

# 2025 STATE OF SOFTWARE SECURITY FINANCIAL SERVICES SECTOR SNAPSHOT



**VERACODE**

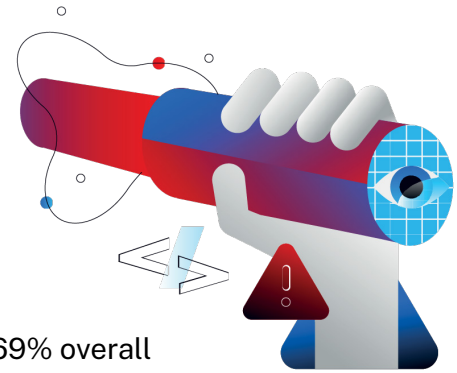
**Discover trends about where the most software risk resides in the financial services sector and what metrics can be used to gauge progress against it.**

Veracode recently published the [2025 State of Software Security \(SOSS\)](#), marking the 15th edition of the annual report that has become a staple in the industry. This SOSS offered a new view of maturity, sharing insights drawn from analysis of 126 million security findings across 1.3 million unique applications. In particular, this edition establishes key AppSec risk metrics and benchmarks the performance of leading and lagging organizations.

While the 2025 SOSS findings are broadly applicable to all types of organizations, they didn't focus on specific industries. The role of this Snapshot is to feature insights specific to managing security debt in the Financial Services sector.



# Finding flaws



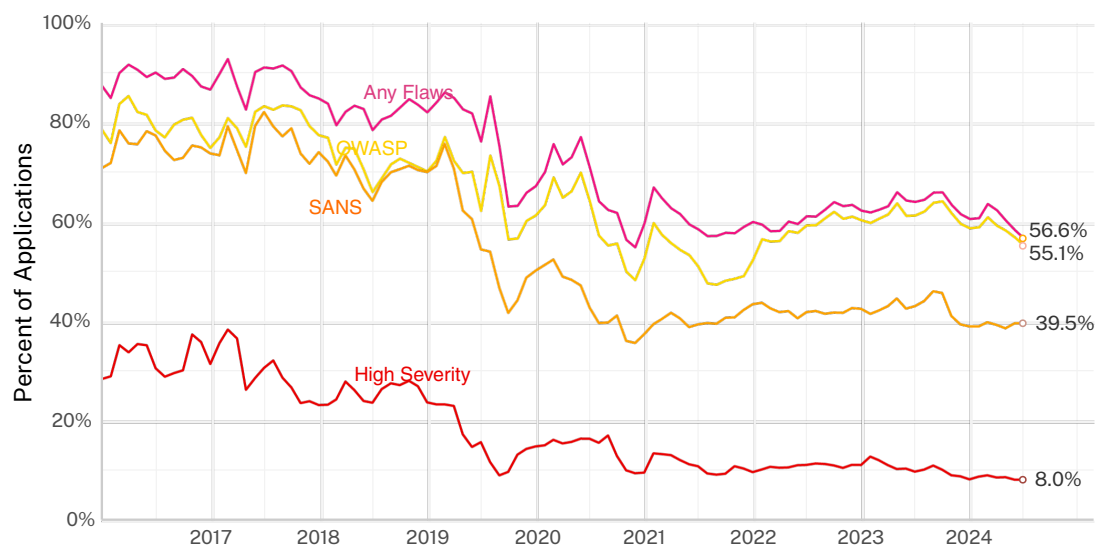
- 55% of apps have OWASP Top 10 flaws vs. 69% overall
- 8% of apps have high-severity flaws vs. 16% overall

As of their latest SAST<sup>1</sup> scan, 57% of financial services sector applications had at least one flaw. Just under that proportion had flaws included in the [OWASP Top 10](#) risks and 40% contained those considered most dangerous per the [CWE Top 25](#). Thankfully, a much lower 8% exhibit high or critical severity flaws according to our [own rating system](#).

If you're reading this Snapshot, it likely won't be surprising to see that most applications — including those in the financial services sector — have security flaws. The chart below puts a few qualifying statistics around that statement.

FIGURE 1

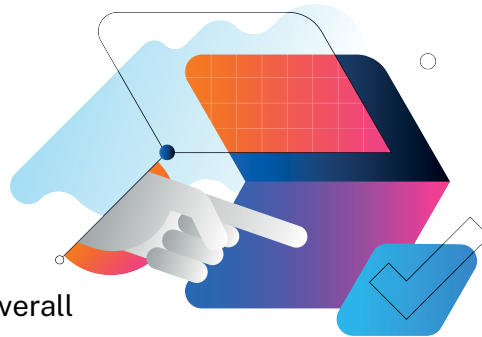
Prevalence of security flaws in financial services sector applications over time (SAST only)



Encouragingly, we see the prevalence of security flaws affecting financial services applications has decreased over time. Though even with the progress made since 2017, we see a stagnation in progress since 2021. Financial service organizations seem to be struggling to make headway in reducing AppSec risk in recent years.

