logic technology

Open Source Management zonder omkijken

Gevorg Melikdjanjan – Product Manager

Logic Technology

Het ontwerpen van innovatieve elektronica

Woensdag 19 april 2023

1931 Congrescentrum 's-Hertogenbosch

Logic Solutions

- Static Code Analysis
- Test Automation
- MISRA®-C Checkers
- Architectural Analysis
- Application Lifecycle Management
- **Software Composition Analysis**

software quality

software components

- Flash Management Software
- High-Performance File Systems
- Embedded In-Memory Database Management
- UEFI BIOS & Bootloaders

- Computer Modules
- Boundary Scan Testing
- Device Programmers
- Connectors
- SBC
- Development Kits

board & solutions

development tools

- Model Driven Software Engineering
- Embedded GUI Development
- JTAG Debuggers
- Intel IDE’s
- CI/CD Build Time Optimization

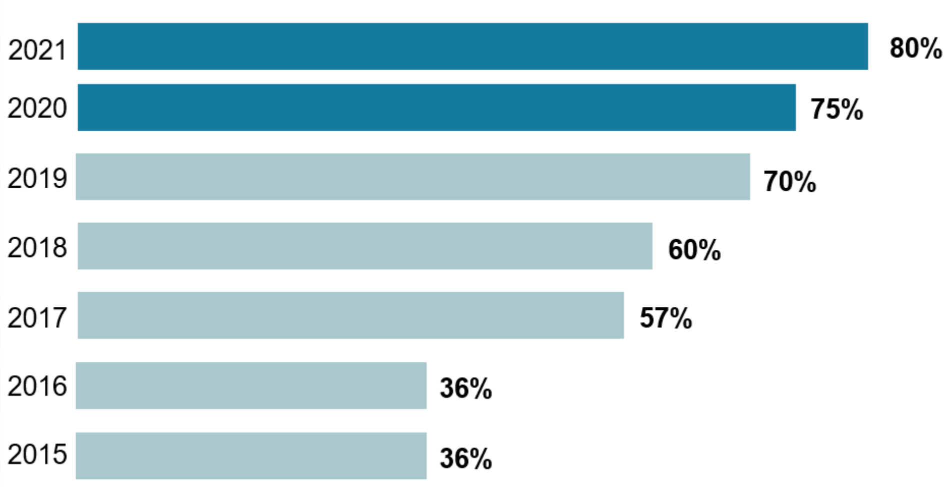
logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

The Problem with Open Source

60% to 80% of an average app’s code base is comprised of open source you didn’t write,

But **you own** managing the risks



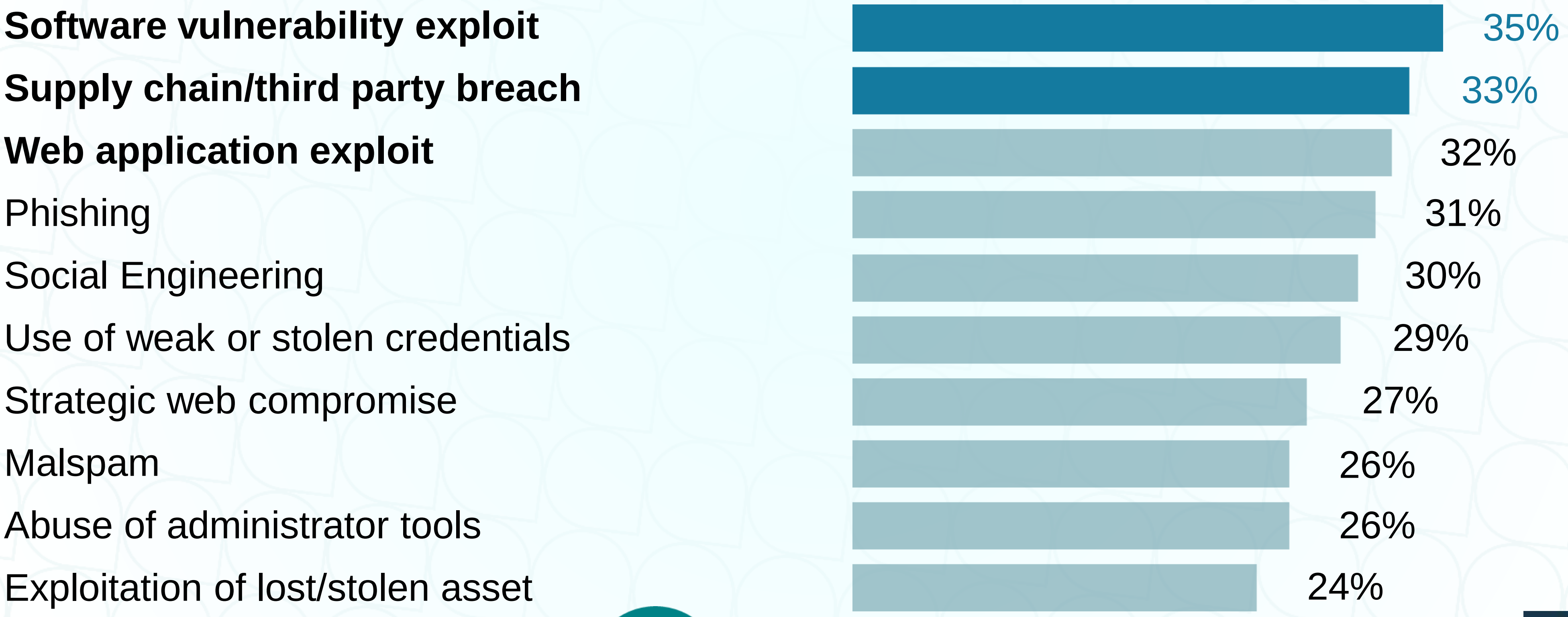
Source: Synopsys OSSRA reports, 2015 through 2021

logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

Today’s reality: Attackers Target Applications

How was the external attack carried out?



