RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **PS-W02**

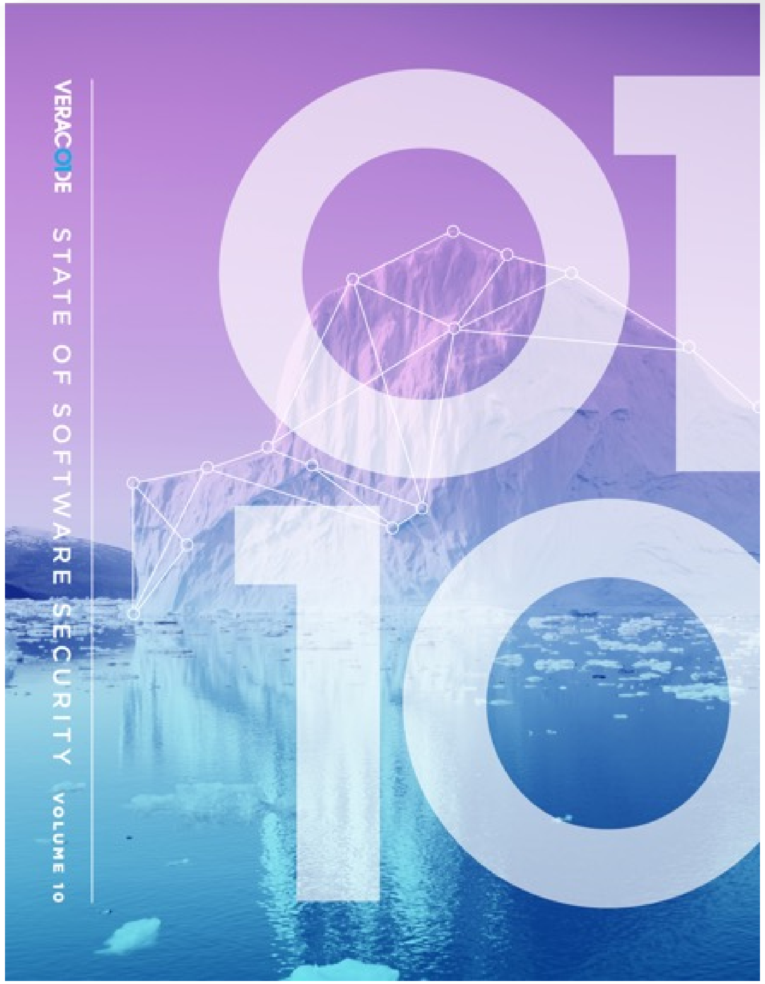
# 8 Million Findings in One Year: Fresh Look at the State of Software Security

**Chris Wysopal**

Co-Founder and CTO
Veracode
@WeldPond

**Jay Jacobs**

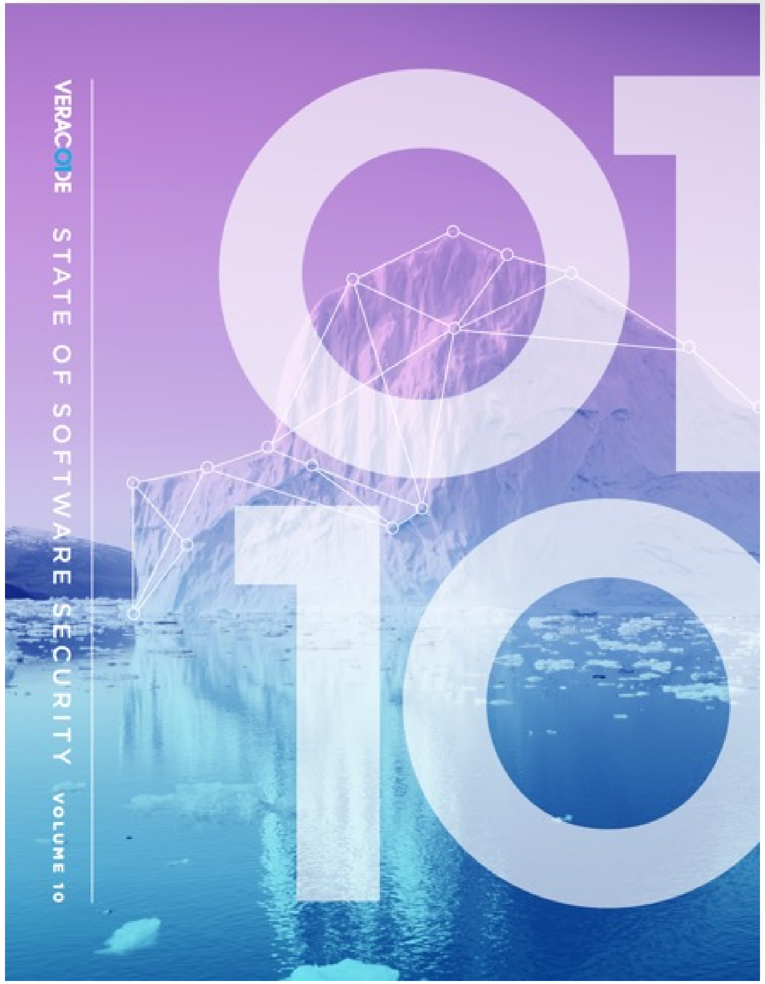
Co-Founder and Chief Data Scientist
Cyentia Institute
@jayjacobs



# What is this research?

- Veracode State of Software Security (SoSS), Vol. 10
- Largest quantitative study of application security findings
- Partnered with data scientists at Cyentia Institute

State of Software Security Report
State of Software Security Report
State of Software Security Report
State of Software Security Report
State of Software Security Report
STATE OF SOFTWARE SECURITY
STATE OF SOFTWARE SECURITY 2016
STATE OF SOFTWARE SECURITY 2017
STATE OF SOFTWARE SECURITY



## The Why:

- Insights into industry performance, and impact of DevSecOps on fix rates
- Provide data for customers to benchmark themselves against their peers
- Generate actionable advice for improving application security programs

## The How:

- Formulate questions that might be answerable given the available data
- Stand back and use science

State of Software Security Report
State of Software Security Report
State of Software Security Report
State of Software Security Report
State of Software Security Report
STATE OF SOFTWARE SECURITY
STATE OF SOFTWARE SECURITY 2016
STATE OF SOFTWARE SECURITY 2017
STATE OF SOFTWARE SECURITY

# The Data…

Vol. 1

1,591

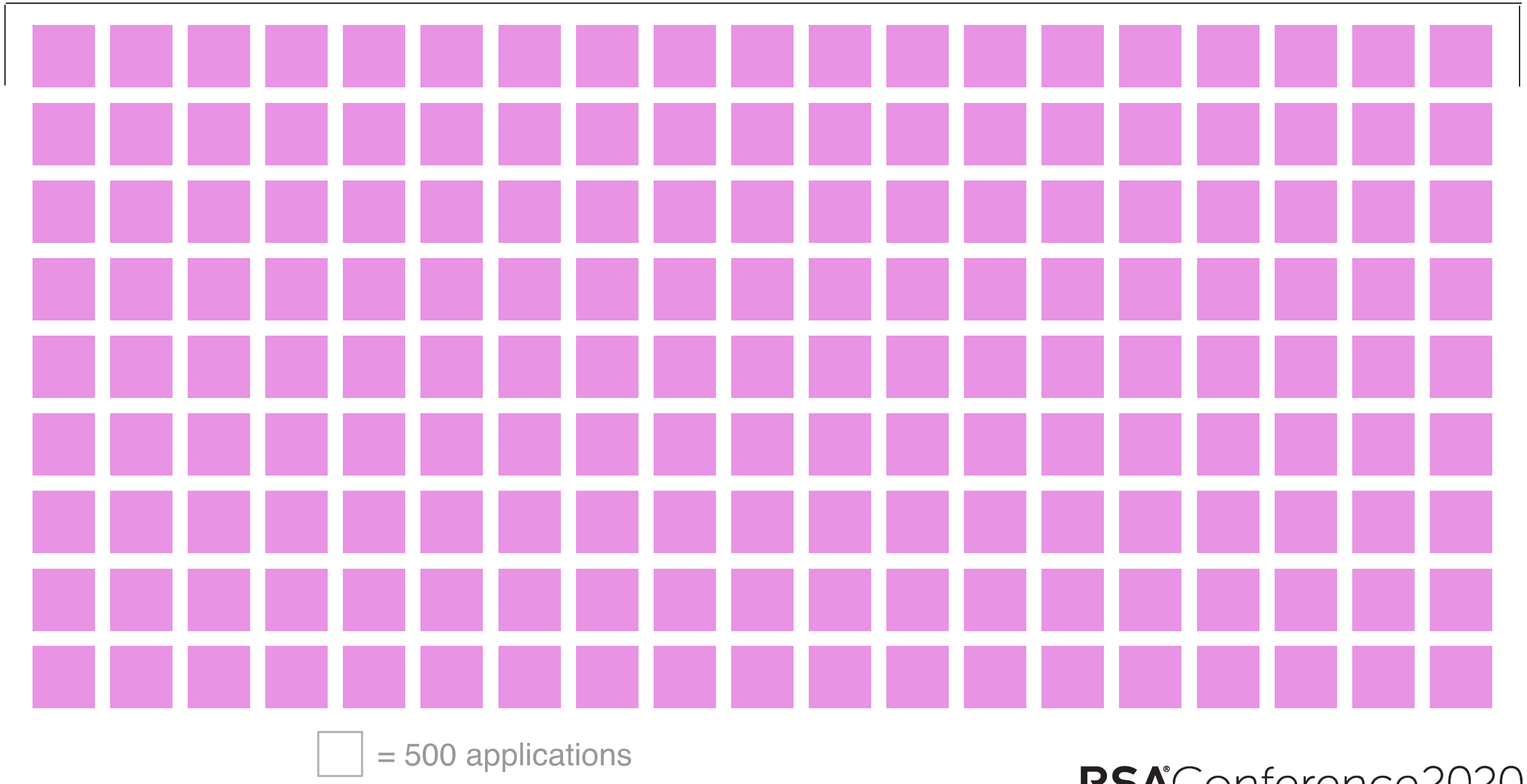
software tested

Vol. 10

85,000+

software tested

That’s over a 50-fold increase in sample size!



