State of Software Security 2023: BFS&I

Securing the Future:

The State of Software Security in Banking, Financial Services, and Insurance

VERACODE

Each year we publish a series of cuts of the data specific to verticals or geographic regions as companion research to the State of Software Security (SoSS). These cuts allow us to narrow the lens slightly and explore where we are, how we got there, and how we could do things better. It also provides an excellent chance to tease out relevant trends that get somewhat buried in the aggregate data view of the main report. This search for the signal also lets us track progress, which is particularly interesting this year for our Financial Services cut.

Over the last year, the security performance of applications within the Financial Services vertical has gone from solid middle of the pack to generally outperforming the other industries across all four of the types of flaws we are tracking. Looking at Figure 1 you will notice that the position of High Severity doesn't quite match what you just read. How do we reach the "generally outperforming" conclusion when High Severity flaws are still performing at average? The average performance of the introduction of High Severity flaws is technically also slightly better than the average of the collective non-financials by 1%, but yes- financials are still lagging in that one area.

As we continue to examine Figure 1, just under 72% of applications had at least one security flaw found in their last scan over the last 12 months, compared to over 76% of the non-financial organizations. That is an improvement of 1% down from 73%. Nicely done. At 67.1% of OWASP Top 10 it came close to a two-way tie, but both manufacturing and financials are ahead of the pack. CWE Top 25 shows up at the top spot with 51.7% of applications introducing at least one flaw in their last scan over the last 12 months. All in all, it is a good showing for Financial Services.

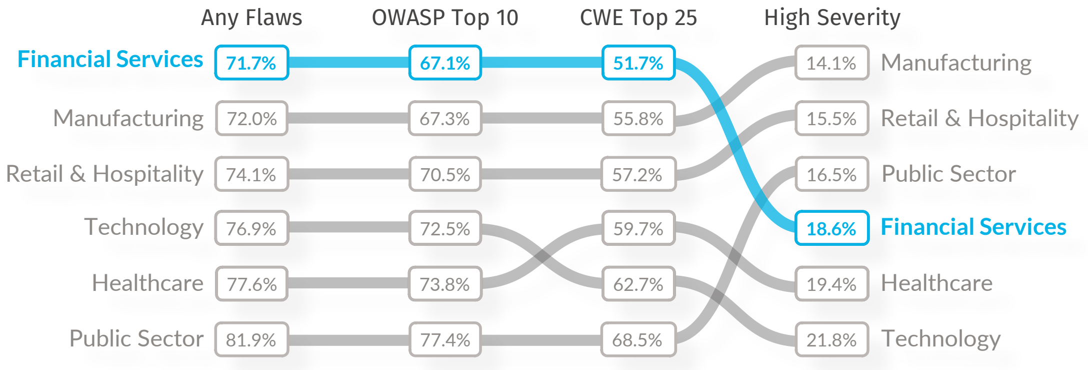
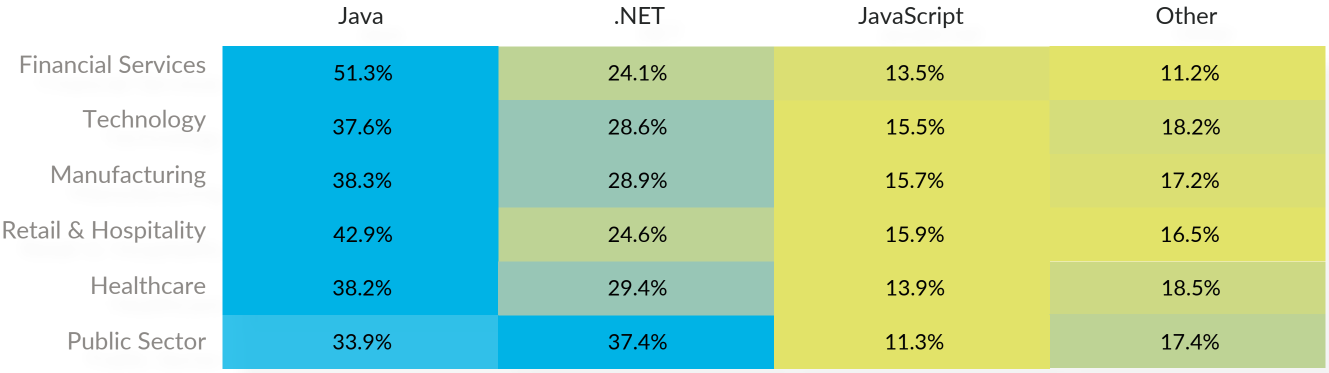


