ALERT LOGIC®

TAKING A DEVOPS APPROACH TO SECURITY

10 PRACTICAL SECURITY TIPS FOR DEVOPS

More organisations are embracing DevOps and automation to realise compelling business benefits, such as more frequent feature releases, increased application stability, and more productive resource utilization. However, many security and compliance monitoring tools have not kept up. In fact, they often represent the largest single remaining barrier to continuous delivery.

By working with the DevOps team, you can ensure that the production environment is more predictable, auditable and more secure than before. The key is to integrate your security requirements into the DevOps pipeline; however, as part of that integration you will need to change the way you work. A normal approach of checklists, templates, manual processes etc will not scale. With the speed of cloud deployments, you will need to automate and use tools and scripts. This will allow you to move as fast as the DevOps team needs you to.

10 PRACTICAL SECURITY TIPS FOR DEVOPS

For security professionals, it is key to understand that instead of validating the end solution, you need to validate the pipeline. If you are happy that the pipeline is building the solution in a way that meets you security goals, you can be confident that this will be repeated every time a developer needs to get source code into production.

1. ARCHITECTURE AND DESIGN

During architecture and design the development teams will be attempting to rapidly iterate against the requirements whilst building out the Cloud infrastructure. It is at this point that security teams need to get involved to understand the scope of what teams are looking at, different elements of the infrastructure need protection in different ways. Learn and understand the shared security model - Amazon S3 is very different to protecting EBS storage on an EC2 instance, the barriers between IaaS and PaaS are rapidly breaking down, and each has a different security paradigm.

Threat Modelling can be done against the different components. This will allow security teams to define the threats against the different components, and what elements are going to be needed further up the DevOps pipeline to secure them.

ACTION: Work with the architecture to understand the cloud components being used, and the security controls required for each. Take this further by using techniques like Threat Modelling

2. STATIC CODE ANALYSIS + CODE REVIEWS

Code reviews are a common part of DevOps, Security team should educate colleagues on secure coding techniques so that they can include this into their secure code reviews. However, many of these items can be validated by using Static Code analysis as part of the build process. This is where the source code or a partially compiled version of the source code can be checked for potential vulnerabilities. Many potential security vulnerabilities can be picked up at this point, and if they fail the checks - it breaks the build. Developers will quickly change coding techniques to meet the requirements.

ACTION: Understand what the current code review process is and ensure that there are security elements within that. Likewise investigate what Static Code Analysis tools are available and if they can be used

3. AUDIT OF CHEF COOKBOOKS / CLOUDFORMATION SCRIPTS

You will hear the phrase 'infrastructure as code' a lot in the DevOps world. This is where the infrastructure is built in a highly automated way using scripts and configuration files. The advantage of this for a security professional is that automated checks can be run against these scripts. If a developer creates an infrastructure script to create a storage bucket with public access to the internet, this can raise an error. Combine this with the threat modelling where you have identified potential issues (like marking a storage bucket as public) and you have a very powerful tool to validate the infrastructure every time a developer makes a change.

ACTION: Use the automation tools to ensure that the infrastructure is being built to meet the security standards

4. SECURITY TESTING POST BUILD

Automated builds and unit tests running after check-in are a core part of DevOps. This is where security teams can add-in security testing tools to automate the validation of the build (https://www.owasp.org/index.php/Appendix_A:_Testing_Tools). The reason why automated build and testing is so key in DevOps is that the shorter the time between a developer checking in code and a test failing - the less time it will take for the developer to fix the issue. The same holds true for security vulnerabilities, running testing at the end of the project can inject significant delays as developers struggle to identify the issues and fix the bug. Identifying the issue within minutes of a developer checking the code in, reduces the time taken to identify and fix the issue.

ACTION: Investigate automated security testing tools and integrate into the build process

5. SECURE AND HARDEN THE OPERATING SYSTEM

Let's move from 'Infrastructure as Code' to 'Secure Infrastructure as Code'. If you are creating and building servers via scripting, lets also add in the scripts to lockdown the OS as well. The risk of applying OS hardening at the end of the project is that the application stops working. If it is applied at the beginning of the project, firstly issues are identified up front, and secondly if the hardening has to be relaxed, it can be identified early, and security teams can work with the developer to potentially find another way of performing the function. If it occurs late in the project, the development team will force the issue through.

