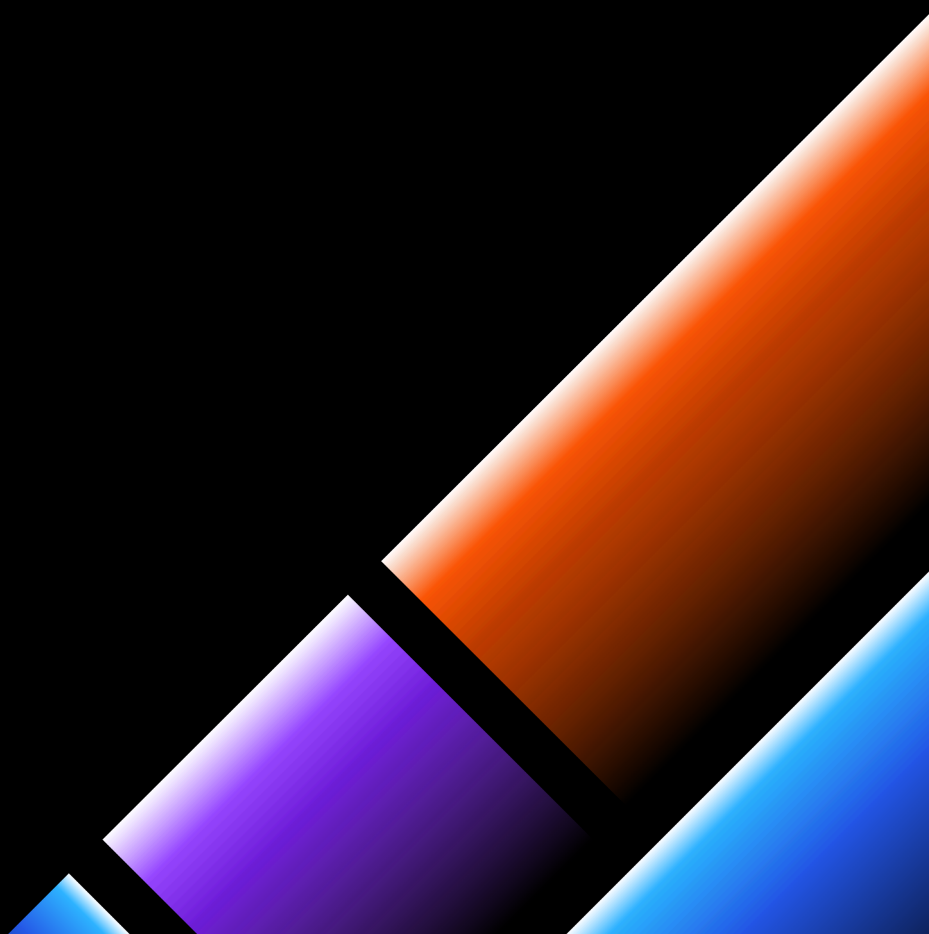
RAPIDFORT

WHITE PAPER

Your Path to Near Zero CVE Images

A Practical Approach



Executive Summary

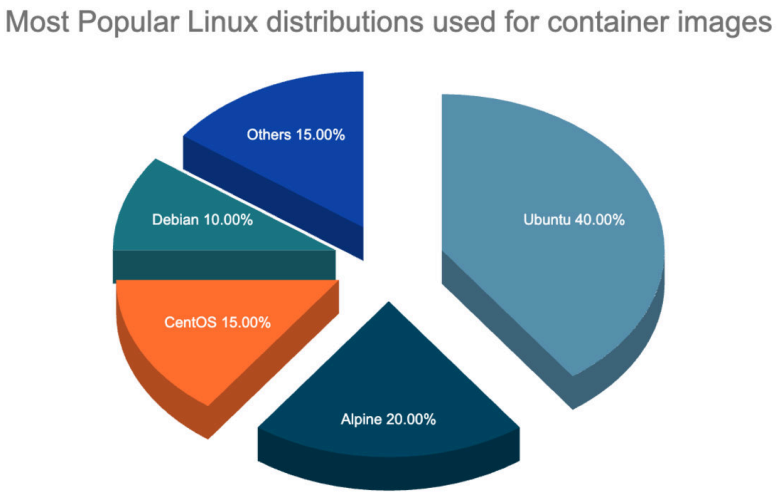
In today's cybersecurity landscape, the focus has shifted left, emphasizing the importance of securing software earlier in the CI/CD pipeline rather than waiting until after deployment. One critical strategy in achieving this is using near zero CVE OS, framework, and third-party application images. These are container images that are regularly patched and hardened to eliminate known Common Vulnerabilities and Exposures (CVEs), making them essential building blocks for modern containerized software development.

