**DOCKER BENCH**

* **Ngữ cảnh**

Docker giúp phân phối nhanh chóng, nhất quán các ứng dụng; cho phép chạy nhiều khối lượng công việc hơn trên cùng một phần cứng. Docker tiện lợi nhưng nó cũng tiềm ẩn nhiều lỗ hỏng bảo mật, như là nó cho phép kẻ tấn công có thể thực thi code từ xa hay là leo thang đặc quyền…

Do đó, điều quan trọng là phải bảo vệ Docker Engine chống lại các mối đe dọa có thể xảy ra, đặc biệt nếu chúng ta đang chạy một máy chủ Docker trong sản xuất, thương mại.

Vậy cần phải làm gì để có môi trường Docker an toàn

* **Docker Security Cheat Sheet**

1. **Keep Host and Docker up to date**

Điều đầu tiên luôn là cập nhật các bản vá mới nhất cho Host và Docker để ngăn chặn các lỗ hỏng đã biết.

1. **Do not expose the Docker daemon socket (even to the containers)**

Docker thường hoạt động như là một client giao tiếp với tiến trình Daemon. Daemon có đặc quyền rất cao, thường là quyền root. Vì vậy nếu user bất kì có quyền truy cập vào Daemon đồng nghĩa với user đó có được quyền root.

1. **Set User**

Cấu hình container cho người dùng không có đặc quyền là cách tốt nhất để ngăn chặn các cuộc tấn công leo thang đặc quyền.

1. **Limit capabilities**

Ta chỉ nên set cho Docker các đặc quyền vừa đủ, không nên cho nó quá nhiều đặc quyền.

1. **Add –no-new-privileges flag**

Luôn chạy Docker với cờ --no-new-privileges để ngăn chặn leo thang đặc quyền

1. **Disable inter-container communication (--icc=false)**

Theo mặc định, inter-container communication (icc) được bật - có nghĩa là tất cả các container có thể nói chuyện với nhau. Điều này có thể làm lộ thông tin không mong muốn cho các container khác.

1. **Use Linux Security Module (seccomp, AppArmor, or SELinux)**

Ta nên sử dụng các security module có sẵn trên linux kernel như là seccomp, AppArmor, SELinux

1. **Some other rules**

* Hạn chế tài nguyên (Tránh tấn công Dos)
* Set file system và volume thành read-only
* Sử dụng các công cụ phân tích tĩnh
* Đặt mức ghi log thành ít nhất là INFO

Ý thức được sự cần thiết của việc đảm bảo cho môi trường Docker đủ “cứng” để có thể đối mặt với các mối đe dọa, đội ngũ phát triển Docker đã tạo ra Docker Bench như là một công cụ kiểm tra bảo mật mã nguồn mở. Docker Bench for Security là một automated script giúp ta có thể tìm ra các vấn đề với cấu hình của mình. Script scan host để tìm điểm yếu trong việc thiết lập Docker Engine.

* **What does Docker Bench for Security do?**

Docker Bench for Security cho phép quản trị viên xây dựng security baseline trong quá trình triển khai Docker. Công cụ quét bảo mật này nên được sử dụng để làm cứng các host configurations.

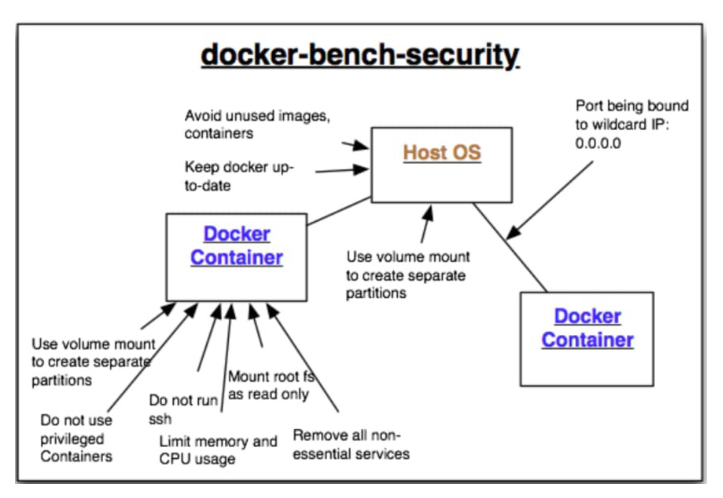
Docker Bench scan Docker host để tìm các vấn đề cấu hình phổ biến, chẳng hạn như cài đặt lỏng lẻo trong tệp cấu hình, hay là quyền hệ thống và các giá trị mặc định có vấn đề. Công cụ này dựa trên cơ sở dữ liệu về các CVE để kiểm tra các thư viện và tệp thực thi trên hệ thống được đề cập.