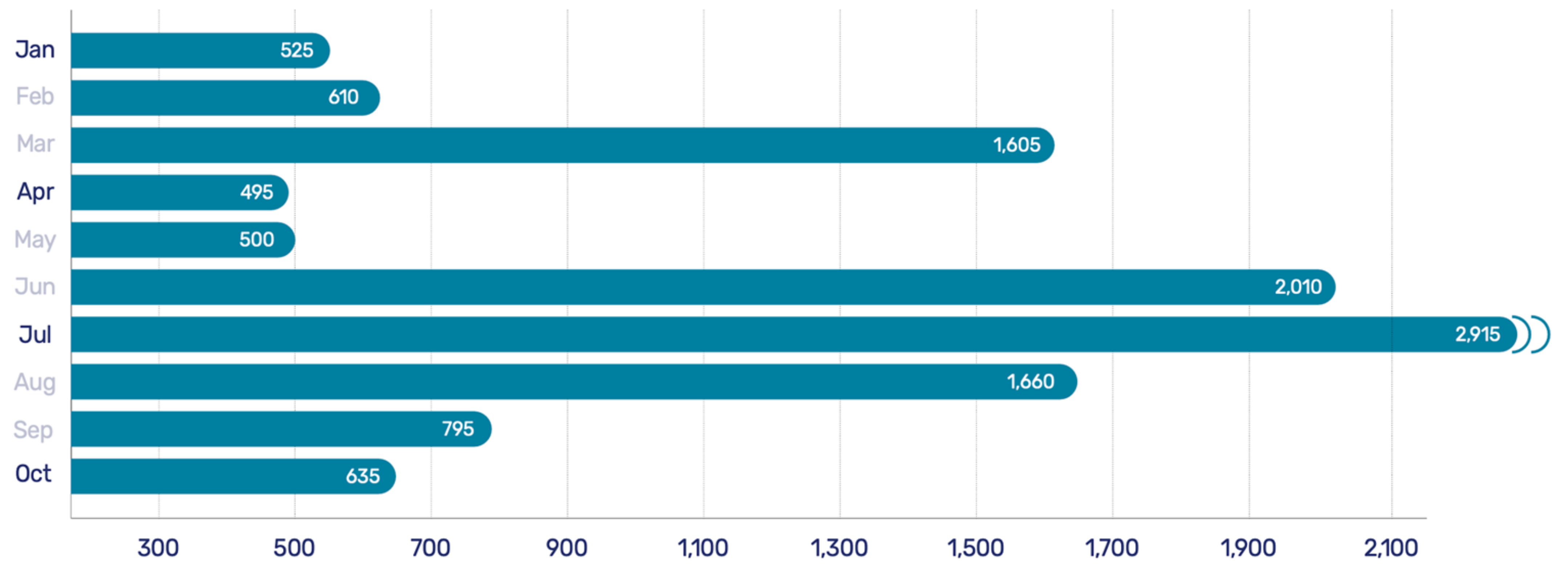
Source: Forrester Analytics Business Technographics Security Survey, 2021
Base: 530 Security decision-makers with network, data center, app security, or security ops responsibilities who experienced an external attack when their company was breached

logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

Malicious Open Source Packages Are Growing Fast

2022 growth of malicious packages across npm and Rubygems



Source: 2022 Mend Open Source Risk Report

Identify and Fix OSS Vulnerabilities

- Manual research
- Build automation tools
- Apply security patches promptly
- **Software composition analysis (SCA)**

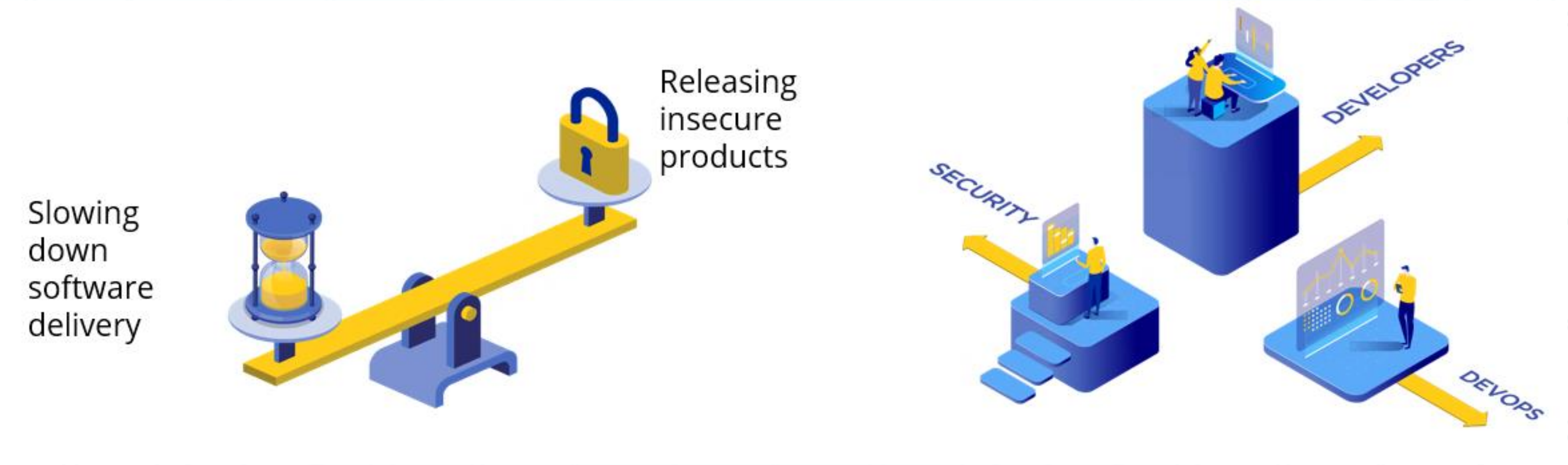
logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

