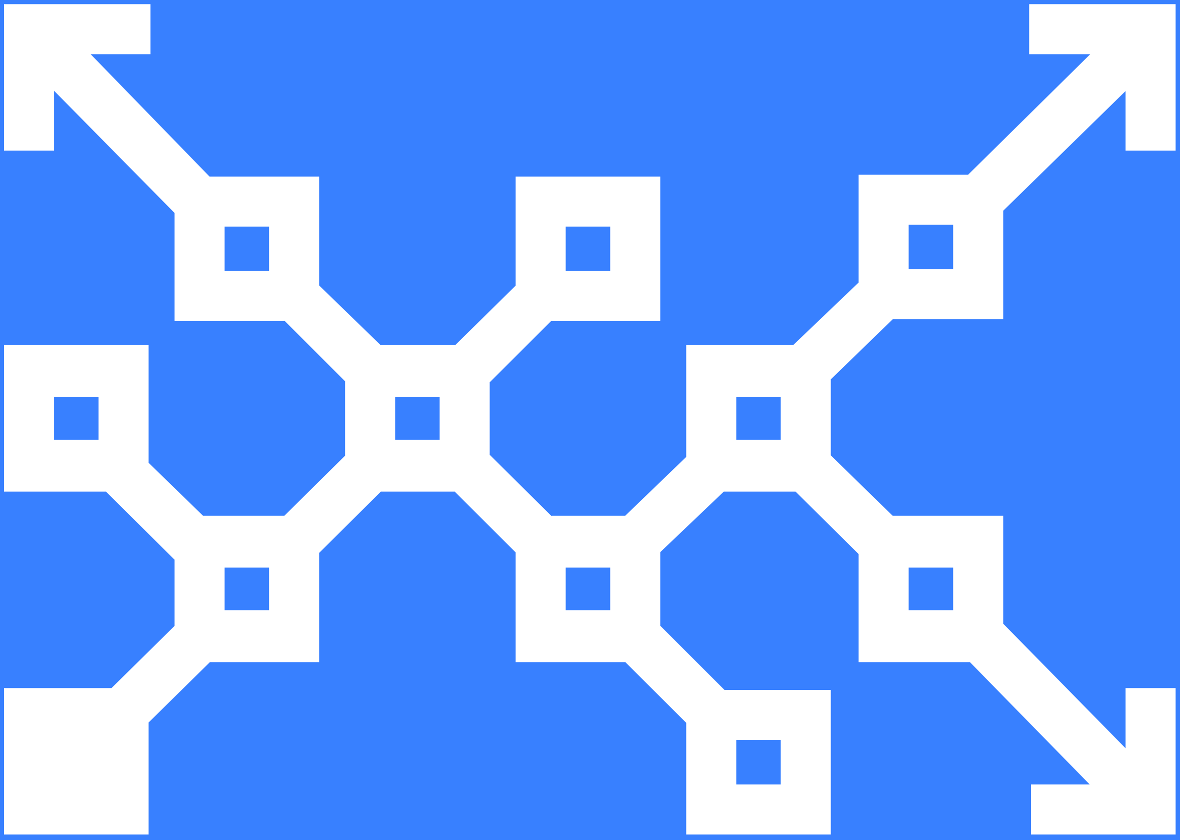
puppet

WHITE PAPER

DevOps - The Path to Continuous Compliance and Better Security



Contents

Introduction

With the proliferation of high-profile hacks, data breaches and ransomware, it's easy to feel insecure about your organization's security these days. You have to protect your organization and its reputation like never before — particularly as your infrastructure grows and diversifies, presenting a broader front for attackers.

Still, not all security issues have to do with purposeful hacks and attacks. For many IT teams, the challenge is maintaining strict rules and regulatory requirements for everything from credit card data to health information privacy. Failing to maintain compliance can put your organization at risk of everything from lost business to substantial fines — or worse.

