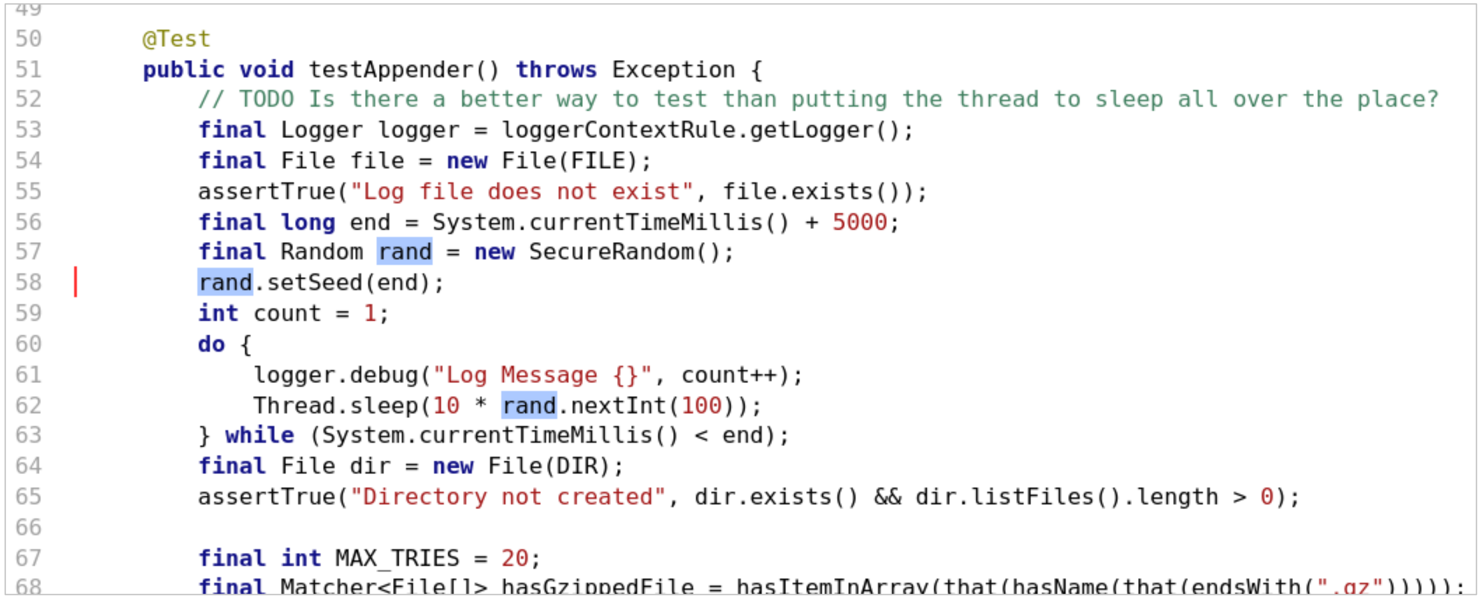
□ Empty database password 4

□ Server hostnames should be verified during SSL/TLS connections (java:S5527) 1

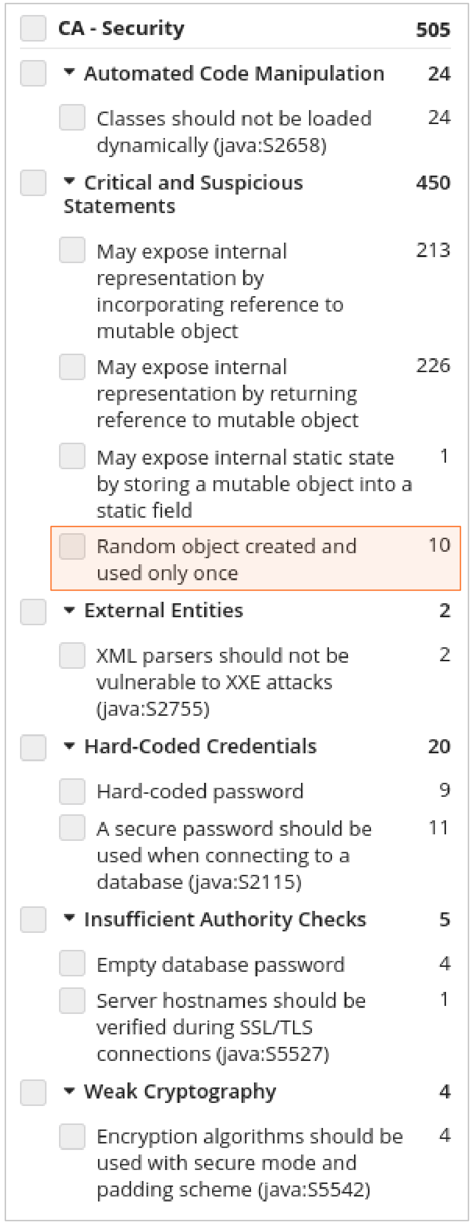
□ ▾ Weak Cryptography 4

□ Encryption algorithms should be used with secure mode and padding scheme (java:S5542) 4

SpotBugs

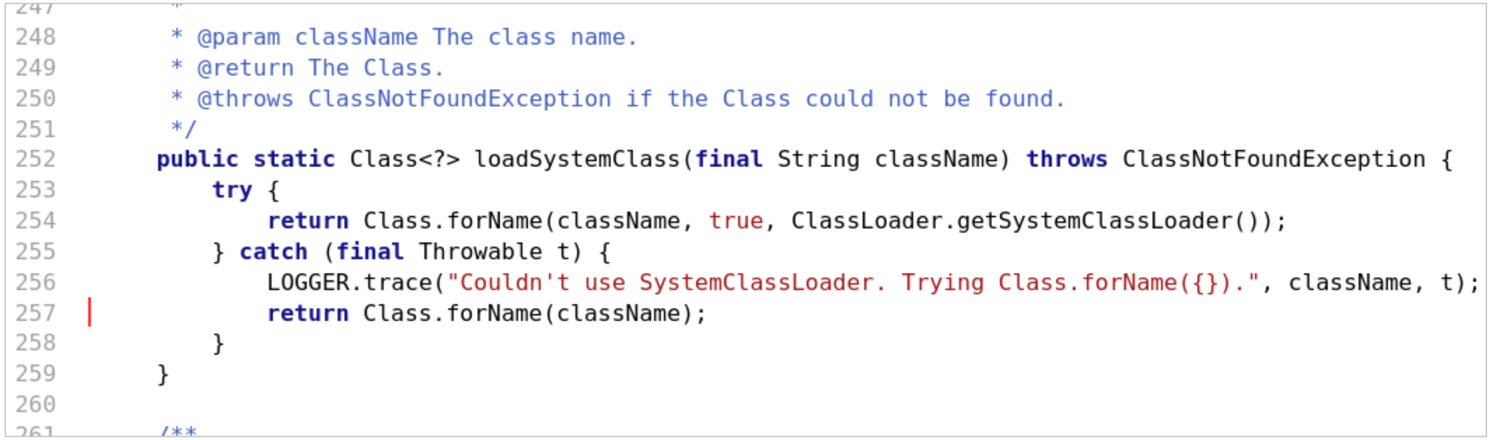


```java

    @Test
    public void testAppender() throws Exception {
        // TODO Is there a better way to test than putting the thread to sleep all over the place?
        final Logger logger = loggerContextRule.getLogger();
        final File file = new File(FILE);
        assertTrue("Log file does not exist", file.exists());
        final long end = System.currentTimeMillis() + 5000;
        final Random rand = new SecureRandom();
        rand.setSeed(end);
        int count = 1;
        do {
            logger.debug("Log Message {}", count++);
            Thread.sleep(10 * rand.nextInt(100));
        } while (System.currentTimeMillis() < end);
        final File dir = new File(DIR);
        assertTrue("Directory not created", dir.exists() && dir.listFiles().length > 0);

        final int MAX_TRIES = 20;
        final Matcher<File[]> hasGzippedFile = hasItemInArray(that(hasName(that(endsWith(".gz")))));
```

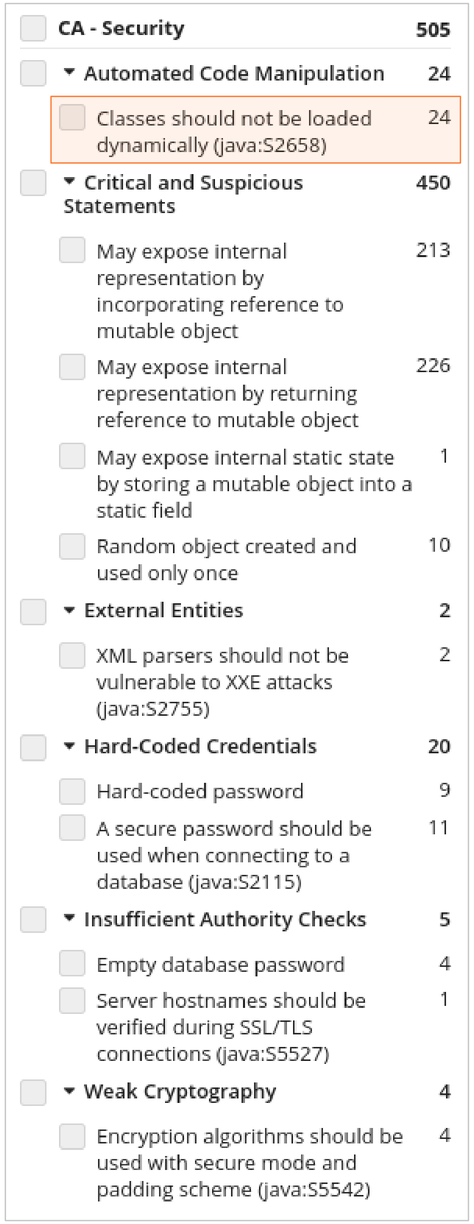


□ **CA - Security**	**505**
□ ▾ Automated Code Manipulation	24
□ Classes should not be loaded dynamically (java:S2658)	24
□ ▾ Critical and Suspicious Statements	450
□ May expose internal representation by incorporating reference to mutable object	213
□ May expose internal representation by returning reference to mutable object	226
□ May expose internal static state by storing a mutable object into a static field	1
□ Random object created and used only once	10
□ ▾ External Entities	2
□ XML parsers should not be vulnerable to XXE attacks (java:S2755)	2
□ ▾ Hard-Coded Credentials	20
□ Hard-coded password	9
□ A secure password should be used when connecting to a database (java:S2115)	11
□ ▾ Insufficient Authority Checks	5
□ Empty database password	4
□ Server hostnames should be verified during SSL/TLS connections (java:S5527)	1
□ ▾ Weak Cryptography	4
□ Encryption algorithms should be used with secure mode and padding scheme (java:S5542)	4

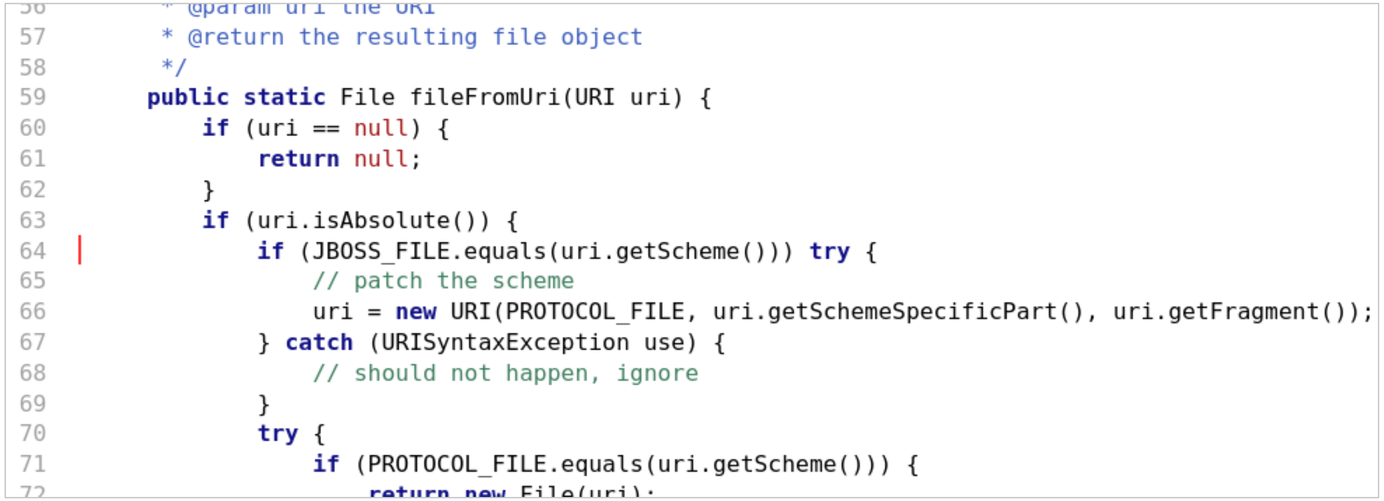
sonarlint



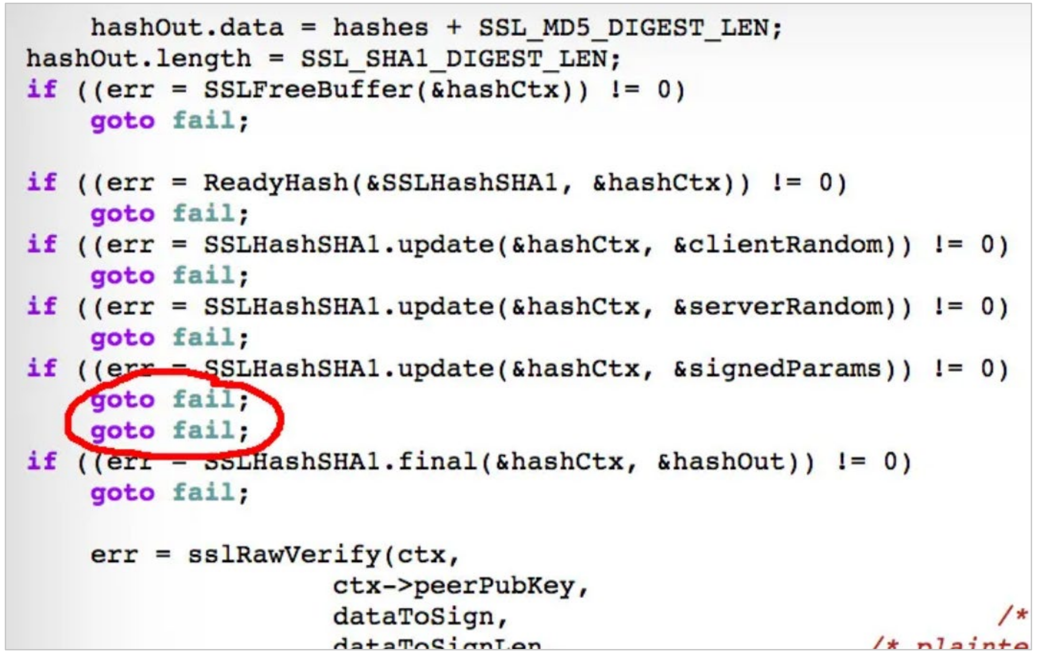
```
     *
     * @param className The class name.
     * @return The Class.
     * @throws ClassNotFoundException if the Class could not be found.
     */
    public static Class<?> loadSystemClass(final String className) throws ClassNotFoundException {
        try {
            return Class.forName(className, true, ClassLoader.getSystemClassLoader());
        } catch (final Throwable t) {
            LOGGER.trace("Couldn't use SystemClassLoader. Trying Class.forName({}).", className, t);
            return Class.forName(className);
        }
    }

    /**
```



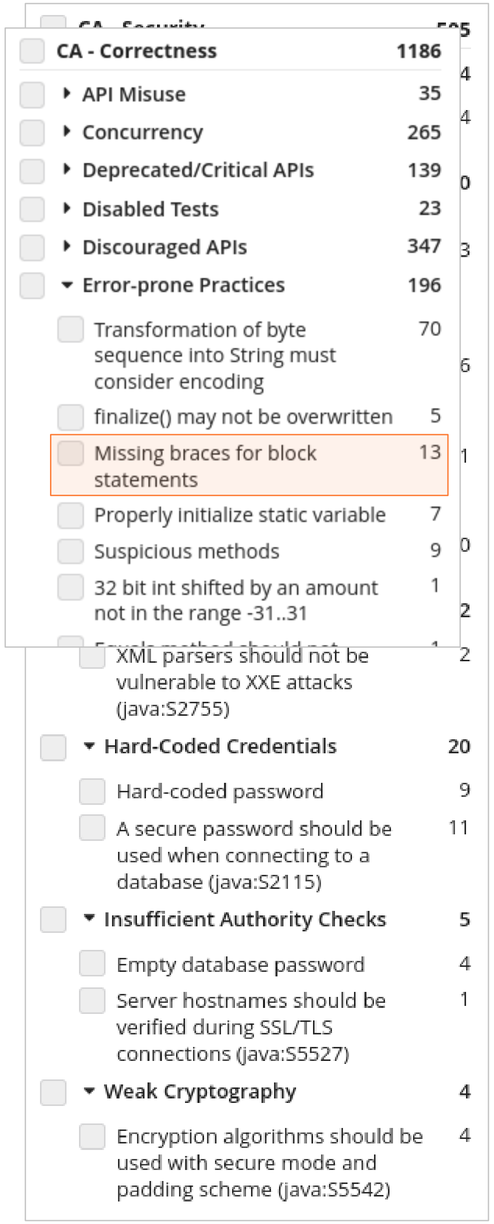
☐ **CA - Security**	**505**
☐ ▾ **Automated Code Manipulation**	**24**
☐ Classes should not be loaded dynamically (java:S2658)	24
☐ ▾ **Critical and Suspicious Statements**	**450**
☐ May expose internal representation by incorporating reference to mutable object	213
☐ May expose internal representation by returning reference to mutable object	226
☐ May expose internal static state by storing a mutable object into a static field	1
☐ Random object created and used only once	10
☐ ▾ **External Entities**	**2**
☐ XML parsers should not be vulnerable to XXE attacks (java:S2755)	2
☐ ▾ **Hard-Coded Credentials**	**20**
☐ Hard-coded password	9
☐ A secure password should be used when connecting to a database (java:S2115)	11
☐ ▾ **Insufficient Authority Checks**	**5**
☐ Empty database password	4
☐ Server hostnames should be verified during SSL/TLS connections (java:S5527)	1
☐ ▾ **Weak Cryptography**	**4**
☐ Encryption algorithms should be used with secure mode and padding scheme (java:S5542)	4



