Prioritizing security in our software is more important now than ever. DevSecOps is all about incorporating security practices into our DevOps process from the start, moving these activities "left" in the development lifecycle so they don't delay delivering value later on. This eGuide rounds up a collection of resources to help you build security into your existing deployment pipeline, so you can reduce risk, expose vulnerabilities early, and prioritize security in your software from day one.

In this DevSecOps eGuide

DevSecOps: Incorporate Security into DevOps to Reduce Software Risk

DevSecOps is a growing movement to incorporate security into DevOps practices in order to ensure flaws and weaknesses are exposed early on through monitoring, assessment, and analysis, so remediation can be implemented far earlier than traditional efforts. By failing fast with security testing, organizations reduce risk of a security incident and decrease the cost of rework.

4 Keys to Protecting Your Data in a DevOps World

It may seem like the desires for end-to-end DevOps and protection of sensitive data are in conflict, but if done correctly, they can be two sides of the same coin. DevOps processes such as version control and delivery automation introduce the very measures needed to properly protect production data. The key to keeping data safe while using it during your DevOps process is to focus on these four areas.

Using the Principles of the CIA Triad to Implement Software Security

If you're starting or improving a security program for your software, you probably have questions about the requirements that define security. Data need to be complete and trustworthy, and also accessible on demand, but only to the right people. The CIA triad defines three principles—confidentiality, integrity, and availability—that help you focus on the right security priorities.

DevOps and Security: 5 Principles for DevSecOps

With the trend toward a more continuous delivery and deployment process, late-lifecycle activities like security assurance present a significant hurdle to continuously delivering value to customers. DevSecOps addresses this by shifting security assurance activities, personnel, and automation closer to development.

A Definition of Done for DevSecOps

In DevOps, we have a software delivery pipeline that checks, deploys, and tests every build. The goal is to produce a viable candidate for production, so we have to look at many different aspects of quality, including security. To be sure we hit all the crucial marks, we should have a definition of done for DevSecOps.

Managing Security Testing in Agile Software Development

One of the biggest myths in the world of agile development is that there is not enough time to do security testing. Sanjay Zalavadia shows you the most efficient and cost-effective way of performing security testing in an agile environment: by rolling it into each sprint incrementally, from day one.

Insight from the Industry

Additional Resources

DevSecOps: Incorporate Security into DevOps to Reduce Software Risk

By Alan Crouch

By now, most organizations have heard of DevOps, and many have begun to adopt DevOps practices as a key enabler of software delivery. Organizations that employ an agile approach find DevOps practices a natural extension, and DevOps truly enables agile practices to flourish.

Organizations typically start with implementing continuous integration, test-driven development, and test automation early on. Agile delivery teams embrace small, iterative development and increased code quality, and with these practices in place, continuous delivery and continuous monitoring practices start taking hold. Teams focus on faster delivery with less human interaction, and successful organizations build more collaboration between development and operations teams as they start working toward a shared goal.

While all these practices provide strategic benefits like breaking down the traditional silos between development, test, and operations, it is unfortunate that many organizations stop there. In my experience, most organizations fail to integrate their security programs into their development efforts, resulting in lengthy security compliance activities and lots of vulnerabilities identified late in the delivery lifecycle. Does that sound like DevOps to you?

These issues are why the concept of DevSecOps is making such a large impact in the security community. DevSecOps is a growing movement to incorporate security into DevOps practices in order to ensure flaws and weaknesses are exposed early on through monitoring, assessment, and analysis, so remediation can be implemented far earlier than traditional efforts. By failing fast with security testing, organizations reduce risk of a security incident and decrease the cost of rework.

Exploring a DevSecOps Workflow

Here's an example of DevSecOps in action:

1. Developers create the code and tests, which are managed by a version control system like Git.
2. Changes are committed to the Git.
3. Jenkins pulls the code from the repository and builds and runs unit tests, as well as static code analysis to identify code quality bugs and security defects.
4. An infrastructure-as-code tool, like Chef, provisions an environment, deploys the application, and applies security configurations to the system.
5. Jenkins runs a test automation suite against the newly deployed application, including UI, back-end, integration, API, and security tests.
6. If the application successfully passes all tests, the application is deployed to production using the same infrastructure-as-code tool used in the previous environments.
7. The production environment is continuously monitored by tools like New Relic and Splunk to detect active cyber security threats.

DevSecOps provides a number of benefits among development, security, and operations. It eliminates silos, promotes collaboration and teamwork, and identifies vulnerabilities early while still providing better, faster delivery. DevSecOps also contributes business value through dollars and resources saved, improved operations,