Developers Shouldn’t become Security Experts

- Growing friction – DevOps, security,
  and developers are not on the same page

- Security by exception, not interruption



logic technology

4 Main Challenges Managing Open Source Risks

1 Knowing which open source components you are using

2 Ensuring up-to-date and accurate risk info about open source inventory

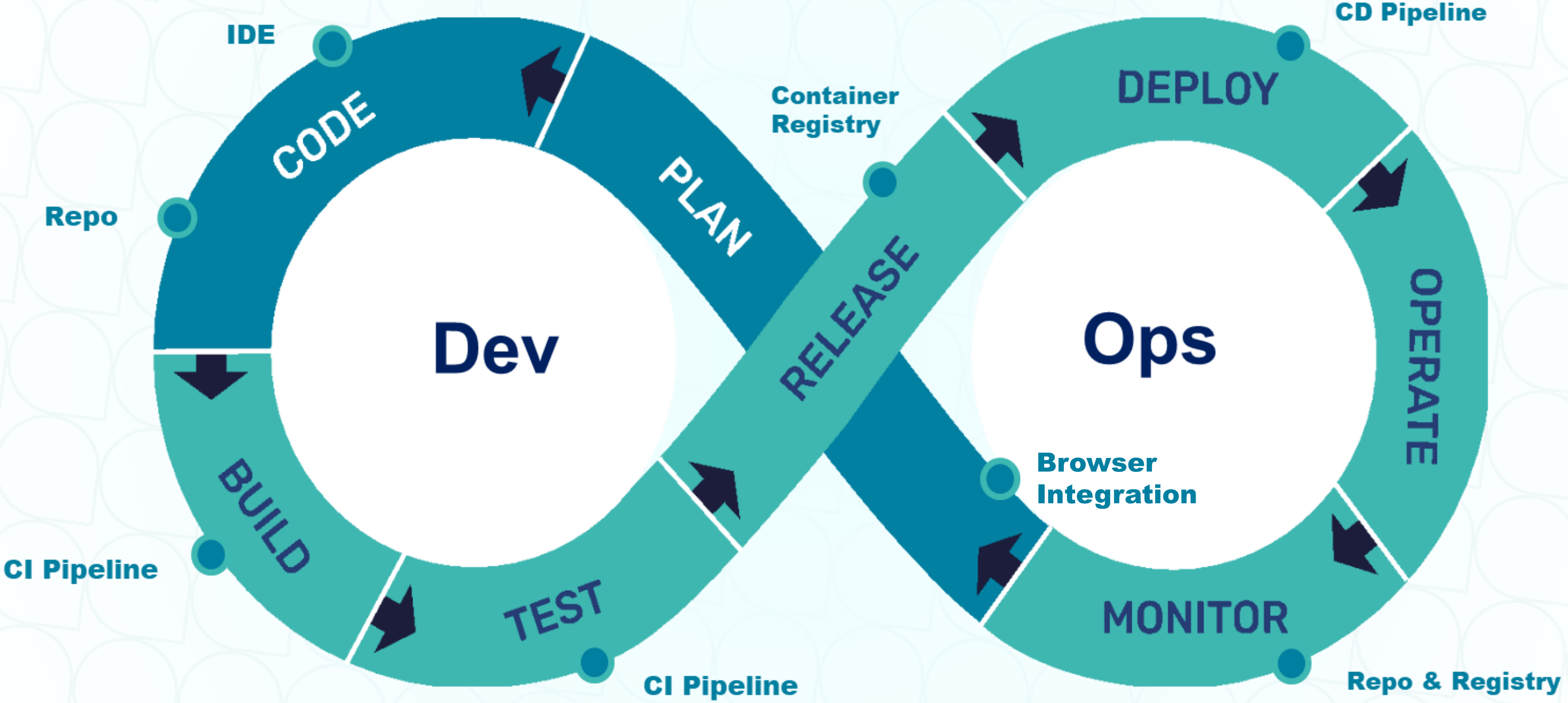
3 Filtering and prioritizing which issues need action