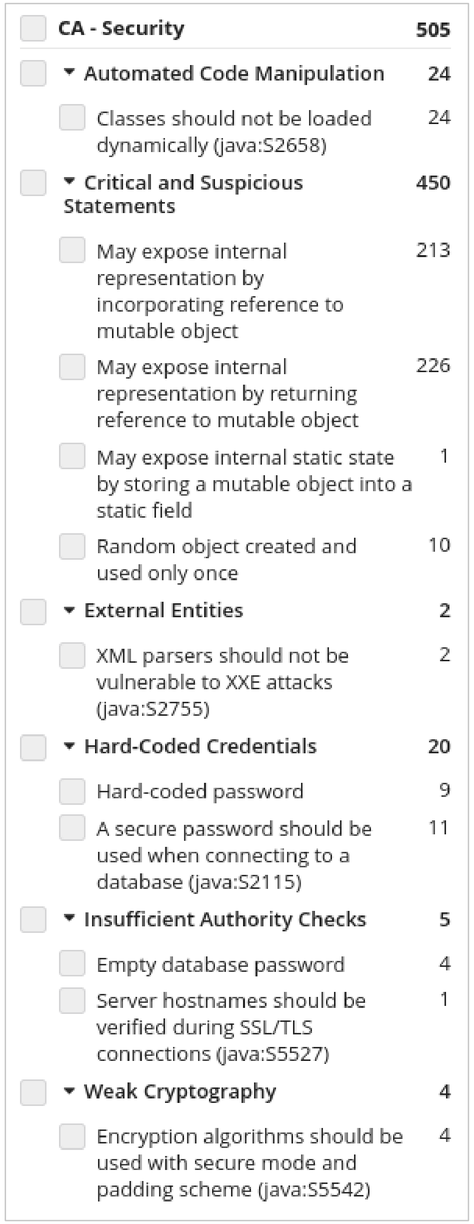
```
     * @param uri the URI
     * @return the resulting file object
     */
    public static File fileFromUri(URI uri) {
        if (uri == null) {
            return null;
        }
        if (uri.isAbsolute()) {
            if (JBOSS_FILE.equals(uri.getScheme())) try {
                // patch the scheme
                uri = new URI(PROTOCOL_FILE, uri.getSchemeSpecificPart(), uri.getFragment());
            } catch (URISyntaxException use) {
                // should not happen, ignore
            }
            try {
                if (PROTOCOL_FILE.equals(uri.getScheme())) {
                    return new File(uri);
```



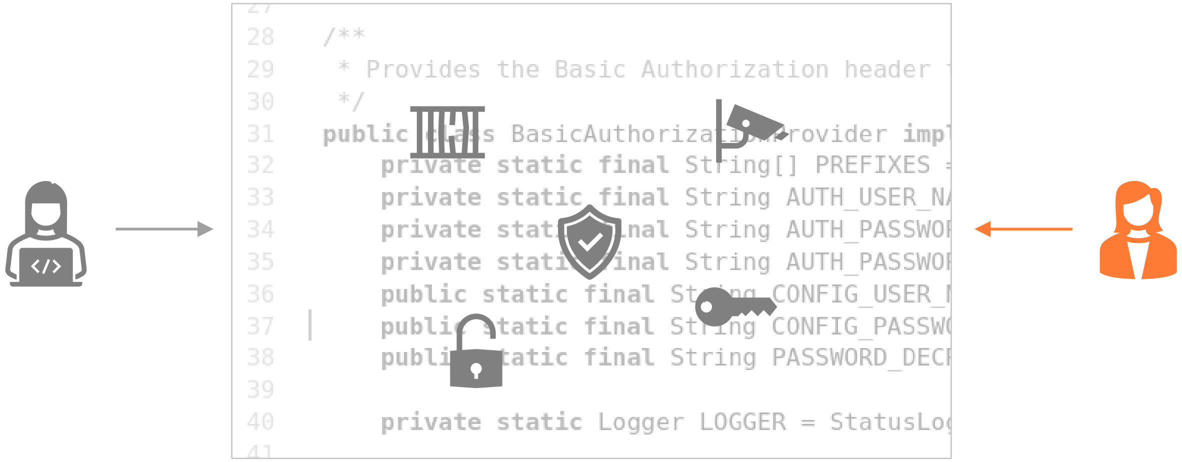
```
    hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
hashOut.length = SSL_SHA1_DIGEST_LEN;
if ((err = SSLFreeBuffer(&hashCtx)) != 0)
    goto fail;

if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
    goto fail;
if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
    goto fail;
    goto fail;
if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
    goto fail;

    err = sslRawVerify(ctx,
                       ctx->peerPubKey,
                       dataToSign,                              /*
                       dataToSignLen,                    /* plainte
```



□ CA - Security	505
□ CA - Correctness	1186
□ ▸ API Misuse	35
□ ▸ Concurrency	265
□ ▸ Deprecated/Critical APIs	139
□ ▸ Disabled Tests	23
□ ▸ Discouraged APIs	347
□ ▾ Error-prone Practices	196
□ Transformation of byte sequence into String must consider encoding	70
□ finalize() may not be overwritten	5
□ Missing braces for block statements	13
□ Properly initialize static variable	7
□ Suspicious methods	9
□ 32 bit int shifted by an amount not in the range -31..31	1
□ XML parsers should not be vulnerable to XXE attacks (java:S2755)	2
□ ▾ Hard-Coded Credentials	20
□ Hard-coded password	9
□ A secure password should be used when connecting to a database (java:S2115)	11
□ ▾ Insufficient Authority Checks	5
□ Empty database password	4
□ Server hostnames should be verified during SSL/TLS connections (java:S5527)	1
□ ▾ Weak Cryptography	4
□ Encryption algorithms should be used with secure mode and padding scheme (java:S5542)	4



□ **CA - Security**	**505**
□ ▾ **Automated Code Manipulation**	**24**
□ Classes should not be loaded dynamically (java:S2658)	24
□ ▾ **Critical and Suspicious Statements**	**450**
□ May expose internal representation by incorporating reference to mutable object	213
□ May expose internal representation by returning reference to mutable object	226
□ May expose internal static state by storing a mutable object into a static field	1
□ Random object created and used only once	10
□ ▾ **External Entities**	**2**
□ XML parsers should not be vulnerable to XXE attacks (java:S2755)	2
□ ▾ **Hard-Coded Credentials**	**20**
□ Hard-coded password	9
□ A secure password should be used when connecting to a database (java:S2115)	11
□ ▾ **Insufficient Authority Checks**	**5**
□ Empty database password	4
□ Server hostnames should be verified during SSL/TLS connections (java:S5527)	1
□ ▾ **Weak Cryptography**	**4**
□ Encryption algorithms should be used with secure mode and padding scheme (java:S5542)	4



```
/**
 * Provides the Basic Authorization header f
 */
public class BasicAuthorizationProvider impl
    private static final String[] PREFIXES =
    private static final String AUTH_USER_NA
    private static final String AUTH_PASSWOR
    private static final String AUTH_PASSWOR
    public static final String CONFIG_USER_N
    public static final String CONFIG_PASSWO
    public static final String PASSWORD_DECR

    private static Logger LOGGER = StatusLog
```

AUTOSAR

BaFin
Bundesanstalt für
Finanzdienstleistungsaufsicht

OWASP
Open Web Application
Security Project

Bundesamt
für Sicherheit in der
Informationstechnik

SANS CWE TOP 25

. . .

MISRA

Bundesamt für Informatik und
Telekommunikation BIT

https://tmscl.me/a2310-oop

Wie zukunftssicher ist Ihr Softwaresystem?

