SecDevOps Leveraging CI/CD

ATTACKIQ

The Challenge & Opportunity - SecDevOps

The adoption of CI/CD has changed how organizations develop and test software. First there was Waterfall, next came the flexibility and quick cycles of Agile, and now we see wide adoption of DevOps. Modern development organizations recognize the many benefits of this approach and continue to evolve and fine-tune their processes within this model. The rise of DevOps has increased focus on improving the Continuous Delivery, Continuous Integration, (CI/CD) and Continuous Deployment processes. Conventional software development and delivery methods are rapidly becoming obsolete. Historically, in the agile development process, most companies would deploy/ship software in monthly, quarterly, bi-annual, or annual intervals. Now however, in the DevOps era, weekly, daily, and even multiple times a day is the norm. This is especially true as SaaS is taking over the world and you can easily update applications on the fly without forcing customers to download new components. Often times, they won't even realize things are changing.

Development teams have adapted to the shortened delivery cycles by embracing automation across their software delivery pipeline. Most teams have automated processes to check in code and deploy to new environments. This has been coupled with a focus on automating the testing process along the way as well. And that is where SecDevOps can come into play heavily. But before we discuss that, lets briefly define the terms CI/CD.

Continuous integration focuses on blending the work products of individual developers together into a repository. Often, this is done several times each day, and the primary purpose is to enable early detection of integration bugs, which should eventually result in tighter cohesion and more development collaboration. The aim of ***continuous delivery*** is to minimize the friction points inherent to deployment or release processes. Typically, the implementation involves automating each of the steps for build deployments such that a safe code release can be done—ideally—at any moment in time. ***Continuous deployment*** is a higher degree of automation, in which a build/deployment occurs automatically whenever a major change is made to the code.

DevSecOps is the process of integrating a security-centric testing workflow as part of CI/CD. Thus, the integration of the AttackIQ FireDrill™ continuous security validation platform into an existing CI/CD process can ultimately facilitate the creation of a secure development best practices and automated testing methodology.

Leveraging Automated Security Testing in CI/CD

An existing AttackIQ customer integrated FireDrill™ into their continuous delivery and continuous integration (CI/CD) model, as it was important to ensure security was validated as part of their overall delivery model thus creating a DevSecOps workflow. While able to implement security controls via their current delivery model, testing was a tedious task not aligned with the model of automating everything.

Discover and Evaluate on Change

An extensive process of knowledge transfer between stakeholders of the DevOps and security team were the first step to implementation of automated security testing into the DevOps process. These were extensive sessions where a detailed review of the application took place. Knowledge transfer discussions covered hosting infrastructure, operating system environments, frameworks, tools and application services. Through these conversations it was also identified that security practices and remediation needed to be
integrated into the development workflow in order to unify previously disparate parties. While still a valuable transfer of knowledge, the rapid development pace of the team quickly eroded the value and relevance of the knowledge gained.

Historically, security testing has been limited to automated vuln scan and ad-hoc testing from a security engineer. While a critical piece, it does not provide an exhaustive security assessment of the security application with each release. An adversarial approach to evaluate and discover the application landscape provided continuous insight and understanding of the application environment. NMAP scans provided the team visibility into asset count, types, networks segments and flow control. Enumeration tactics on the local host provided insights into the application based on listening ports, environment variables and services.
Understanding risk includes quantifying and measuring the attack surface. Any kind of change or increase to the attack surface represents a security incident and should be reviewed. This approach provides
continuous baseline measurement to give a holistic overview of change. As new services or listening ports were exposed a discussion around the impact of such changes. If new ports and services were found as part of continuous testing the developer could identify it as a known change moving forward without
explicit approval, or simply address the Security Group modification that cause drift from a previously known state. This allowed organization identify and remediate in a matter of hours what would otherwise go unnoticed.

Move Security Decision Up to the Developer

There are many disparate security tools that could be leveraged to execute tests in this environment. As new gaps were identified, corresponding shell scripts were injected into the delivery pipeline. While a valuable indicator, identification of the protection failure prompted an extensive exercise by both the DevOps and security teams to identify the cause and mitigations strategies.

By codiftiyng those same scripts already written into the AttackIQ Scenario Framework testing can be automated and scheduled.
Every test extensively logs and parses steps taken via Activity Details providing a summary of findings based on context at execution time.

As part of the Scenario development process Mitigation Recommendation that are both generic and specific based on context are codified. This allowed users to have the context necessary to quickly identify what occurred while gaining insights into mitigation strategies based on that context. Protection failures identified utilize AttackIQ Cloud Alerts to generate tickets directly into the JIRA backlog. This moved the identification and recommendation all the way up to the developer to ensure gaps are prioritized and addressed accordingly.