Sau khi scan, nó sẽ cung cấp điểm bảo mật. Quản trị viên có thể theo dõi điểm này để đánh dấu các cải tiến theo thời gian. Điểm số đó được tăng lên khi cấu hình Pass và ngược lại, nó sẽ giảm khi bị Warn, và điểm cao hơn rất có thể sẽ bảo mật tốt hơn.

* **What needs to be secured?**

Vậy, những thành phần nào cần được kiểm tra để có môi trường docker an toàn?

Như hình bên dưới, các kiểm tra nó sẽ liên quan đến các cấu hình xoay quanh Host và DockerContainer.

****

* **Host Configuration**

Bắt đầu với Host Configuration:

Phần này bao gồm các khuyến nghị bảo mật mà ta nên tuân theo để chuẩn bị cho máy host. Nó làm theo các phương pháp tốt nhất về bảo mật cơ sở hạ tầng để xây dựng một nền tảng vững chắc và an toàn để thực hiện các khối lượng công việc trên container.

Phần này tập trung vào các điểm yếu trong quá trình kiểm tra bảo mật trên host. Nó sẽ kiểm tra các thư mục Docker, đảm bảo sử dụng phân vùng dành riêng cho containers và đảm bảo cài đặt phiên bản Docker cập nhật.

Cụ thể thì nó sẽ kiểm tra host operating system, kiểm tra phiên bản docker và phân vùng của nó, chỉ cho phép users tin cậy được truy cập vào docker

Ảnh có chứa văn bản

Mô tả được tạo tự động

* **Docker Daemon Configuration**

Phần này sẽ kiểm tra Docker’s socket có bị lộ qua một kết nối không an toàn hay không. Lưu lượng mạng giữa các container trên mạng bridge mặc định nên bị hạn chế và loại bỏ registries không an toàn.

Phần này cũng tìm kiếm các đặc quyền không phù hợp cho các containers. Containers không nên có được các đặc quyền mới, vì điều này có thể cho phép kẻ tấn công phát triển vượt quá container.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* **Docker Daemon Configuration Files**

Các tệp cấu hình Docker Daemon có độ nhạy cảm cao và có thể cho phép kẻ tấn công kiểm soát tất cả các containers trên host.

Phần này bao gồm các quyền hạn và quyền sở hữu các tệp và thư mục liên quan đến Docker. Giữ an toàn cho các tệp và thư mục có thể chứa các tham số nhạy cảm là điều quan trọng để Docker daemon hoạt động chính xác và an toàn.

Ảnh có chứa bàn

Mô tả được tạo tự động

* **Container Images and Build File**

Container base images và build files chi phối các nguyên tắc cơ bản về cách một cá thể container từ một images cụ thể sẽ hoạt động. Đảm bảo rằng ta đang sử dụng base images và các build files thích hợp có thể rất quan trọng để xây dựng cơ sở hạ tầng dựa trên container.

Docker Bench thực hiện kiểm tra cơ bản các Dockerfiles cho các image hiện có. Nó sẽ tìm kiếm container users chuyên dụng, sự hiện diện của các lệnh HEALTHCHECK và việc sử dụng Content Trust để xác minh tính toàn vẹn của dữ liệu.

Phần kiểm tra này cũng sẽ phát ra các cảnh báo nhắc nhở về các bước làm cứng image cơ bản. Sử dụng base images đáng tin cậy, áp dụng các bản vá bảo mật mới và tránh cài đặt các gói không cần thiết. Các biện pháp này giúp loại bỏ các lỗ hổng bên trong các containers.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* **Container Runtime**

Container Runtime sẽ kiểm tra các containers đang hoạt động. Phần này bao gồm hơn 30 bài tests, từ tính khả dụng của SELinux và AppArmor đến việc sử dụng các tùy chọn mạng và mount file system thích hợp.