Figure 1: Percent of applications that had a flaw found in their last scan over the last 12 months, by category. Lower numbers are better.

Having a look at Figure 2, we can see that.NET and JavaScript take the second and third position for most-used language and are the least likely to stray to some other language from those three. At 51% Java is almost a de facto standard within Financial Services organization. The clear preference for Java is interesting and not in the least due to the remediation timeline that we saw for Java in the main SoSS 2023 research. Application teams working with Java remediated flaws at a slower rate than .NET and JavaScript, and this could be construed as a call to action for Financial Services organizations to examine their flaw remediation profile.



	Java	.NET	JavaScript	Other
Financial Services	51.3%	24.1%	13.5%	11.2%
Technology	37.6%	28.6%	15.5%	18.2%
Manufacturing	38.3%	28.9%	15.7%	17.2%
Retail & Hospitality	42.9%	24.6%	15.9%	16.5%
Healthcare	38.2%	29.4%	13.9%	18.5%
Public Sector	33.9%	37.4%	11.3%	17.4%

Figure 2: Development Language Usage by Vertical

To break things out by the scan type we see general alignment within Financial Services with their overall flaw types and the proportion of applications with each of those flaw types. In Figure 3 below, we can see that Static Application Security Testing (SAST) scans show a lower percentage of applications than non-financials are introducing almost every category of flaw . That is where the better performance seems to end though. When examining the percentages of applications where flaws were discovered by Dynamic Application Security Testing (DAST) and Software Composition Analysis (SCA), financial organizations are not faring so well. Why? Is it a measure of maturity in the usage of those tools? Is this some sort of a statement about the use of open source in Financial Services organizations? Does it mean that Financial Services organizations leverage third-party software at a higher rate?

At first, we thought that there might not be a data-driven answer here, but there may very well be one. Java applications are overwhelmingly (>95%) made up of third-party code (see SoSS version 12 Figure 6), and Financial Services organizations are big users of Java. (See Figure 2 above). Given that SCA picks up flaws in the composition of applications, the probability of finding publicly reported flaws with SCA rises commensurately with higher percentages of open-source code. This does not suggest that these applications are full of flaws though. It only indicates that scans found flaws and reported them. After the report of flaws is delivered, SCA will also advise upgrades to versions of the libraries that do not contain those flaws (if applicable). We have reported on this different SCA flaw profile in previous editions of the State of Software Security as well, but it is an interesting correlation to be sure, and one we saw again in a different light with .NET and the Public Sector cut published earlier this year.

Based on the SCA findings in Figure 3, those reading from Financial Services organizations might wish to baseline their third-party software practices and check how many flaws are being introduced and carried forward from these scans. Another recommendation is to encourage teams to be more judicious in adding libraries that perform simple tasks that developers could code themselves. This minimizes the exposure to dependency chains. We are not saying to roll your own crypto or write a new database.