In this white paper, we discuss how your organization can use IT automation to boost security and adopt continuous compliance as part of your DevOps workflow. With IT automation solutions like Puppet, you can:

- Establish and maintain a baseline of security rules and compliance requirements across your entire infrastructure, more easily than with any other solution.
- Improve collaboration between your IT ops and InfoSec teams.
- Incorporate a security review into the QA process, rather than leaving security considerations to the end of the software development cycle.
- Apply DevOps practices to your software development, and cut the time your team spends remediating security issues.

Create a baseline

Establishing a common baseline is a great way to improve security, because it forces you to define what you want and need. If you've already established a security and compliance baseline and you're enforcing it on your infrastructure, you are actually well ahead of most. Many teams don't do this, even though security experts tell us the surest way to detect a problem is to know what you have in the first place. For example, is your firewall supposed to open port 22 to the world — or just to your subnet? Do you have that documented? If it's not written down anywhere, how does your team distinguish between a policy and a security hole?

A good baseline includes things like firewall rules, SSH permissions, regulatory requirements and everything you promised your customers in order to be contractually compliant. You might well start by establishing firewall rules — not just universal ones, but rules that are appropriate to each type of system you manage. This can be notoriously difficult to do.

You may also need to broadly implement role-based access control (RBAC) — the approach whereby you assign privileges to groups and assign users to those groups, instead of assigning specific privileges to each user. This helps manage security more closely by giving you a clear distinction between group rules — admins vs. super users, say — plus the ability to quickly add or remove authority for individual users.

Maintain your baseline

Firewall rules and user privileges can be changed manually, but it's quite a job to manage even the simplest routine tasks over hundreds or thousands of servers, particularly after they've been deployed. Your VM templates or containers may have had great baseline security and compliance rules when they were initially deployed, but changes happen over time, and internally- or externally-made modifications can be difficult to detect and mitigate.

By establishing baselines for your different server types, you can make sure all your servers and containers remain compliant from the moment they're first deployed to the moment you turn them off.

The reality of security audits

Time spent logging and auditing systems is time that can't be spent on core tasks, and that time comes at a premium. Audits slow development as you and your team work to meet the needs of your security team and other internal auditors, rather than working on the needs of product delivery. This conflict of priorities quickly manifests itself as resentment from both the auditors and developers.

You need a clear way to engage your security team from the beginning of the software development cycle, rather than waiting until you're ready to deploy. Too often, security considerations hold up deployment, and result in a lot of re-work, too.

The 2018 State of DevOps Report showed that organizations in the more advanced stages of DevOps evolution place high emphasis on security practices. Teams at these higher levels of DevOps practice have **automated their security policies**, and they involve the security experts in their organizations very early in the software development lifecycle — actually, from the planning and design phases.

Breaking down these silos helps turn time spent auditing servers and logs into far more productive development time. When changes need to be made, the security team has a better understanding of what the operations team needs to do to make the changes happen. And the operations team can make changes across your entire infrastructure quickly and reliably.

Dealing with diminishing confidence over time

The reality is, your level of confidence in your overall security posture becomes less certain as time passes. You may well know what you put in place initially, and what you want to have in place continually, but you don't necessarily know if every node and server is really in compliance. Perhaps you feel confident about your baselines when you first deploy servers, containers and nodes, but what about a month out? Three months out? A year? The fact is, the longer it's been since you deployed, the more vulnerable your servers become to anything from outdated patches to seemingly innocuous manual changes.

Even if you and your team are diligent about performing regular updates and log reviews, that still leaves a lot of room for malicious hacks. No one wants to be the person who has to tell the world that a massive breach his or her company just found was actually planted two years earlier.