4 Choosing which remediation actions to take

logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

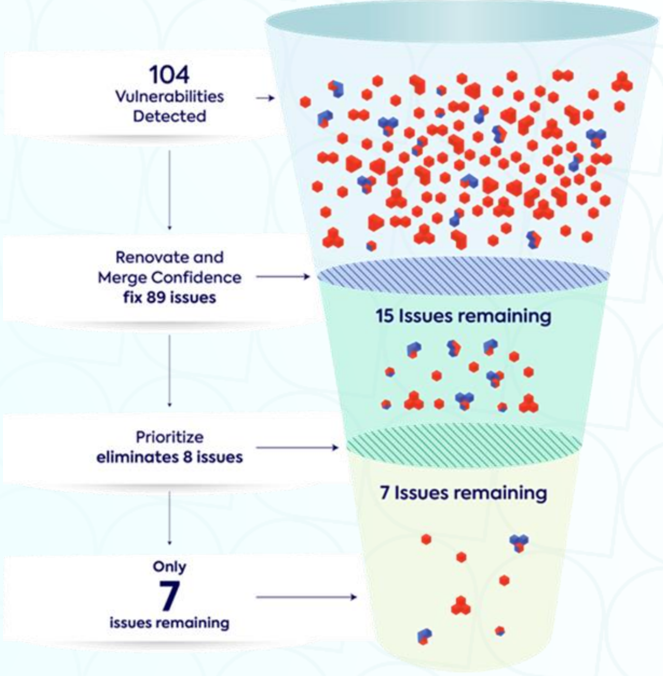
Shift-left Principle



logic technology

D&E EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

The Mend Approach



104 Vulnerabilities Detected

Renovate and Merge Confidence fix 89 issues

15 Issues remaining

Prioritize eliminates 8 issues

7 Issues remaining

Only 7 issues remaining

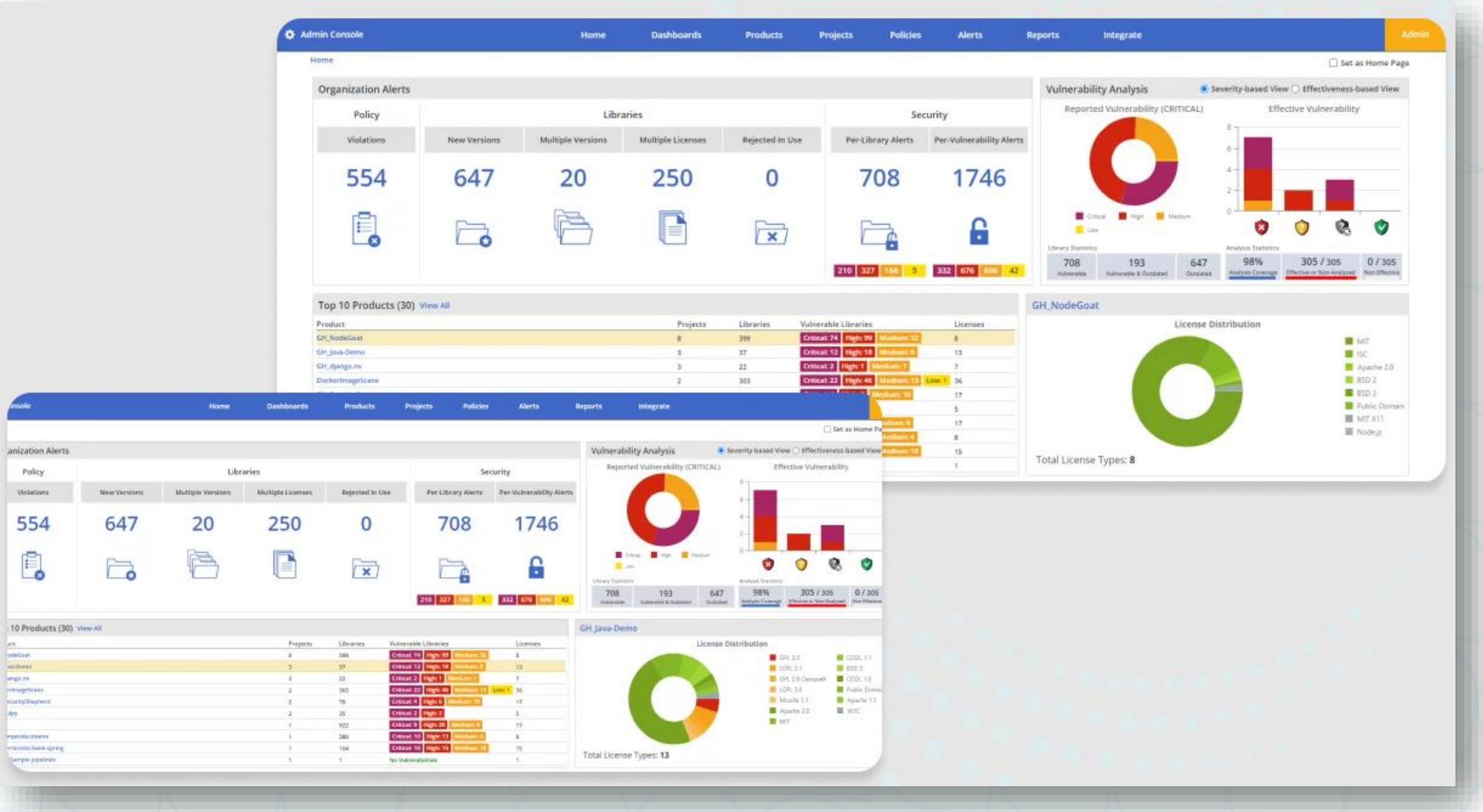
Legacy security tools
Focus on **detection**

- X Tell developers about everything that’s wrong
- X Make dev responsible for all remediation
- X Make developers leave their native environment
- X Security backlog continues to grow

Mend
Focus on **remediation**

- ✓ Alert developers to newly introduced vulns
- ✓ Automate remediation — we do the research
- ✓ Developers never leave their familiar tools
- ✓ MTTR reduced, backlog reduced

MEND Application Security Platform



Organization Alerts

Policy	Libraries				Security	
Violations	New Versions	Multiple Versions	Multiple Licenses	Rejected In Use	Per-Library Alerts	Per-Vulnerability Alerts
554	647	20	250	0	708	1746
					210 327 186 5	332 676 696 42

Vulnerability Analysis ● Severity-based View ○ Effectiveness-based View

Library Statistics

708	193	647
Vulnerable	Vulnerable & Outdated	Outdated

Analysis Statistics

98%	305 / 305	0 / 305
Analysis Coverage	Effective or Non-Analysed	Non Effective

Top 10 Products (30) View All

Product	Projects	Libraries	Vulnerable Libraries	Licenses
GH_NodeGoat	8	399	Critical: 74 High: 99 Medium: 32	8
GH_Java-Demo	3	37	Critical: 12 High: 18 Medium: 8	13
GH_django.nv	3	22	Critical: 2 High: 1 Medium: 1	7
DockerImageScans	2	303	Critical: 22 High: 46 Medium: 13 Low: 1	36