Over 2,300 Veracode customers

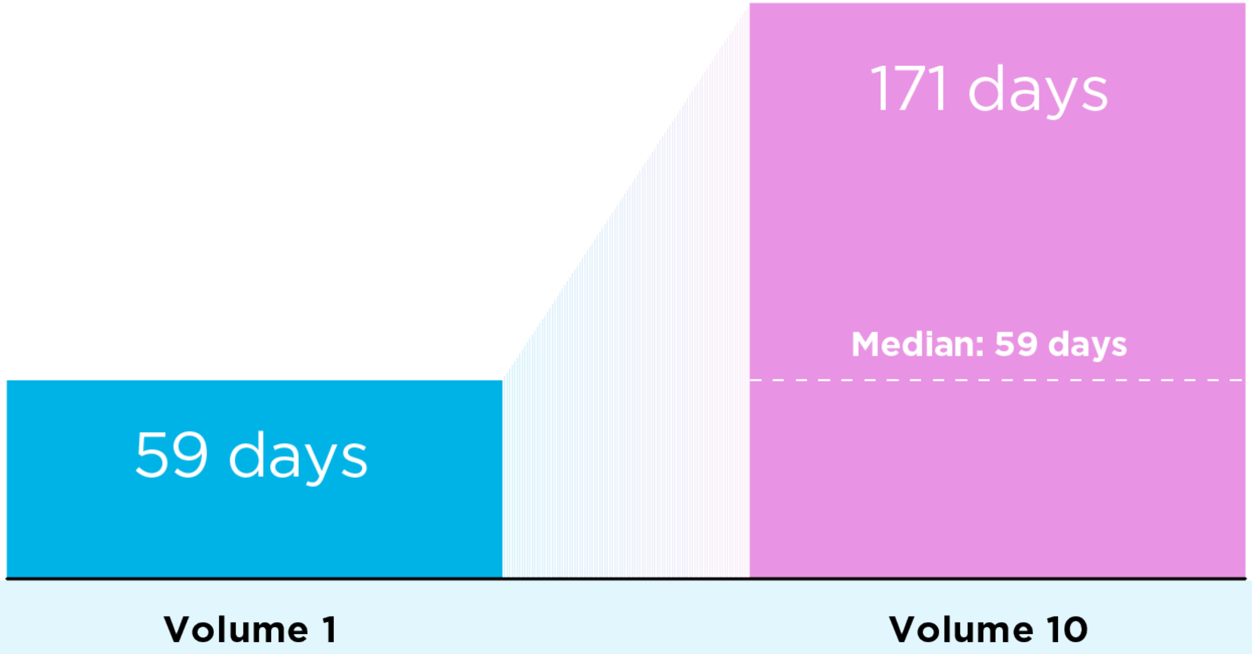
12 months of software scan data: April 1, 2018 – March 31, 2019

Over 85,000 unique pieces of software and 1.4 million individual assessments

VERACODE

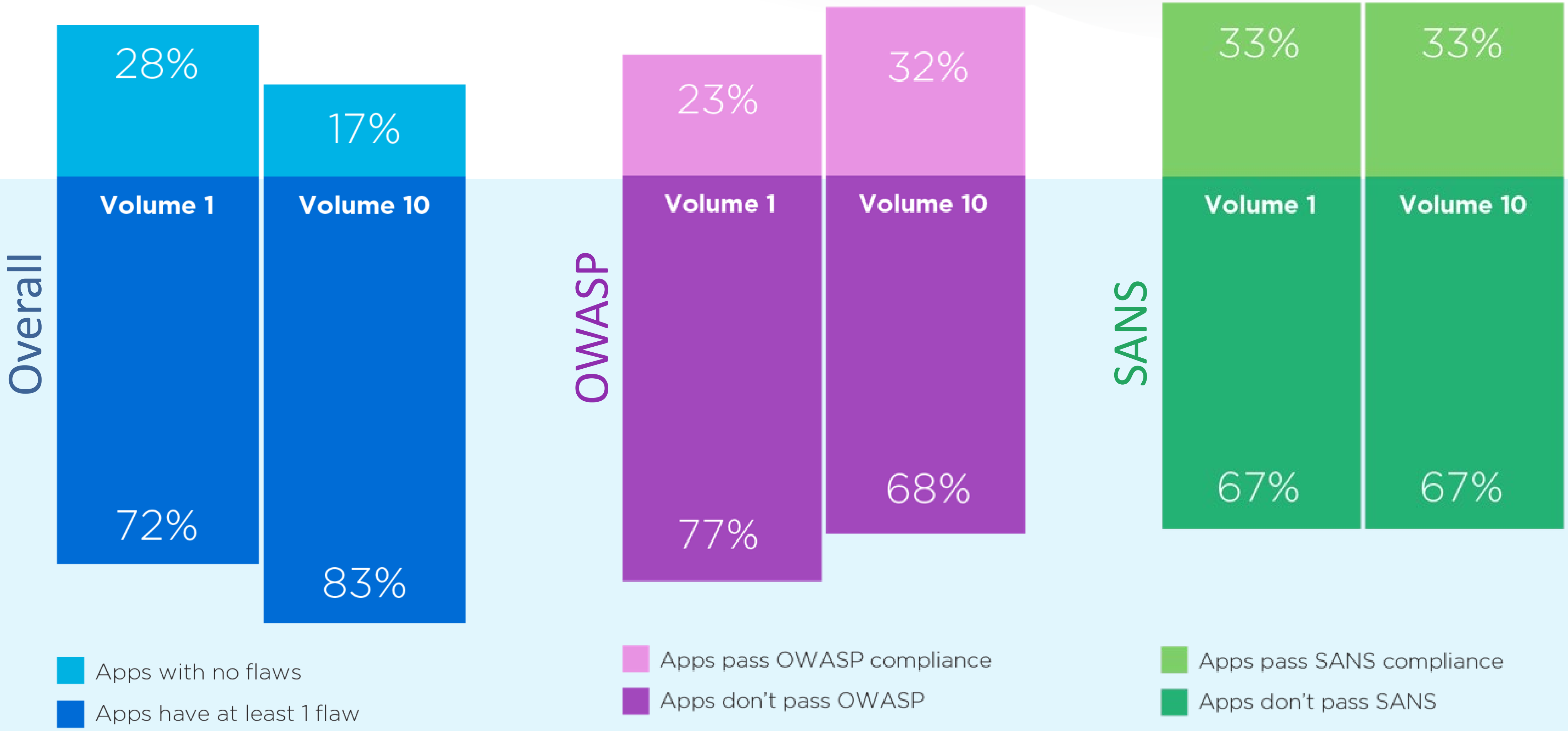
# The State of Software Security

# Mean Time to Remediation among closed findings

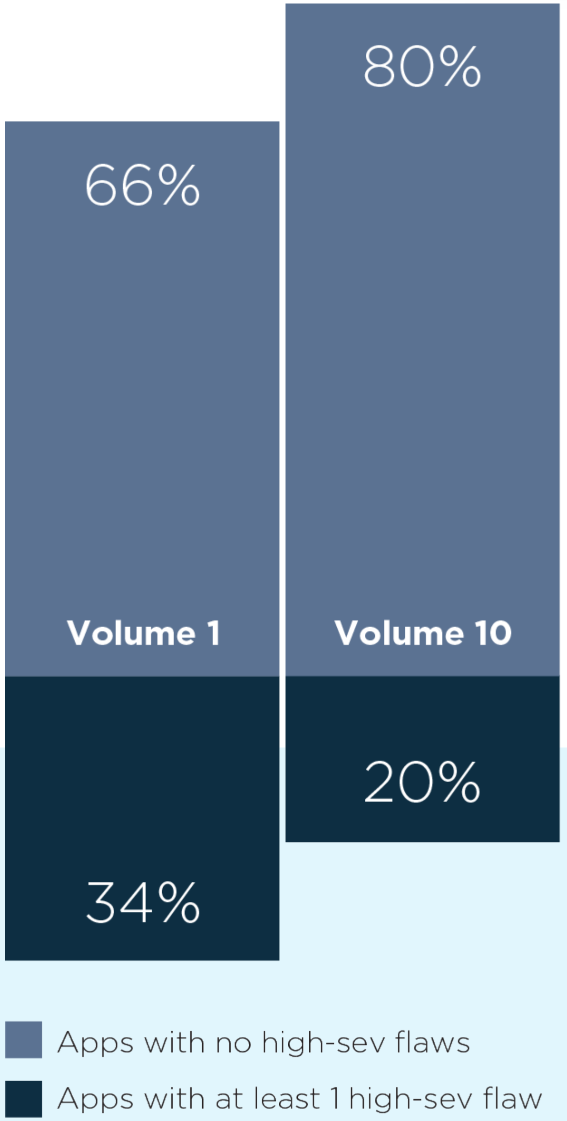


- The median fix time remains relatively unchanged from 10 years ago.
- However, the tail of ever-accruing "security debt" just got a lot longer, causing the mean closed time to stretch out.

