8 Best Practices for Securing Containers from Source to Run

Streamline your Container Security with Snyk and Sysdig

Traditional security approaches are incapable of handling the distributed and ephemeral nature of containers. A different approach is emerging that focuses on secure code, runtime scanning, and threat prioritization. Here are 7 best practices for implementing this modern approach to container security:

01 Code security

Source code is where developers have the most control, so application security should start at home. Source code scanning tools offer an efficient way to accomplish this by:

- Working directly in the tools that developers use day in, day out
- Spotting issues early in development
- Integrating security scans into existing workflows so developers can automate the process of finding and fixing issues
- Providing ongoing code monitoring

02 Open source security

Open source code introduces security risks since you're relying on other developers to maintain the package. To mitigate these risks, developers need to be able to identify and fix issues in the packages they're consuming. Software composition analysis (SCA) tools help developers track dependencies by flagging any issues by referencing a vulnerability database.

03 Image security

Containers provide a standardized way to package applications, but container images themselves can be a source of vulnerabilities. Image security requires:

- Identifying a trustworthy, minimal base image
- Automatically scanning it in the CI/CD pipeline
- Monitoring running images for newly discovered vulnerabilities or updates

04 Runtime security

Containers are opaque, and 44% live less than five minutes, so securing running containers can be tricky. Look for a runtime security tool that:

- Monitors containers in a lightweight way (such as using audit logs)
- Delivers insights and context about security events
- Allows you to automate policies and response actions

05 Network security

Enterprises are increasingly moving to a zero-trust approach to network security, but classic firewall approaches fail in dynamic, cloud native environments. Network security needs to go beyond the physical communication layer to employ effective security policies using native controls like Kubernetes NetworkPolicy. Look for solutions that allow you to:

- Map network topology
- Establish baseline policies
- Automate new policy generation
- secrets are not committed by mistake

06 Kubernetes and cloud security

Infrastructure as code (IaC) introduces the possibility for issues such as:

- Misconfigurations that can lead to overly permissive workload configurations
- Configuration changes that end users might implement to make it easier to carry out tasks (but might not be secure)
- Default configurations that might not be in-line with organizational security protocols

IaC security focuses on detecting and fixing these configuration issues as early as possible, often using a policy engine such as Open Policy Agent (OPA) for governance and compliance.

07 Vulnerability prioritization using runtime signals

Developers can be overwhelmed by the number of vulnerability alerts they receive from security and operations teams. Prioritization techniques can identify the software packages actually executed in the running containers, allowing developers to instantly eliminate up to 95% of the vulnerabilities they would otherwise have to consider.

08 Security from code to runtime with Snyk & Sysdig

Snyk finds and automatically fixes vulnerabilities in your code, open source dependencies, containers, and IaC. Purpose-built to work with the process and tools developers use every day, Snyk cultivates a collaborative security approach that scales and aligns with the iterative DevOps model, even as security teams remain lean and stable.

Together, Snyk and Sysdig help developers secure code and containers in development, protect the runtime Kubernetes environment, and deliver feedback and visibility from production back to developers, eliminating the noise of container vulnerabilities.

To learn more about how the Snyk+Sysdig integration helps development teams secure containers from within the development pipeline, **read the full checklist "Container Security from Code to Runtime."**