Objective Measure of Best Practices

The organization approached the inclusion of all stakeholders in the software development lifecycle in the form of governance via best practices. Every stakeholder from compliance to security was able to provide the expected best practices. While the approach provided baseline expectations, any deviation required review and approval halting the otherwise continuous delivery model. Developers required the ability to measure fulfillment of criteria for every single modification and rapid identification of any deviation for approval.

Best practices entailed areas which encompassed base image specifications, GDPR, segmentation of application/services/networks, least privileged accounts, secrets management, third party services and AWS infrastructure. Rather than audit the configuration of least privileged roles or service, AttackIQ assessment codified and represented the best practices as established by a given stakeholder. I AM role scope of accepted privilege were defined via Assessments. I AM roles with privileges not meeting best practices can be identified as part of the testing process. The developer would adjust the I AM role to fit the organizational best practices. If the increase in privilege is by design, individual stakeholder can be directly communicated to and testing within their governance adjusted.

The Solution (Why AttackIQ?)

The customer provides collection and analysis of big data sets for large organizations and governmental bodies. They provide applications, services and bespoke consulting solutions to help their client manage and make use of extremely large data sets. Their products consists a variety of components managed globally on AWS infrastructure. This enabled a variety of components to be dynamically deployed and scaled on-demand based on customer load. While insights are delivered via UI, full functionality is API driven enabling rapid development of microservices for their core business offering. The customer uses CircleCI and the full Hashicorp stack to automate testing and delivery from git commits to production. The customer sought security testing process that would fit in this framework and provide the ability to programmatically implement security testing and validation.

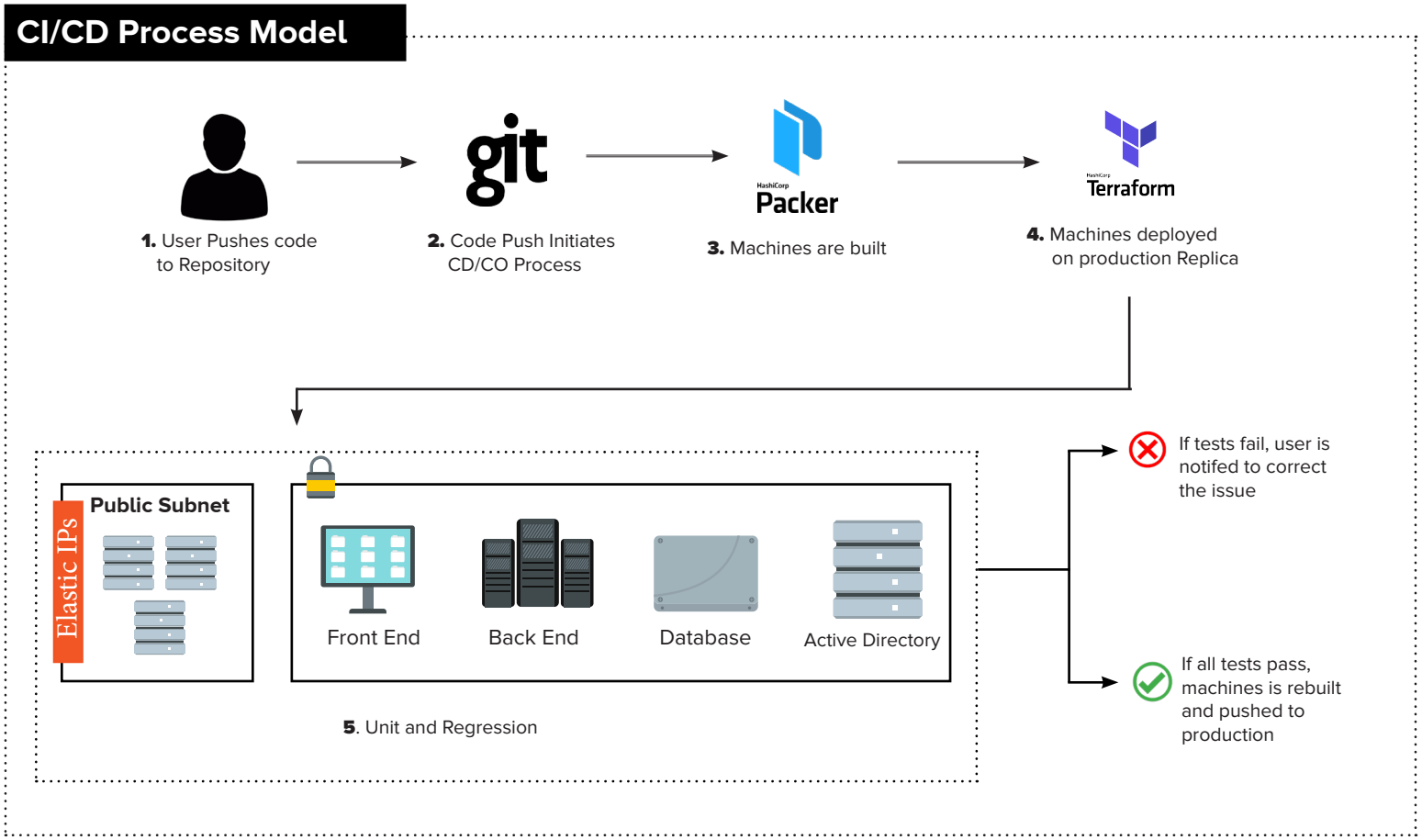
The customer recognized that AttackIQ enabled them to implement security testing and validation at the earliest stages of the software development life cycle. With all FireDrill functionality available via REST API and SDK, AttackIQ fit into their model of automate
everything and the flexibility allowed for testing across the multiple layers of their DevOps process and products.