Unser Vorgehen und Erfahrungen aus 10 Jahren Software-Audits

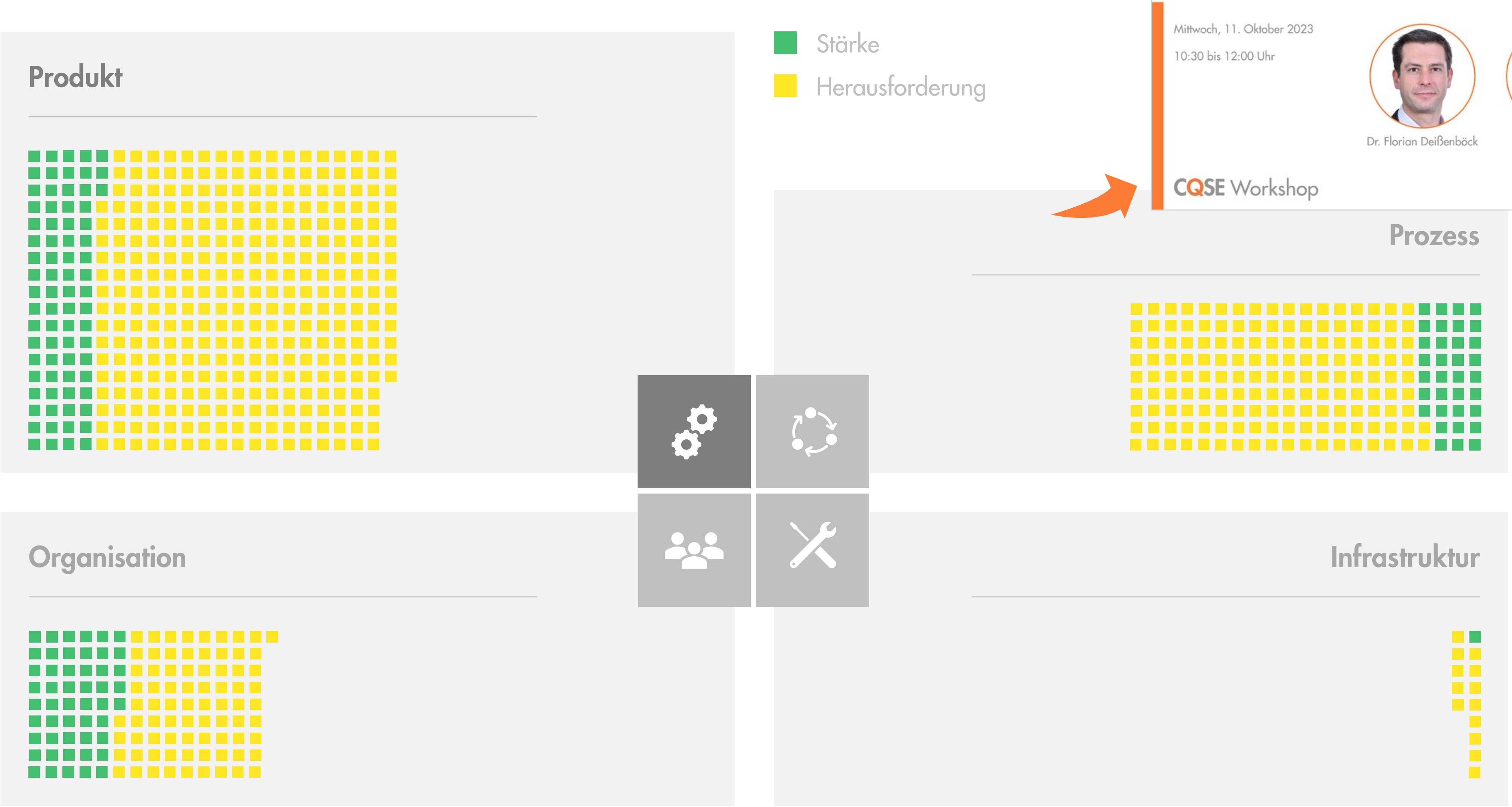
Mittwoch, 11. Oktober 2023

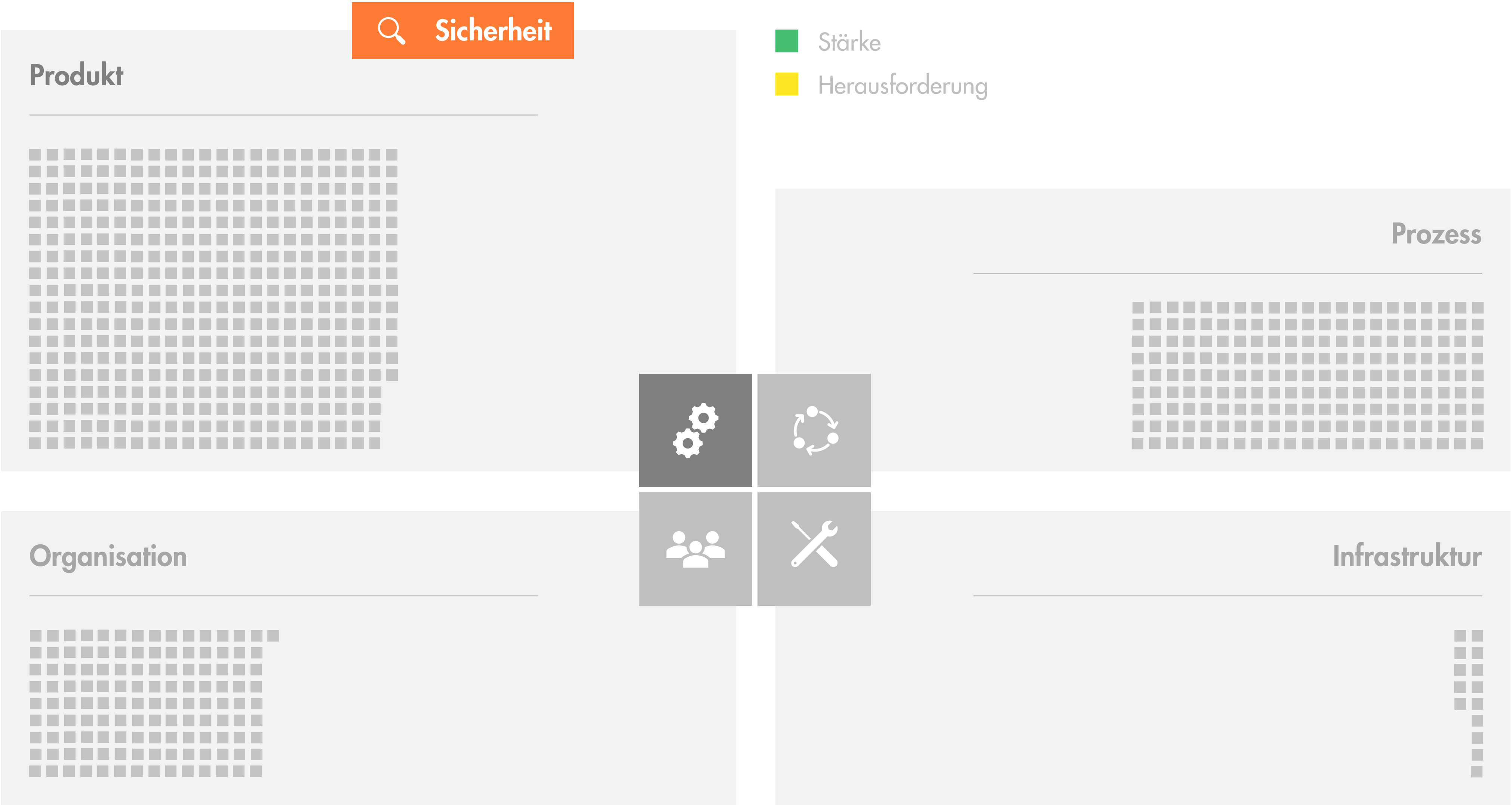
10:30 bis 12:00 Uhr

Dr. Florian Deißenböck

Dr. Nils Göde

CQSE Workshop





Sicherheit

🟩 Stärke

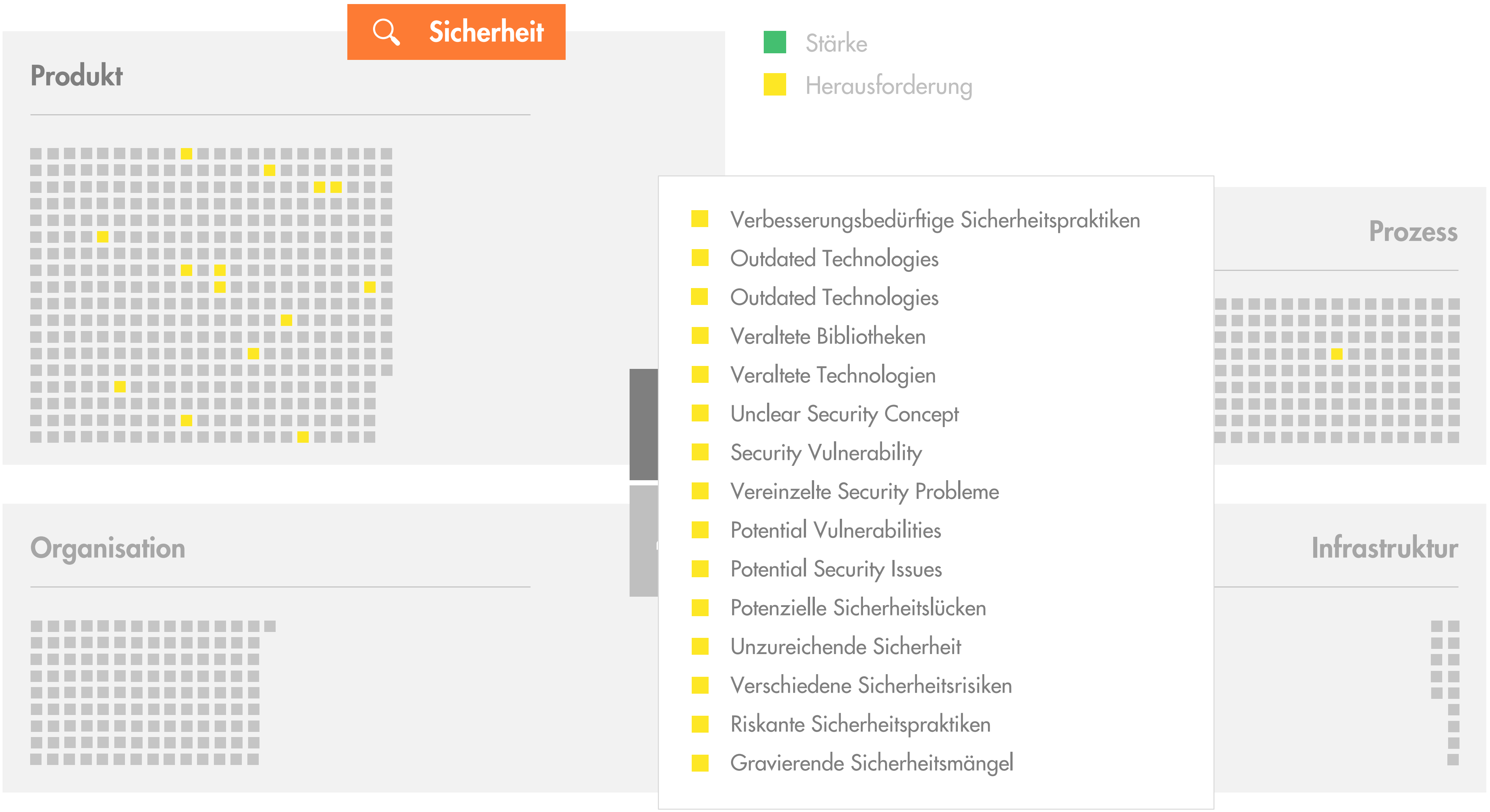
🟨 Herausforderung

Produkt

Prozess

Organisation

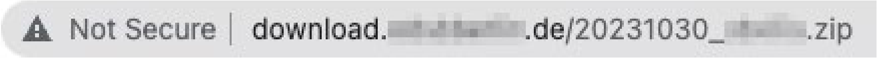
Infrastruktur



🟩 Stärke

🟨 Herausforderung

- 🟨 Verbesserungsbedürftige Sicherheitspraktiken
- 🟨 Outdated Technologies
- 🟨 Outdated Technologies
- 🟨 Veraltete Bibliotheken
- 🟨 Veraltete Technologien
- 🟨 Unclear Security Concept
- 🟨 Security Vulnerability
- 🟨 Vereinzelte Security Probleme
- 🟨 Potential Vulnerabilities
- 🟨 Potential Security Issues
- 🟨 Potenzielle Sicherheitslücken
- 🟨 Unzureichende Sicherheit
- 🟨 Verschiedene Sicherheitsrisiken
- 🟨 Riskante Sicherheitspraktiken
- 🟨 Gravierende Sicherheitsmängel



Not Secure | download.[redacted].de/20231030_[redacted].zip

🟥 Gravierende Sicherheitsmängel

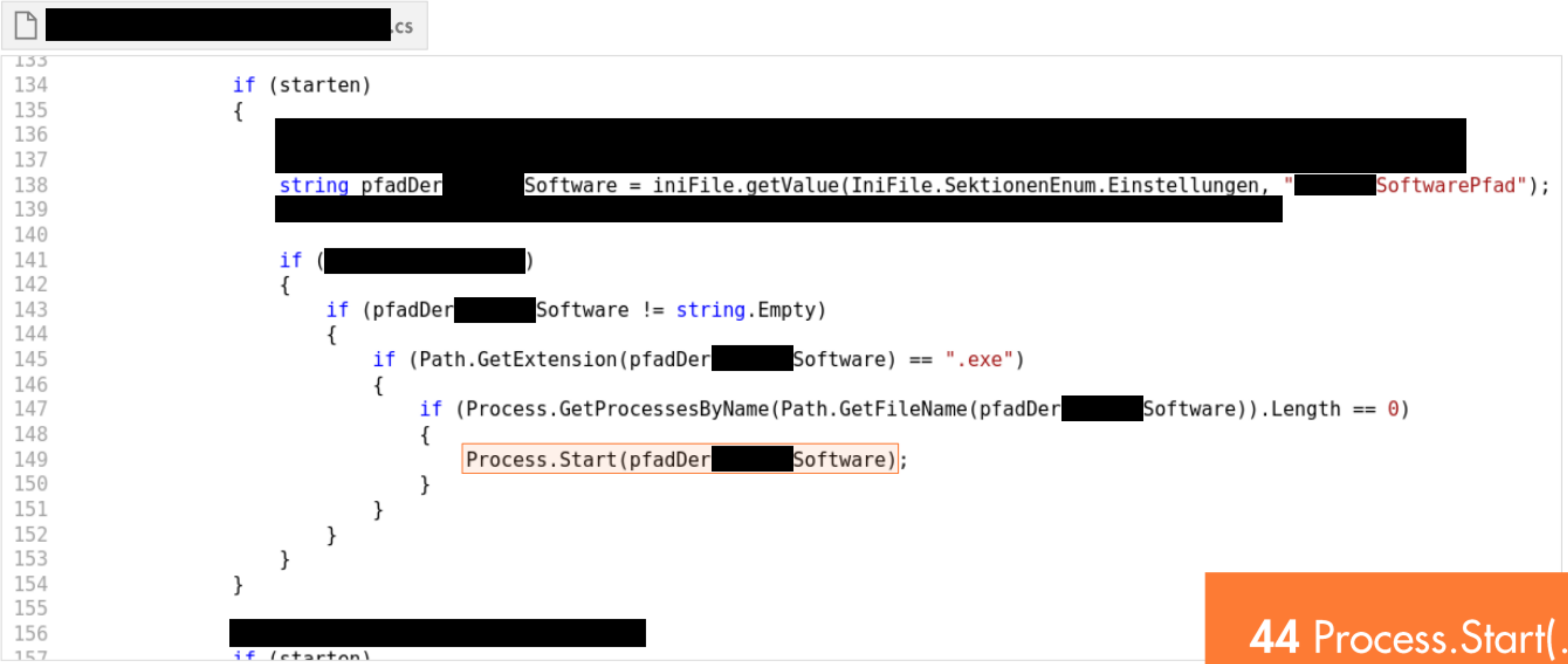
Open Web Application Security Project

- 🟩 Durchführung **Penetrationtest**
- 🟨 Hart-kodierte **Passwörter**
- 🟨 Fehlende Statische Analyse für **Security-Probleme**
- 🟨 **Inkorrekte Nutzung der** Java Kryptographie Architektur (JCA)
- 🟥 Ungesicherte **Prozessausführung**
- 🟥 Hohe Anzahl an **Sicherheits-relevanten Findings**

🔒 Sicherheit

305	Zugriff auf interne Repräsentation
13	Mögliche XML-Parser XXE Angriffe
7	Nicht gesicherter XML-Transformer
5	Veränderbares Objekt in statischen Feld
4	Server-Zertifikat sollte verifiziert werden
4	Passwörter im Quelltext
2	Nutzung von SSL als Protokoll
2	HostnameVerifier liefert immer **true**
...	...
349	

Riskante Sicherheitspraktiken

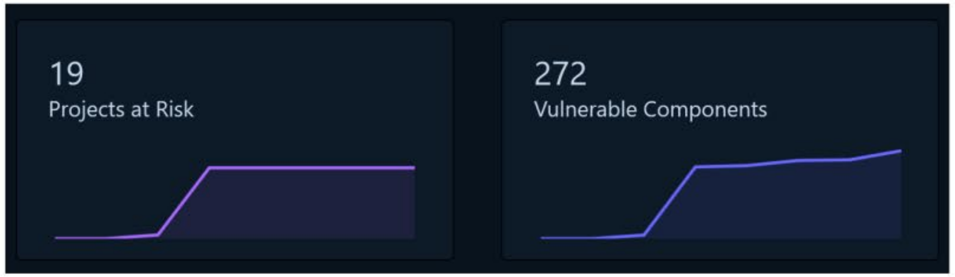


[redacted].cs

```
            if (starten)
            {
                [redacted]
                string pfadDer[redacted]Software = iniFile.getValue(IniFile.SektionenEnum.Einstellungen, "[redacted]SoftwarePfad");
                [redacted]

                if ([redacted])
                {
                    if (pfadDer[redacted]Software != string.Empty)
                    {
                        if (Path.GetExtension(pfadDer[redacted]Software) == ".exe")
                        {
                            if (Process.GetProcessesByName(Path.GetFileName(pfadDer[redacted]Software)).Length == 0)
                            {
                                Process.Start(pfadDer[redacted]Software);
                            }
                        }
                    }
                }
            }

            [redacted]
            if (starten)
```

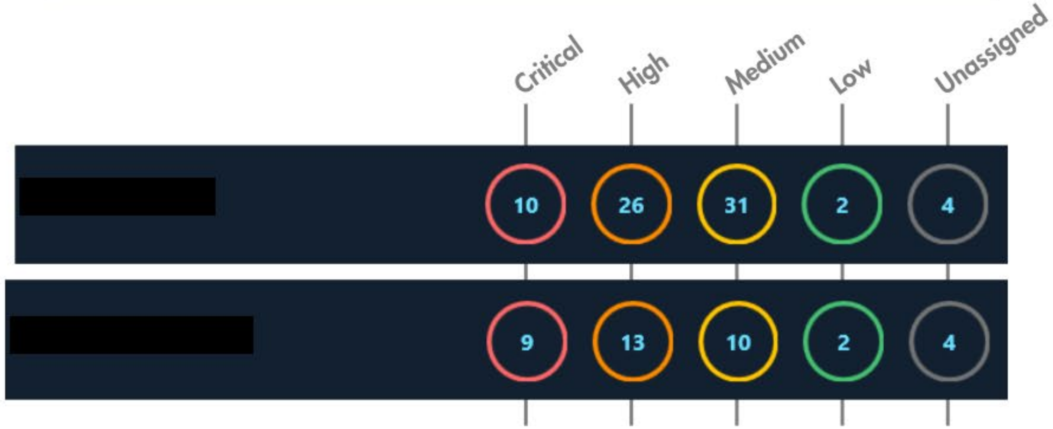
44 Process.Start(…)

Technology Assessment – Vulnerabilities in Dependencies



19
Projects at Risk

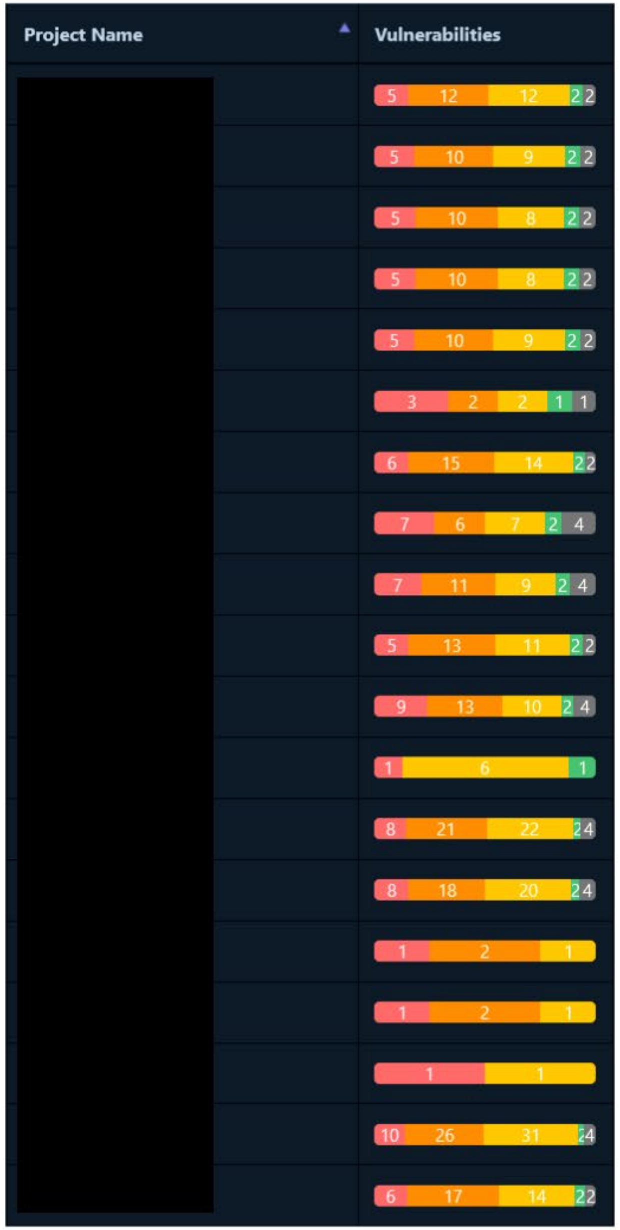
272
Vulnerable Components



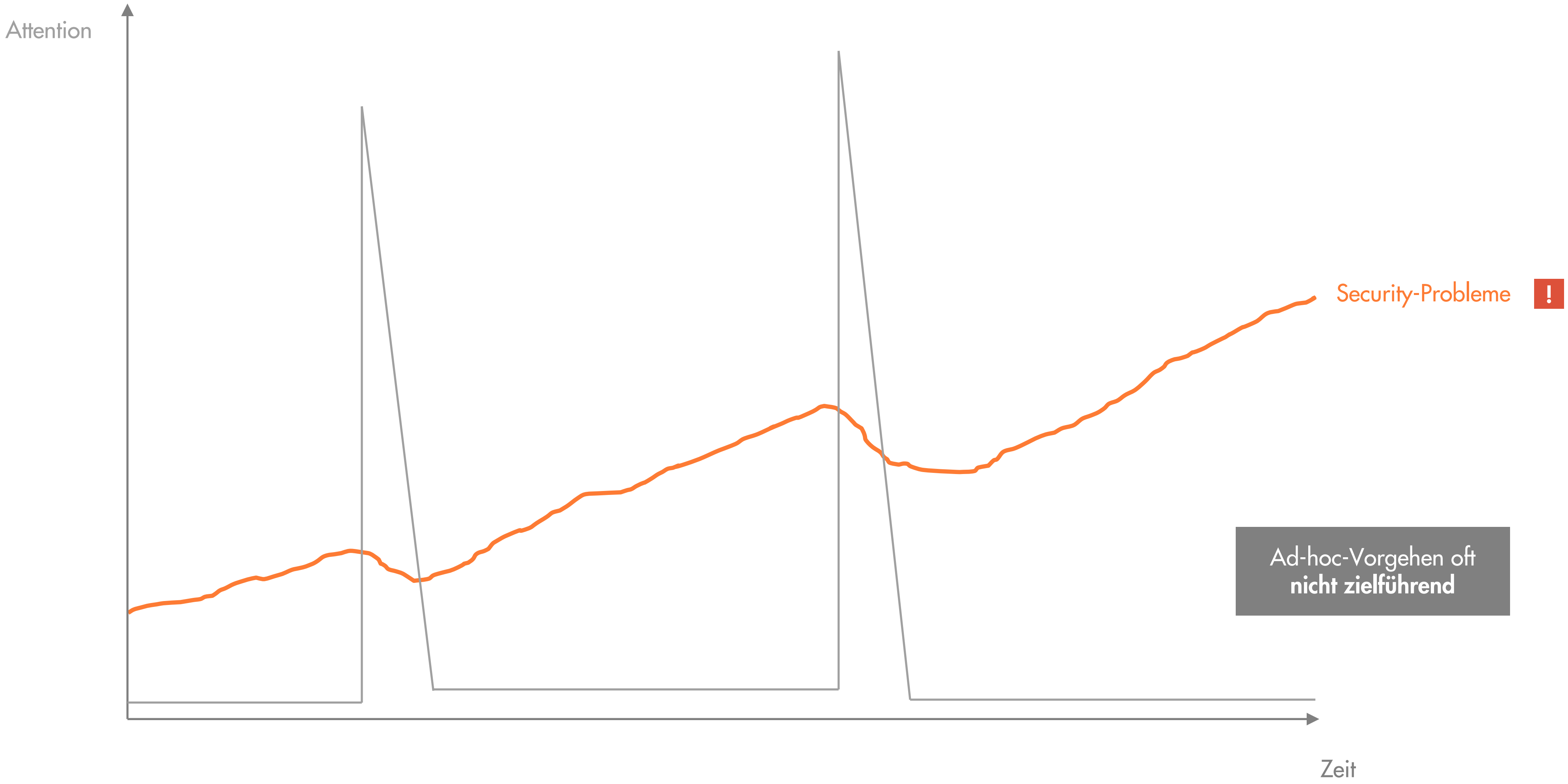
	Critical	High	Medium	Low	Unassigned
[redacted]	10	26	31	2	4
[redacted]	9	13	10	2	4

