VERACODE

VOLUME 11

The State of Software Security

Industry Snapshot: Government and Education

Veracode's State of Software Security (SOSS) Volume 11 focused on how developer actions can influence software security. It also looked at the impact that application attributes (their "nature") and the development environment (the "nurturing" of the app) can have on how quickly flaws are fixed. Our analysis uncovered key differences in software security between industries and found that these differences affect how quickly flaws were addressed, adding another dimension to our "nature vs. nurture" discussion. This infosheet provides a summary of factors that shape software security for the government and education sector.

Figure 1 describes how applications compare across industries in finding and fixing flaws. From left to right, the columns rank each industry by the proportion of applications with flaws, the proportion of applications with high-severity flaws, the percentage of flaws that are fixed, and the median speed at which those flaws are fixed. In this figure, the government and education sector is among the poorer-performing groups, as it fares poorly with fixing flaws and has the second-longest median time for fixing flaws. Even though the government and education sector has the greatest proportion of applications with flaws, the proportion of applications with high-severity flaws are less, compared to other industries. So, while there are a lot of applications to fix and it takes a long time to make those fixes, many of the issues are not catastrophic. That is somewhat comforting in an otherwise bleak picture.

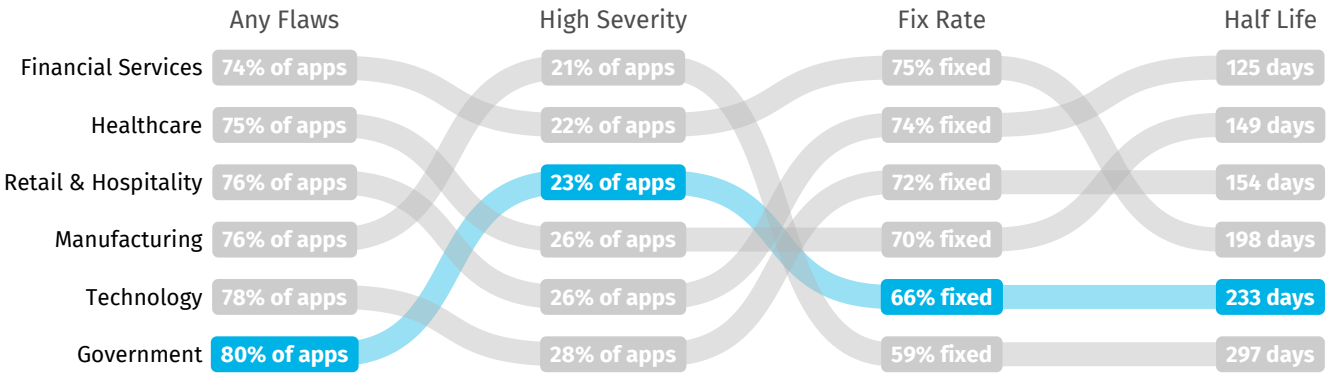

Figure 1: Values and rankings for key software security metrics by industry.

In Figure 2, we show a more detailed view of the flaw types discovered in applications. The left axis shows the overall prevalence of each flaw category across all sectors, and the right axis shows the prevalence of the category for the government and education industry. The ranking is very different for the industry, as shown by the number of crossed lines in this figure. Five of the top 10 flaw types show a lower prevalence in government and education applications. The data shows that applications for the government and education industry have bigger problems with Cross-Site Scripting and input validation compared to the overall trends. In contrast, the prevalence of CRLF injection flaws drops significantly in the government and education sector. SQL injection is an exception, as that flaw type is more common among government and education applications compared to the overall prevalence.