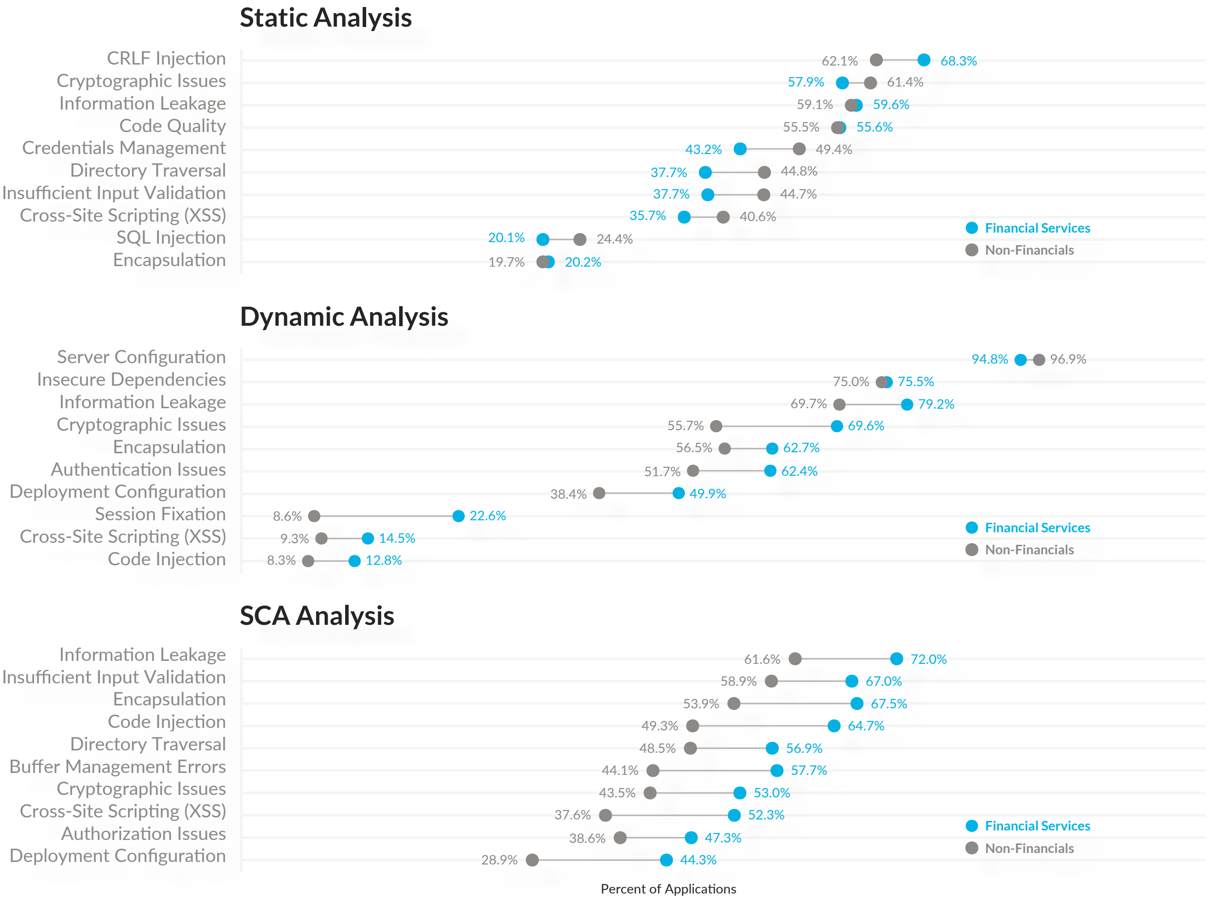


Figure 3: Top Flaw Types by Scan Type

When we examine the lifecycle of applications in Financial Services organizations, we see that their profile closely mimics that of the general population. The starting point and the corresponding drop where applications are onboarded are near identical. The drop is then followed by the “honeymoon period” where fewer flaws are introduced. Where the general population is up around 23% of applications with new flaws, teams in Financial Services trend closer to 20%. As the applications age though, we also see the familiar upward creep begin at about the 1.5-year mark. While the percentages stay lower at first, the lines seem to wander up at the same point. Once this happens, applications in Financial Services organizations join the Everyone Else trend and climb above the 30% mark somewhere between the three- and four-year mark. The shading you see on the blue line is the range of certainty which matches the general profile of Everyone Else as well.