# Proportion of software applications/products with at least one flaw in the initial scan

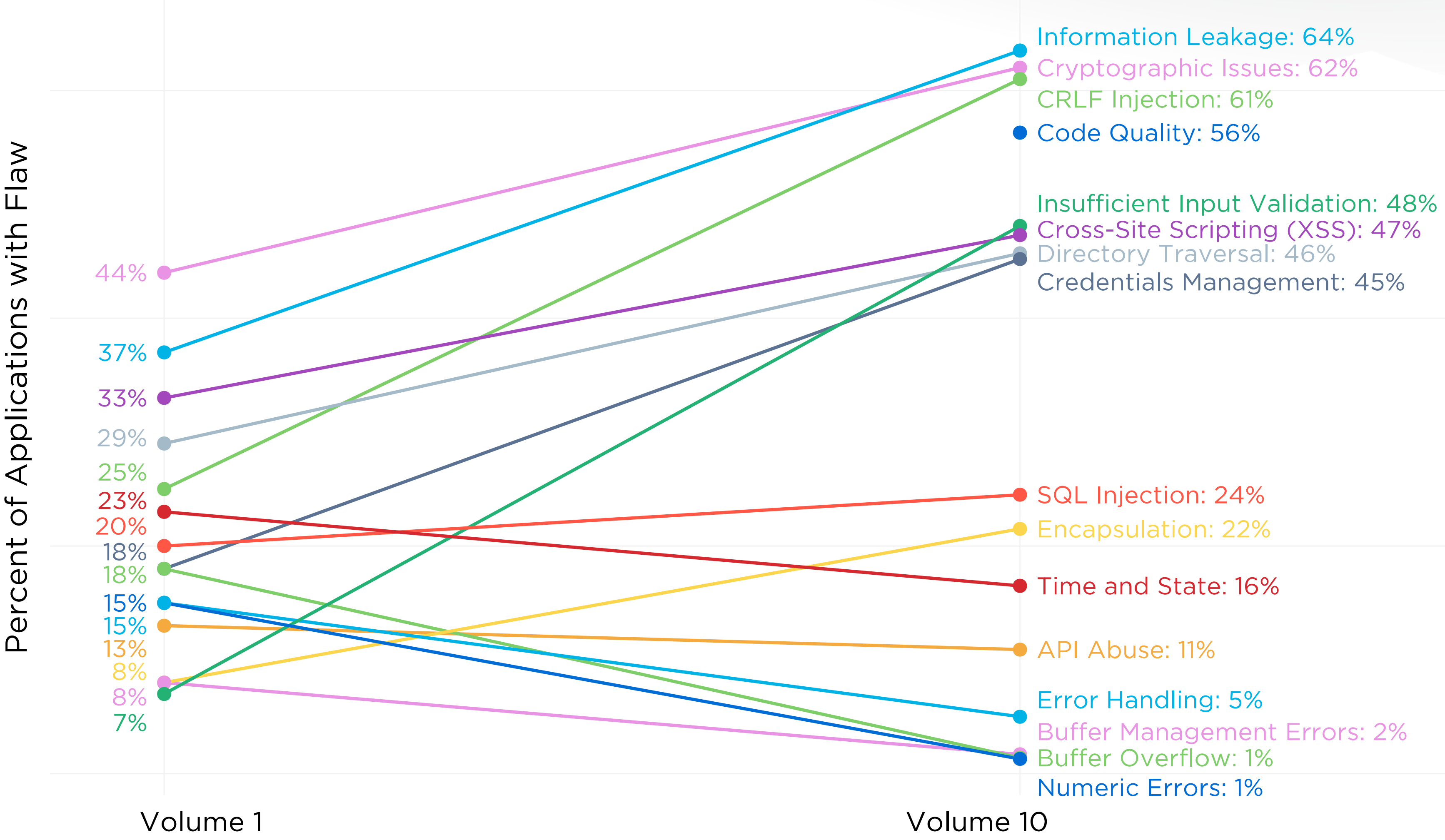


# Software with high-severity flaws



Majority of products/applications are free from high/critical flaws

# Prevalence of flaw categories in SOSS Volume 1 and 10



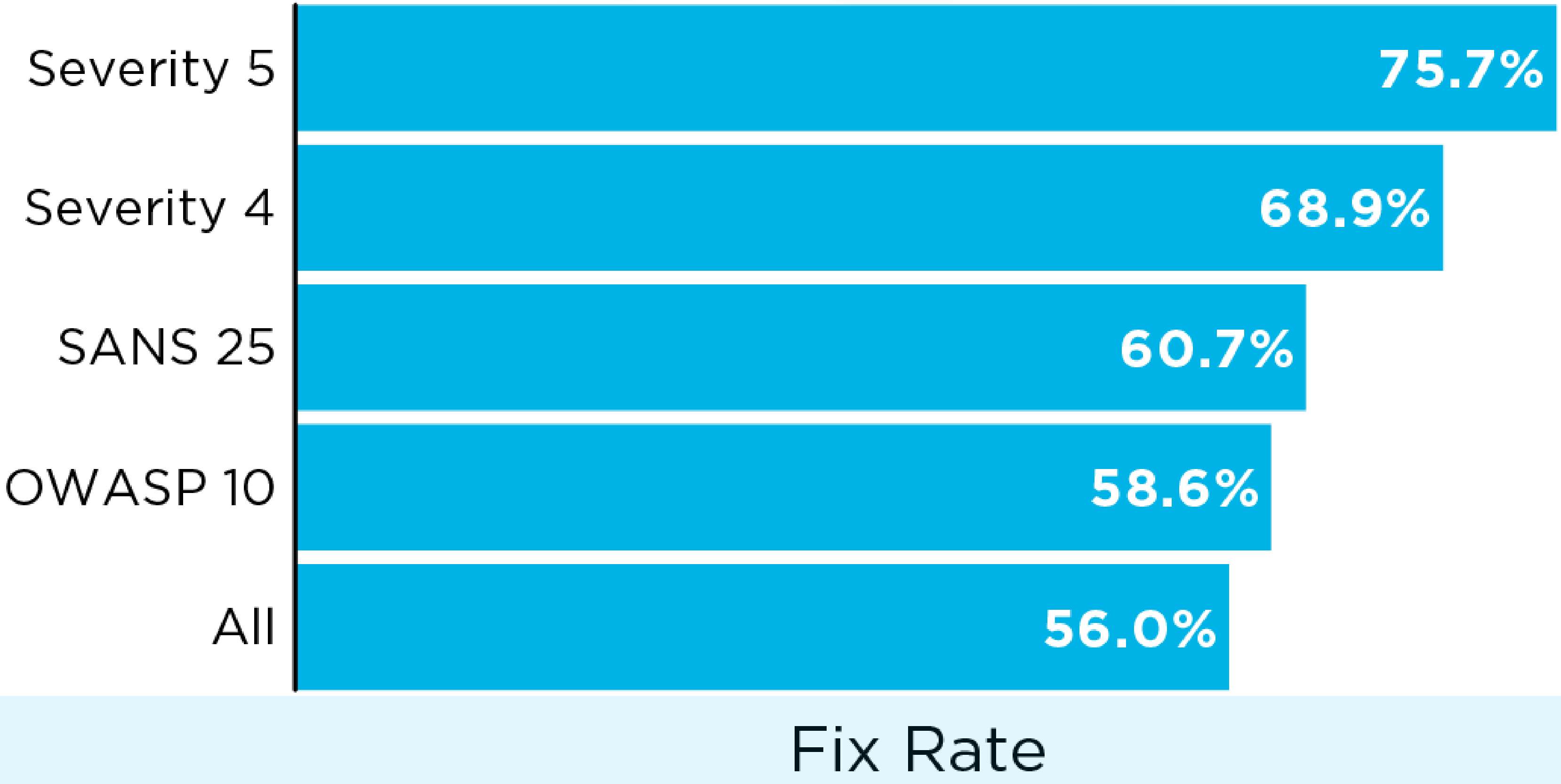
There is a general increase in web-related categories, likely due to a lot more web applications being written.

Less code is being written in C/C++ so buffer overflows, buffer management errors, and numeric errors are way down.

# Fix rate across all flaws

"Fix rate" is the proportion of discovered flaws that are successfully closed or

remediated.