Containers không được có thêm đặc quyền hoặc can thiệp vào host system.

DockerBench cũng tìm kiếm các SSH server chạy bên trong các container. Điều này không được phép xảy ra, vì ta nên tránh truy cập trực tiếp vào container. Tốt hơn hết là sử dụng trình điều khiển docker từ host để tương tác với các container.

Nó cũng xem xét việc sử dụng CPU và giới hạn bộ nhớ. Một container không giới hạn có thể tiêu tốn quá nhiều tài nguyên và gây ra tình trạng hết bộ nhớ trên host.

Và cuối cùng, networking checks sẽ gắn cờ các port không cần thiết trong container.

Ảnh có chứa văn bản, máy tính xách tay, tài liệu

Mô tả được tạo tự động

* **Docker Security Operations**

Phần này bao gồm một số vấn đề bảo mật hoạt động liên quan đến triển khai Docker. Đây là những phương pháp tốt nhất nên được tuân theo nếu có thể. Hầu hết các đề xuất trong phần này chỉ đóng vai trò là lời nhắc nhở cho các tổ chức nên mở rộng các chính sách và thực tiễn tốt nhất về bảo mật.

* **Docker Swarm Configuration**

Phần này liệt kê các đề xuất thay đổi và bảo mật hành vi của Docker Swarm. Nó sẽ đưa ra cảnh báo nếu chế độ Swarm được bật nhưng không thực sự được sử dụng.

Ảnh có chứa văn bản

Mô tả được tạo tự động

* **Remediation**
* **1.1.1**

sudo mount --rbind /var /var/lib/docker

* **1.1.3 – 1.1.18**

sudo apt install auditd

sudo nano /etc/audit/rules.d/audit.rules

Thêm các lệnh sau vào cuối file

-w /usr/bin/docker -p wa

-w /var/lib/docker -p wa

-w /etc/docker -p wa

-w /lib/systemd/system/docker.service -p wa

-w /lib/systemd/system/docker.socket -p wa

-w /etc/default/docker -p wa

-w /etc/docker/daemon.json -p wa

-w /usr/bin/docker-containerd -p wa

-w /usr/bin/docker-runc -p wa

-w /run/containerd -p wa

-w /usr/bin/dockerd -k docker

-w /etc/containerd/config.toml -k docker

-w /usr/bin/containerd -k docker

-w /usr/bin/containerd-shim -k docker

-w /usr/bin/containerd-shim-runc-v1 -k docker

-w /usr/bin/containerd-shim-runc-v2 -k docker

-w /usr/bin/runc -k docker

-w /etc/sysconfig/docker -k docker

Lưu file và khởi động lại auditd

systemctl restart auditd

Ảnh có chứa bàn

Mô tả được tạo tự động

Các cờ [Info] (File not found) không cần quan tâm vì một số cấu hình không cần file này. Nếu file này tồn tại thì cần phải kiểm tra.

* **2.2, 5.2**

Truy cập vào /etc/docker/daemon.json và thêm các trường bên dưới

sudo nano /etc/docker/daemon.json

{

"icc": false,

"userns-remap": "default",

"log-driver": "syslog",

"live-restore": true,

"userland-proxy": false,

"no-new-privileges": true ,

"selinux-enabled": true

}

**Ảnh có chứa văn bản

Mô tả được tạo tự động**

sudo dockerd

* **4.5**

Truy cập vào /etc/sudoers và thêm trường DOCKER\_CONTENT\_TRUST

sudo nano /etc/sudoers

Defaults env\_keep += "DOCKER\_CONTENT\_TRUST"

Sau đó thực hiện lệnh sau:

export DOCKER\_CONTENT\_TRUST=1

* **5.2, 5.10, 5.11, 5.12, 5.14, 5.26, 5.28**

Luôn chạy container với các tùy chọn sau

docker run -it -d --cpu-shares 512 -d --memory 256m --read-only --restart=on-failure:5 --security-opt label=level:TopSecret --health-cmd='stat /etc/passwd || exit 1' --pids-limit 100 <image\_name or id>