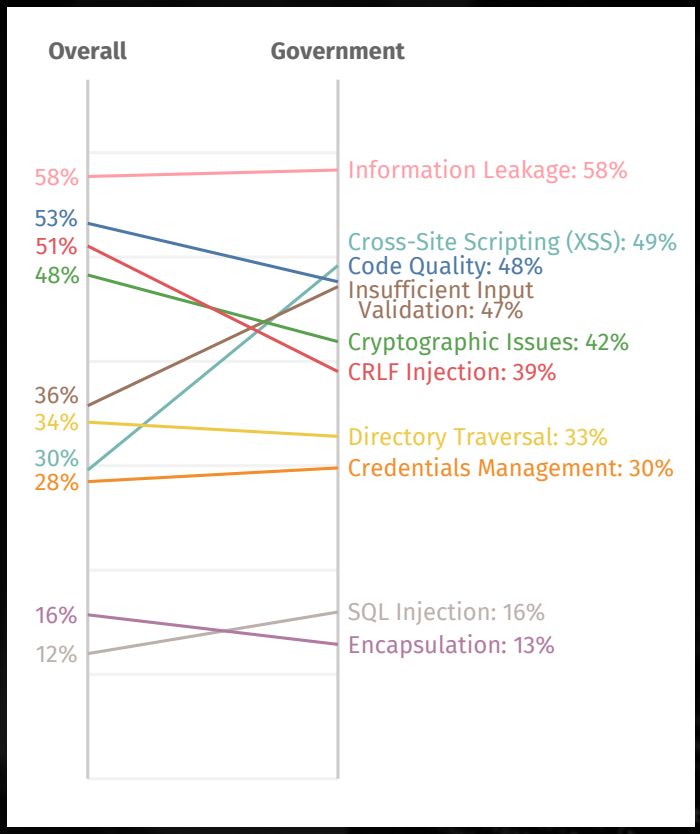

Figure 2: Prevalence of flaw categories in the government and education sector

In this year's SOSS, we see strong evidence that certain developer behaviors associated with DevSecOps yield substantial benefits to software security. In Volume 10, we discovered that teams scanning applications most frequently carry five times less security debt than infrequent scanners. In this volume, we see other developer actions also make a difference in fixing flaws in software.

For example, we looked at the impact that application attributes and the development environment can have on how quickly flaws are fixed in software. There are some attributes that developers have no control over and others, such as best practices, that developers can influence.

In Figure 3, we look at how entities in the government and education sector compare with companies in other industries in their development attributes (nature) and utilizing helpful developer behaviors (nurture). Going from left to right for attributes, firms in the government and education sector are set up fairly well compared to other industries, ranking first with newer applications and small organization sizes, and with a middle-of-the-road ranking for application sizes.

Government and education have the worst flaw density, meaning flaws are found throughout the application. In contrast, government and education organizations are all over the map when it comes to adopting DevSecOps behaviors, ranking first in scanning frequency and integrating security testing into the development process (using the API), in the middle for using software composition analysis, and dead last in both using dynamic analysis scanning technology and in scan cadence.

While developers in government and education are scanning frequently, they are not doing so consistently – so there are periods of frequent scanning interspersed by periods of no (or little) scanning. This suggests that for many government and education organizations, security testing is still being saved for specific points after coding, perhaps just before a major release. They are also less likely to have a comprehensive view of the flaws in the application because they typically do not utilize multiple scanning methods.

In the most recent SOSS, we see that certain factors are likely to lead to flaws getting fixed faster, and others lead to slower fixes. Industry is something developers don't have control over, so how does being part of the government and education sector affect time to remediate flaws?