GH_NodeGoat

Total License Types: 8

GH_Java-Demo

License Distribution

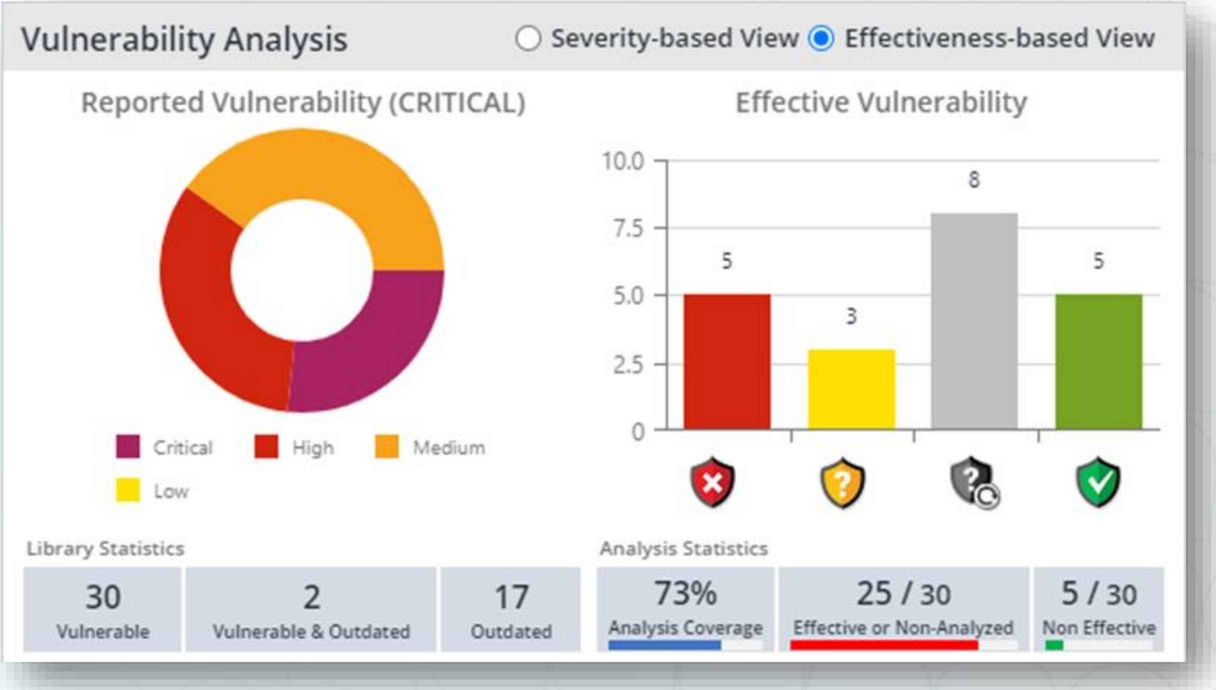
Total License Types: 13

logic technology

Effective Usage Analysis

Reachability path analysis discovers which OSS vulnerabilities matter, and which can be ignored