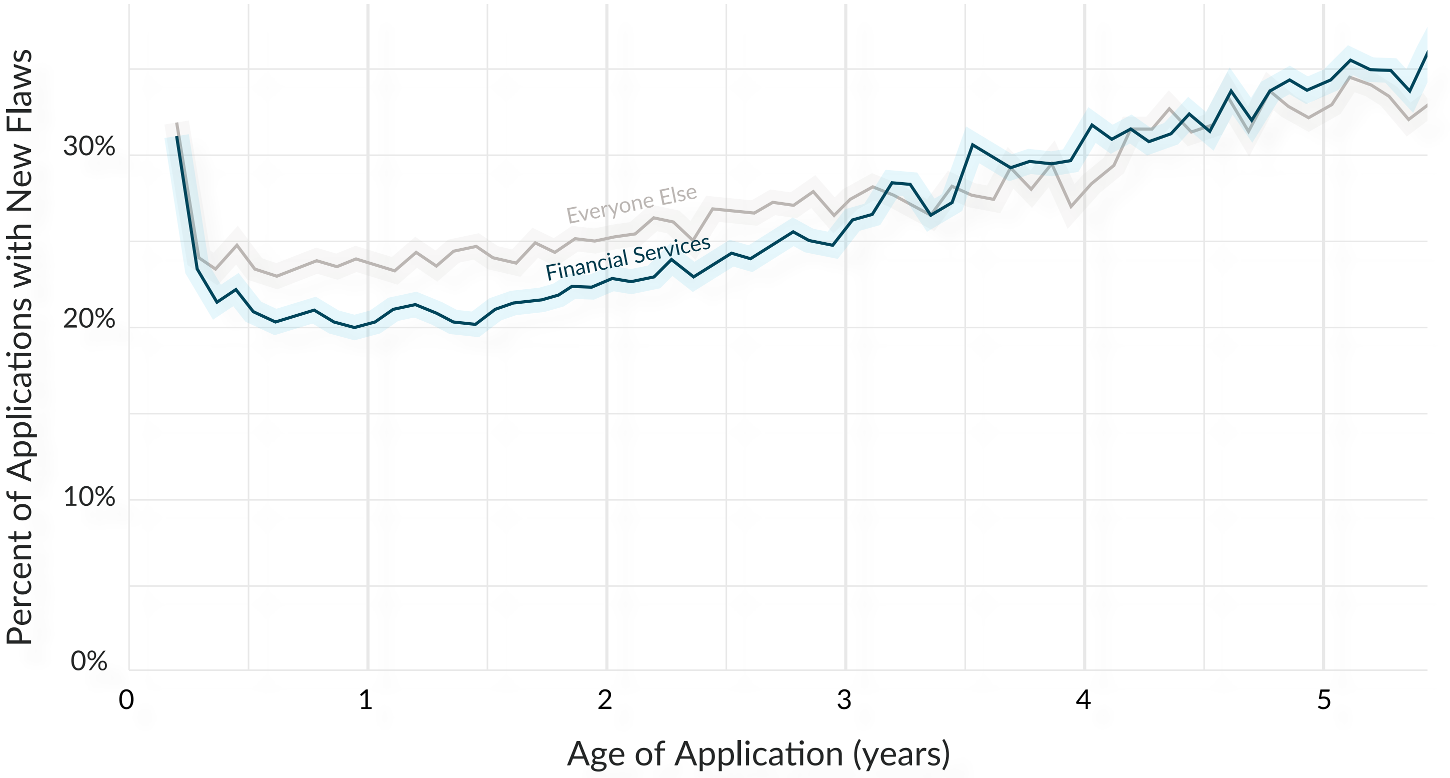


Figure 4: Application Size by Age of Application

If we think back to Figure 1 where we saw that development teams in Financial Services are generally performing better from a security perspective, we see in Figure 5 some potential correlation as to why. We see stronger effects from the positive elements of scanning via API and security training than with the general population of applications. Mind you, this is a correlation, but it is rather interesting. The base probability of introducing one or more flaws per month is 27% and Figure 5 depicts the elements that affect that base probability.