- Dependencies of all 19 Maven projects contain known vulnerabilities
- Compile time and runtime dependencies
- Deployable WAR artifact includes 288 JAR artifacts

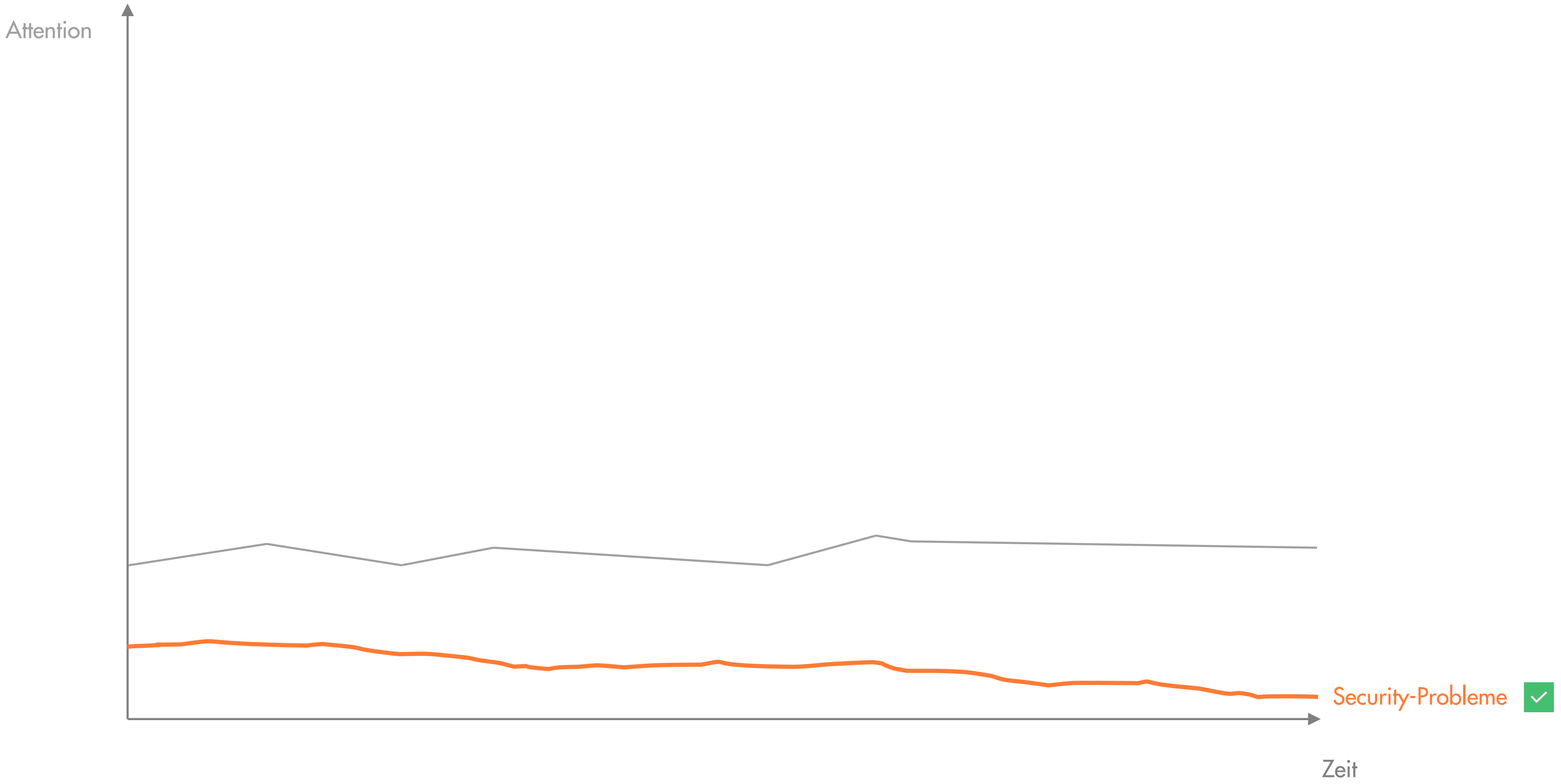
Information extracted with Dependency Track

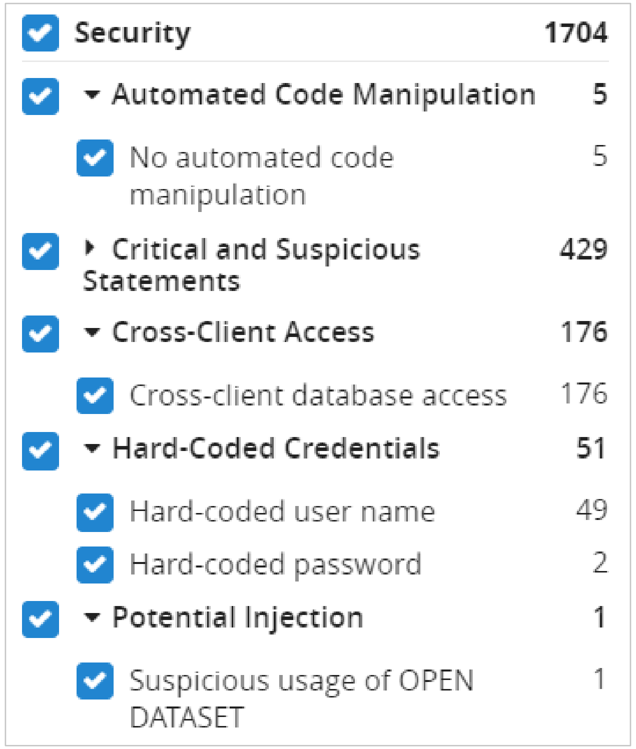


Project Name ▲	Vulnerabilities
[redacted]	5 12 12 2 2
[redacted]	5 10 9 2 2
[redacted]	5 10 8 2 2
[redacted]	5 10 8 2 2
[redacted]	5 10 9 2 2
[redacted]	3 2 2 1 1
[redacted]	6 15 14 2 2
[redacted]	7 6 7 2 4
[redacted]	7 11 9 2 4
[redacted]	5 13 11 2 2
[redacted]	9 13 10 2 4
[redacted]	1 6 1
[redacted]	8 21 22 2 4
[redacted]	8 18 20 2 4
[redacted]	1 2 1
[redacted]	1 2 1
[redacted]	1 1
[redacted]	10 26 31 2 4
[redacted]	6 17 14 2 2



Ad-hoc-Vorgehen oft **nicht zielführend**


Attention
Security-Probleme ✓
Zeit



☑ Security	1704
☑ ▾ Automated Code Manipulation	5
☑ No automated code manipulation	5
☑ ▸ Critical and Suspicious Statements	429
☑ ▾ Cross-Client Access	176
☑ Cross-client database access	176
☑ ▾ Hard-Coded Credentials	51
☑ Hard-coded user name	49
☑ Hard-coded password	2
☑ ▾ Potential Injection	1
☑ Suspicious usage of OPEN DATASET	1

C++

Fortify

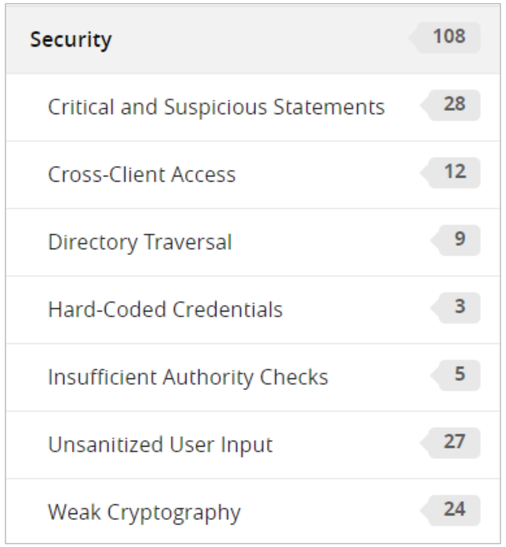
klocwork

BLACKDUCK

Teamscale

SpotBugs

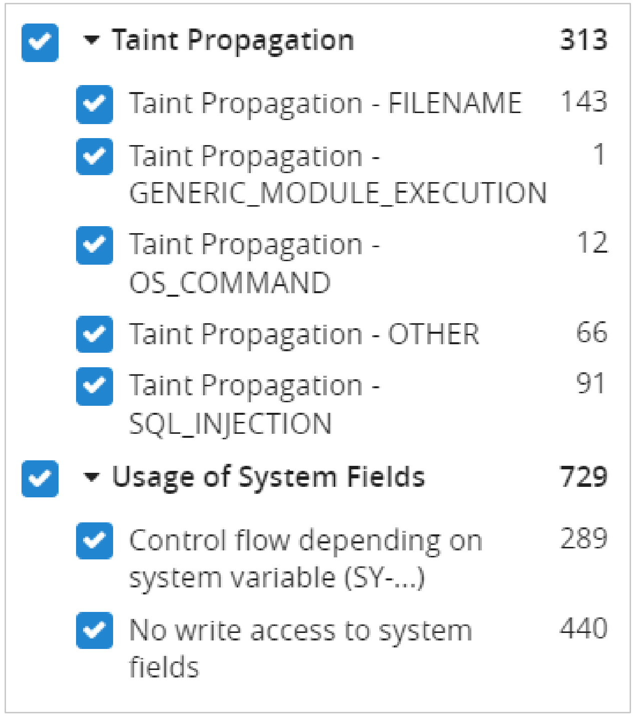
SANS CWE TOP 25



Security	108
Critical and Suspicious Statements	28
Cross-Client Access	12
Directory Traversal	9
Hard-Coded Credentials	3
Insufficient Authority Checks	5
Unsanitized User Input	27
Weak Cryptography	24

sonarlint

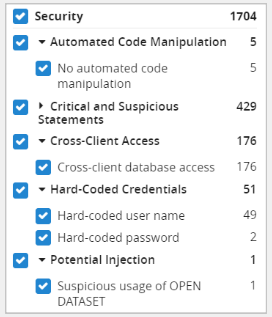
OWASP TOP 10



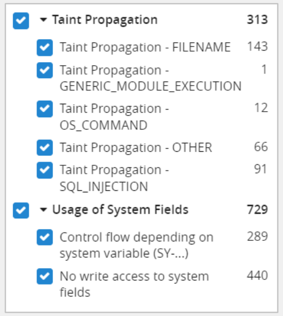
☑ ▾ Taint Propagation	313
☑ Taint Propagation - FILENAME	143
☑ Taint Propagation - GENERIC_MODULE_EXECUTION	1
☑ Taint Propagation - OS_COMMAND	12
☑ Taint Propagation - OTHER	66
☑ Taint Propagation - SQL_INJECTION	91
☑ ▾ Usage of System Fields	729
☑ Control flow depending on system variable (SY-...)	289
☑ No write access to system fields	440

Transparenz schaffen

Regelset sinnvoll wählen

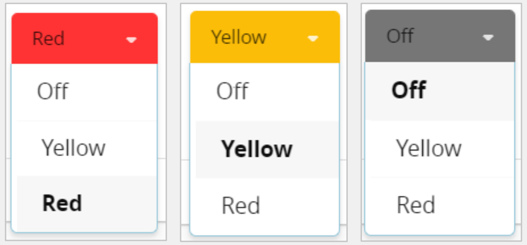


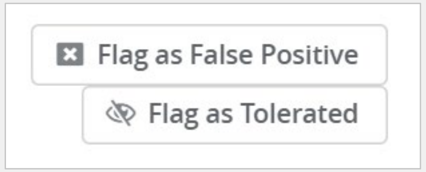
☑ Security 1704
☑ ▾ Automated Code Manipulation 5
☑ No automated code manipulation 5
☑ ▸ Critical and Suspicious Statements 429
☑ ▾ Cross-Client Access 176
☑ Cross-client database access 176
☑ ▾ Hard-Coded Credentials 51
☑ Hard-coded user name 49
☑ Hard-coded password 2
☑ ▾ Potential Injection 1
☑ Suspicious usage of OPEN DATASET 1



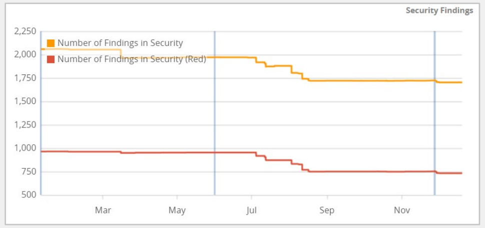
☑ ▾ Taint Propagation 313
☑ Taint Propagation - FILENAME 143
☑ Taint Propagation - GENERIC_MODULE_EXECUTION 1
☑ Taint Propagation - OS_COMMAND 12
☑ Taint Propagation - OTHER 66
☑ Taint Propagation - SQL_INJECTION 91
☑ ▾ Usage of System Fields 729
☑ Control flow depending on system variable (SY-...) 289
☑ No write access to system fields 440

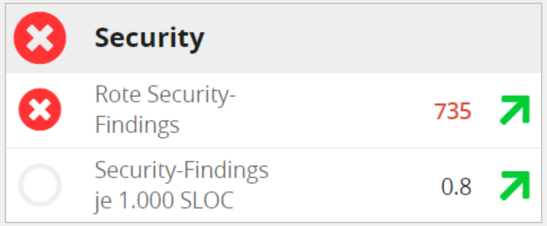
Priorität und Relevanz der Findings festlegen





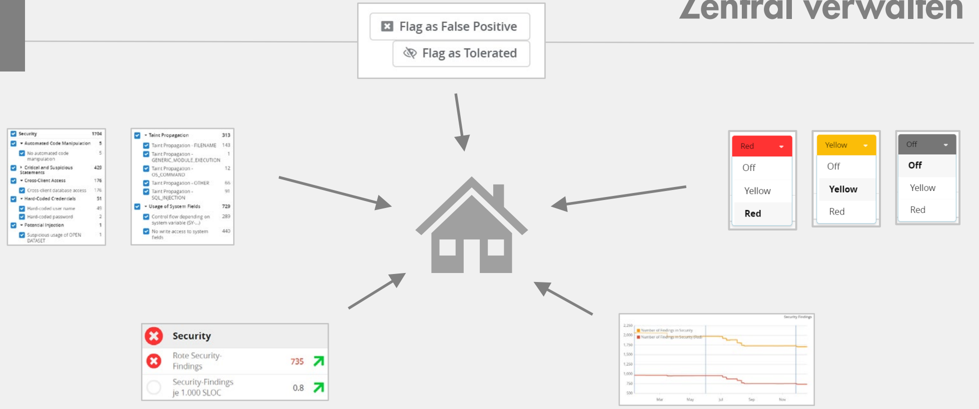
Monitoring aufsetzen



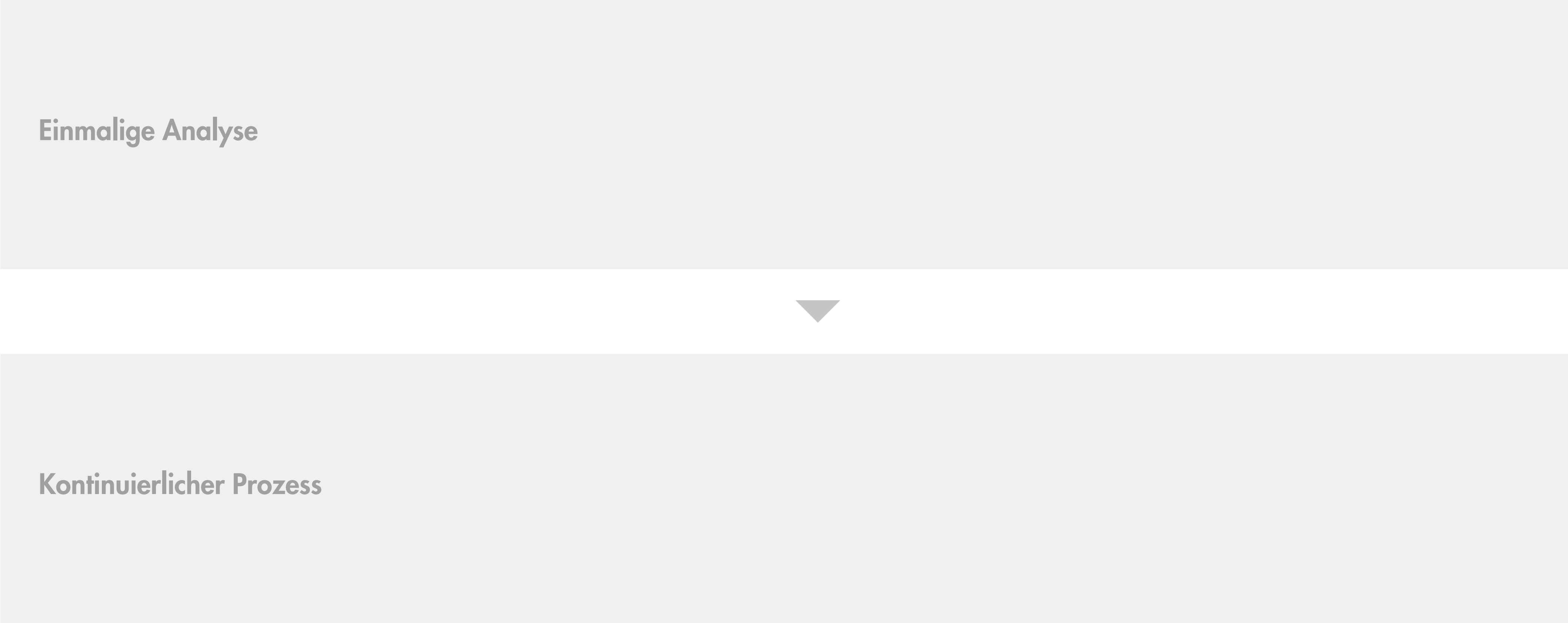


☒	**Security**		
☒	Rote Security-Findings	735	↗
○	Security-Findings je 1.000 SLOC	0.8	↗