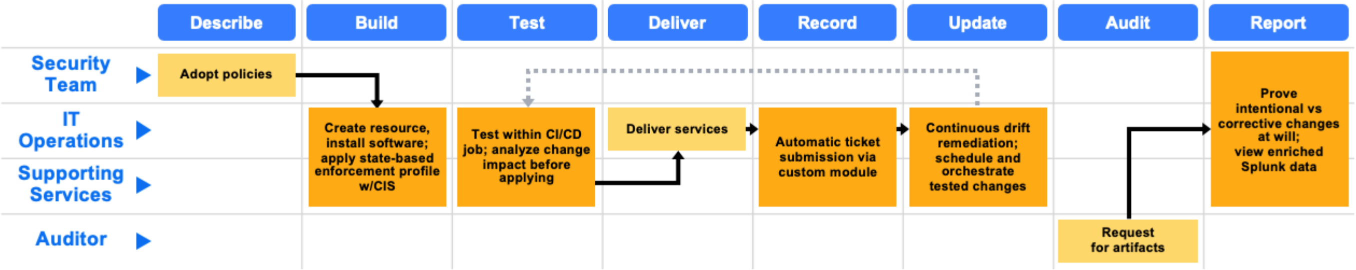
You also should consider the time involved in checking and proving compliance. Think of it this way: If you had to create a report for auditors showing who could log in to all the machines in your infrastructure — perhaps because of a breach — could you do it quickly? Would it take hours — or weeks? And what impact will this have on collaboration between ops and security teams as the clock ticks on?

The key is having visibility into infrastructure changes as they happen, and homing in on the types of changes that could be malicious. Having this visibility will enable your operations team to work more closely with your security team to provide a clear audit of everything that's happening.

How Puppet helps

As you continue to look for ways to reliably make your growing infrastructure more secure against external and internal threats, as well as more compliant with business and customer requirements, Puppet can help you get there. We can help you report and continuously enforce configuration policies and prove compliance. Puppet streamlines the compliance process, helping you manage configuration policies that are related to security and regulatory compliance, so you can reduce the risks associated with security misconfigurations and failed audits.

Puppet's simple declarative language, and its usefulness for managing everything in the data center and beyond, makes Puppet a key solution for collaboration between teams — including security teams. That's why Puppet is a natural part of the DevOps toolchain.


Describe
Build
Test
Deliver
Record
Update
Audit
Report
Security Team
IT Operations
Supporting Services
Auditor
Adopt policies
Create resource, install software; apply state-based enforcement profile w/CIS
Test within CI/CD job; analyze change impact before applying
Deliver services
Automatic ticket submission via custom module
Continuous drift remediation; schedule and orchestrate tested changes
Request for artifacts
Prove intentional vs corrective changes at will; view enriched Splunk data

With Puppet, you can:

- Define and deploy your policies as code. Build your organization's policies right into your configurations, and know they'll be deployed and enforced by Puppet. Among the thousands of modules on the Puppet Forge, you'll find security modules to help you get it done faster.
- Monitor and remediate drift. Continuously monitor your infrastructure for compliance and verify that changes to systems are correctly enforcing your policies. When differences are detected, Puppet automatically remediates systems back to their compliant state.
- Analyze the impact of changes before they happen. Easily view the effect and risk of any proposed Puppet code changes before they happen, so you can quickly approve and deploy lower-risk changes while enabling more scrutiny for higher-risk changes.
- Flexible job scheduling and orchestration. Schedule jobs to run at a specific time, such as during maintenance windows to improve workflows and eliminate running tasks in the middle of the night.
- Prove compliance. Avoid surprises and give auditors confidence with reports that clearly demonstrate compliance. Easily audit your infrastructure, reporting on the number of systems, how they're configured, and which configurations fulfill security requirements. Because reports let you trace intent and verifications, audits are faster and less costly.

Talk with us, and see how Puppet can work for you.

Resources

2018 State of DevOps Report

https://www.puppet.com/resources/report/2018-state-devops-report

Puppet Enterprise

https://puppet.com/products/puppet-enterprise/

Bolt

https://puppet.com/open-source/bolt/

Pre-built compliance modules from Puppet Forge

https://forge.puppet.com