No more false positives


Vulnerability Analysis

○ Severity-based View ● Effectiveness-based View

Library Statistics

30 Vulnerable

2 Vulnerable & Outdated

17 Outdated

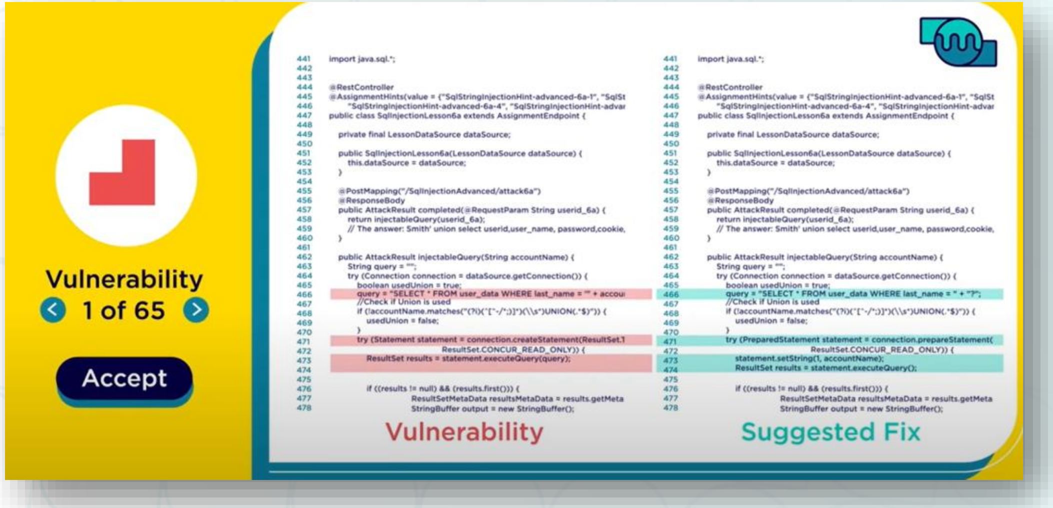
Analysis Statistics

73% Analysis Coverage

25 / 30 Effective or Non-Analyzed

5 / 30 Non Effective

Automated Remediation

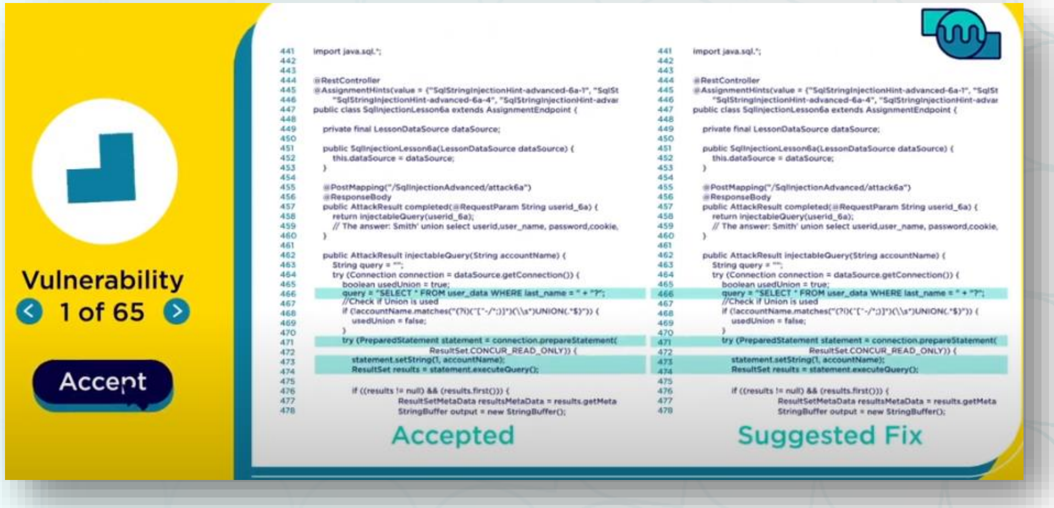


Vulnerability

```
import java.sql.*;


@RestController
@AssignmentHints(value = {"SqlStringInjectionHint-advanced-6a-1", "SqlSt
     "SqlStringInjectionHint-advanced-6a-4", "SqlStringInjectionHint-advar
public class SqlInjectionLesson6a extends AssignmentEndpoint {

   private final LessonDataSource dataSource;

   public SqlInjectionLesson6a(LessonDataSource dataSource) {
      this.dataSource = dataSource;
   }

   @PostMapping("/SqlInjectionAdvanced/attack6a")
   @ResponseBody
   public AttackResult completed(@RequestParam String userid_6a) {
      return injectableQuery(userid_6a);
      // The answer: Smith' union select userid,user_name, password,cookie,
   }

   public AttackResult injectableQuery(String accountName) {
      String query = "";
      try (Connection connection = dataSource.getConnection()) {
         boolean usedUnion = true;
         query = "SELECT * FROM user_data WHERE last_name = '" + accou
         //Check if Union is used
         if (!accountName.matches("(?i)(^[^-/*;)]*)(\\s*)UNION(.*$)")) {
            usedUnion = false;
         }
         try (Statement statement = connection.createStatement(ResultSet.1
                              ResultSet.CONCUR_READ_ONLY)) {
            ResultSet results = statement.executeQuery(query);


            if ((results != null) && (results.first())) {
                         ResultSetMetaData resultsMetaData = results.getMeta
                         StringBuffer output = new StringBuffer();
```

Suggested Fix

```
import java.sql.*;


@RestController
@AssignmentHints(value = {"SqlStringInjectionHint-advanced-6a-1", "SqlSt
     "SqlStringInjectionHint-advanced-6a-4", "SqlStringInjectionHint-advar
public class SqlInjectionLesson6a extends AssignmentEndpoint {

   private final LessonDataSource dataSource;

   public SqlInjectionLesson6a(LessonDataSource dataSource) {
      this.dataSource = dataSource;
   }

   @PostMapping("/SqlInjectionAdvanced/attack6a")
   @ResponseBody
   public AttackResult completed(@RequestParam String userid_6a) {
      return injectableQuery(userid_6a);
      // The answer: Smith' union select userid,user_name, password,cookie,
   }

   public AttackResult injectableQuery(String accountName) {
      String query = "";
      try (Connection connection = dataSource.getConnection()) {
         boolean usedUnion = true;
         query = "SELECT * FROM user_data WHERE last_name = " + "?";
         //Check if Union is used
         if (!accountName.matches("(?i)(^[^-/*;)]*)(\\s*)UNION(.*$)")) {
            usedUnion = false;
         }
         try (PreparedStatement statement = connection.prepareStatement(
                              ResultSet.CONCUR_READ_ONLY)) {
            statement.setString(1, accountName);
            ResultSet results = statement.executeQuery();

            if ((results != null) && (results.first())) {
                         ResultSetMetaData resultsMetaData = results.getMeta
                         StringBuffer output = new StringBuffer();
```



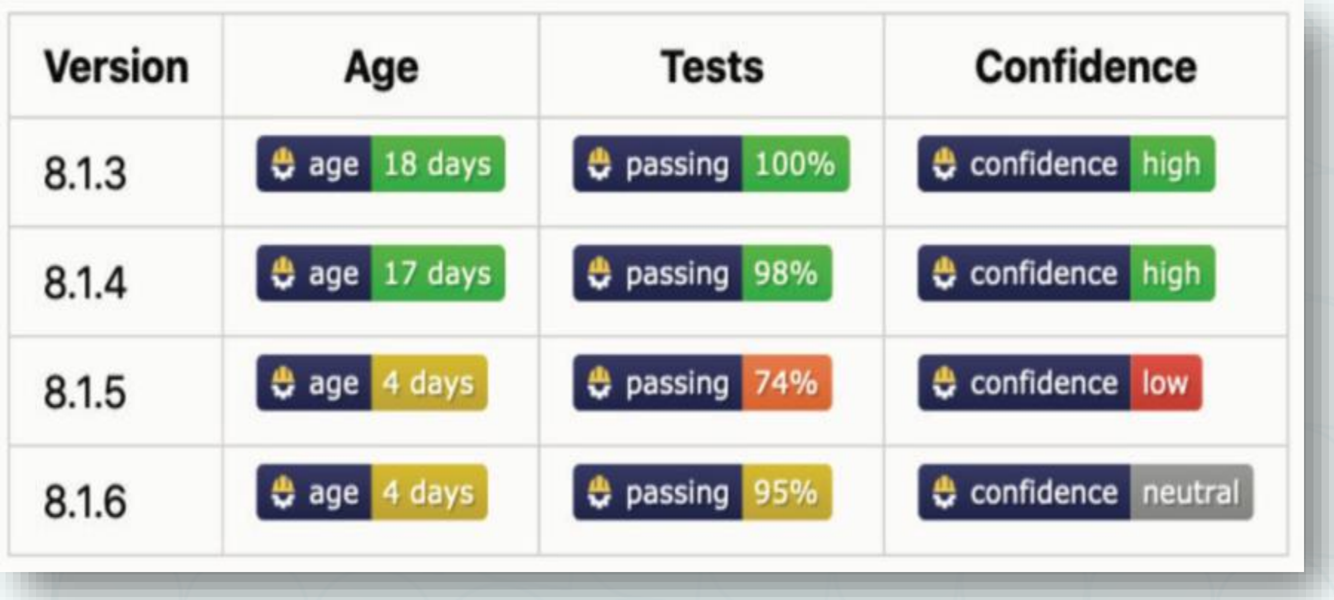
Accepted

```
import java.sql.*;


@RestController
@AssignmentHints(value = {"SqlStringInjectionHint-advanced-6a-1", "SqlSt
     "SqlStringInjectionHint-advanced-6a-4", "SqlStringInjectionHint-advar
public class SqlInjectionLesson6a extends AssignmentEndpoint {

   private final LessonDataSource dataSource;

   public SqlInjectionLesson6a(LessonDataSource dataSource) {
      this.dataSource = dataSource;
   }

   @PostMapping("/SqlInjectionAdvanced/attack6a")
   @ResponseBody
   public AttackResult completed(@RequestParam String userid_6a) {
      return injectableQuery(userid_6a);
      // The answer: Smith' union select userid,user_name, password,cookie,
   }

   public AttackResult injectableQuery(String accountName) {
      String query = "";
      try (Connection connection = dataSource.getConnection()) {
         boolean usedUnion = true;
         query = "SELECT * FROM user_data WHERE last_name = " + "?";
         //Check if Union is used
         if (!accountName.matches("(?i)(^[^-/*;)]*)(\\s*)UNION(.*$)")) {
            usedUnion = false;
         }
         try (PreparedStatement statement = connection.prepareStatement(
                              ResultSet.CONCUR_READ_ONLY)) {
            statement.setString(1, accountName);
            ResultSet results = statement.executeQuery();

            if ((results != null) && (results.first())) {
                         ResultSetMetaData resultsMetaData = results.getMeta
                         StringBuffer output = new StringBuffer();
```