This guide will explain how to choose near zero CVE images and the factors to consider when selecting a vendor.

Start with a Mature and Widely Supported Operating System

Start by selecting a vendor that curates images on a well-supported, well-maintained, and widely adopted OS distribution. It's crucial to choose a distribution with an active and broad community that regularly identifies vulnerabilities, contributes patches, and provides long-term support, CIS benchmarking, and other security and stability measures.

Ubuntu and Alpine's dominance reflects their widespread adoption of container use cases. Ubuntu provides a familiar base, and Alpine offers a very lightweight option. CentOS and Debian also have a significant presence in the container ecosystem.



Take Ubuntu as an example: it controls **40% of the Linux market and powers over a quarter of all websites worldwide.** Its widespread adoption and active community result in continuous testing, rapid vulnerability detection, and timely patching. Known for its stability, security, and ease of use, Ubuntu comes with built-in protections and regular updates. Users receive security patches for over 25,000 software packages, along with long-term support for security, hardware, and bug fixes. The extensive Ubuntu community provides comprehensive technical documentation and support, making it easy to access a vast software library and install additional applications.

Avoid operating systems that are a "community of one"—those maintained by a single company without broader market adoption. This creates vendor lock-in and the potential for pricing abuse. If the company experiences liquidity issues, the OS and its unique packages are left unsupported, creating significant risks.

Selecting Scratch vs. Curated Base Images

Scratch images are suitable for greenfield projects, where a few coders working closely together start from scratch and add only the specific packages needed for their code. This approach can lead to leaner, more secure images. But in practice, it entails continuous updates as dependencies change with every code drop, and the probability of breaking the image over time increases with every iteration.

In addition, Security Technical Implementation Guides like the CIS benchmark or DISA-STIG have hundreds of configuration recommendations, so hardening and auditing a Linux system manually can be very tedious. Ubuntu Security Guide (USG) is a tool available with Ubuntu 20.04 LTS that greatly improves the usability of hardening and auditing and allows for environment-specific customizations.

The reality is that most coders are working in brownfield environments, dealing with legacy codebases or ones built by developers who may have long since moved on. Understanding the precise dependencies in these cases can be challenging and time-consuming.

In these cases, it's often better to start with curated base images that include all the necessary packages and have already been patched. This reduces the risk of runtime breakage due to missing dependencies. One can still optimize these images later with tools that harden them based on runtime profiles, ensuring more reliable results than manual dependency tracking. With every iteration of the code, profiling the runtime behavior of a container across several instances will enforce a relevant and accurate dependency tree and ensure that the workload is hardened effectively for subsequent builds.

Choose an OS distribution with Trustworthy Security Advisories

When selecting an OS distribution, it's crucial to choose one with transparent and reliable security disclosure, tracking, and remediation processes. Linux distributions like Ubuntu or Red Hat benefit from large, diverse user communities where a broad community reports vulnerabilities and patches are reviewed by a wide range of contributors and pushed through mature QA cycles. This broad involvement ensures an accurate, transparent assessment of security risks.

In contrast, a distribution managed by a single company poses a conflict of interest. When the same team that builds the distribution is also responsible for reporting its vulnerabilities, objectivity may be compromised. It's like asking a "wolf to guard the sheep." A large, active, third-party community brings greater trust, accountability, and credibility to security and CVE reporting, helping ensure you're aware of all potential risks.

Use Widely Supported Distributions Recognized by Mainstream Scanners

Image scanning products are only as capable as the components that make them operate—a scanner to detect software components, a database against which detected components can be evaluated, and a front-end tool to view and report on results. Image scanners may not detect every third-party component in your scanned codebase. Vulnerability databases may also not document particular libraries bought from small vendors or unpopular open-source projects. If your "near zero CVE" image vendor uses an obscure or custom distribution, scanners may fail to recognize the container's origin, potentially skipping the scan altogether.