Scanning via API is a rough measure of maturity. Teams that integrate scanning via API likely have more automation and control over the development pipeline. We see that the Financial Services development teams leveraging scanning via API reduce the chance of flaw introduction per month by 2.9%. That is almost a percent better than non-Financials and given that the base chance is 27% that's a significant reduction. The next difference is the benefit of security training on the probability of flaw introduction. Again this is almost a full percent better as seen in Figure 5. While the other elements tend to stick to the trend for all other industries these two (API-launched scanning and training) stand out and combined drop the base probability of flaw introduction down to less than 22%. Further, completing 10 training courses is not a magical number. As we discussed in the main SoSS 2023 research, there is a benefit at one training and that benefit continues to grow past our demarcation point of 10, but eventually flattens out.

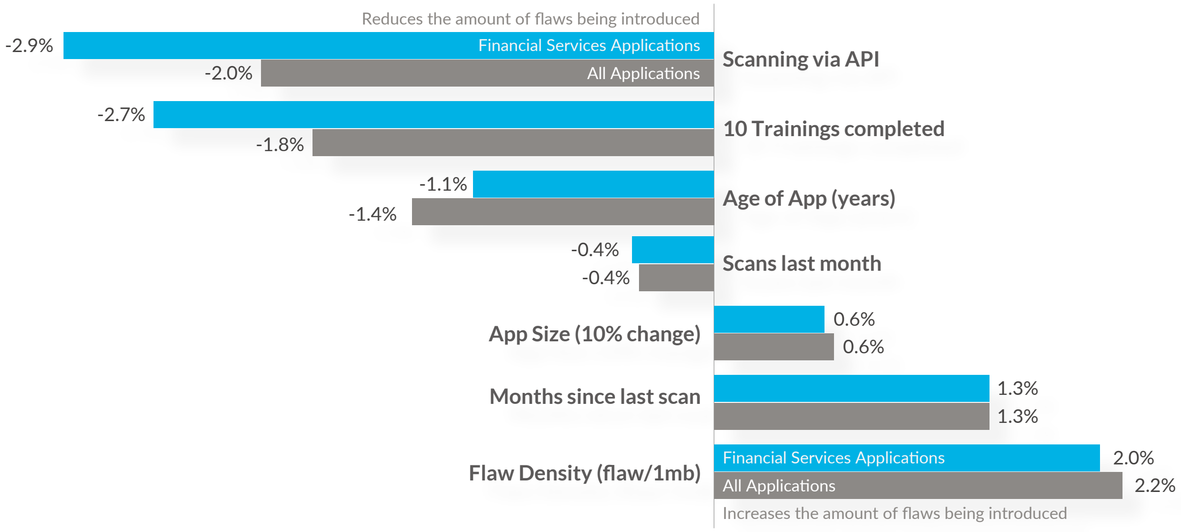


Figure 5: Factors Influencing the Probability of Flaw Introduction

We noted in the main SoSS 2023 research that the factors in Figures 5 and 6 pay off in multiple ways, but it is really clear this is true for Financial Services teams. Sort of like double dipping, only in this case we have explored in previous years that developer training also has a positive effect on remediation speed and number of flaws remediated for a quadruple dip. Where Figure 5 is examines the probability of introducing flaws in any given month, Figure 6 is examines how those same factors affect how many flaws are introduced when they are introduced. The number that jumps off the page in Figure 6 is that when flaws are introduced, teams that completed 10 trainings introduced 26% fewer flaws. That is a quarter fewer flaws which is well above the all-industry average. Similarly launching scans via API, which is our rough measure of automation and maturity, seems to have a stronger influence on Financial Services than the all-industry average. For the rest of the beneficial and detracting factors, Financial Services closely align with the averages.