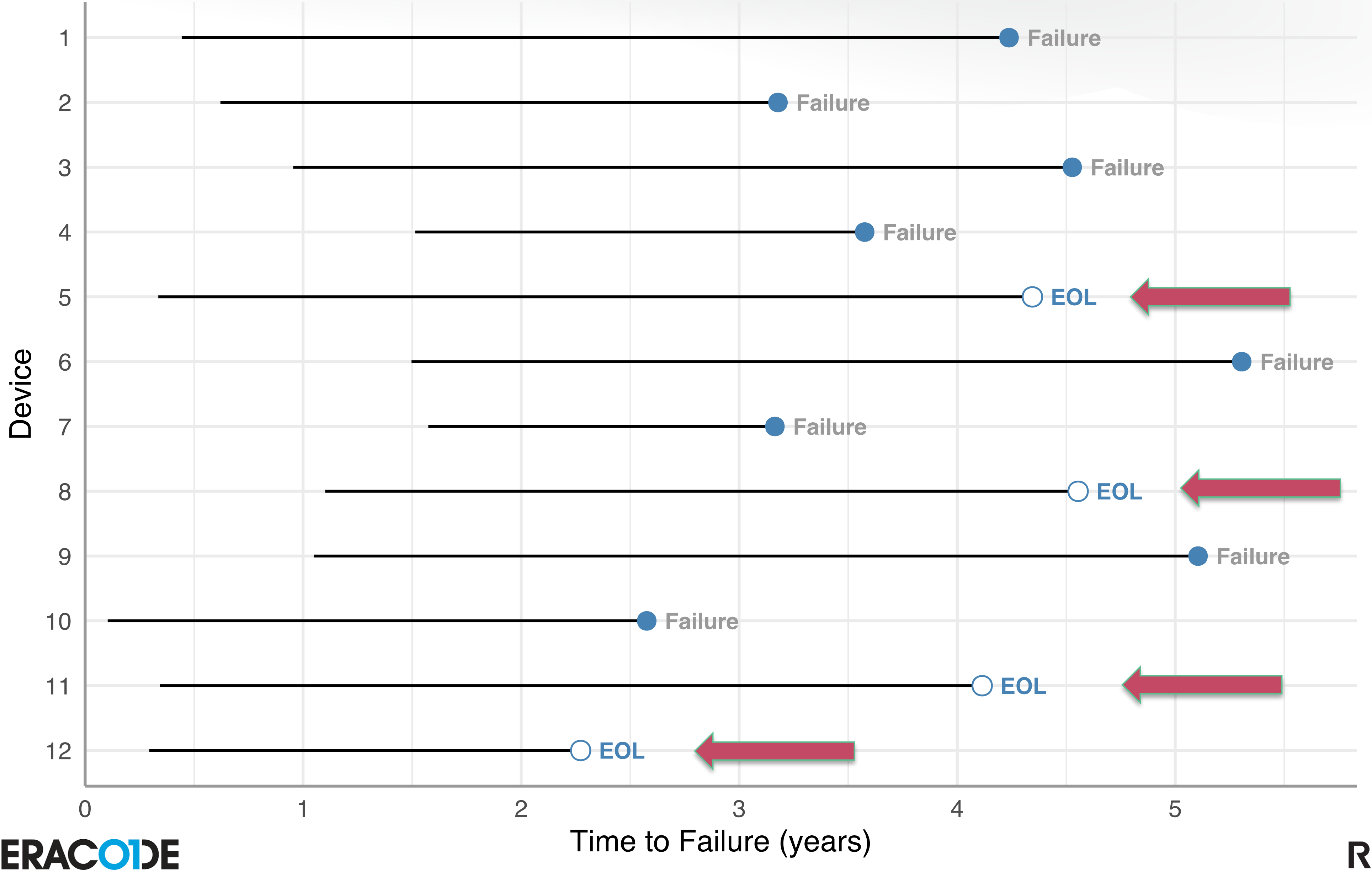


# Fix Behavior

# Measuring time to remediate is challenging…

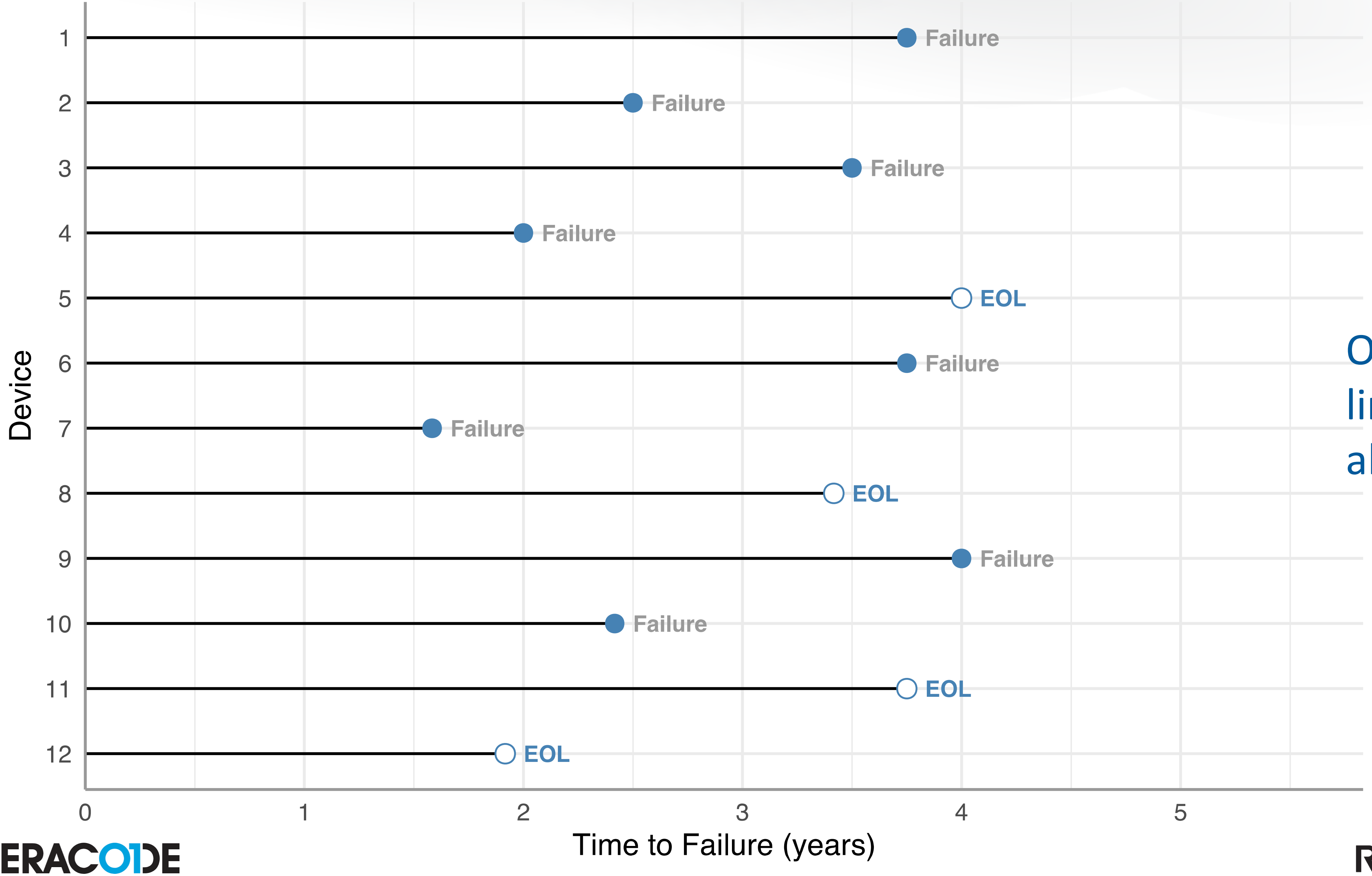
- Simple approach is to calculate time for remediated findings
  - Ignores the still-open (security debt)
  - But it’s simple and intuitive
- Survival analysis studies the time to an event
  - Accounts for findings that are still open (security debt)
    - Team stopped scanning
    - Not closed yet, was still open at last scan

# Time to Failure (example)



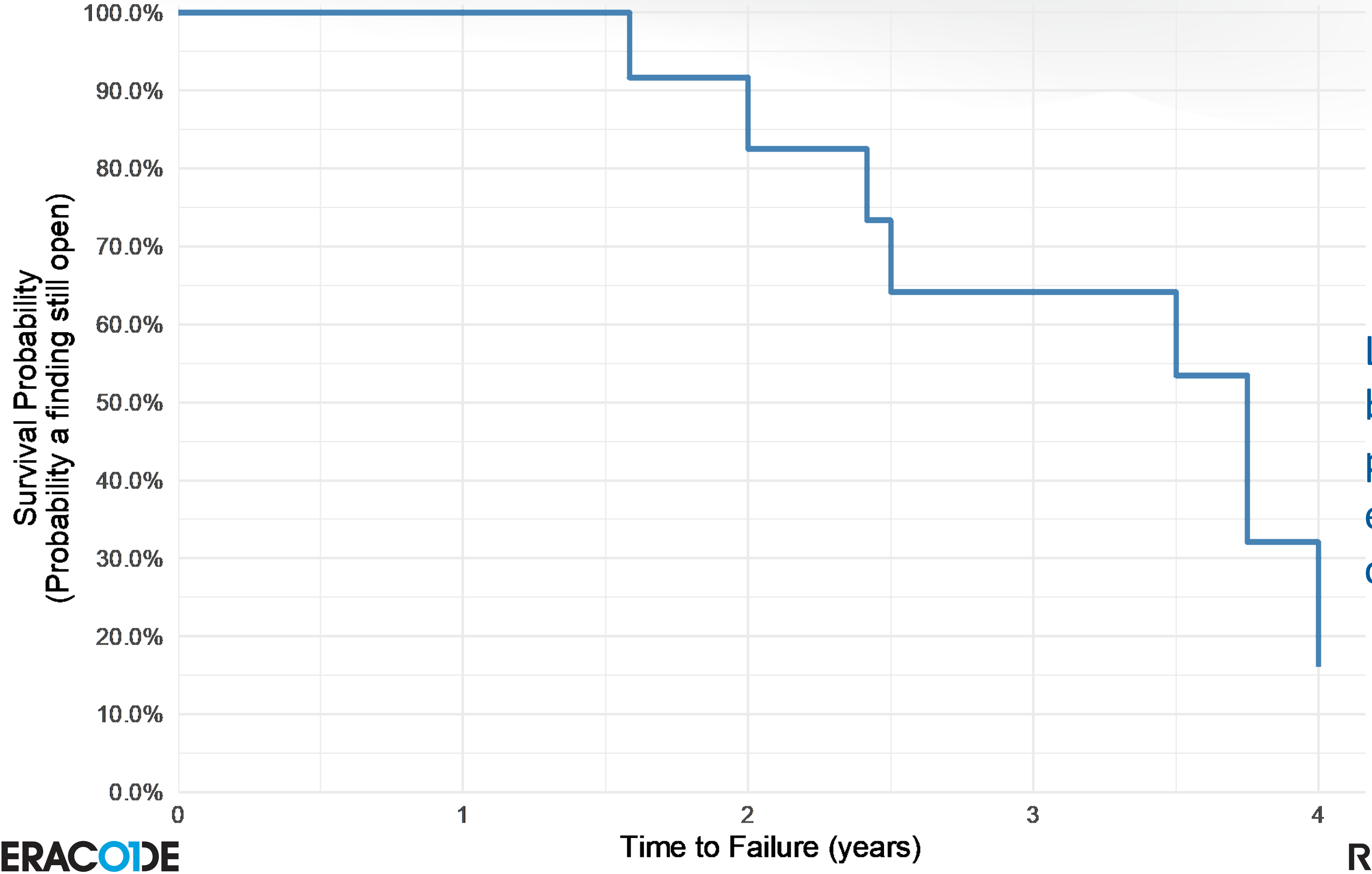
These are "censored" - all we know is they lasted "at least" this long.

# Time to Failure (example)



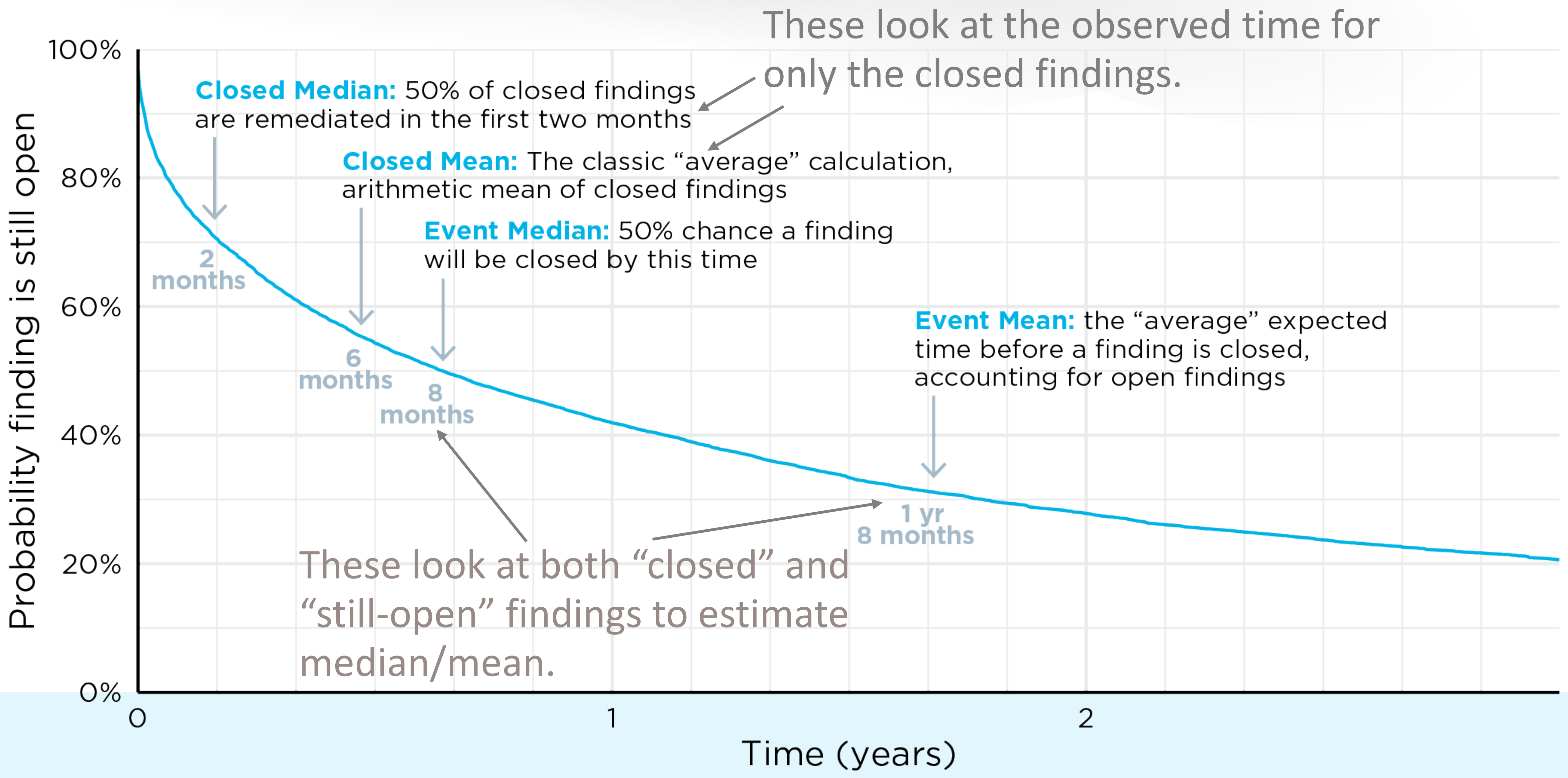
Observations are lined up so they all start on day 0.

