

# OBSERVABILITY OF INACTIVE CODE

HIGHLIGHTS FROM THE 2021 STATE OF OPEN-SOURCE SECURITY REPORT

## NOT ALL LIBRARIES IN AN APPLICATION ARE USED!

**62%**

OF ALL LIBRARIES PRESENT IN APPLICATIONS ARE **INACTIVE** (76% FOR NODE, 58% FOR JAVA, 33% FOR .NET).

## AND NOT ALL PARTS OF ACTIVE LIBRARIES ARE INVOKED

**69%**

OF ALL CLASSES IN ACTIVE LIBRARIES ARE **INACTIVE** (95% FOR NODE, 68% FOR JAVA, 33% FOR .NET).

## NOT EVERY LIBRARY POSES THE SAME LEVEL OF RISK

**25**

ALL OF THE TOP 25 JAVA LIBRARIES ARE **PRESENT** IN A MAJORITY OF APPLICATIONS. HOWEVER, ONLY 12 OF THE 25 ARE **ACTIVE** IN FEWER THAN HALF OF APPLICATIONS.

**49%**

OF .NET APPLICATIONS HAVE JUST ONE **ACTIVE LIBRARY**.

**90+%**

TOP 25 NODE LIBRARIES ARE PRESENT IN **90+%** OF APPLICATIONS. HOWEVER, MOST COMMON LIBRARY IS ONLY IN **42%** OF APPLICATIONS.

## KEY TAKEAWAYS

VULNERABILITIES IN INACTIVE LIBRARIES AND CLASSES POSE NO RISK.

TRADITIONAL SCA TOOLS **DO NOT DIFFERENTIATE ACTIVE VS. INACTIVE CODE.**

OBSERVABILITY IS KEY TO PRIORITIZING REMEDIATION FOR SECURITY AND EFFICIENCY.

CONTRAST OSS PROVIDES THE OBSERVABILITY AND DEEP INSIGHTS NEEDED TO **PINPOINT WHICH OPEN-SOURCE VULNERABILITIES POSE RISK—AND WHICH ONES DO NOT.**

GET THE FULL 2021 STATE OF OPEN-SOURCE SECURITY REPORT TODAY.