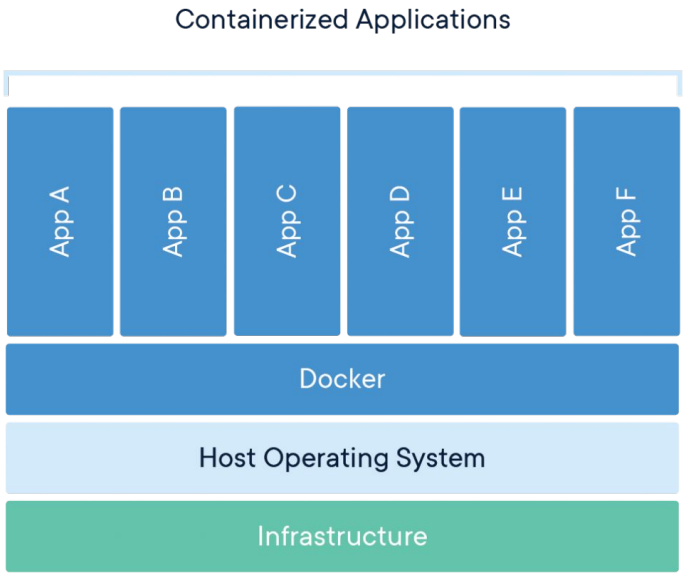
Basics Forensics of Dockers and Malware

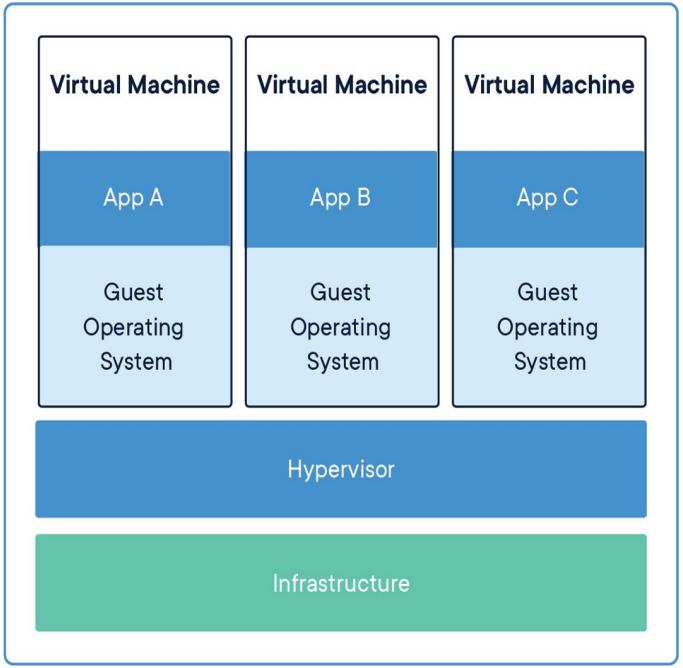
By Saket Thombre

What is a Docker?

- Docker, which is one of the various virtualization technology in server systems, is getting popular as it provides more lightweight environment for service operation than existing virtualization technology.
- A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.
- Available for both Linux and Windows-based applications, containerized software will always run the same, regardless of the infrastructure.

Containers Vs. Virtual Machines





Understanding Docker containers

In a way, Docker is a bit like a virtual machine. But unlike a virtual machine, rather than creating a whole virtual operating system, Docker allows applications to use the same Linux kernel as the system that they're running on and only requires applications be shipped with things not already running on the host computer.

Containers can be thought of as necessitating three categories of software:

- Builder: technology used to build a container.
- Engine: technology used to run a container.
- Orchestration: technology used to manage many containers.

Forensics in the age of Containers

- Containers differ from bare metal or virtual machines in a number of ways that impact obtaining actionable evidence.
- There is no default “container snapshot” function available; containers must be captured as a set of distinct components.
- In contrast with virtual machine (VM) environments, containers running on a host share the underlying operating system kernel.
- There isn’t any hardware isolation at play

File system

- Leading container runtimes use a copy-on-write file system. This is a great advantage to forensic acquisition.
- Any changes since the container started are stored in a separate directory from the original image. Furthermore, any deletion of original files from the image is also recorded.
- Docker will capture all file modifications, since the container was started into a new layer on the copy-on-write file system. Simply commit an existing container, running or not, into a new image: `docker commit $CONTAINER_ID imagename`.

Memory

- Containers are not a first-class primitive in linux; they are an abstraction on top of multiple subsystems, such as process namespaces and groups.
- Processes running inside a container manage memory the same as any other process. By default, processes inside a container may not interact or interfere with memory controlled by any process outside the container.
- The utility gcore suspends the process during acquisition but does not modify it; it executes it quickly, and the output format is compatible with YARA.

Shared volumes

- Containers are ephemeral. Malware might not be aware of this and write some interesting data to the container's allocated disk space.
- Ultimately the most sensitive data is likely stored via a volume, typically mounted against persistent online storage. This is where container isolation breaks down.
- It is not recommended to use a container that also mounts this volume to enumerate files – that may tarnish the file access and modification metadata. If you absolutely must access the volume this way, be sure to mount it into the container as 'read-only'.

Related microservices

- Containers are typically employed within a microservices-based architecture. While great for speed and scalability, this means that suspicious activity will likely involve multiple containers pertaining to multiple services, across various hosts and with access to different data stores.
- While investigating a container breach, make note of any data stores or services that this container accessed.
- The web of lateral movement across containers will differ according to the sophistication of an attack hence it is better to acquire more.

Container escape

There are vulnerabilities and misconfigurations that could allow malware to escape a container.

Container isolation falls into the following categories:

- Network isolation
- Process namespacing
- File system chroot
- Device access control
- Default seccomp profile

Containers in a forensic environment

- At this time, there isn't a formal mechanism for running a captured container. Once they're shut down, even if both file system and memory contents are exported, there is no mechanism for combining the two back into the previous running state.
- Interaction during execution is mandatory, and it is possible to preserve the container and remove it from a production environment. There are a few mechanisms one can use to persist the running container without the container further impacting the cluster in a significant way.
- Once a container is paused or quarantined (if one desires a clean operating environment), it's possible to scale up a cluster.

Conclusion

Forensic analysis is a critical capability of any enterprise security program, and it must take into account the characteristics of the significantly different attack surface and environment that containers form. Understanding container ephemerality and isolation is required in order to identify infected containers, capture components, and analyze them safely.