# Time to Failure (example)



Line represents best estimate of probability an event hasn't occurred yet.

# Flaw persistence curve



These look at the observed time for only the closed findings.

**Closed Median:** 50% of closed findings are remediated in the first two months

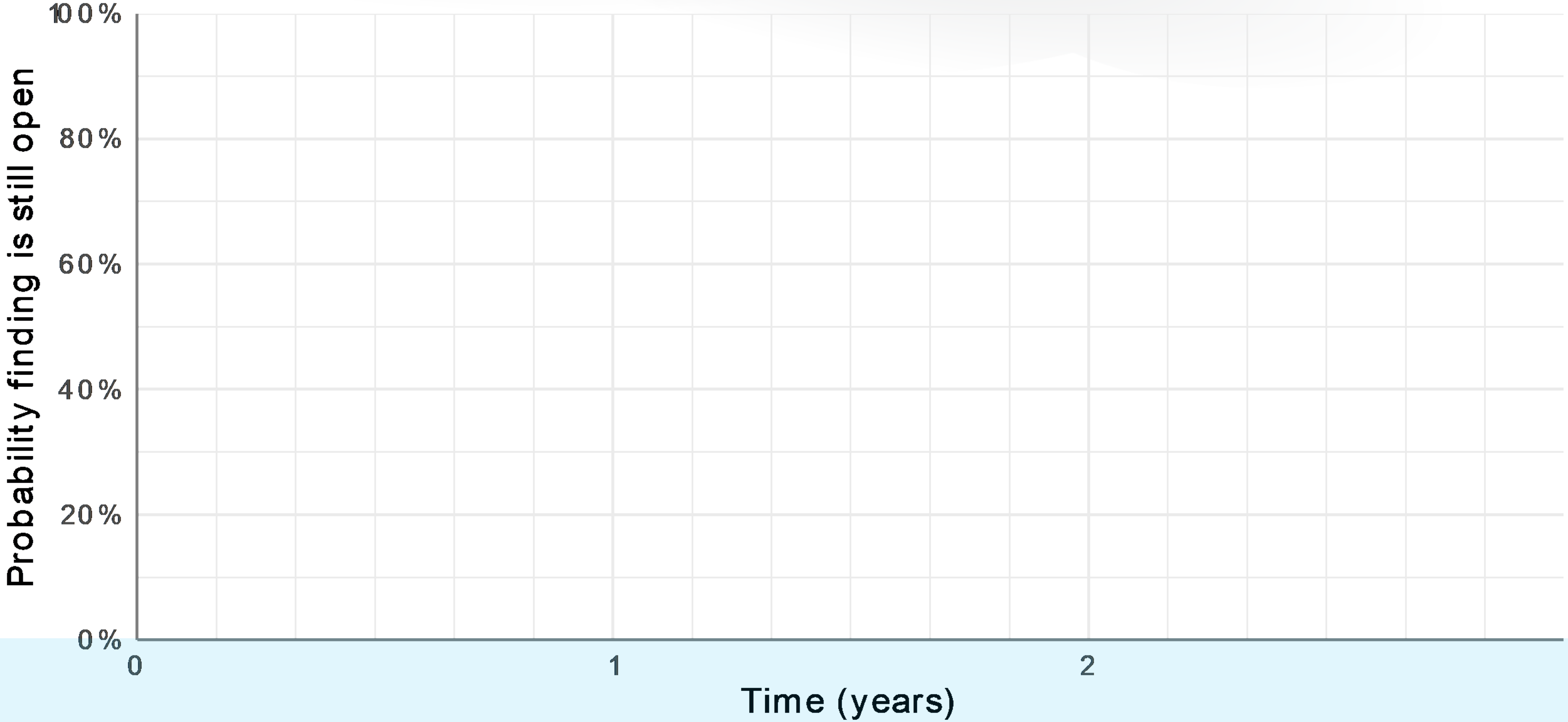
**Closed Mean:** The classic "average" calculation, arithmetic mean of closed findings

**Event Median:** 50% chance a finding will be closed by this time

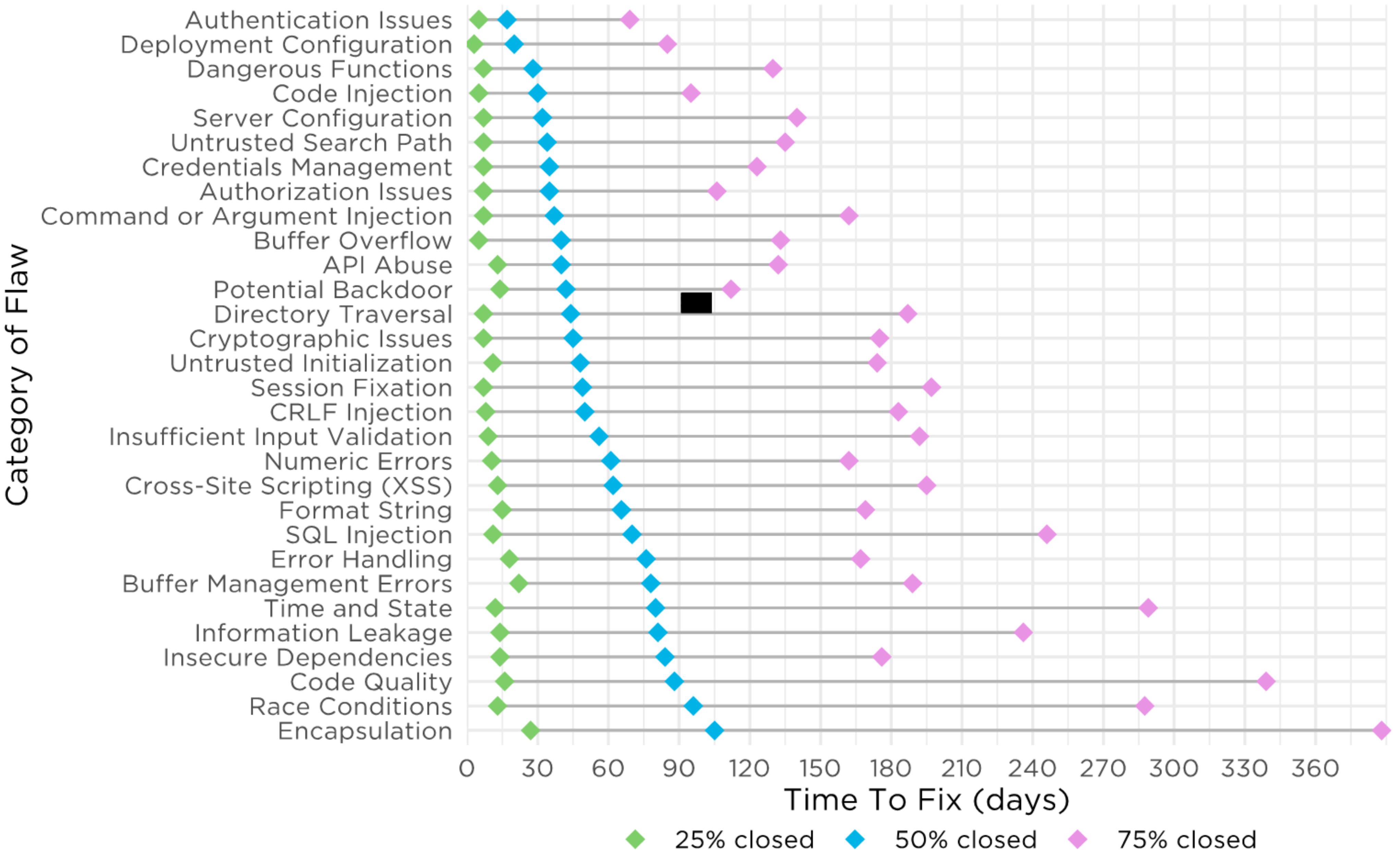
**Event Mean:** the "average" expected time before a finding is closed, accounting for open findings

These look at both "closed" and "still-open" findings to estimate median/mean.

