Your SCA Is Slowing You Down

Your existing SCA delivers a limited analysis, missing the mark on contextual risk analysis, automated risk severity grading and prioritization.

Malicious Code Blindness

Open-source packages and systems are your weakest link because **traditional SCA is blind to the malicious code they can deliver.**

Alert Fatigue

Irrelevant and non-actionable alerts your SCA generates **lack context and proof of usage in your application.**

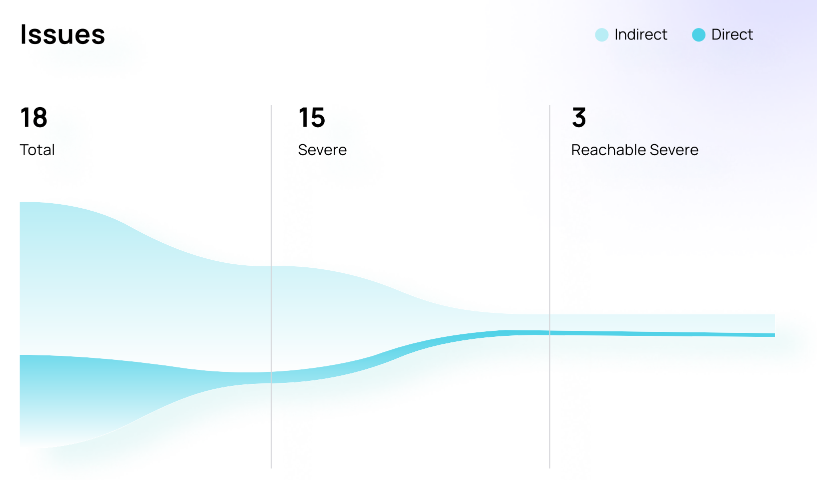
Remediation Guesswork

When your team cannot present the engineering organization with a concrete remediation plan, **developer's time is wasted on fighting the wrong battles.**

Detect Attacks, Prioritize Reachable Vulnerabilities

Myrror helps you detect a variety of supply chain attacks, prioritize the risk, and act decisively with proprietary, multi-dimensional SCA engines.

- Software Supply Chain Attack Detection
- Vulnerability detection
- Vulnerability Prioritization
- Optimal Remediation Generator
- SBOM



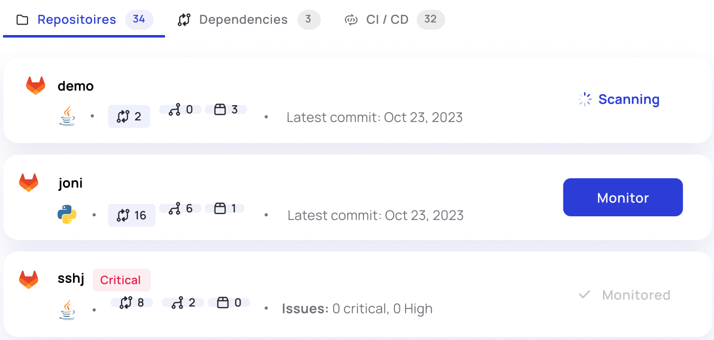
Focus on Your Most Pressing Threats

How It Works

1

Discover Your Assets (SBOM)

Gain ongoing visibility into your development repositories, open-source packages and CI/CD tools.

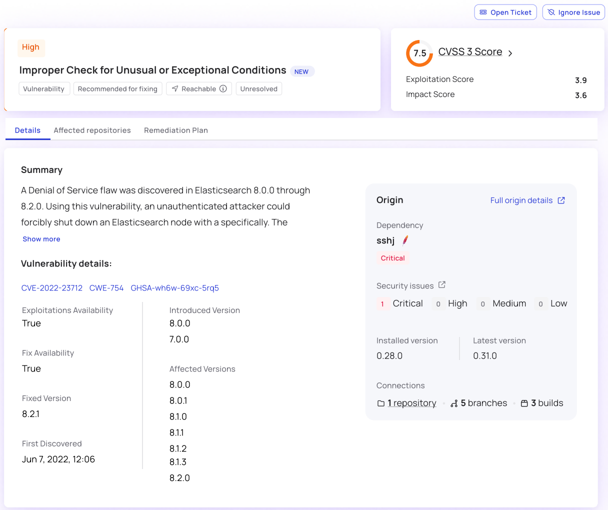


Repositoires 34 | Dependencies 3 | CI / CD 32

demo — 2 · 0 · 3 · Latest commit: Oct 23, 2023 — Scanning

joni — 16 · 6 · 1 · Latest commit: Oct 23, 2023 — Monitor

sshj Critical — 8 · 2 · 0 · Issues: 0 critical, 0 High — Monitored



Open Ticket | Ignore Issue

High

Improper Check for Unusual or Exceptional Conditions NEW

Vulnerability | Recommended for fixing | Reachable | Unresolved

7.5 CVSS 3 Score ›

Exploitation Score	3.9
Impact Score	3.6

Details | Affected repositories | Remediation Plan

Summary

A Denial of Service flaw was discovered in Elasticsearch 8.0.0 through 8.2.0. Using this vulnerability, an unauthenticated attacker could forcibly shut down an Elasticsearch node with a specifically. The

Show more

Vulnerability details:

