Secure Software Lifecycle in the Continuous Deployment pipeline

Vishwas Manral (vishwas_manral@mcafee.com)
Chief Cloud Architect & Head of Container Security
McAfee

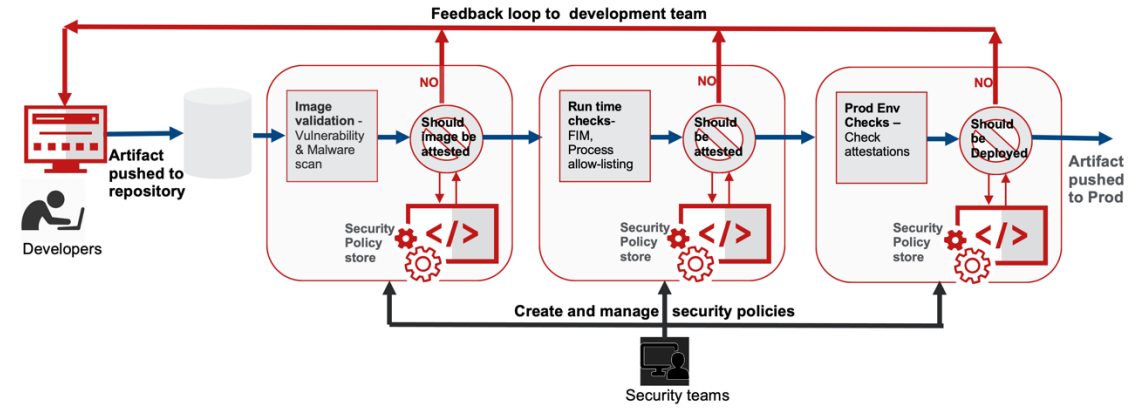
In Area #2, NIST strives to address the guidelines and best practices around development of software for purchase in the workshop. It is important to address not just legacy and on-premises software development processes, but also those Continuous Integration / Continuous Deployment environments. These environments are hosted by the USG directly or by Cloud Service Providers. U.S. Federal Departments and Agencies are rapidly embracing the cloud. It is important protections are in place for cloud development, including 3rd party open-source usage as well as establishing secure software lifecycles with continuous development processes.

Why focus on Continuous Deployment

Continuous Deployment (CD) process enables value to be deployed immediately into the operational environments by enabling the deployment of artifacts once they are built. These CD environments requires the Automated and Continuous Testing and security to allow for validated, secure, correct, and stable deployment to production environments. While it is best to catch security issues as soon as the code is being developed – it is easier said than done. In a world where applications are being built in diverse environments, Security teams are often unaware and less familiar with where the code pipelines are run by the development teams, and where the code for the same has come. The place where security teams can first insert themselves is on the CD pipeline. In cases where COTS software is to be deployed its often wise to do security checks in the CD pipeline, to validate the security of the software to be deployed.

CD pipeline security steps

CD pipeline is triggered as soon as the artifact is deployed into the artifact repository. The checks include



Repository checks
Periodic checks need to be done to the repository to see if it is correctly configured. These checks can include, a) not allowing developer credentials rights to push artifacts, instead only tools should allow that; b) not allowing overwrite of artifact versions, as this can lead to other tools picking up wrong versions of artifacts; c) not allowing change of change to repository configurations and permissions without oversight.

Build checks
Functions registered to scan the deployed artifacts for build time checks are triggered as soon as an artifact is pushed. This can trigger a scan of the artifacts for:

1. Vulnerability assessment of the built package
2. Malware scans of the build package

Infrastructure-as-Code (IaC) Deployment template checks
In dynamic environments infrastructure is created using IaC deployment templates like CloudFormation, Terraform, Helm charts etc. These templates define the infrastructure on which the artifact is to be deployed, as well as the permissions required for the artifacts to run. IaC policies which include security posture checks like open security groups, network capabilities needed are validated. Once the policies are verified the deployment can proceed.

Run time preproduction/ staging checks
In preproduction environments the templates are deployed and run. The run time behavior of the application is monitored this includes checking what processes are run, what system calls are made, what files are modified, what network connection are made by the running artifact. Checks and updates based on previous versions of the artifacts need to be highlighted and archived. It is at this stage that checks are Fuzz testing, availability testing and overall system sanity testing are done. Validation policies are checked for the various functions, based on the same attestation is either done or disallowed and the rightful developers notified of the reasons of non-attestation.

Deployment controller checks
Deployment controllers and admission controllers allow functions to be run before any template is deployed. The functions can check the attestation of every template that is deployed in production environments. Only the templates with the matching policies are deployed. Typical policies could be only allowing deployment of artifacts when the right attestations are in the system. Checks need to be made that only artifacts from validated repositories are deployed. Based on Controller polices the controller can modify artifact templates and configurations before doing any deployment.

Periodic checks
Every policy or configuration change on the repository needs to be checked periodically and every access logged. Policies for every security check needs to be looked at. Every admission controller policy deployed needs to be validated and periodically checked and logged.

Attestation
After any security function is run, crypto attestation of the artifact needs to attach to the artifact. Attestation should be done for individual binary and one that cannot be ported for another version of the same binary. Different functions should have different attestation keys. An example policy for attestation could be to attest any artifact only if there is no High Severity vulnerability or any known Malware that is found. In case the policy checks are met the artifact is signed with the right attestation. Every template can have multiple attestations each for a different function. All vulnerability data is archived and managed for future use. If the attestation policies are not met the attestation is not done, though the artifact is still stored in the repository.

If selected, McAfee requests Vishwas Manral, Head of Container Security and the Chief Cloud Architect at McAfee participate on the panel. Previously Vishwas was the CEO of Nanosec, a container security acquired by McAfee. Vishwas is the co-chair of the Cloud Security Alliance (CSA), Silicon Valley, and leads the effort on Serverless Security for CSA Global. Vishwas was the Chief Technologist at HPE & the founder of Ionos Networks. He is an advisor to security companies such as Bootup Ventures and H.A.C.K. a government accelerator for Cyber Security in Bangalore.