# Flaw persistence curve

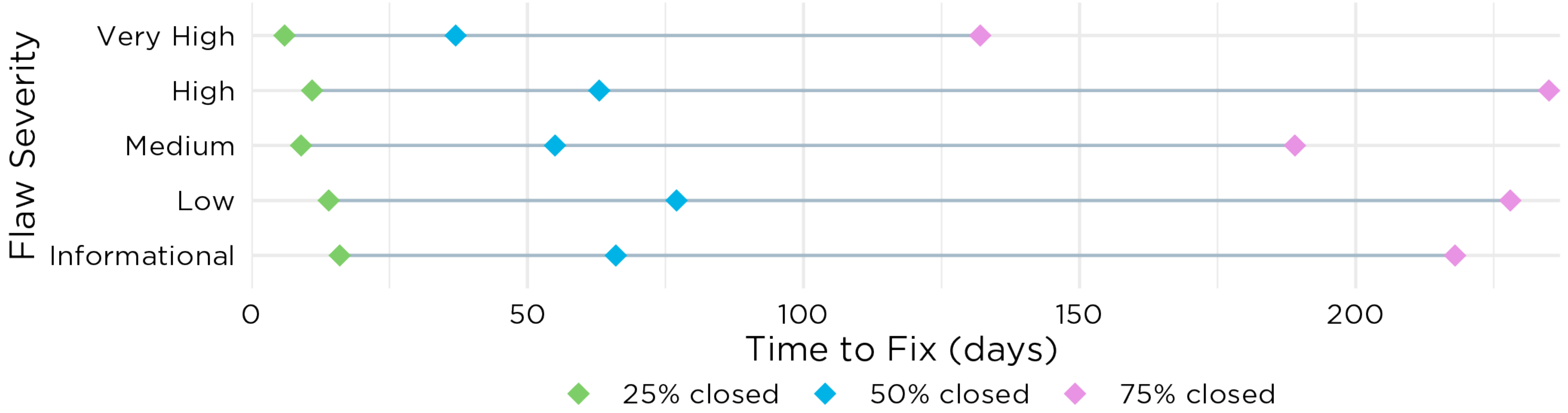


# Median Time-to-remediate across flaw categories



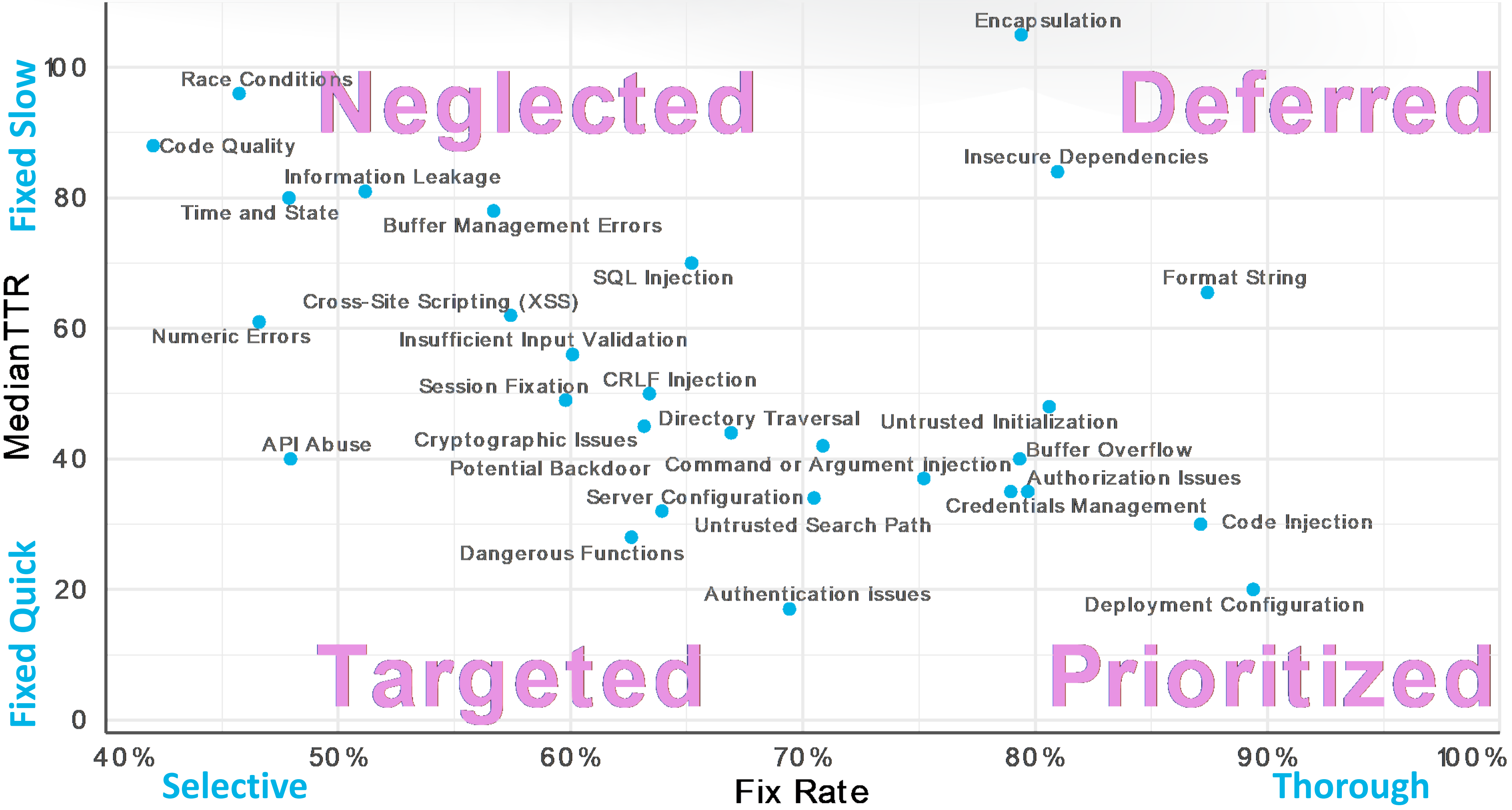
# Time-to-remediation across flaw severity scores

Surprisingly, flaw severity doesn’t correlate strongly with fix speed

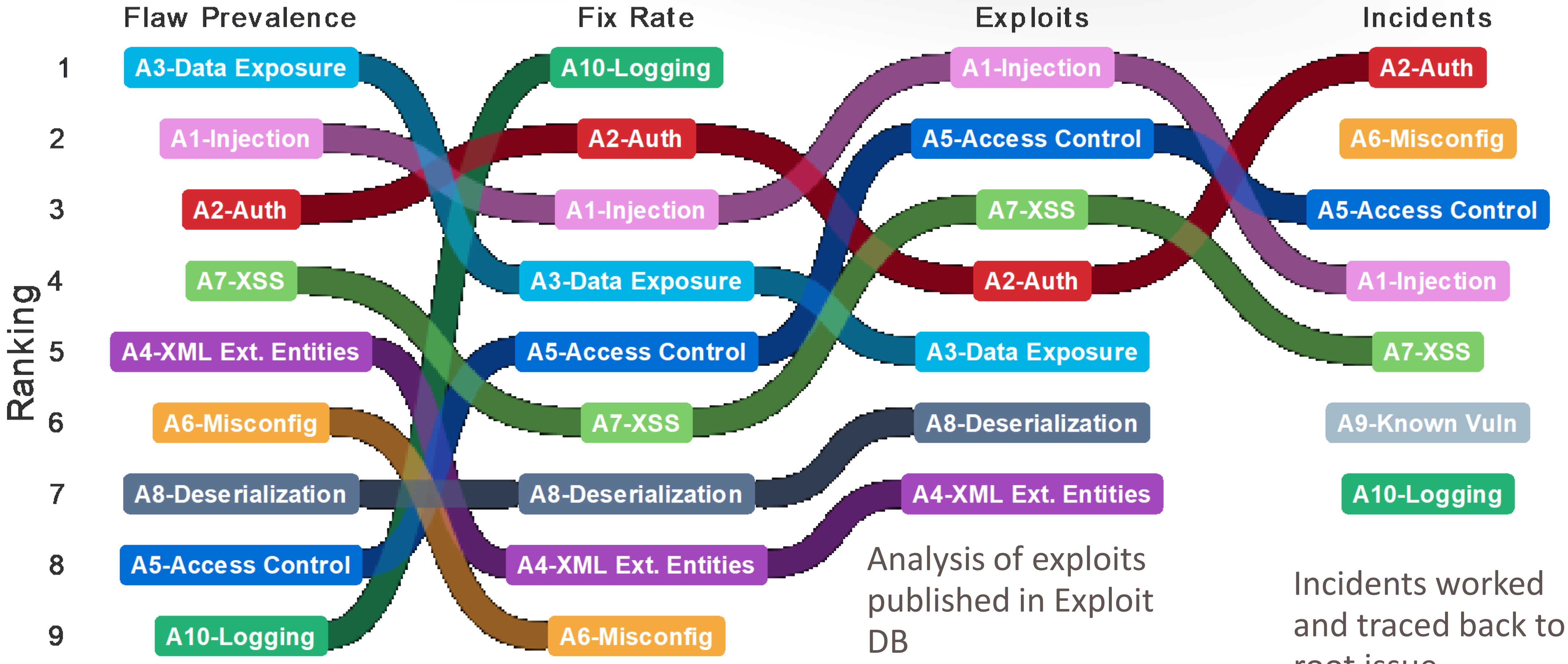


Even the Very High severity flaws have a long tail for fix time, taking over 130 days to reach the 75% closed milestone.

# Speed and comprehensiveness for flaw categories



# OWASP Top 10: Rankings



| Ranking | Flaw Prevalence | Fix Rate | Exploits | Incidents |
|---|---|---|---|---|
| 1 | A3-Data Exposure | A10-Logging | A1-Injection | A2-Auth |
| 2 | A1-Injection | A2-Auth | A5-Access Control | A6-Misconfig |
| 3 | A2-Auth | A1-Injection | A7-XSS | A5-Access Control |
| 4 | A7-XSS | A3-Data Exposure | A2-Auth | A1-Injection |
| 5 | A4-XML Ext. Entities | A5-Access Control | A3-Data Exposure | A7-XSS |
| 6 | A6-Misconfig | A7-XSS | A8-Deserialization | A9-Known Vuln |
| 7 | A8-Deserialization | A8-Deserialization | A4-XML Ext. Entities | A10-Logging |
| 8 | A5-Access Control | A4-XML Ext. Entities | | |
| 9 | A10-Logging | A6-Misconfig | | |