AttackIQ provided the ability to write custom scenarios which enabled the organization to incorporate previously written scripts and common security tools into their DevSecOps program. The open platform enabled not only development of attacks for systems proprietary to the organization, but also allowed them to develop custom integrations that would ensure that the security pipeline was validated continuously.

Action

Prior to deploying AttackIQ, the organization utilized CircleCI (circleci.com) to manage the deployment process from GIT commit through production roll-out. Each new commit initiated a process that would build a full production replica environment via Terraform (https://www.terraform.io/) while incorporating the new code.

This replica environment included the same set of security controls implemented in production environment. Based on this methodology, the application functionality was receiving a fully automated QA cycle, but security testing was ad-hoc and executed manually by a mix of the security team and user validation.



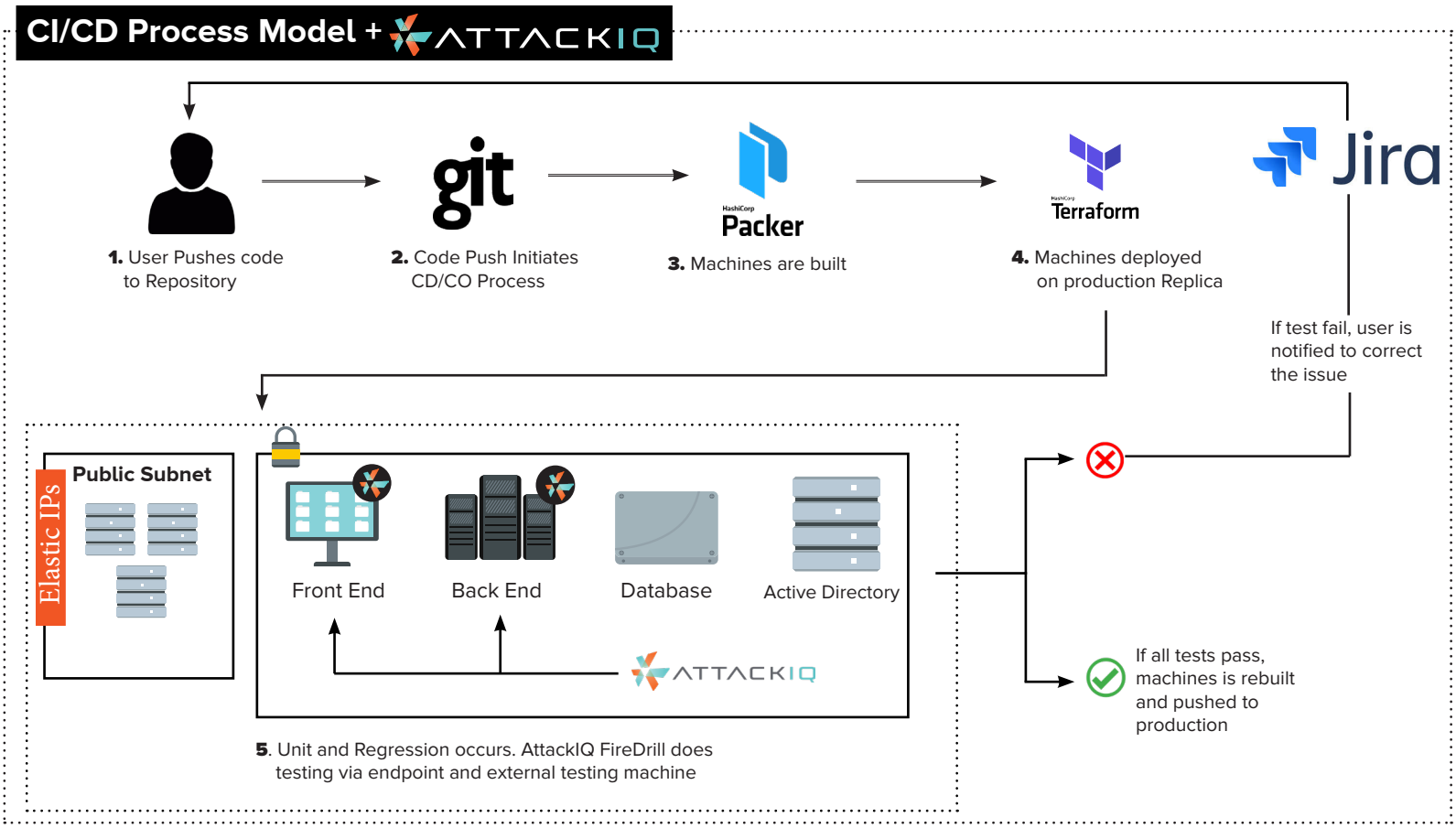
The organization recognized a need for a program to automate the security implementation validations along with the unit and regression testing. AttackIQ was integrated into this process by being deployed in the replica environment on the newly generated hosts, as well as a standalone agent deployed in the replica environment. This allowed the organization to programmatically validate their prevention and detection capabilities on the endpoint which included:

