BUILDING SMARTER DEFENSES

There are security weaknesses in most applications' code. But how do you make sure you find them before the bad guys do?

Infosys®

WHY PREVENTION IS BETTER THAN CURE

When one of our client's most strategic web applications was hacked, it was something of a wake-up call.

Application security is surprisingly often an afterthought, and serious investment is only sometimes made when security breaches are discovered. Effectively, companies try to close the stable door after the horse has bolted.

However, preventive security maintenance of apps is far more cost-effective and less damaging than reactive security. Our client therefore asked us to put in place an app security testing program early in the development lifecycle to reduce the chances of similar breaches in future.

Infosys®

LOOKING FOR THE ACHILLES' HEEL

We began with static application security testing (SAST) followed by dynamic application security testing (DAST) scanning, using a range of tools including Fortify, Burp, Zap, and Nessus. These testing programs were combined with network vulnerability assessments (NVA), designed to provide a complete picture of each app's security profile, and to highlight any weaknesses in the code. This was conducted first on the main app that was affected by the security breach, and then moved onto other strategic application portfolios.

OOPS! FALSE ALARM!

However, using these tools on their own, the testing process was long and time-consuming because the tools would scan the code and flag up all the potential issues that they found. Each issue would need to be manually checked by a security analyst, even though many of them were false alarms. We knew that, by checking thoroughly, we could find the points of failure that might cause future security issues – but at a high cost.

We realized that if we could filter out the false alarms, we could increase productivity immensely. We therefore created a filter set for our client that focused the attention of the team on the most important weaknesses. These were then addressed, and we passed our findings back to the development team. Lessons were thus learned, and future code became gradually more secure and robust.

BREAKTHROUGH

If we could filter out the false alarms we could increase productivity immensely.

Infosys®

STRONGER AND STRONGER

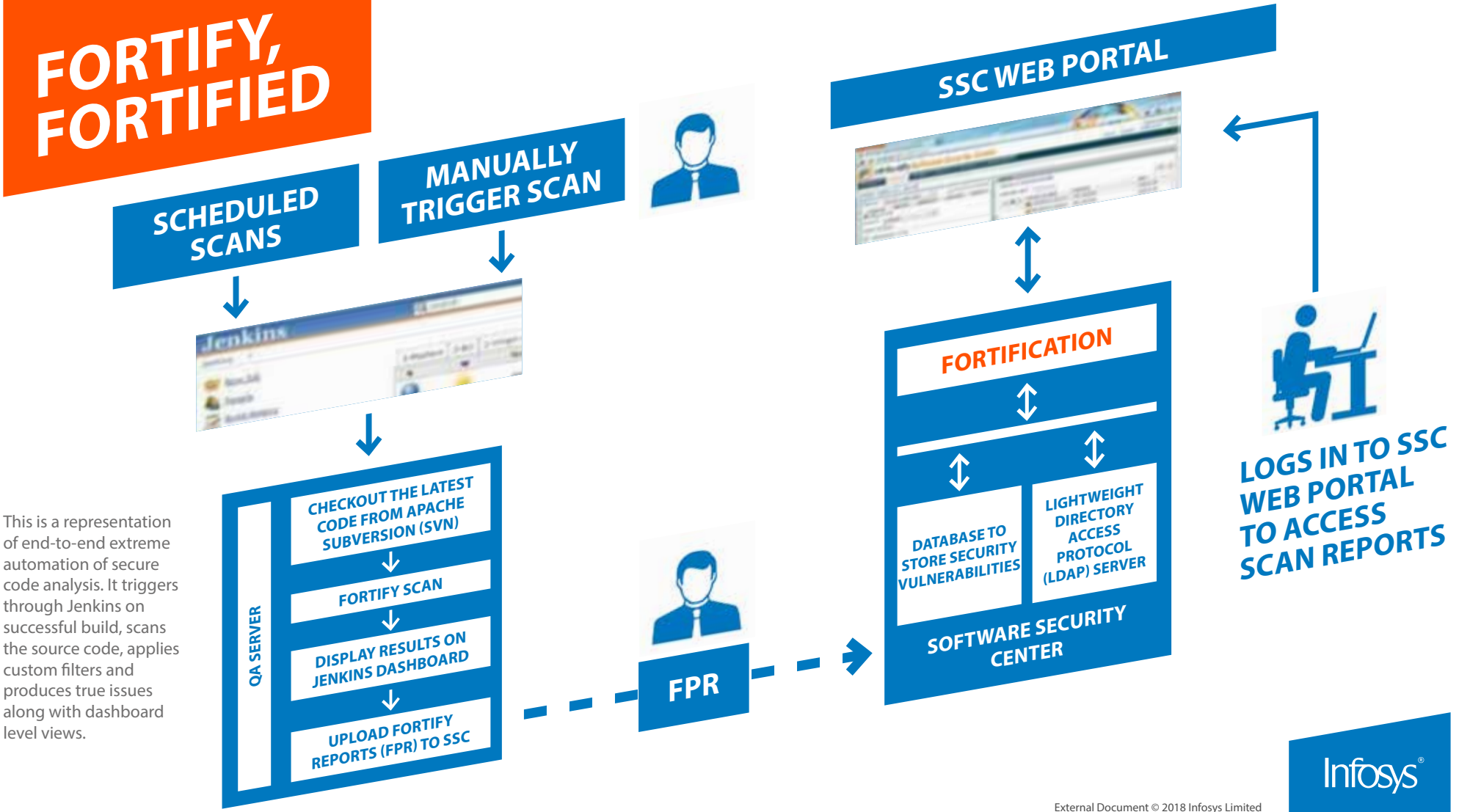
One tool in particular, Fortify (from MicroFocus), was already being used effectively. But we wanted to make life easier for security analysts by automating a part of the checking process. We used the connectors available within the tool to enable Fortify to filter out the 'false alarms' and only report genuine issues that required manual attention. This capability was continually refined over time as part of a continuous integration/continuous deployment (CI/CD) process where new learnings were gradually incorporated into the system.

The result was a platform that we refer to as **Fortification**, because we have taken Fortify and made it even stronger, and we are now making that innovation available to other clients. Some modification is required, but the principles of the platform can be immediately applied to apps developed for all clients.

Infosys®

FORTIFY, FORTIFIED

This is a representation of end-to-end extreme automation of secure code analysis. It triggers through Jenkins on successful build, scans the source code, applies custom filters and produces true issues along with dashboard level views.



Infosys®

REMOVED

90

PERCENT OF MANUAL CHECKING EFFORT

INCREASED COVERAGE TO

85

PERCENT APPS

ZERO

PERCENT DEFECT LEAKAGE IN PRODUCTION

We used the connectors available within the tool (Fortify), to filter out the 'false alarms' and only report genuine issues that required manual attention.

WE DID THIS FOR THEM. WE CAN DO IT FOR YOU.

Find out more about how Fortification can be a smarter approach to improve app security. Reach out to us at askus@infosys.com

Infosys®