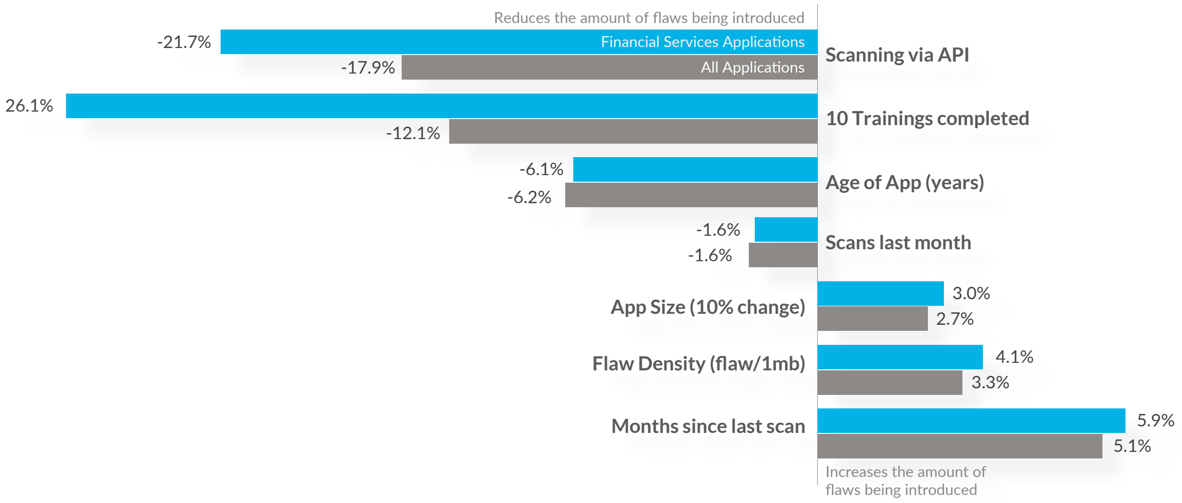


Figure 6: Factors Influencing the Amount of Flaws Introduced

Recommendations

Examine the Remediation Timeline

Financial Services organizations top the charts when it comes to Java usage. Based on our analysis in the main SoSS report this means that the Financial Services vertical likely directionally influenced a fair portion of the remediation timeline as a larger constituent in the overall total. This means that Financial Services organizations should definitely baseline their own program to examine the potential room for improvement in not just Java but also, other languages if there are similarities in how development teams function.

If teams can claim to hit both targets then they will be able to state they are performing well by SoSS 2023 benchmarks. This was approximately the remediation timeline for teams that developed applications in JavaScript in the main SoSS 2023 report. Teams that cannot achieve those numbers likely will eventually carry an increasing balance of flaws that will accumulate over time.

Questions to ask:

How many days does it take to address 50% of flaws that are introduced? **(target <120 days)**

What percentage of flaws survive three months after introduction? **(target <55%)**

What percentage of flaws survive two years? **(target <14%)**

Training and Maturity

The data indicates that, as in the general population, Financial Services organizations benefit from training and the automation that API usage brings in the scanning environment. If only to double down on the advice from the main SoSS 2023 report, it makes sense to improve the overall level of secure coding capabilities in the organization. Getting to the self-service automation is aspirational for many organizations but, similarly, we see that launching scans via API correlates with very good results in the probability that flaws will be introduced at all and then by reducing the number of flaws introduced when they *are* introduced.

Write simple code yourself

We saw the SCA outlier in Figure 3, and that also occurred in our EMEA data cut. Both EMEA and Financial Services organizations are big Java users. We plan on testing in the next SoSS research to find out if this is indeed a Java thing. For now, a recommendation from our Open Source section of the State of Software Security 2023 is a safe bet. Since Java applications are overwhelmingly open source, teams need to discuss a purposeful way of when they should include relatively simple libraries that bring dependency chains of questionable value. If it is simple code, write it yourself, but don't roll your own crypto or dive into a proprietary database. Fewer dependencies, by their nature, shoud help.

VERACODE

Veracode is intelligent software security. The Veracode Software Security Platform continuously finds flaws and vulnerabilities at every stage of the modern software development lifecycle. Prompted by powerful AI trained by trillions of lines of code, Veracode customers fix flaws faster with high accuracy. Trusted by security teams, developers, and business leaders from thousands of the world's leading organizations, Veracode is the pioneer, continuing to redefine what intelligent software security means.

Learn more at www.veracode.com, on the Veracode blog and on Twitter.