Analysis of exploits published in Exploit DB

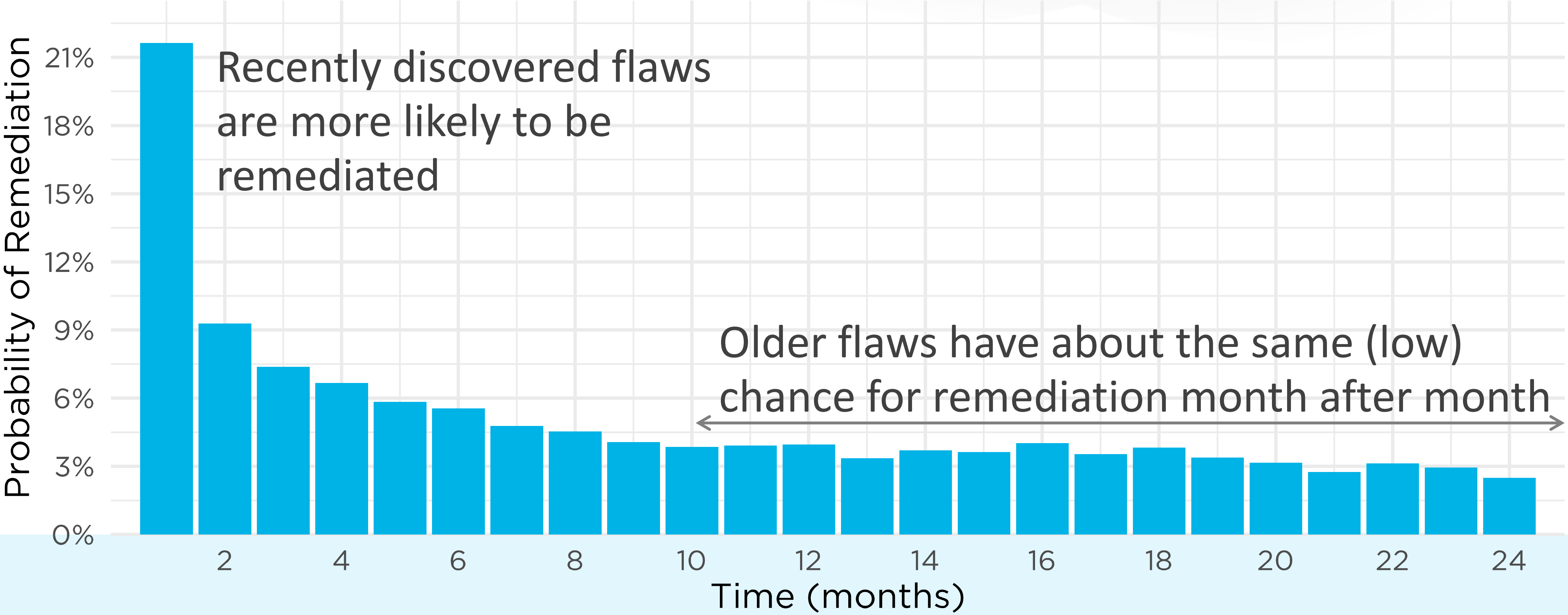
*(courtesy of F5)*

Incidents worked and traced back to root issue

*(courtesy of F5)*

# Security Debt

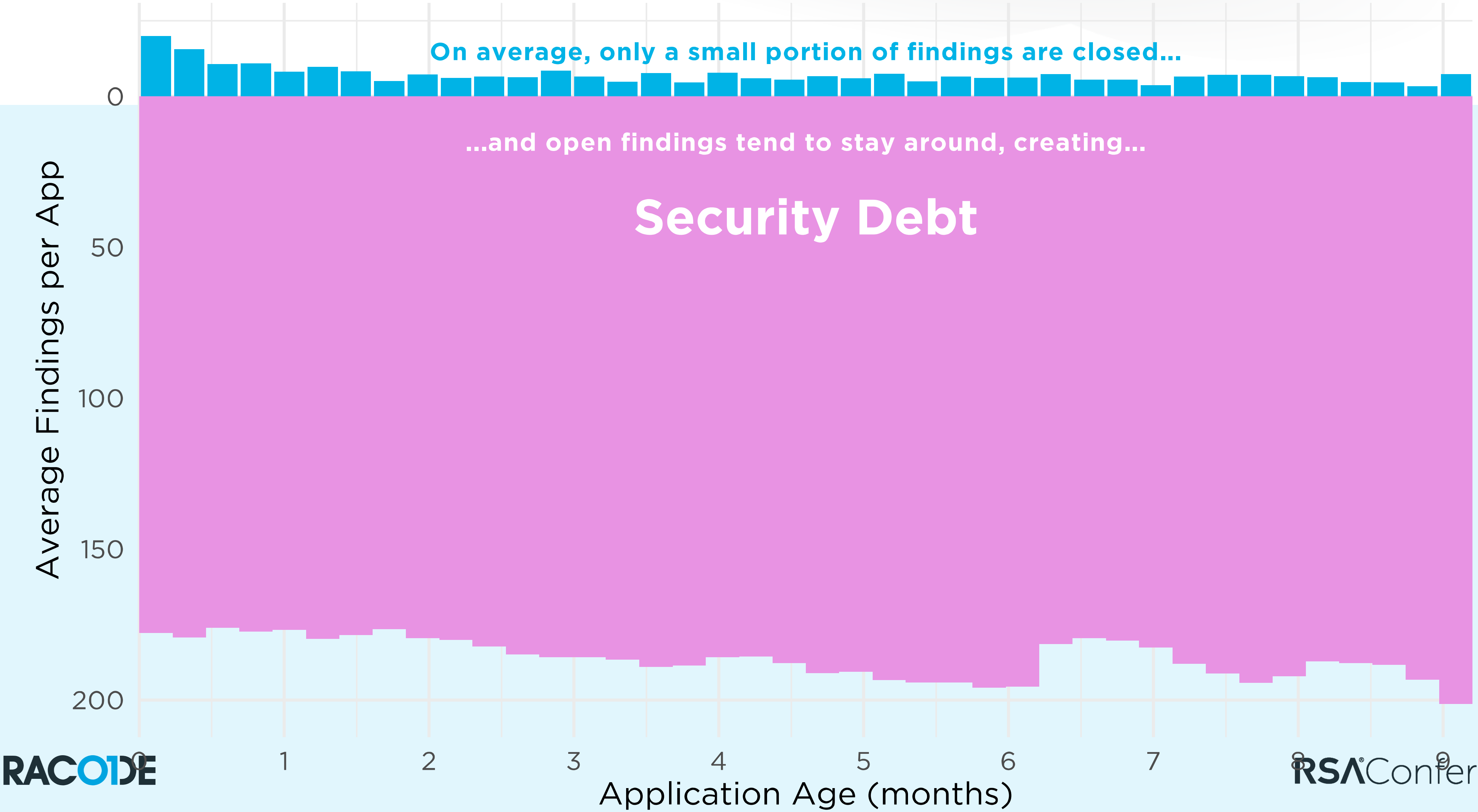
# Probability of remediation over time



Recently discovered flaws are more likely to be remediated

Older flaws have about the same (low) chance for remediation month after month

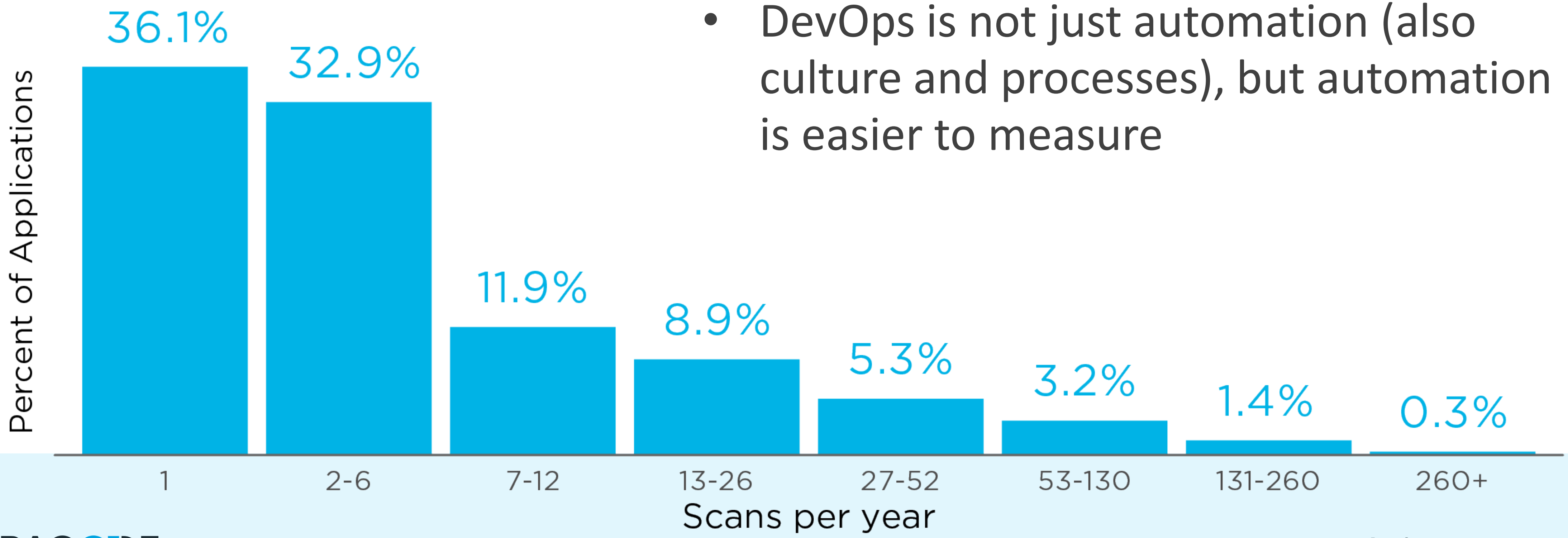
# Flaw fix capacity and accumulation (security debt) over time



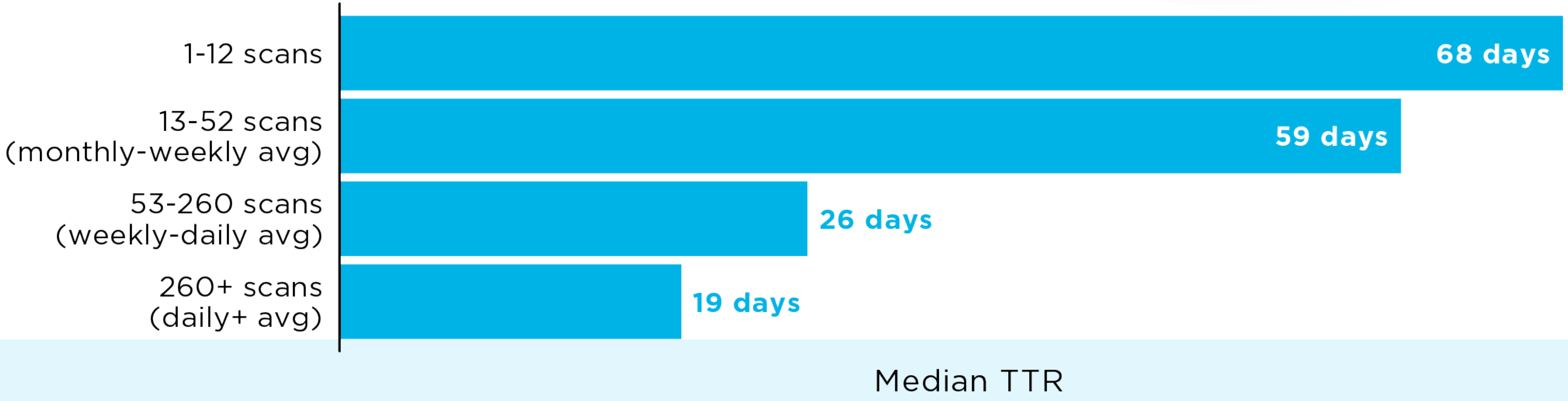
# Does DevOps make a difference?