Zentral verwalten



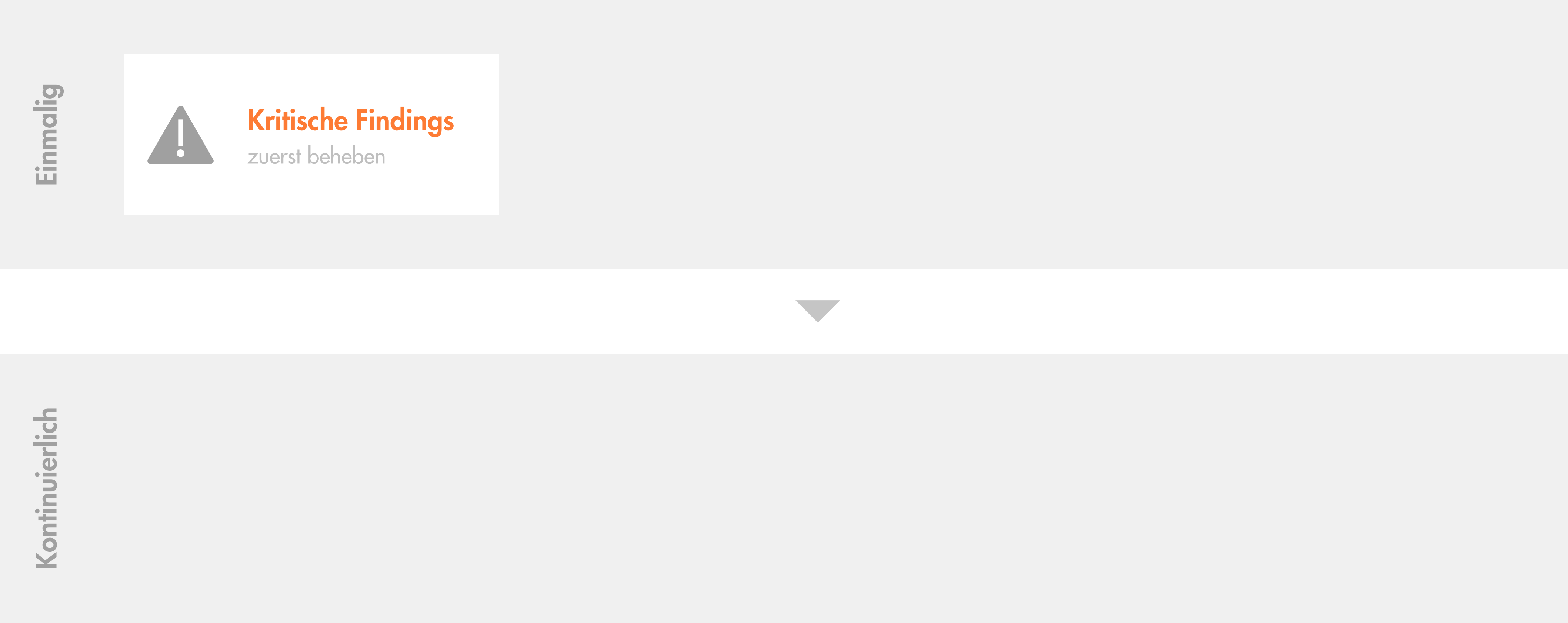
Security-Analyse



Einmalige Analyse

Kontinuierlicher Prozess

Security-Analyse



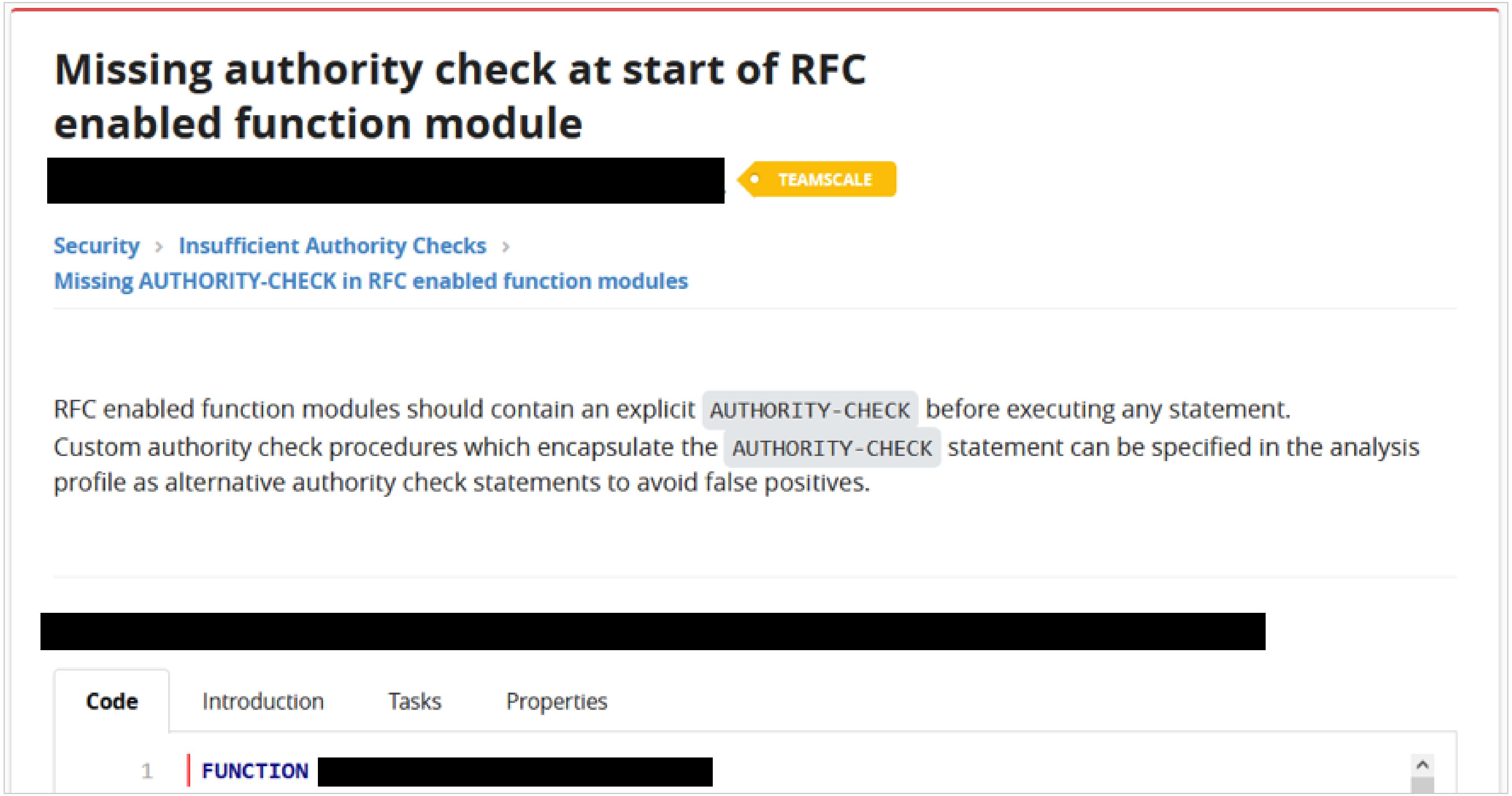
Einmalig

Kritische Findings

zuerst beheben

Kontinuierlich

⚠ Fehlende Berechtigungsprüfung



Missing authority check at start of RFC enabled function module

[redacted]

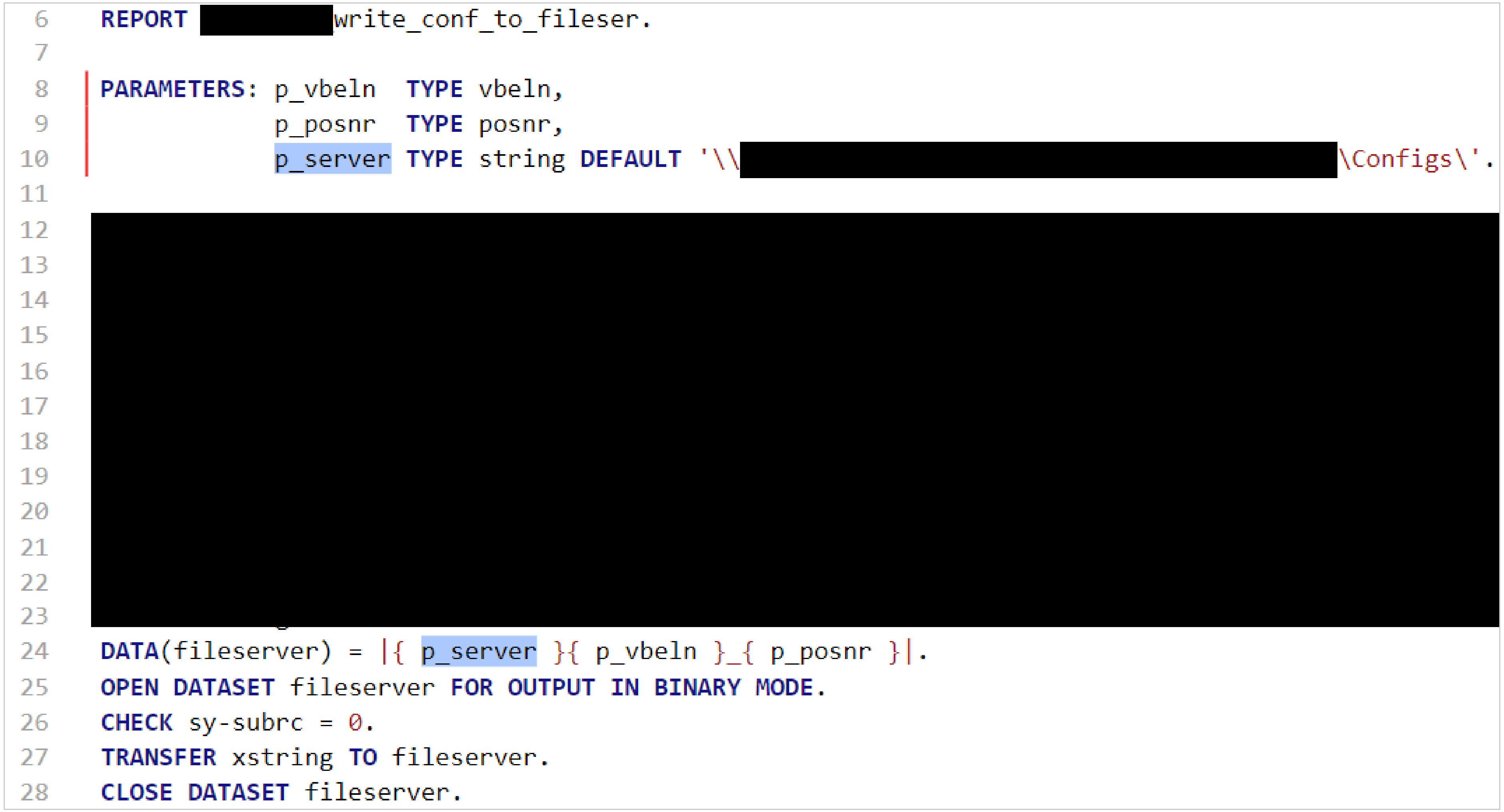
Security › Insufficient Authority Checks › Missing AUTHORITY-CHECK in RFC enabled function modules

RFC enabled function modules should contain an explicit `AUTHORITY-CHECK` before executing any statement. Custom authority check procedures which encapsulate the `AUTHORITY-CHECK` statement can be specified in the analysis profile as alternative authority check statements to avoid false positives.

[redacted]

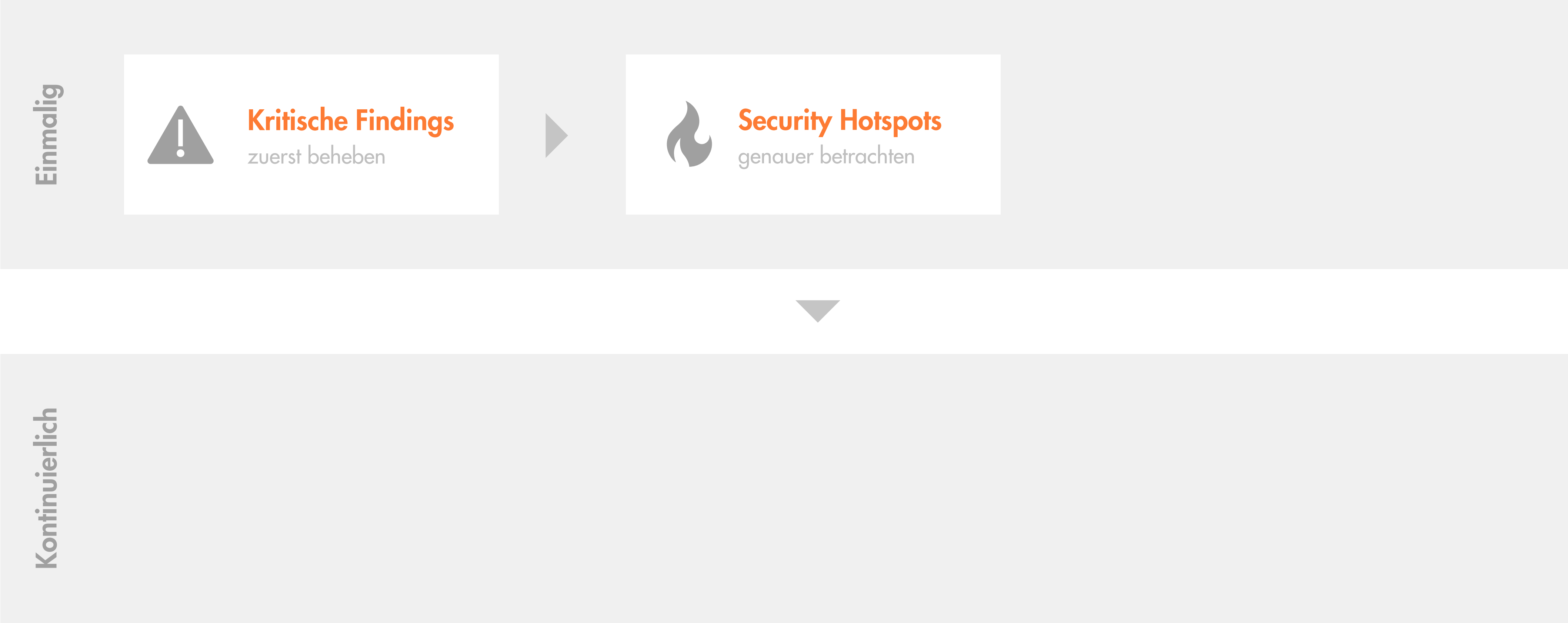
```
FUNCTION [redacted]
```

⚠ Mögliches Path-Traversal



```
REPORT [redacted]write_conf_to_fileser.

PARAMETERS: p_vbeln  TYPE vbeln,
            p_posnr  TYPE posnr,
            p_server TYPE string DEFAULT '\\[redacted]\Configs\'.

[redacted]
DATA(fileserver) = |{ p_server }{ p_vbeln }_{ p_posnr }|.
OPEN DATASET fileserver FOR OUTPUT IN BINARY MODE.
CHECK sy-subrc = 0.
TRANSFER xstring TO fileserver.
CLOSE DATASET fileserver.
```

