Brian Sowers

12/17/2022

DevSecOps

1. Explain the three security layers that must be part of a DevSecOps and effective API security design.

An effective API security design will have the following security layers: Application security, system security, and network security, . Application security is defined as “security approaches taken during a product development lifecycle to ensure security goals and prevent security gaps being built into an application” (API Security in the Enterprise, 5). System security is defined as “security approaches that control access to a computer system’s resources, especially data and operating system files” (API Security in the Enterprise, 5). Network security is defined as “security approaches to prevent and monitor unauthorized access, misuse, data modification or denial of access to an enterprise computer network and network-stored resources” (API Security in the Enterprise, 5). Furthermore, to achieve security goals and effectively prevent attacks across the three security layers, API security must uphold the following principles: data validity, integrity, confidentiality, availability, authentication, authorization, audit, and non-repudiation. API security is needed as an insecure API, which is commonly used to facilitate third-party services and access for trusted partners, leaves open attack vectors such as cross site scripting, SQL injections, MiTM attacks, and dependency vulnerabilities. The best way, currently, to address these security shortcomings is to assume a DevSecOps approach. A holistic DevSecOps approach takes 3 core actions: democratization,

collaboration, and industrialization. Democratization is the process of assigning APIs as they are designed pre-approved and pre-tested security policies. Collaboration occurs when development, security, and operations teams get together and discuss the API cycle with their respective inputs. Industrialization is the process of automating security related workflows for the purpose of continuous delivery and security testing processes. (API Security in the Enterprise, 11).

2. In your own words, explain two or more obstacles that a DevSecOps development team would need to overcome in the future.

Two obstacles facing a DevSecOps team in the future will be security keeping pace with development, and the inherent risks associated with containers. To keep pace with development, security will need to continue automating as much of its workflow as possible. Since business operations can't always slow down development to do extra security checks, you will have to "accelerate security efforts to match the pace of development" (Bellairs, 2018). Security Orchestration, Automation and Response (SOAR) is a current solution that can "automate enrichment of indicators by querying different threat intelligence tools for context. By running this playbook at the outset of incident response, security teams have the enrichment data available for study within seconds, shaving off wasted time that can be used towards proactive investigation" (Goh, 2018). It will be necessary for the security community to create similar automation tools in the future to keep pace with development. One reason why development efforts are outpacing security efforts is because developers are using containers. Containers are "the modern way of packaging, sharing, and deploying an application" (Chen, 2019). They contain all the necessary dependencies for an application and are platform-agnostic, meaning it is readily portable. However, the architectural make-up of containers creates a significant attack

surface. A container is an OS-level virtualization that shares both the hardware and the host's OS kernel. This sharing causes a weak trust boundary between the containers and the host. Consequently, it is easier for kernel exploits to gain privilege escalation and allow the compromised processes to gain control outside its intended namespaces (Chen, 2019). There are current projects such as Google gVisor, Amazon Firecracker, and OpenStack Kata attempting to create stronger trust boundaries, but none have fully resolved this problem yet.

Works Cited

"API Security in the Enterprise: How a DevSecOps Approach Delivers Reliable API Security." *PanAcademy,* 2017. Accessed: 12 December 2022. http://panacademy.net/CourseFiles/sbo/DevSecOps_API_Security.pdf

Bellairs, Richard. "DevSecOps Best Practices – 3 Tips To Use Today." *Perforce,* 17 October 2018, https://www.perforce.com/blog/qac/devsecops-best-practices-3-tips-use-today.

Chen, Jay. "Making Containers More Isolated: An Overview of Sandboxed Container Technologies." *Unit 42,* 06 June 2019, https://unit42.paloaltonetworks.com/making-containers-more-isolated-an-overview-of-sandboxed-container-technologies/.

Goh, Jane. "Security Orchestration Use Case: Automating IOC Enrichment." *Palo Alto Networks,* 09 October 2018, https://www.paloaltonetworks.com/blog/security-operations/security-orchestration-use-case-automating-ioc-enrichment/.