# Frequency of security scanning across SDLC

- We use scan activity as an indicator that an organization may be following DevOps practices
- DevOps is not just automation (also culture and processes), but automation is easier to measure

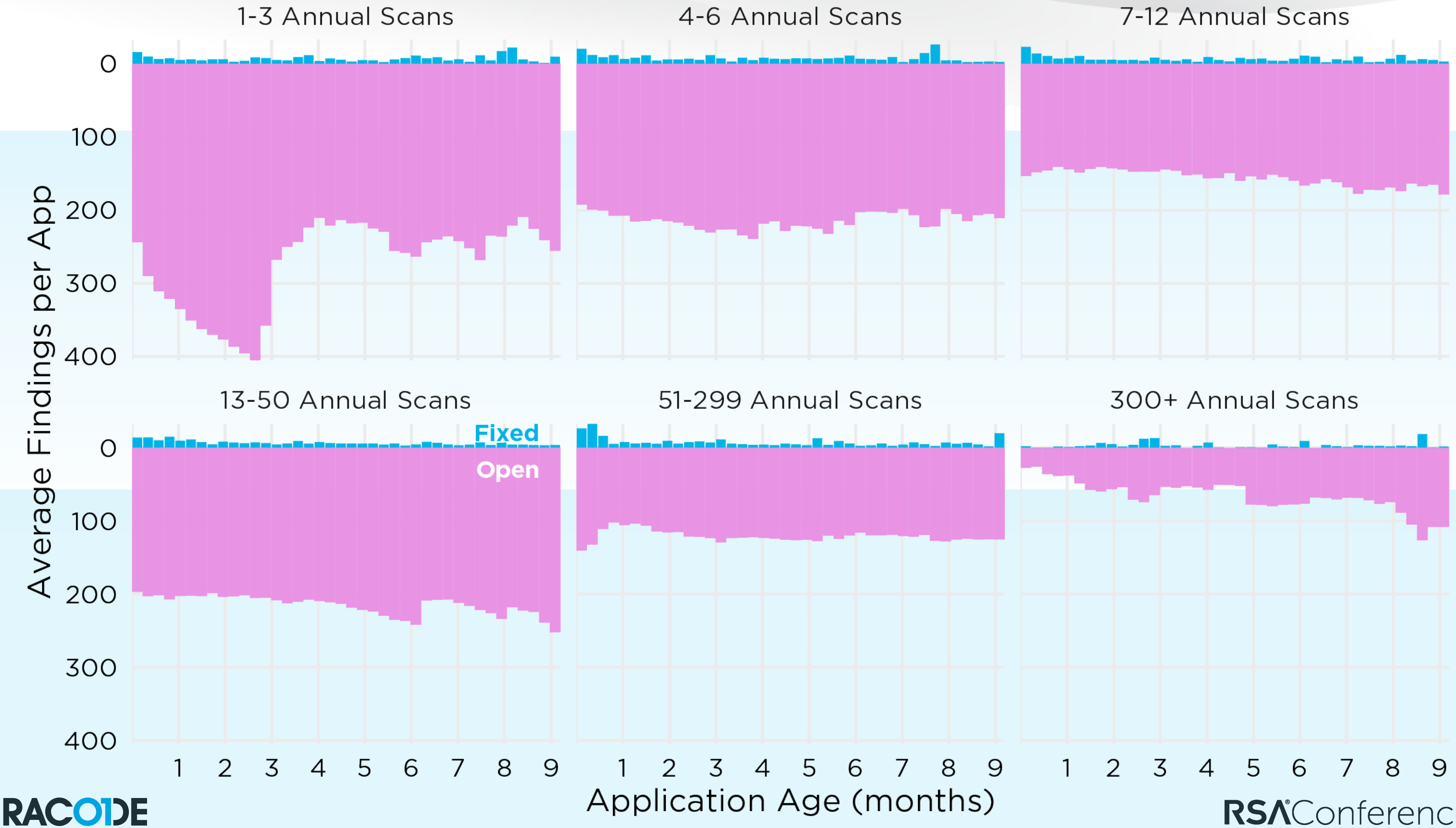


# Effect of annual scan frequency on median time-to-remediation

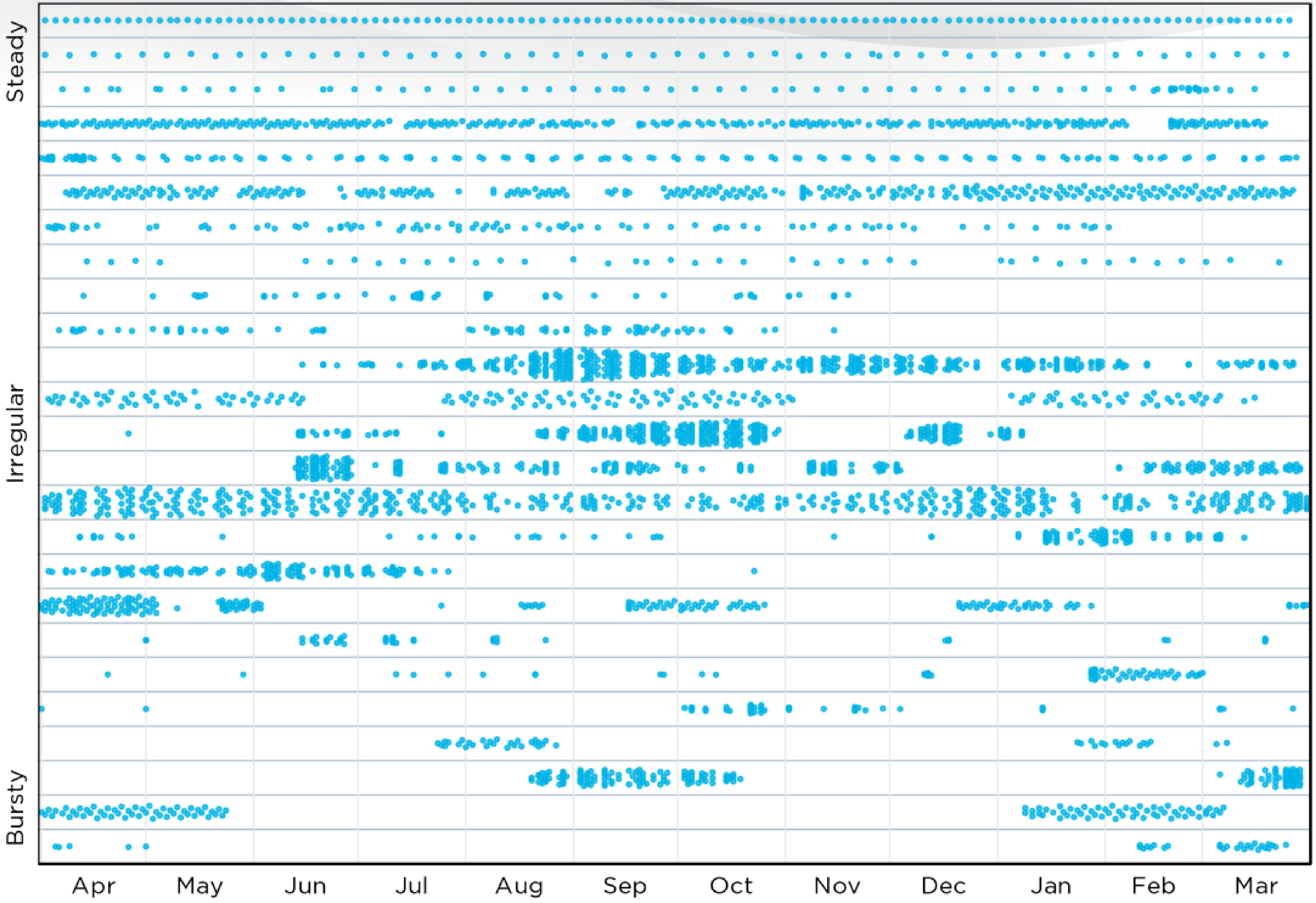


- Frequent scanners closed flaws much quicker.
- Fix rate was tripled
- Security debt reduced three-fold.

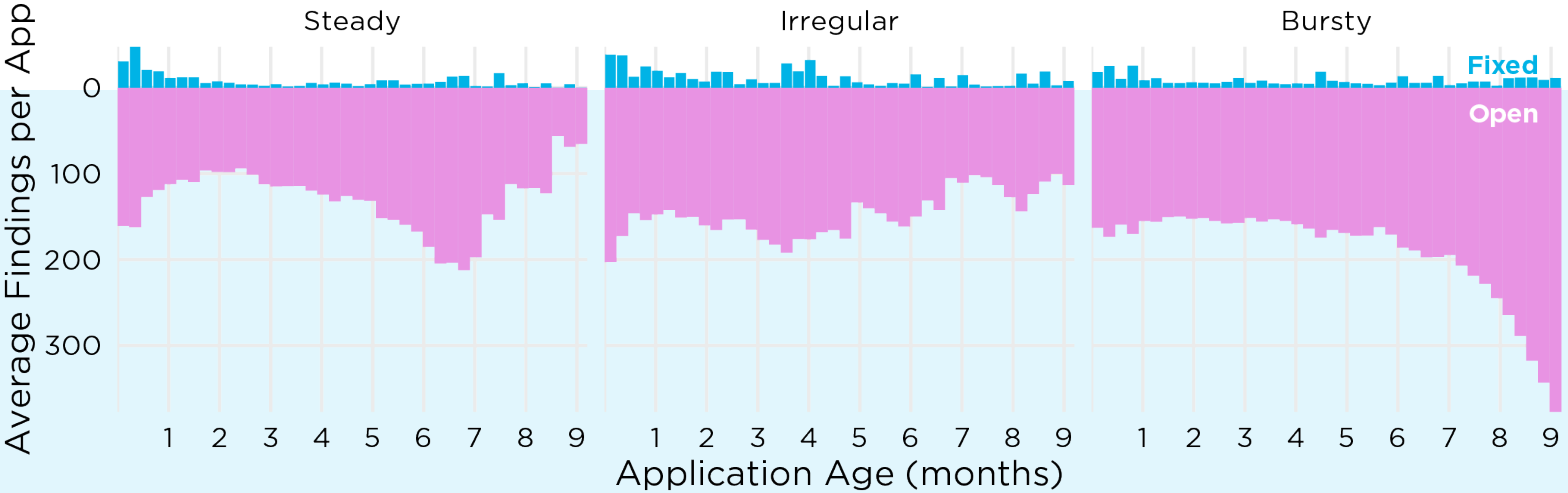
# Fix capacity and security debt by scan frequency



# Security scanning across a sample of SDLCs



# Fix capacity and security debt by scan cadence



Source: Veracode SOSS Volume 10

# Conclusions

# Our data suggests:

- Security automation (as measured by scan frequency) continues to significantly lag the widespread and accelerating adoption of DevOps
- Developers do not prioritize fixes in a security-appropriate manner; recency appears to outweigh every other factor
- Incorporating daily application testing improves MedianTTR by 3x relative to weekly testing
- Steady testing facilitates chipping away at security debt, while bursty testing allows security debt to balloon

# Apply What You Have Learned Today

- Next week you should:
  - Identify DevOps pipelines in your organization that could have AST added
- In the first three months following this presentation you should:
  - Add AST to the DevOps pipelines that are ready
  - Get on a steady cadence for finding and fixing security flaws
- Within six months you should:
  - Work to get all development teams using an automated build process
  - Integrate AST into the build and defect tracking system and process

RSAConference2020
San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: **PS-W02**

# 8 Million Findings in One Year: Fresh Look at the State of Software Security

**Chris Wysopal**

Co-founder and CTO
Veracode
@WeldPond

**Jay Jacobs**

Co-founder and Chief Data Scientist
Cyentia Institute
@jayjacobs

#RSAC