Security-Analyse



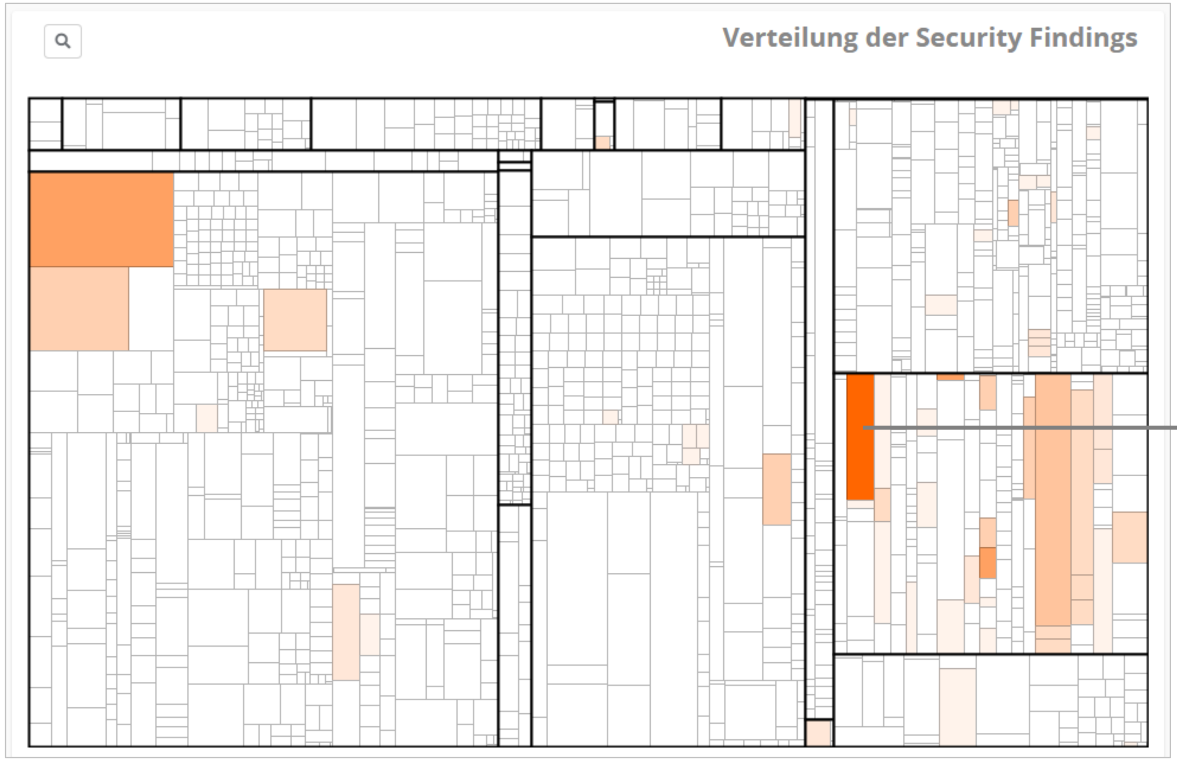
Einmalig

Kritische Findings
zuerst beheben

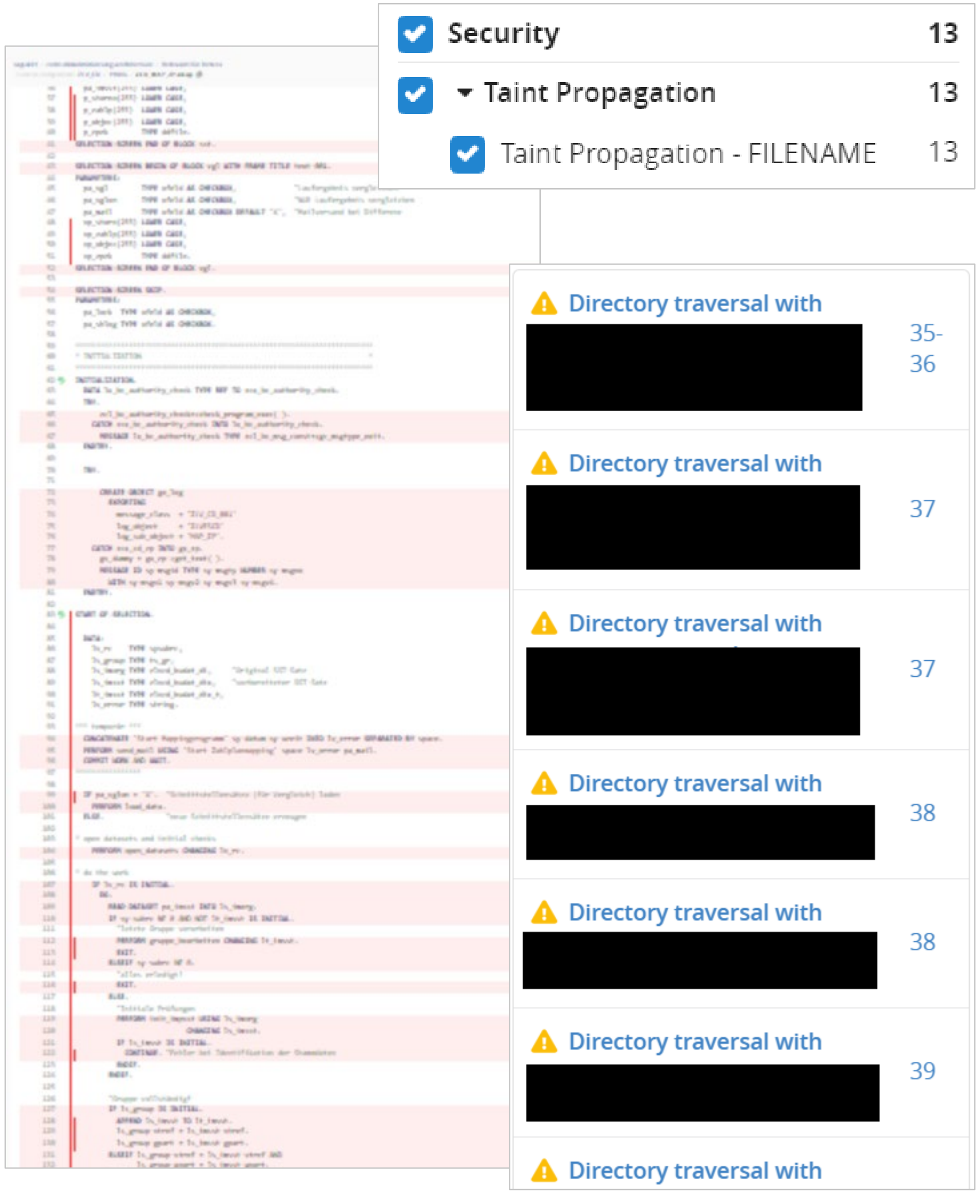
Security Hotspots
genauer betrachten

Kontinuierlich

Security Hotspots



Verteilung der Security Findings



☑	Security	13
☑	▾ Taint Propagation	13
☑	Taint Propagation - FILENAME	13

⚠ Directory traversal with [redacted] 35-36

⚠ Directory traversal with [redacted] 37

⚠ Directory traversal with [redacted] 37

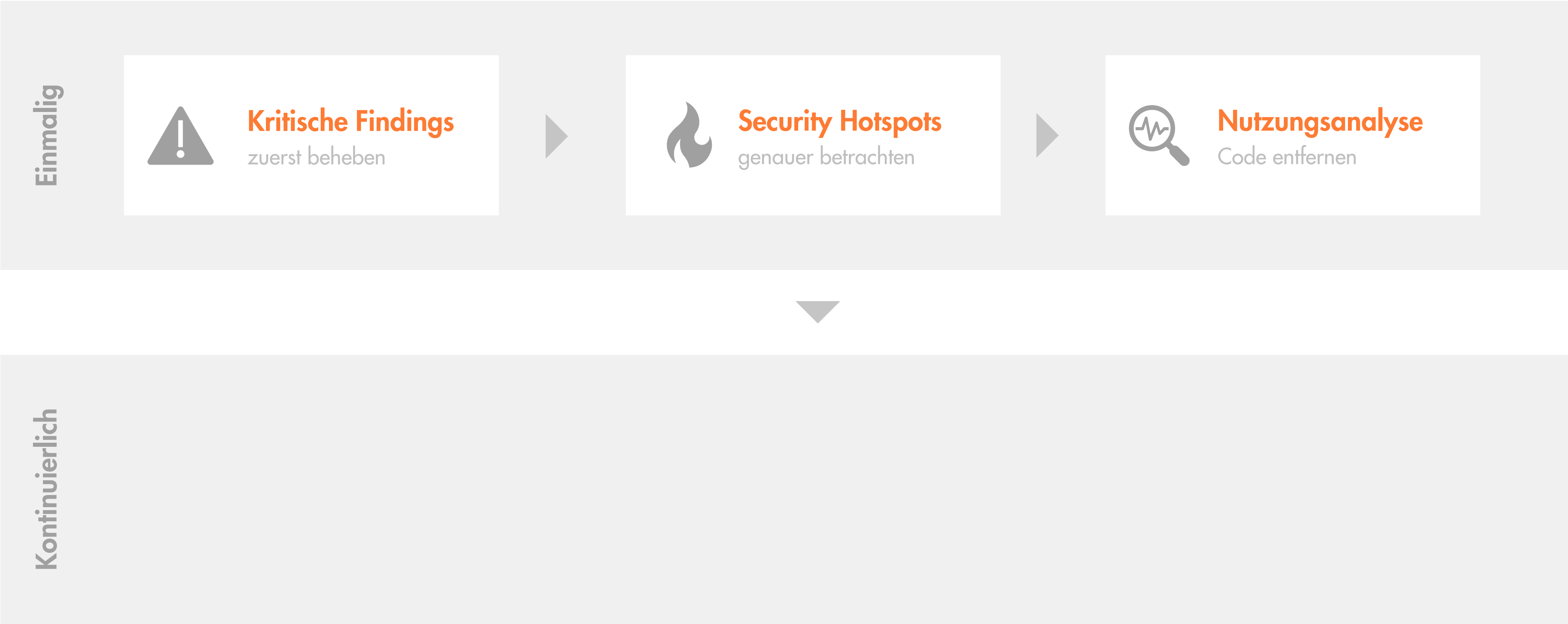
⚠ Directory traversal with [redacted] 38

⚠ Directory traversal with [redacted] 38

⚠ Directory traversal with [redacted] 39

⚠ Directory traversal with

Security-Analyse



Einmalig

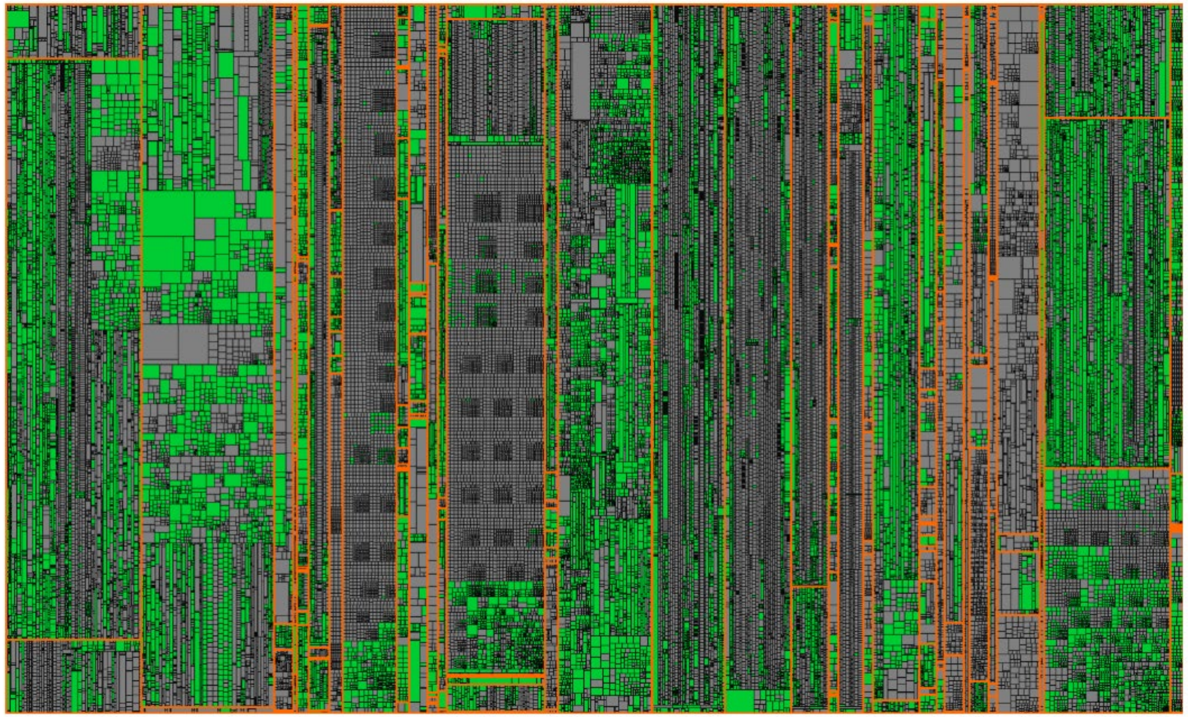
Kritische Findings
zuerst beheben

Security Hotspots
genauer betrachten

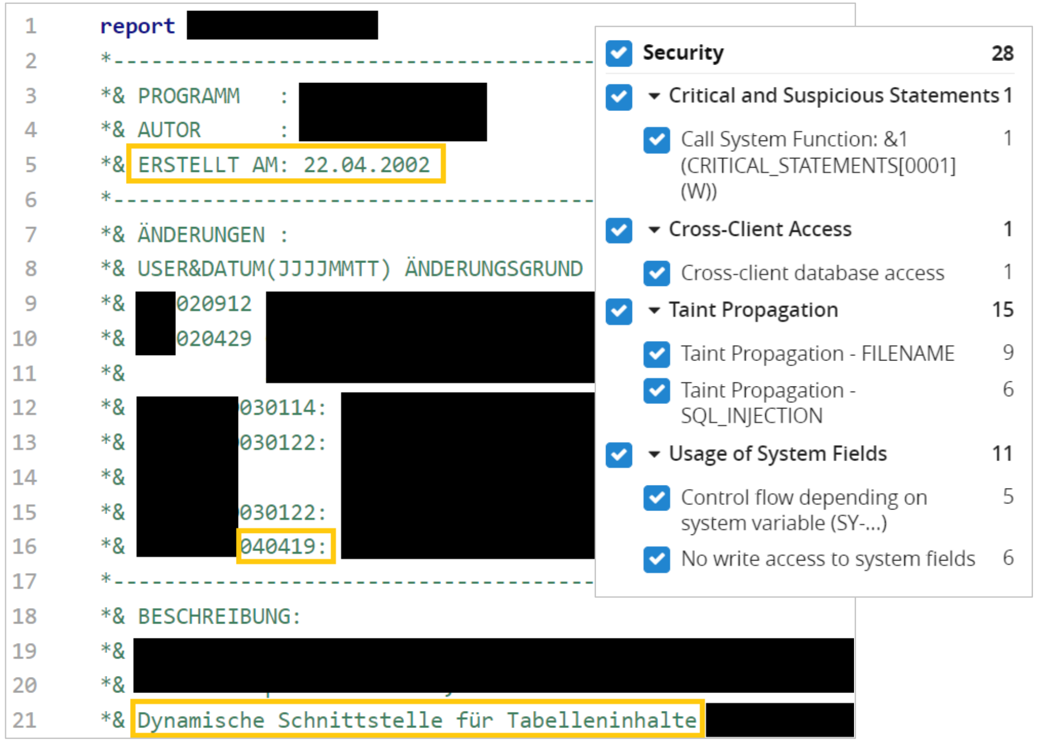
Nutzungsanalyse
Code entfernen

Kontinuierlich

Nutzungsanalyse



37% ausgeführter Code innerhalb eines Jahres



```
report [redacted]
*-----------------------------------------
*& PROGRAMM    : [redacted]
*& AUTOR       :
*& ERSTELLT AM: 22.04.2002
*-----------------------------------------
*& ÄNDERUNGEN :
*& USER&DATUM(JJJJMMTT) ÄNDERUNGSGRUND
*& [redacted]020912 [redacted]
*&   020429
*&
*& [redacted]030114: [redacted]
*&          030122:
*&
*&          030122:
*&          040419:
*-----------------------------------------
*& BESCHREIBUNG:
*& [redacted]
*&
*& Dynamische Schnittstelle für Tabelleninhalte [redacted]
```

