**Docker Tightens Security Over Container Vulnerabilities**

**Docker unveils three ways to make containers more secure, especially when code is changed during its update cycle.**

[](http://www.informationweek.com/cloud/cloud-vs-on-premises-6-benefits-of-keeping-data-private/d/d-id/1323089)

**Cloud Vs. On-Premises: 6 Benefits Of Keeping Data Private**

*(Click image for larger view and slideshow.)*

Docker has added a hardware signing feature, YubiKey, a USB device, for developers of container images and updates to ensure that the code they file to a repository arrives untampered with and intact.

It was one of three major container security improvements added to the Docker Platform announced November 16 and 17 at[DockerCon Europe](http://europe-2015.dockercon.com/) in Barcelona.

Docker has already implemented [The Update Framework](http://blog.docker.com/2015/08/content-trust-docker-1-8/) (TUF), a method of confirming that a digital signature applied to a container image in a repository matches the signature on the code arriving at an enterprise's Docker system. TUF is tougher than mere public key encryption because it can restore the security system's integrity, even if the signature-assigning server is compromised. Docker calls its system [Docker Content Trust](http://www.informationweek.com/cloud/infrastructure-as-a-service/docker-applies-secure-key-model-on-code-handling/d/d-id/1321724).

At Barcelona's DockerCon, Docker announced a new layer in the code- and identity-confirming process. Developers and system administrators can use a keychain fob or [YubiKey 4](https://www.yubico.com/2015/11/4th-gen-yubikey-4/), plugged into the USB port of their laptop or workstation, to upload their unique identifier to the container. As the code moves along its journey to a production system, that identifier continually ensures the recipient that only the intended hands have touched the code.

[Yubico](https://www.yubico.com/)'s YubiKey 4 is the current state of the art.

Its two-factor authentication requires the device to recognize the user's fingerprint before it will issue the user identification to a containerized application, said Scott Johnston, senior vice president of product at Docker. Even if a developer's Yubikey were lost or stolen, it would be worthless without the correct fingerprint.

[](http://www.informationweek.com/cloud/platform-as-a-service/docker-tightens-security-over-container-vulnerabilities/d/d-id/1323178?image_number=1)

(Image: Wavebreak/iStockphoto)

Two-factor authentication makes it extremely difficult for someone to abduct code in transit or spoof it to deliver malware to the intended recipient, Johnston said.

In another move, Docker has added image scanning to the Docker Hub.

As users assemble container workloads using source code from publicly available repositories such as Ubuntu's, Docker image scanning checks it for correct release number and vulnerabilities. If the code is a release with known vulnerabilities, the downloader and the supplier are notified, with the latter expected to fix it.

With image scanning, "IT organizations can rely on Official Repos (like the Ubuntu repository) as a curated source for secure, high integrity content," Johnston said.

Previously a system admin would have to know what information on vulnerabilities had been published by each Linux distributor and other sources of online code. With Docker Hub providing scans, independent software vendors can now deliver what recipients will regard as secure content because the code origins have been confirmed. The Docker Hub downloads approximately 4,000 containers a minute.

**[Want to learn more about Docker's previous moves to shore up container security? See**[**Docker to Defang Root Privilege Access**](http://www.informationweek.com/software/enterprise-applications/docker-to-defang-root-privilege-access/d/d-id/1321023?itc=edit_in_body_cross)**.]**

In a third security improvement, Docker's latest 1.9 Experimental release (the early preview version) enables operations managers to assign privileges by user group for each container. For the first time, the containers have been separated from root access on the host. Only the Docker daemon has root access, and that access to the Docker daemon can be restricted to a defined set of system administrators.

In the past, each container had root access to the host, meaning it could access all the host's resources if its code instructed it to do so. By using Linux namespaces to separate the container from the Docker daemon, this old vulnerability in container operations is walled off from further mischief.

In addition, IT operations can establish granular access-control rights, giving explicit permission to certain departments or teams to use certain Dockerized services. This new control prevents one organization from inadvertently being given control over another organization's application services, Johnston said.