ACTION: Review the automation scripts to ensure that the OS is being deployed in a secure way and any changes to this standard are controlled. Use resources like SANS Linux Security Checklist or CIS Benchmark

6. HARDEN YOUR CLOUD DEPLOYMENT (Standard AMIs, Security Groups, IAM roles, MFA tokens)

Cloud services can deliver incredibly secure infrastructures 'if' done correctly. However it is also very quick and easy to open up significant security holes. You need to review how your company is using the cloud. This include the segregation of roles - do developer have the rights to change the production environment. If so why? I am sure you do not let your server administrators walk around using Domain Admin accounts; so why should people have root access in the AWS Console. You need to review everything from the development environment through to production

ACTION: Review how teams are accessing the console and what permissions that they have. People should only have the permission they need to do their job, and if they have significant permissions they should be using two factor authentication

7. DEPLOYMENT OF SECURITY TOOLS

Once you get to deploying applications to production, are you going to be able to keep up with multiple teams deploying multiple applications to production. In the same way to can use automation to ensure that security is as you require it, you can ensure that your security tools are deployed at the same time.

You should be looking at deploying network detection for threats on the network, monitoring of HTTP for attacks as well as monitoring log files. With Alert Logic Cloud Defender you can monitor these three different feeds as well and at the same time have a 24x7 SOC investigate the threats and escalate if required.

ACTION: Script the deployment of your security tools so that all environments have a baseline coverage

8. VULNERABILITY SCANNING OF OS AND APPLICATIONS

One of the most common attack vectors is for people to exploit the vulnerabilities in the OS or applications that are running on the servers. As part of a DevOps pipeline servers can be checked for vulnerabilities. This ensure that you know what state your servers are at any point. In addition with Alert Logic Cloud Defender this information feeds into Alert Logic's ActiveAnalytics platform, which identifies valid security events and suppresses false positives. When ActiveAnalytics determines a series of events to be a valid security threat, an incident is created. Depending on incident severity, escalation with remediation recommendations will be delivered via email or through an Alert Logic security analyst. The Alert Logic approach dramatically reduces false positives and keeps analysts and customers focused on real, actionable incidents

ACTION: Run regular vulnerability scans against the environments and remediate any vulnerabilities

9. PHOENIX UPGRADES

Instead of deploying patches to production, you should be burning and redeploying servers as required. This not only increases your agility to roll out new versions, but also increases your ability to rapidly respond to security issues. You can deploy a new patched version across your entire cloud environment rapidly and safely; and with the phoenix upgrade strategy you also reduce the risk of technical debt and configuration drift

ACTION: Work with the DevOps team to support them using Phoenix Upgrades and ensure this gives you the ability to patch security issues and roll them out.

10. ON-GOING AND REAL TIME AUDIT OF PRODUCTION ENVIRONMENT

Visibility post deployment is often down to the level of auditing that has been put in place. You should have standard auditing levels across different Server roles and Applications. Your goal is to get a level of auditing that can be fed into a security tool like Alert Logic to give it the data that is needed, but not swamp your servers with too much auditing.

Once all of these elements are in place, it will allow you to audit production to ensure that at any point in time you understand what state production is in, and if it has drifted from it defined security profile. The cloud is often referred to as a programmable datacenter. Developer can use this to create huge IT systems in very short timeframes - you can use this same power to audit these systems multiple times a day.

ACTION: Work with the development team to set logging levels and use a tool like Chef to ensure that your configuration does not drift

The evolution of DevOps should be extended to embrace Security – providing speed and agility to securing critical applications, assets and services in a more predictable, auditable and secure way.

ABOUT ALERT LOGIC

Alert Logic, the leader in security and compliance solutions for the cloud, provides Security-as-a-Service for on-premises, cloud, and hybrid infrastructures, delivering deep security insight and continuous protection for customers at a lower cost than traditional security solutions. Fully managed by a team of experts, the Alert Logic Security-as-a-Service solution provides network, system and web application protection immediately, wherever your IT infrastructure resides. Alert Logic partners with the leading cloud platforms and hosting providers to protect over 3,000 organizations worldwide. Built for cloud scale, our patented platform stores petabytes of data, analyses over 400 million events and identifies over 50,000 security incidents each month, which are managed by our 24×7 Security Operations Center. Alert Logic, founded in 2002, is headquartered in Houston, Texas, with offices in Seattle, Dallas, Cardiff, Belfast and London. For more information, please visit www.alertlogic.com.

ALERT LOGIC®

032715UK