☑ Security 28

☑ ▾ Critical and Suspicious Statements 1

☑ Call System Function: &1 (CRITICAL_STATEMENTS[0001] (W)) 1

☑ ▾ Cross-Client Access 1

☑ Cross-client database access 1

☑ ▾ Taint Propagation 15

☑ Taint Propagation - FILENAME 9

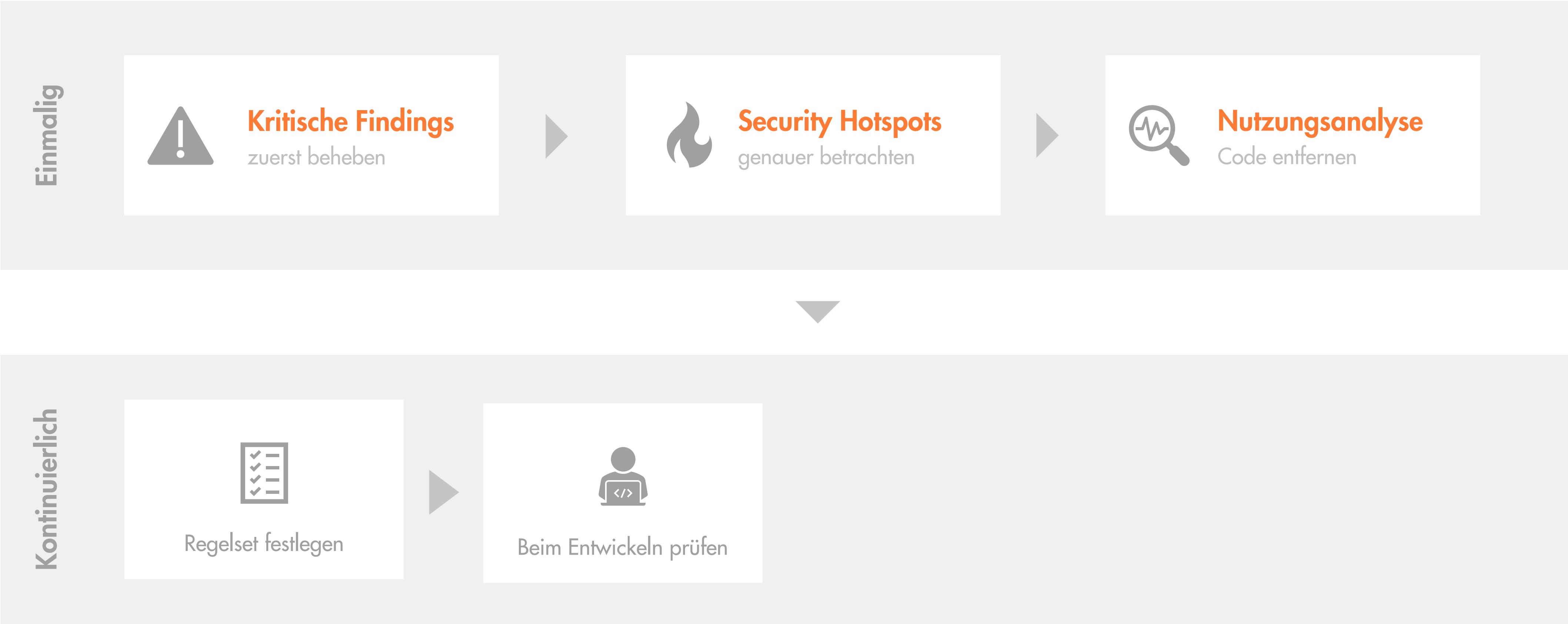
☑ Taint Propagation - SQL_INJECTION 6

☑ ▾ Usage of System Fields 11

☑ Control flow depending on system variable (SY-...) 5

☑ No write access to system fields 6

Security-Analyse



Einmalig

Kritische Findings
zuerst beheben

Security Hotspots
genauer betrachten

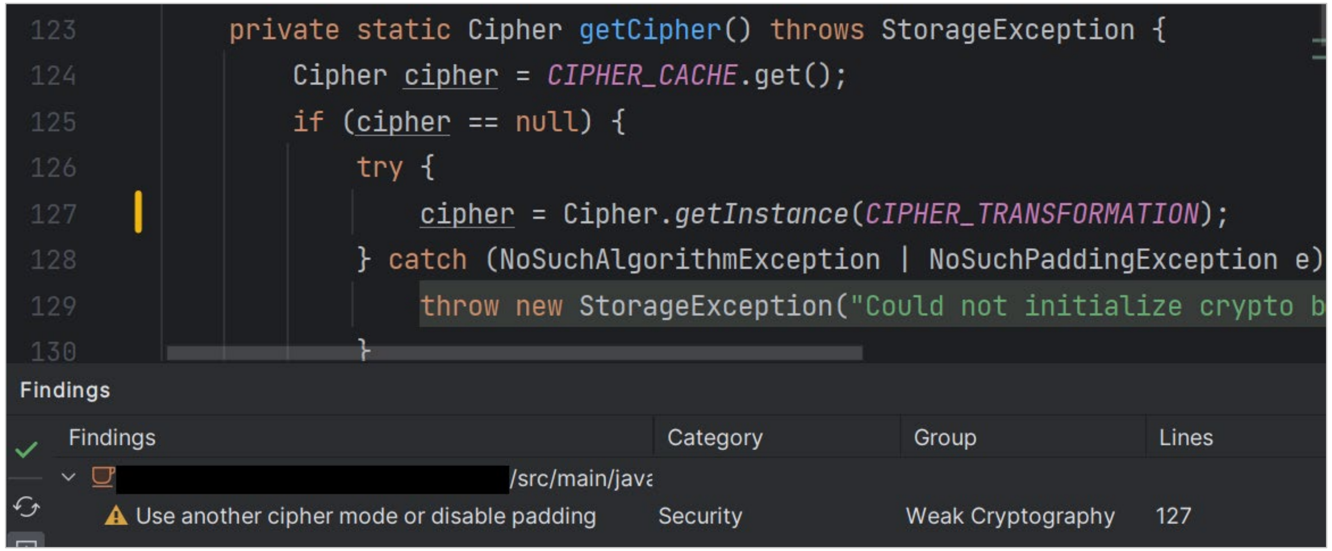
Nutzungsanalyse
Code entfernen

Kontinuierlich

Regelset festlegen

Beim Entwickeln prüfen

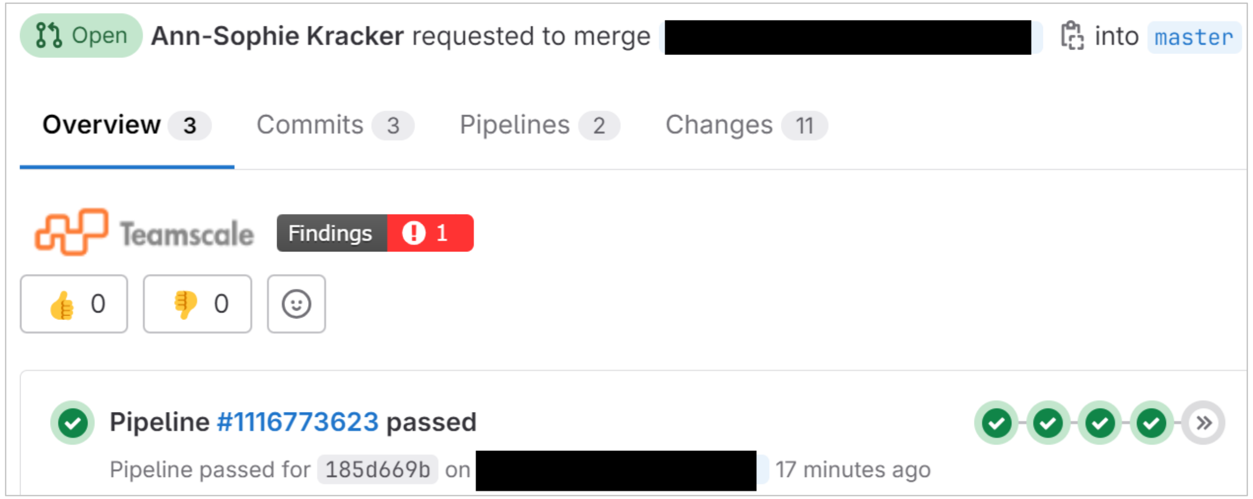
Beim Entwickeln prüfen



```
    private static Cipher getCipher() throws StorageException {
        Cipher cipher = CIPHER_CACHE.get();
        if (cipher == null) {
            try {
                cipher = Cipher.getInstance(CIPHER_TRANSFORMATION);
            } catch (NoSuchAlgorithmException | NoSuchPaddingException e)
                throw new StorageException("Could not initialize crypto b
            }
```

Findings

Findings	Category	Group	Lines
[redacted]/src/main/java			
⚠ Use another cipher mode or disable padding	Security	Weak Cryptography	127

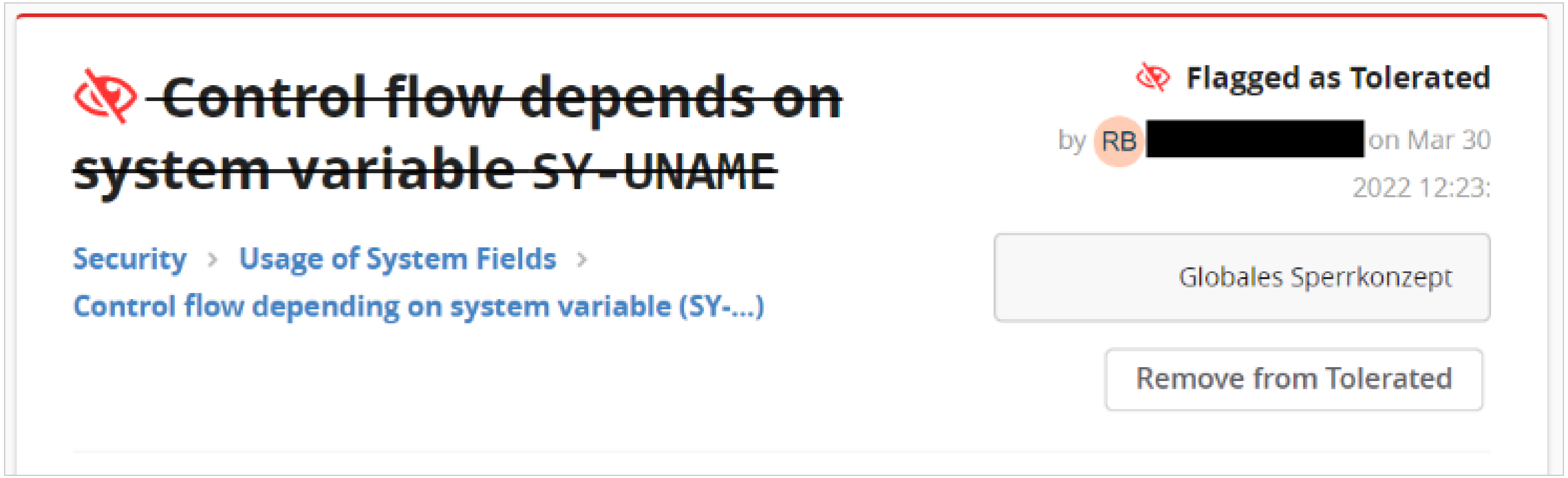


Open Ann-Sophie Kracker requested to merge [redacted] into master

Pipeline #1116773623 passed

Pipeline passed for 185d669b on [redacted] 17 minutes ago

Beim Entwickeln prüfen



~~Control flow depends on system variable SY-UNAME~~

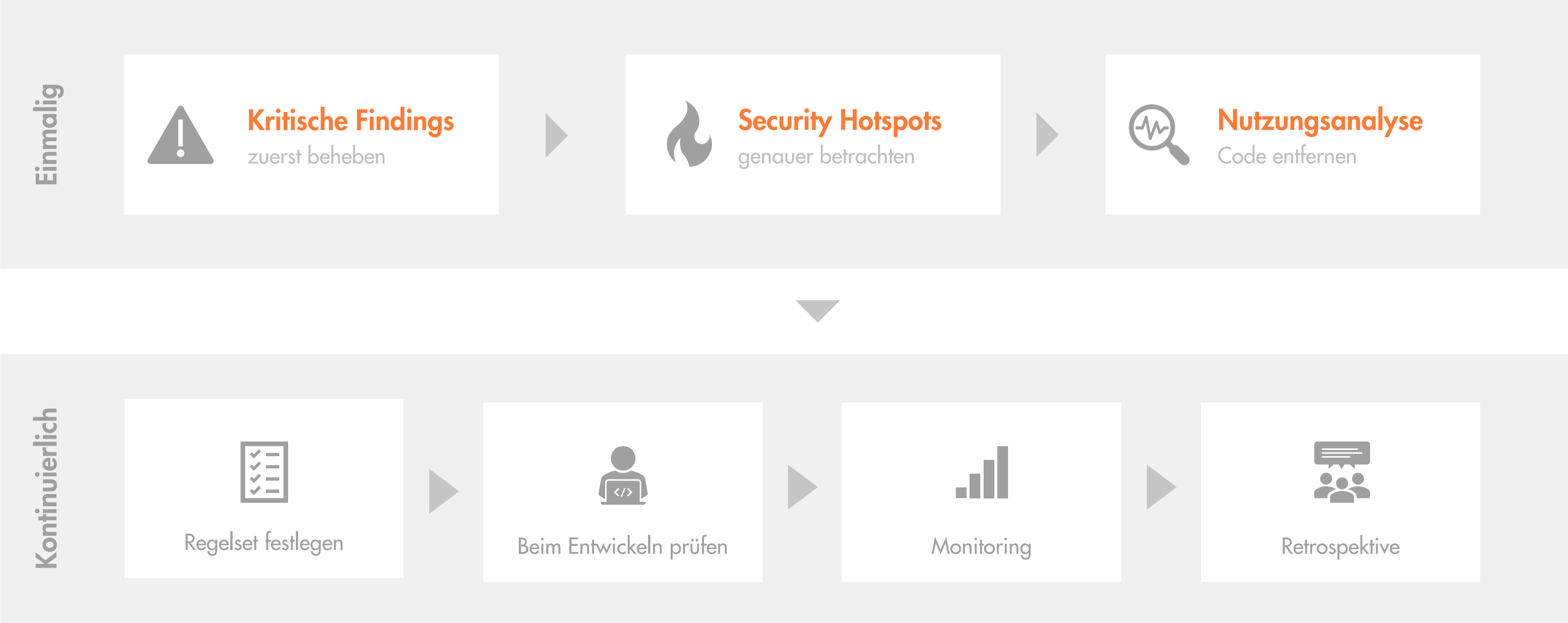
Security › Usage of System Fields › Control flow depending on system variable (SY-...)

Flagged as Tolerated

by RB [redacted] on Mar 30 2022 12:23:

Globales Sperrkonzept

Security-Analyse



Einmalig

Kritische Findings zuerst beheben

Security Hotspots genauer betrachten

Nutzungsanalyse Code entfernen

Kontinuierlich

Regelset festlegen

Beim Entwickeln prüfen

Monitoring

Retrospektive

Retrospektive

Team-Retrospektive

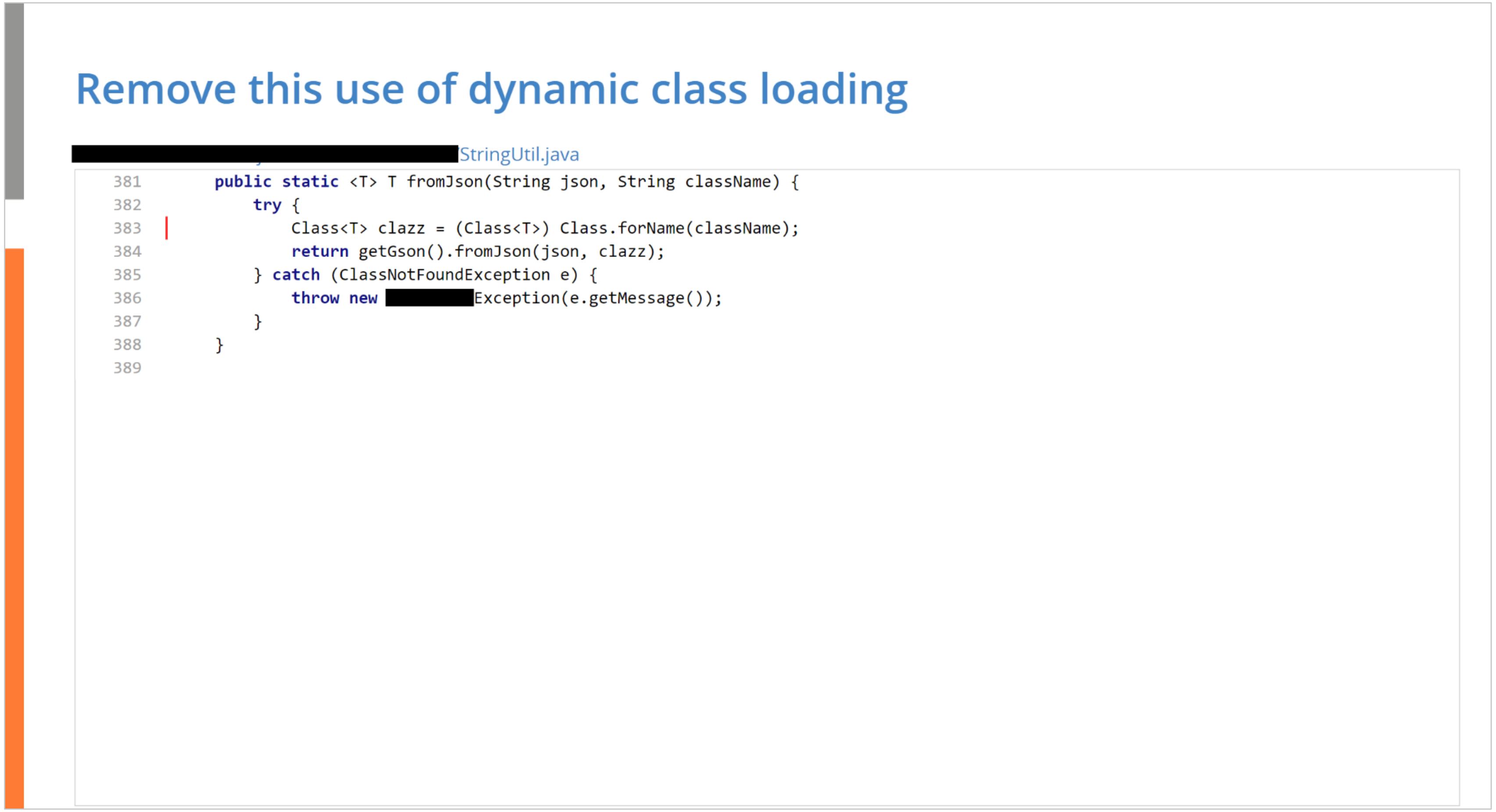
November 2023

Retrospektive

Team-Retrospektive

November 2023

CQSE

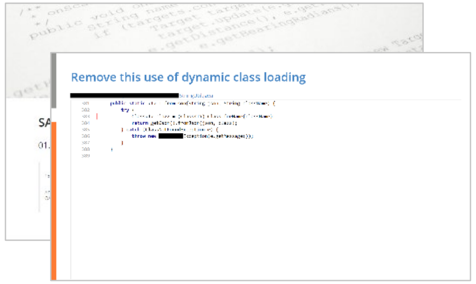


Remove this use of dynamic class loading

[redacted]StringUtil.java

```java
    public static <T> T fromJson(String json, String className) {
        try {
            Class<T> clazz = (Class<T>) Class.forName(className);
            return getGson().fromJson(json, clazz);
        } catch (ClassNotFoundException e) {
            throw new [redacted]Exception(e.getMessage());
        }
    }