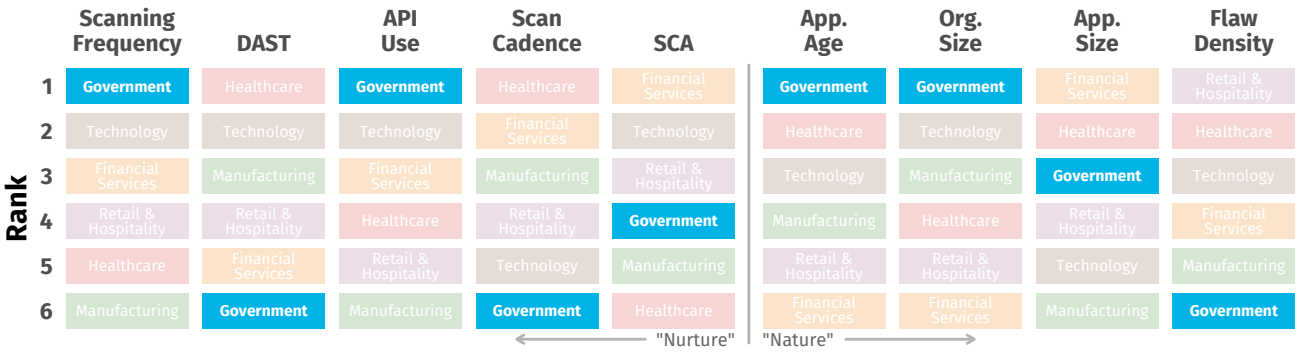

Figure 3: Relative ranking of the government and education sector for attributes and actions associated with application security performance

It's not pretty, at all. Figure 4 shows us that being in the government and education sector means the organization is significantly slower in remediating flaws. We see hints of this in Figure 1, with a median time to flaw remediation of 233 days, or a little less than eight months. Figure 3 suggests that even though developers in the government and education sectors don't have to deal with a challenging set of circumstances, the high flaw density slows down their efforts to fix the flaws and reduce security debt. Government and education organizations were at the extremes in Figure 3 on which DevSecOps behaviors had been adopted, so there is plenty of room for developers to improve.

We clearly see the impact of erratic best practices on a fairly positive environment in Figure 4 as the data trends toward slower flaw remediation. It goes to show that when developers don't consistently utilize best practices for secure coding, there is a negative impact on software security even when the circumstances are somewhat in favor of the developers.

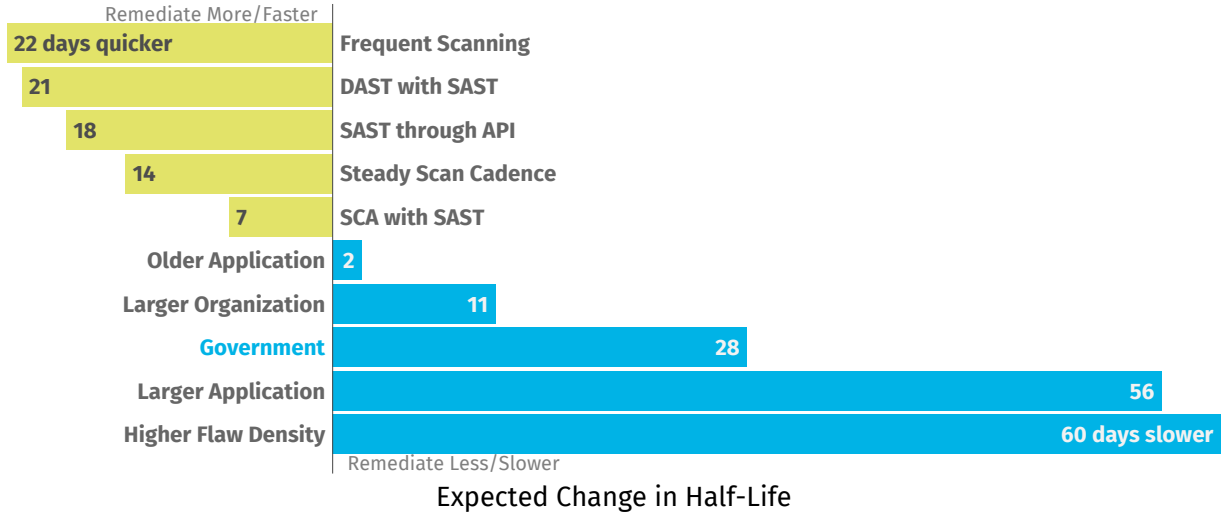

Figure 4: Remediation factors...