This oversight prevents accurate reporting of vulnerabilities, making it difficult to validate, reconcile, and trace the "near zero CVE" claim. Opting for a distribution that circumvents traditional scanning methods can create a false sense of security, especially if the scanner doesn't support or recognize the distribution in question. To ensure comprehensive security assessments, choose distributions that are widely supported and easily identifiable by industry-standard scanning tools.

Final Thoughts

Cybersecurity is paramount, and adopting near zero CVE images is a critical step toward building robust software systems. Starting with a mature and widely supported operating system distribution, using well-curated images to manage dependencies, and choosing systems with transparent CVE advisories with broad scanning support are key steps to building a secure software foundation. Additionally, selecting near zero CVE images that allow for flexibility in adding necessary packages ensures that your software remains adaptable to changing needs. By following these guidelines, you can confidently build modern, containerized applications that are both secure and efficient from the ground up.

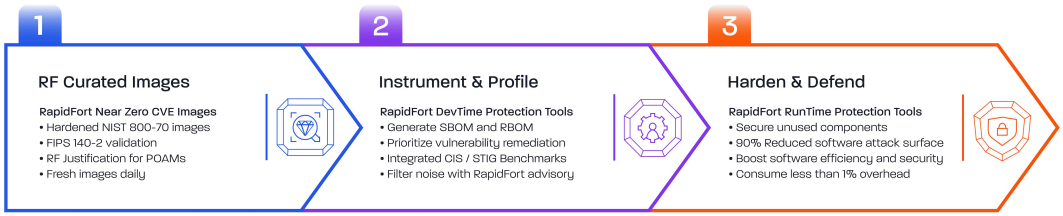
About us

At RapidFort, we are dedicated to empowering organizations to enhance their container security and vulnerability management practices. As a leading provider of cloud-native cybersecurity solutions, we specialize in addressing the unique challenges of securing modern software environments. Through our innovative technologies and expert guidance, we assist our clients in safeguarding their software infrastructure against evolving threats. We trust that the insights presented in this white paper will aid organizations in gaining a deeper understanding of their security landscape and taking proactive steps to fortify their defenses.

To learn more about the RapidFort platform, please visit: https://www.rapidfort.com/

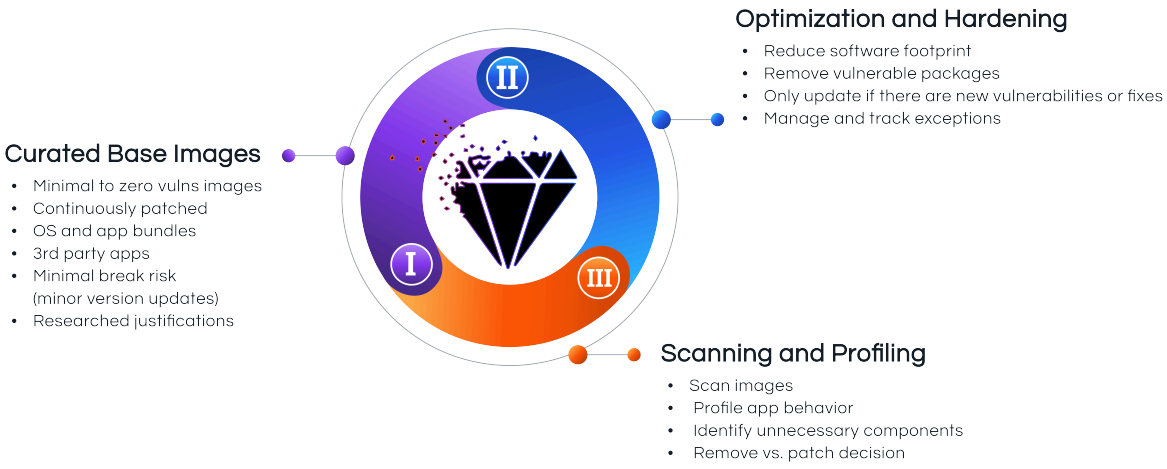
To learn more about RapidFort Curated Images, please visit: https://hub.rapidfort.com/

Three Easy Steps to CVE Remediation
One Unified Platform



RapidFort Flywheel

RapidFort is a comprehensive Software Attack Surface Management (SASM) platform that automates vulnerability remediation without code changes:



www.rapidfort.com

RAPIDFORT