Suggested Fix

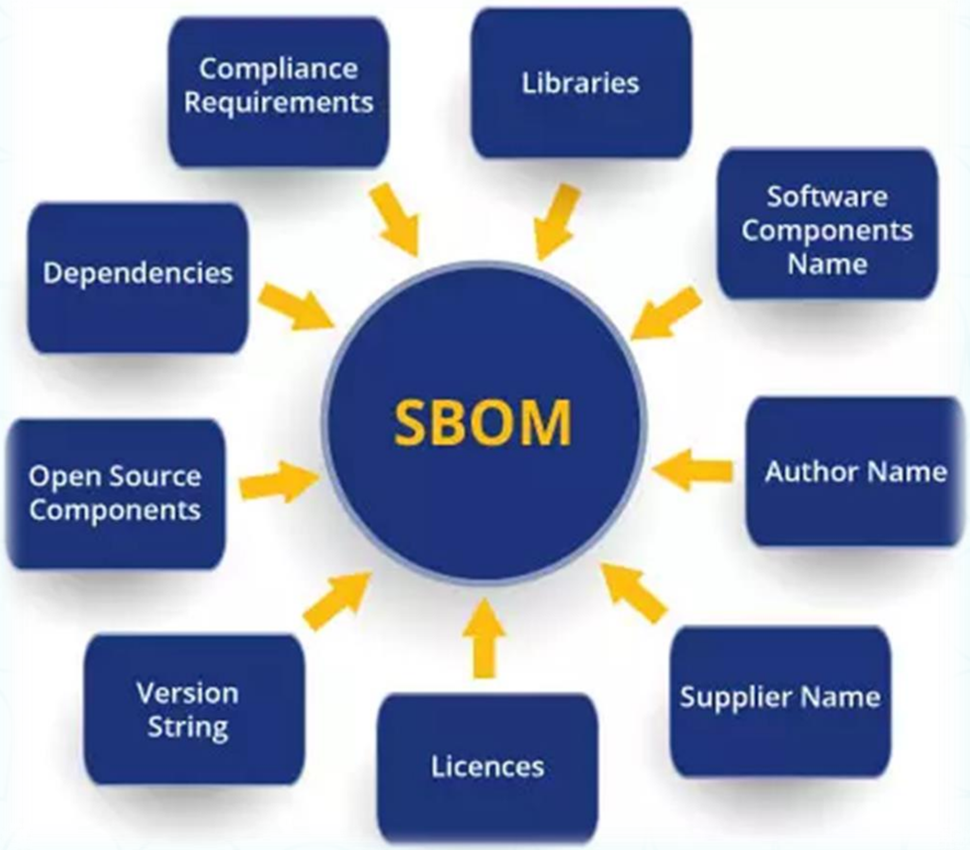
```
import java.sql.*;


@RestController
@AssignmentHints(value = {"SqlStringInjectionHint-advanced-6a-1", "SqlSt
     "SqlStringInjectionHint-advanced-6a-4", "SqlStringInjectionHint-advar
public class SqlInjectionLesson6a extends AssignmentEndpoint {

   private final LessonDataSource dataSource;

   public SqlInjectionLesson6a(LessonDataSource dataSource) {
      this.dataSource = dataSource;
   }

   @PostMapping("/SqlInjectionAdvanced/attack6a")
   @ResponseBody
   public AttackResult completed(@RequestParam String userid_6a) {
      return injectableQuery(userid_6a);
      // The answer: Smith' union select userid,user_name, password,cookie,
   }

   public AttackResult injectableQuery(String accountName) {
      String query = "";
      try (Connection connection = dataSource.getConnection()) {
         boolean usedUnion = true;
         query = "SELECT * FROM user_data WHERE last_name = " + "?";
         //Check if Union is used
         if (!accountName.matches("(?i)(^[^-/*;)]*)(\\s*)UNION(.*$)")) {
            usedUnion = false;
         }
         try (PreparedStatement statement = connection.prepareStatement(
                              ResultSet.CONCUR_READ_ONLY)) {
            statement.setString(1, accountName);
            ResultSet results = statement.executeQuery();

            if ((results != null) && (results.first())) {
                         ResultSetMetaData resultsMetaData = results.getMeta
                         StringBuffer output = new StringBuffer();
```

Merge Confidence



Version	Age	Tests	Confidence
8.1.3	age 18 days	passing 100%	confidence high
8.1.4	age 17 days	passing 98%	confidence high
8.1.5	age 4 days	passing 74%	confidence low
8.1.6	age 4 days	passing 95%	confidence neutral

Software Bill of Materials



logic technology

Adopting OSS Management: 5 Best Practices

1. **Preparation and planning to maximize visibility**

2. **Effective branch strategies**

3. **Don’t just shift left. Shift smart.**

4. **Policies for automated enforcement**

5. **Secure Software Supply Chain**

logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

"With closed source software, you're trusting that the company who wrote it isn't evil. With open source software, you can see for yourself." - Bruce Schneier - Cryptographer

Let’s make your open source security stronger and more resilient!