CVE-2022-23712 CWE-754 GHSA-wh6w-69xc-5rq5

Exploitations Availability
True

Fix Availability
True

Fixed Version
8.2.1

First Discovered
Jun 7, 2022, 12:06

Introduced Version
8.0.0
7.0.0

Affected Versions
8.0.0
8.0.1
8.1.0
8.1.1
8.1.2
8.1.3
8.2.0

Origin — Full origin details

Dependency
sshj
Critical

Security issues
1 Critical 0 High 0 Medium 0 Low

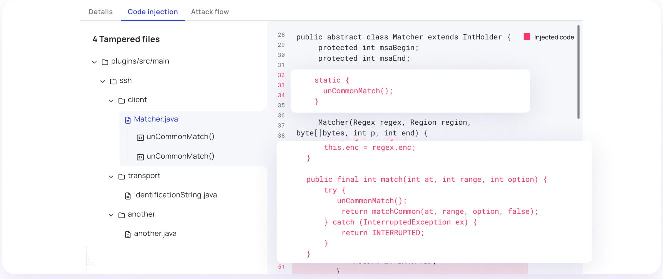
Installed version: 0.28.0
Latest version: 0.31.0

Connections
1 repository · 5 branches · 3 builds

2

Detection Engines

Expose vulnerabilities, malicious dependencies, trojans, and supply chain attacks in both your open-source and your own builds - before they hit production.



Details | Code injection | Attack flow

4 Tampered files

- plugins/src/main
  - ssh
    - client
      - Matcher.java
        - unCommonMatch()
        - unCommonMatch()
    - transport
      - IdentificationString.java
    - another
      - another.java

Injected code

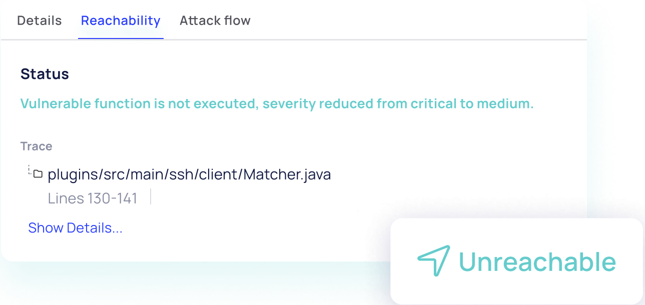
```
public abstract class Matcher extends IntHolder {
    protected int msaBegin;
    protected int msaEnd;

  static {
    unCommonMatch();
  }

    Matcher(Regex regex, Region region,
byte[]bytes, int p, int end) {
      this.enc = regex.enc;
  }

  public final int match(int at, int range, int option) {
      try {
        unCommonMatch();
         return matchCommon(at, range, option, false);
      } catch (InterruptedException ex) {
         return INTERRUPTED;
      }
  }
```

3

Prioritization Engine

Combine CVSS, EPSS & Our Own proprietary static reachability analysis to understand the code and package context.

Focus only on functions that might actually get executed in practice.



Details | Reachability | Attack flow

Status

Vulnerable function is not executed, severity reduced from critical to medium.

Trace

plugins/src/main/ssh/client/Matcher.java

Lines 130-141

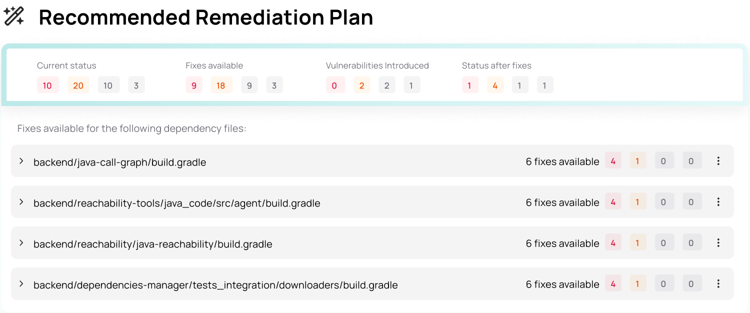
Show Details...

Unreachable

4

Remediation Plan Generator

Reduce MTTR using an actionable mitigation plan that accounts for both existing and newly-introduced risks, and suggests the optimal path to every scenario.



Recommended Remediation Plan

Current status	Fixes available	Vulnerabilities Introduced	Status after fixes
10 20 10 3	9 18 9 3	0 2 2 1	1 4 1 1

Fixes available for the following dependency files:

backend/java-call-graph/build.gradle	6 fixes available	4	1	0	0
backend/reachability-tools/java_code/src/agent/build.gradle	6 fixes available	4	1	0	0
backend/reachability/java-reachability/build.gradle	6 fixes available	4	1	0	0
backend/dependencies-manager/tests_integration/downloaders/build.gradle	6 fixes available	4	1	0	0

Integrations

Language support

Java

C#

JS/TS

Python

C/C++ Upcoming!

Connect Your SCM in 5 minutes

And growing...

Security Driven

As an SDLC Security Solution, Myrror's priority is to maintain a safe and secure environment for its service provision.
To ensure the highest level of security, we continually invest in our overall information security program, resources, and expertise.
As a security service provider, we understand the importance of providing clear information about our security practices, tools, resources, and responsibilities, so that our customers can feel confident in choosing us as a trusted service provider.

We use Amazon Web Services (AWS) Data Centers, and our environments and services uses SSO+MFA and role-based (RBAC) security architecture and requires users of the systems to be identified and authenticated prior to the use of any system resources.

We are SOC2 Compliant. Myrror undergoes a SOC 2 Type 2 Audit on an annual basis.

Myrror transmits data over public networks using strong encryption. This includes data transmitted between BlindSpot's clients and the BlindSpot service. We support the latest recommended secure cipher suites to encrypt all traffic in transit, including the use of TLS protocols, encryption, and hashing algorithms, as supported by the clients. This also applies to all types of data at rest.

Myrror assesses the security risk of each software development project according to our Secure Development Lifecycle.
Before completion of the design phase, we undertake an assessment to qualify the security risk of the software changes introduced. All code is checked into a version-controlled repository. Code changes are subject to peer review and continuous integration testing. Of course, we use our platform on our own services.

Testing and staging environments are logically separated from the Production environment. No Production Data is used in our development or test environments.