One of the big reasons for this is because AppSec is not a static environment. Code is written, flaws are found, flaws are fixed, new code is written, and the cycle begins anew. Or, perhaps more accurately, it never ends. This is why it's critical to view software security as a lifecycle.

1. Most of the analysis in the 2025 SOSS as well as this Snapshot are based on a combination of static code analysis (SAST), dynamic code analysis (DAST), and software composition analysis (SCA). We use SAST only here because sufficient volume and history of data exists to support longer trends over time.



# Fixing flaws

- Half-life of 276 days vs. 252 days overall
- 30% of flaws still open after 2 years vs. 28% overall

Finding flaws is easy; fixing them is where the challenge lies. There are many ways to measure the speed at which flaws are fixed across active applications. We use a statistical technique called survival analysis for this metric. A flaw's lifespan begins at discovery and ends when scans confirm that it has been fixed (which doesn't always happen).

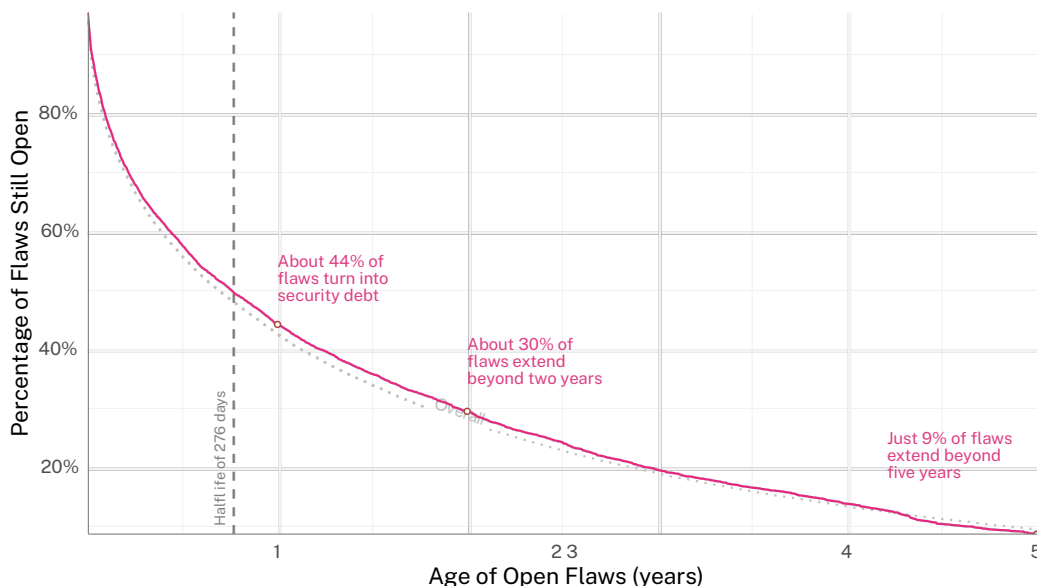
The chart below depicts the survival curve for all flaws across financial services applications. You can determine the survival rate at any point along the curve based on where the x and y axes intersect.

Half-life is a key statistic for fix speed, measuring the time it typically takes to eliminate 50% of flaws.

For the financial services sector, half-life stands at 276 days — less than one month longer than the overall average. After two years, 7 out of 10 of the flaws have been fixed, and 9% remain open for more than five years. Note how the survival curve for the financial service sector is close to the overall curve, indicating that this sector is typical of the overall population in terms of its rate of remediation.

FIGURE 2

Financial service sector flaw remediation timeline based on survival analysis



The concave shape of the financial services survival curve makes it clear that the process of fixing flaws begins in earnest but tapers off over time.

There are numerous reasons for this phenomenon, but the result is that applications gradually become bloated with old, unresolved flaws that we term security debt.



## Managing security debt

- 77% of orgs have security debt vs. 74% overall
- 63% of orgs have critical debt vs. 50% overall

Security debt refers to flaws that remain unfixed for over a year. It's quite pervasive — about three-quarters of all organizations have accrued some level of debt. The level of security debt in the financial service sector is slightly above par, affecting

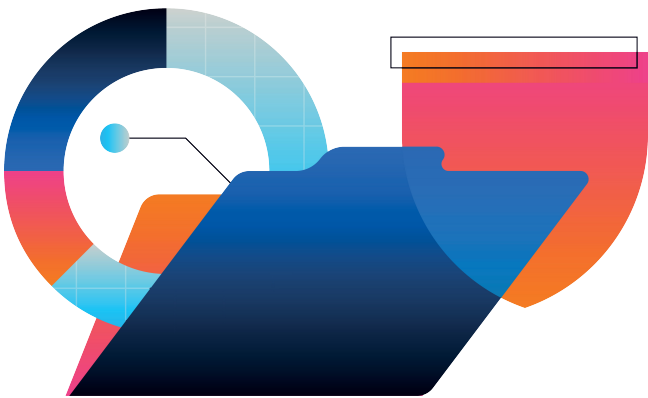
77% of these organizations. Even more concerning, over half (63%) of financial services organizations have critical debt — the risky combination of highly severe and long-unresolved flaws. That's 13% higher than the overall rate.