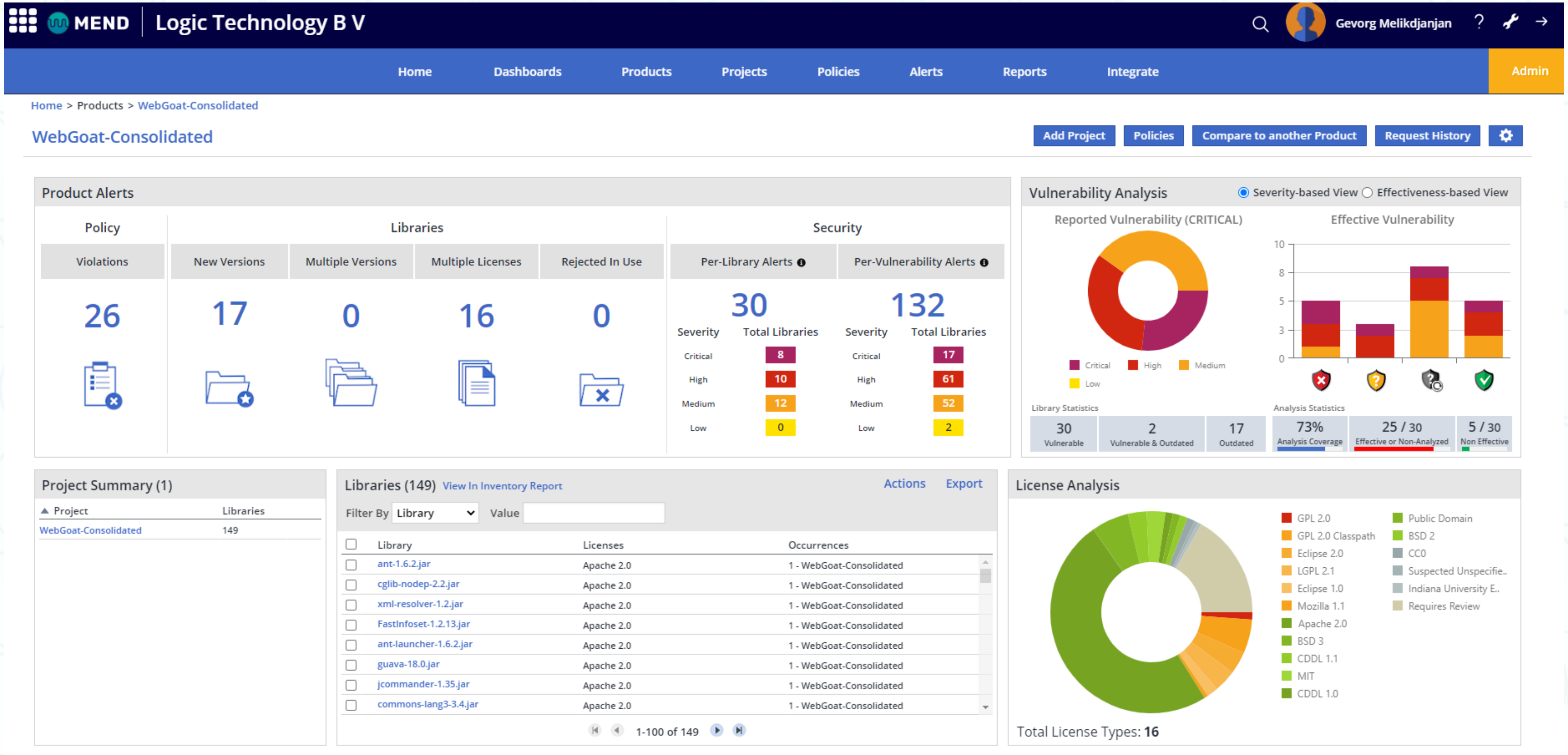
Gevorg Melikdjanjan – Logic Technology

g.melikdjanjan@logic.nl

logic technology

D&E
EVENT
Het ontwerpen van innovatieve elektronica
Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch

Visit our MEND Application Security Demo



Home > Products > WebGoat-Consolidated

WebGoat-Consolidated

Product Alerts

Policy	Libraries				Security	
Violations	New Versions	Multiple Versions	Multiple Licenses	Rejected In Use	Per-Library Alerts 🛈	Per-Vulnerability Alerts 🛈
26	17	0	16	0	30	132

Severity	Total Libraries	Severity	Total Libraries
Critical	8	Critical	17
High	10	High	61
Medium	12	Medium	52
Low	0	Low	2

Vulnerability Analysis

● Severity-based View ○ Effectiveness-based View

Library Statistics

30	2	17
Vulnerable	Vulnerable & Outdated	Outdated

Analysis Statistics

73%	25 / 30	5 / 30
Analysis Coverage	Effective or Non-Analyzed	Non Effective

Project Summary (1)

▲ Project	Libraries
WebGoat-Consolidated	149

Libraries (149) View In Inventory Report

Actions Export

Filter By: Library Value:

□	Library	Licenses	Occurrences
□	ant-1.6.2.jar	Apache 2.0	1 - WebGoat-Consolidated
□	cglib-nodep-2.2.jar	Apache 2.0	1 - WebGoat-Consolidated
□	xml-resolver-1.2.jar	Apache 2.0	1 - WebGoat-Consolidated
□	FastInfoset-1.2.13.jar	Apache 2.0	1 - WebGoat-Consolidated
□	ant-launcher-1.6.2.jar	Apache 2.0	1 - WebGoat-Consolidated
□	guava-18.0.jar	Apache 2.0	1 - WebGoat-Consolidated
□	jcommander-1.35.jar	Apache 2.0	1 - WebGoat-Consolidated
□	commons-lang3-3.4.jar	Apache 2.0	1 - WebGoat-Consolidated

1-100 of 149

License Analysis

Total License Types: **16**

logic technology

D&E EVENT

Het ontwerpen van innovatieve elektronica

Woensdag 19 april 2023
1931 Congrescentrum 's-Hertogenbosch