```

Retrospektive



Remove this use of dynamic class loading

[redacted] [illegible]

```
public static [illegible] fromJson(String json, String [illegible]) {
    try {
        [illegible] Class.forName([illegible]);
        return getGson().fromJson(json, [illegible]);
    } catch ([illegible] e) {
        throw new [redacted]Exception(e.getMessage());
    }
}
```

Dynamisches Laden von Klassen

OWASP Top 10 2017 Category A1 - Injection

Remove this use of dynamic class loading

Findings ❗ 7

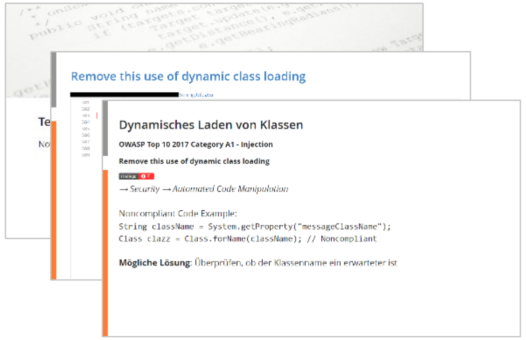
→ Security → Automated Code Manipulation

Noncompliant Code Example:

```
String className = System.getProperty("messageClassName");
Class clazz = Class.forName(className); // Noncompliant
```

Mögliche Lösung: Überprüfen, ob der Klassenname ein erwarteter ist

Retrospektive



Te

No

Remove this use of dynamic class loading

[redacted]

Dynamisches Laden von Klassen

OWASP Top 10 2017 Category A1 - Injection

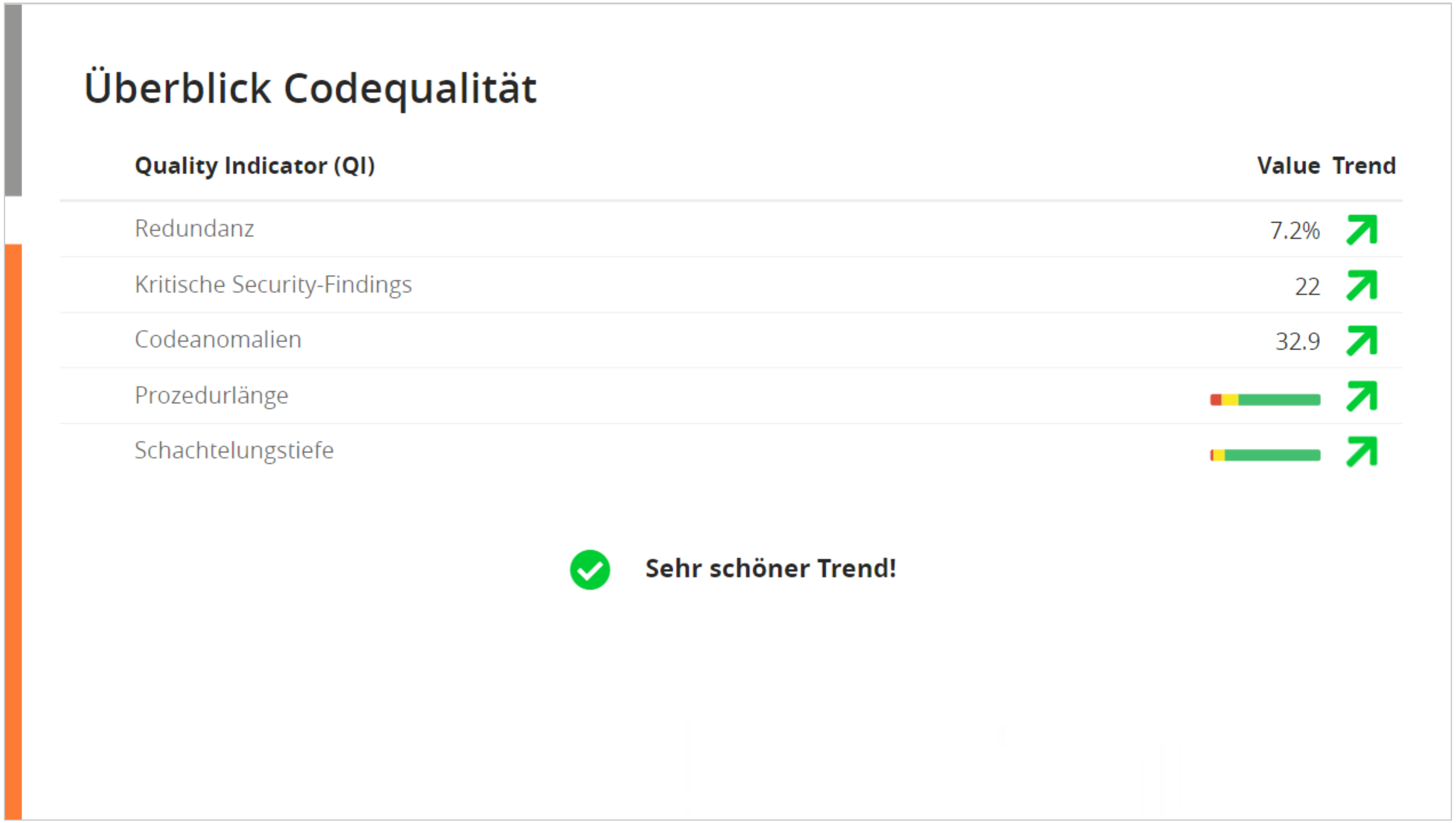
Remove this use of dynamic class loading

→ *Security* → *Automated Code Manipulation*

Noncompliant Code Example:

```
String className = System.getProperty("messageClassName");
Class clazz = Class.forName(className); // Noncompliant
```

Mögliche Lösung: Überprüfen, ob der Klassenname ein erwarteter ist

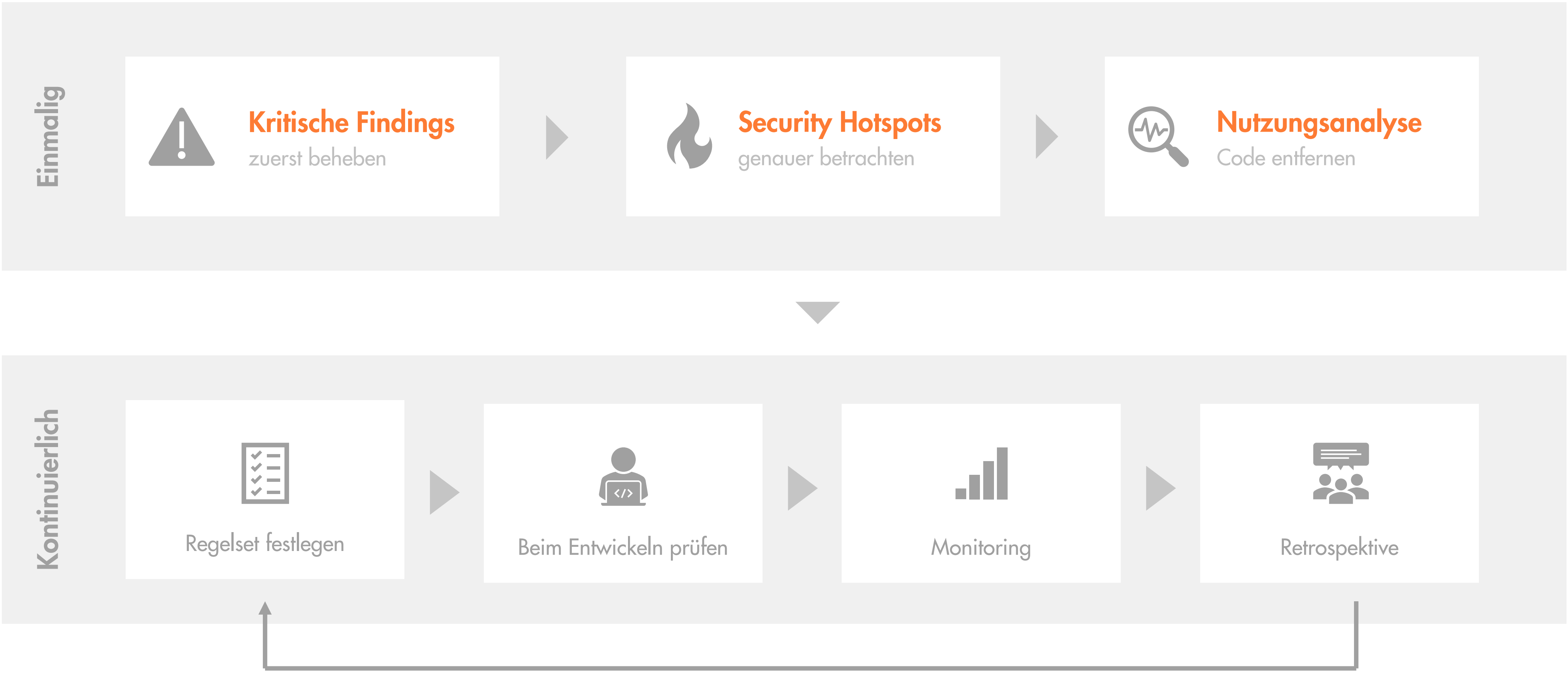


Überblick Codequalität

Quality Indicator (QI)	Value	Trend
Redundanz	7.2%	↗
Kritische Security-Findings	22	↗
Codeanomalien	32.9	↗
Prozedurlänge	🟥🟨🟩	↗
Schachtelungstiefe	🟥🟨🟩	↗

✓ **Sehr schöner Trend!**

Security-Analyse



Einmalig

Kritische Findings
zuerst beheben

Security Hotspots
genauer betrachten

Nutzungsanalyse
Code entfernen

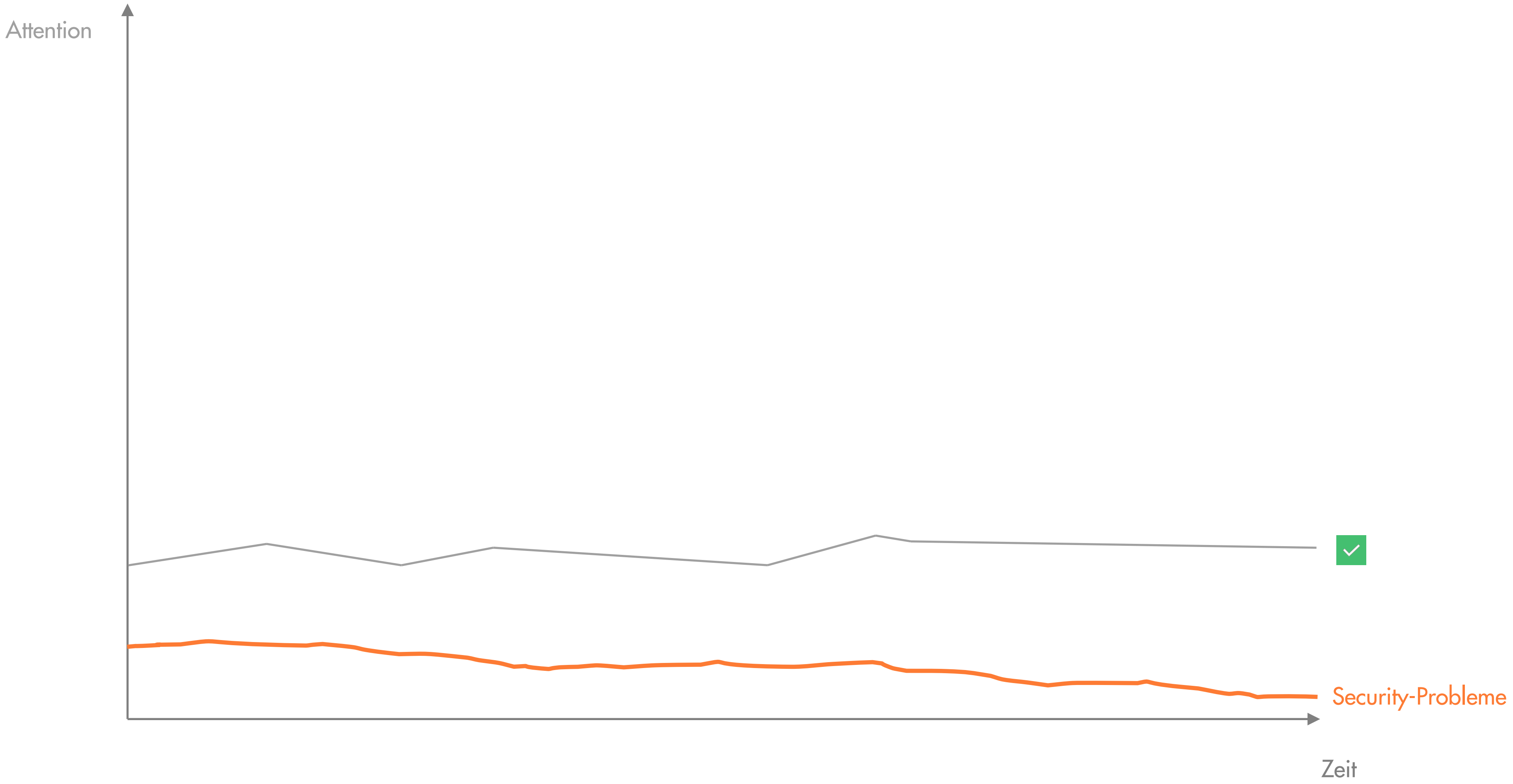
Kontinuierlich

Regelset festlegen

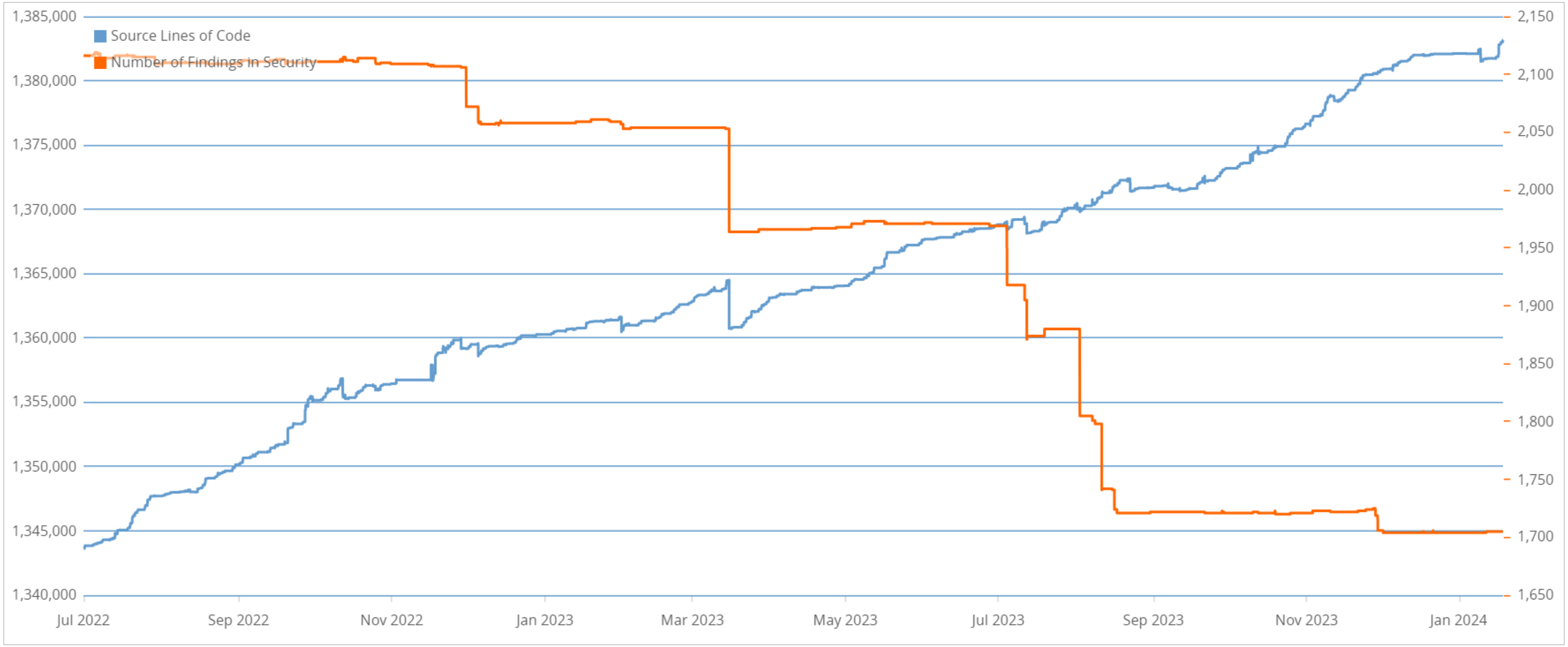
Beim Entwickeln prüfen

Monitoring

Retrospektive


Attention
✓
Security-Probleme
Zeit

Security-Trend



Wir begrüßen Sie gerne an unserem Stand!

Dr. Nils Göde · goede@cqse.eu · +49 176 10452662
Ann-Sophie Kracker· kracker@cqse.eu · +49 172 1860208

CQSE GmbH
Centa-Hafenbrädl-Straße 59
81249 München

CQSE
Continuous Quality in Software Engineering

Ignorieren bis es knallt? Security-Analysen

aus Entwickler- und Management-Perspektive

tmscl.me/oop-2024-talk1

Wie zukunftssicher ist Ihr Softwaresystem?

Unser Vorgehen und Erfahrungen aus 10 Jahren Software-Audits

tmscl.me/a2310-oop

Continuous Quality Control

Qualität trotz immer kürzerer Releasezyklen

12. März (10:30 – 12:00)

tmscl.me/cqc-2024-03-oop