FIGURE 3

Rate of security debt among financial service sector organizations



Wondering where to start to eliminate debt in your organization? Our analysis shows that debt tends to be most concentrated in older, larger applications. Unsurprisingly, there is a relationship between software security debt and broader forms of tech debt that degrade productivity, efficiency, and resilience in many organizations.

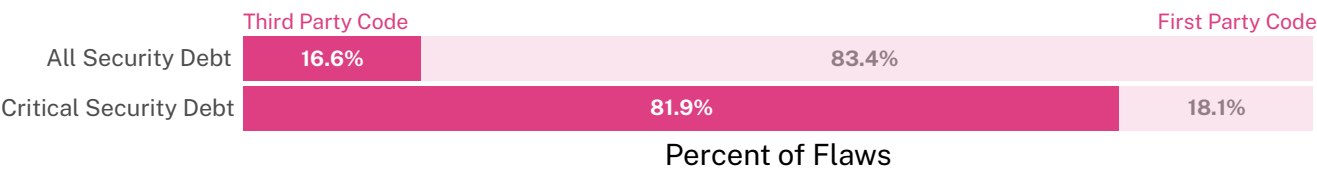


# Open-source debt

Not only does debt affect the code your developers write, but it also creeps in via open-source libraries imported into your applications. Of all security debt in the financial services sector, nearly 17% stemmed from third-party code.

However, if we focus on critical security debt, that ratio exceeds 82%. Any debt-fighting strategy that doesn't extend beyond your team's own code is not one that will ultimately be successful in driving down AppSec risk.

FIGURE 3 Proportion of security debt and critical debt in first-party vs. third-party code



It generally takes about 50% longer to fix flaws in third-party code compared to first-party software. Evaluating open-source libraries and avoiding those riddled with flaws before incorporating them into your codebase can help prevent major issues across applications.



# Benchmarking performance

At the outset of this Snapshot, we promised to benchmark financial services organizations in terms of how they’re managing security debt. Here’s where we make good on that promise for 5 key metrics that we recommend using to measure the maturity of your AppSec program:

- Flaw prevalence: Percentage of applications with at least one unresolved security flaw
- Fix capacity: Average percentage of open flaws fixed each month
- Fix speed: Time it takes to fix half of all discovered flaws (half-life)
- Debt prevalence: Percentage of applications that have accrued debt (flaws >1y old)
- Open-source debt: Percentage of all security debt that exists in third-party code

The stats below compare the performance of “leading” (top 25th percentile) versus “lagging” (bottom 25th percentile) financial service organizations for each metric. On the whole, there’s a marked difference in how leading financial service sector organizations are driving down risk. They have fewer flaws, they fix more of them in a timely manner, and maintain a much lower rate of security debt across their applications.

➤ Leading Organizations	🕒 Lagging Organizations
< 76% of apps have flaws	95% of apps have flaws
Fix capacity: above 9% of flaws monthly	Fix capacity: 0.1% of flaws monthly
Fix speed: half of flaws in 2.5 months	Fix speed: half of flaws in 12.1 months
Security debt in less than 26% of apps	Security debt in more than 85% of apps
71% of apps have open-source critical debt <sup>2</sup>	100% of apps have open-source critical debt

2. This number is much higher than what we observe outside the financial services sector (15%). We suspect this is due to a comparatively low number of financial services organizations using SCA. Those actively doing SCA are likely doing it out of necessity because they heavily utilize open-source libraries, and so are more likely to discover flaws at a higher rate.

How do leading financial services organizations achieve this level of performance? More importantly, how can YOUR team join them?

It starts with having visibility into your SDLC to minimize flaws at the source. Do this through continual scanning as developers write their code before flaws go into production. This is the most cost-effective time to remediate flaws, and it's a great use case for responsible-by-design AI.

Next, correlate and contextualize findings in a single view to prioritize work to burn down security debt. Your AppSec tools are flooding you with information about what's severe, but you need a way to see what's exploitable, reachable, and urgent to help you prioritize further. This requires an open and tool-agnostic [Application Security Posture Management](#) solution.

The Veracode logo is displayed in white text on a dark blue background. The letter 'V' is stylized with a blue diagonal line. The background of the entire page features abstract, overlapping shapes in shades of blue, green, and grey, with a grid pattern visible in some areas.