Attacks

- Advanced Persistent Threats (APT)
- Privilege Escalation
- Lateral Movement
- C&C
- Ex-filtration
- AntiVirus

Validation

- Compliance Testing
- Access/Routing/Availability
- Content/Web Filtering
- Firewall
- Network and Host IPS/IDS
- SSL Certificates
- AWS Infrastructure (S3, AMI, and VPC)
- AWS IAM roles



By exercising their environment, the organization was able to validate their ability to prevent the techniques they expected to. Integrations were able to validate the entire security pipeline. This ensured that the flow of logging, analysis, detection and reporting were all working as expected.

The standalone agent was then leveraged to be representative of an outside actor communicating to the newly deployed application. This allowed the team to codify and continuously validate vulnerabilities such as XSS that were discovered as part of their development and continuous testing process.

The platform provided this organization an opportunity to foster a culture of security within the application development team. Each time a new security vulnerability is discovered, the application developer who introduced the issue will write a customized script to replicate and exploit the introduced security flaw. This allows the organization to curate a library of known vulnerabilities which are then used for automated security regression testing. The organization has built a culture of security awareness within the development team by having the developers recreate all newly discovered security flaws.

The platform provided this organization an opportunity to foster a culture of security within the application development team. Each time a new security vulnerability is discovered, the application developer who introduced the issue will write a customized script to replicate and exploit the introduced security flaw. This allows the organization to curate a library of known vulnerabilities which are then used for automated security regression testing. The organization has built a culture of security awareness within the development team by having the developers recreate all newly discovered security flaws.

Results / Conclusion

By integrating AttackIQ into their current CI/CD process, this organization was able to establish a baseline for their security controls implementation and the entire security pipeline process. This has allowed the full stack of the application's security posture to be continuously measured over time. This allows the team to rapidly deploy changes into the environment knowing that their security capabilities are being measured and monitored. Any changes that may affect security posture are noticed prior to production deployment and appropriate modifications can be made.

Any issues found as part of the build process are not only raised for review in GIT by developers, but JIRA tasks tickets with specific recommendations are generated. This has been successful in building a culture of security in the developers workflow as security testing and remediation tasks are included into their work-flow used today. In establishing this progress developers can remain left of bang when it comes to security issues and address them in a matter of hours -- not days. Instead of making it an approval process we are moving it all the way up to the developers by ensuring they have the context to respond, manage and con-tinue to deliver value.

Resources

a. https://www.attackiq.com/blog/2017/03/06/its-time-to-operationalize-devops-infrastructure-security/

About AttackIQ

AttackIQ was founded to bring automated assurance to security. The rising corporate cost of security debt has proven that the standard approach to IT Security needed to change. The existing methodologies were based on too many assumptions and it is now time to give data-driven assurances to this billion-dollar industry. The AttackIQ mission is to challenge the existing security landscape and enable organizations to measure the resiliency and efficacy of their security posture, gain better visibility into their full IT security infrastructure, and make better data-driven security decisions.

Validate your security infrastructure today with a free trial:

https://www.attackiq.com/register/

+1-415-413-7353
INFO@ATTACKIQ.COM
www.attackiq.com
5703 OBERLIN DRIVE, STE. 306, SAN DIEGO, CA 92121