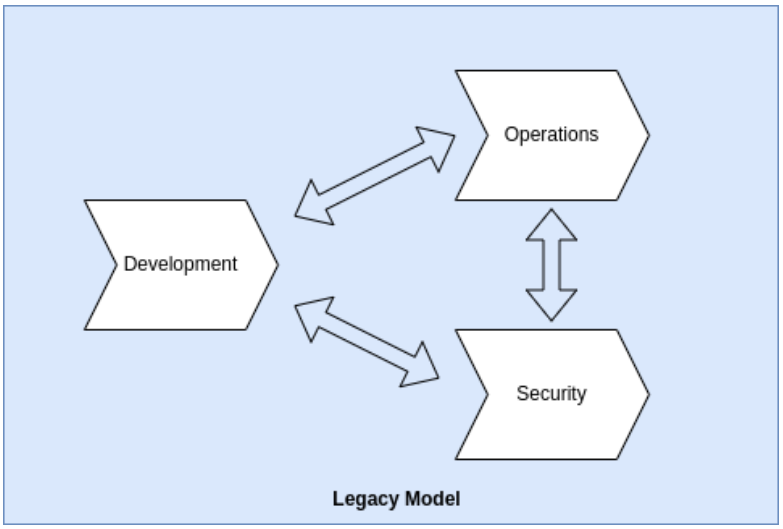
DevSecOps: Necessity of the Modern Application Infrastructure

Baber Rehman
October 19, 2022

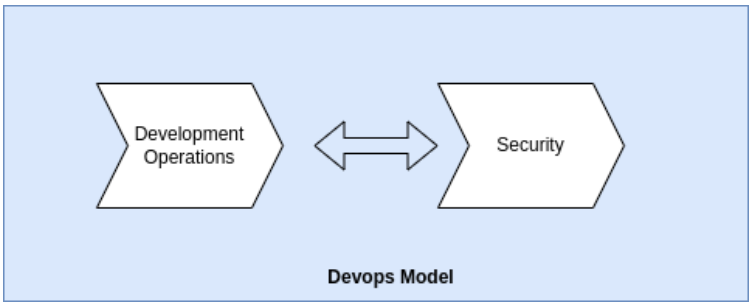
Let's talk about the legacy development, deployment and security models. Developers build an application, then the application goes to the operations team. Operations team is responsible for the deployment and maintenance of the application. This includes logging, monitoring and setting up other essential infrastructure for the application. Security team works side by side to make the application secure and resolve any potential security threats. And all this is happening in a sequential manner with various teams involved in various phases of the application. If the application doesn't behave as expected in the production environment, the operations team passes the necessary details back to the development team. Back and forth communication channels open among the development team and the operations team. Development team resolves the issue and pushes the updated build towards the operations team. Similar patterns repeat if more such issues are encountered.



DevOps

This legacy model works, but is it efficient? Is the communication among various teams timely? Do developers understand operational terminologies and vice versa? If not, then what can be done to make the overall process efficient? Answer to all these questions in the recent past is DevOps. DevOps is a modern application development

and deployment culture where developers are not separated from the operations team. On the contrary, DevOps culture intermingles development and operations roughly in the same team. Developers are more aware of the operational process including deployment, monitoring and logging. Therefore, it's the very start of the build process where developers start thinking about the potential operational challenges and minimize them as far as possible. Moreover, numerous automated tools are available to be used in devops culture for deployment, monitoring and logging. This makes the end-to-end process much more efficient compared to the legacy models.

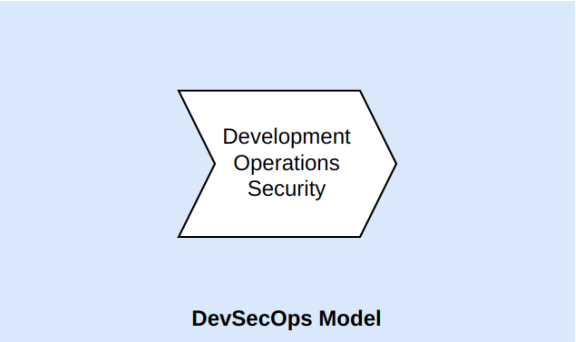


Devops Model

DevSecOps

What about application security? Someone has to be responsible for the security of the application. In the DevOps model, there is a separate security team which manages the security of the application. If the security team encounters any threat, the application goes back to the DevOps team and the process repeats as above. A natural question arises at this stage is can we not merge the security team with the DevOps team as we did for the development team and the operations team? DevSecOps answers this question. DevSecOps intermingles the development, operations and security tasks roughly under a single umbrella.

Security has now become a shared responsibility instead of an independent job for the security team. Developers are now well aware of the common security practices and security considerations are counted since the very start of the application cycle. The gain to all this is the efficient delivery of the applications and the updates. But at the same time, this asserts more pressure on the developers to learn about the operations and the security practices.



DevSecOps Model

DevSecOps Tools

Automated tools are being developed to maximize the automation of repeated tasks in DevSecOps culture. This shifts some pressure from developers to the automated tools and makes the process more efficient and robust, since it minimizes the human involvement upto some extent which cuts down the probability of human errors. There are various areas in DevSecOps and having one article to cover all the areas is unjustified.

Containerized Technology

We will particularly talk about the tools available for **container vulnerability scanning** in this article. Other areas will be discussed in subsequent articles. Containerized technology has revolutionized the modern application deployment workflow where applications run in isolated packages. This isolation of the applications guarantees some level of security since one application is restricted to directly manipulate the files of another application even though running on the same server. However, container build images may itself be compromised.

Code running in a docker container may leak sensitive information to another server. Unnecessary ports may also be used by an application running inside a docker container. Moreover, one or more base images may be compromised. If not impossible, this definitely is a non-trivial job to check each docker container against all such security threats without any automation. Fortunately, several pre-programmed tools have been developed which, given a docker image, can identify the potential security threats.

Primitive Docker Scan Features

Docker is the leader in the containerized applications world. Docker itself provides a **docker scan**[1] plugin along with the docker command which scans the given docker image for any potential threats. It provides an option to scan the base images as well if one wishes to do so. Scan results can be stored in a file and analyzed for remediation. Furthermore, it provides the ability to filter out the vulnerabilities as per severity level, ranging from low, medium to high. Results can also be printed in JSON format for further automated processing. **Docker bench for security**[2] provides an automated script to scan the docker images against standard security practices. This script is available for free to assist the developers to scan docker containers against CIS Docker Benchmark v1.4.0 (at the time of writing this article).

Other Container Vulnerability Scan Tools

Containers are like light-weight virtual machines in which an application runs in isolation. Thus running a vulnerability scan on a container follows the same analogy as running it on the actual machine to identify the unpatched threats.

Numerous commercial and open source tools are available for container vulnerability scanning. This includes Trivy, Grype, Claire, Anchore, Dagda, Falco and Aqua Security among others. Such tools are feature rich and bring forth an interactive interface to look for and remediate vulnerabilities. Some of the tools also provide integration with CI/CD pipelines, which means the vulnerability scan tool will run as soon as the container is built and without human intervention.

[1] https://docs.docker.com/engine/scan/
[2] https://github.com/docker/docker-bench-security