Isolated Container Runtime

Harshal Patil, IBM Linux Technology Center
Kernel Vulnerabilities

- Dirty COW (CVE-2016-5195) race condition allows an attacker in one docker container to access another docker container running on the same system.
- It also allows an attacker from a docker container to gain root access on the host.
- Proof of Concept of Dirty COW working from within a docker container, [https://github.com/scumjr/dirtycow-vdso](https://github.com/scumjr/dirtycow-vdso)
- Compromising EC2 instance, [https://www.youtube.com/watch?v=BwUfHJXgYgo](https://www.youtube.com/watch?v=BwUfHJXgYgo)
- This vulnerability stayed in Linux kernel for 9 years.
- This was not a first kernel vulnerability and probably won't be the last one.
Security Sensitive Workloads

- Financial Transactions
- Validating nodes on permissioned blockchains
- Multi-tenant cloud setups
- Hybrid Cloud deployments
Isolating Container Execution

- Application
- Qemu for Isolation
- runtime
- Host kernel
Various Approaches

- `docker run --isolation option for hyperv`
- Hyperhq runv
- Intel Clear Containers
- Runvm
Since earlier docker didn't have native support on Windows, docker run –isolation=hyperv was introduced to run containers under VM in Windows.

We saw it as a good entry point to run container workloads more securely under a VM using –isolation option.

Docker run –isolation=qemu to run container workloads in a qemu VM in Linux.

Here is the PR - https://github.com/moby/moby/issues/29454

But the community suggested that we take more modular OCI compliant approach by implementing it in a separate runtime
We need have a VM with "init" that can take input from host to,

- Configure VM IP configuration, set hostname etc
- Mount a container rootfs and chroot into it
- Initiate the execution of the workload inside chrooted environment
• One of the first implementation of an idea in Linux that container workloads can be run inside a virtual machine.

• Runv is an OCI compliant runtime now, but earlier they needed custom made containerd to smoothly integrate in the docker workflow.*

• Runv boots customized VM that can communicate with runtime for stdin/stdout, configuration, signal handling etc.

https://github.com/hyperhq/runv

* https://blog.hyper.sh/runv-bring-isolation-to-docker.html
Clear Containers

- Uses similar approach like runv to configure VM using communication channel from host to VM.
- Clear containers has impressive set of performance optimizations. E.g. using DAX for rootfs, qemu-lite etc.
- 9pfs is used to pass container root to qemu VM

https://github.com/clearcontainers
If we are talking about ultra secure workloads, do we really need to keep a communication channel open between host and container workload in VM?

Can we think of a 'fire-and-forget' approach to minimize the attack surface further?
Minimum Requirements
- Do VM IP configuration
- Mount container rootfs
- Chroot into mounted rootfs
- Execute the workload (container entrypoint)

Current solutions either use custom init or an init that starts a custom agent within VM

What about leveraging something which already satisfies the above requirement?
- cloud-init
  - ubiquitous way to configure VM instances in cloud
  - Potential use of encrypted container image, with decryption key made available via cloud-init metadata service

* Challenges: How do we do signal handling using cloud-init?
Proof of Concept runtime which uses cloud-init to configure and run container workloads in VMs
- Can leverage standard cloud images, moby tools (linuxkit) generated images etc.
- No two-way communication between host and guest. In other words, no docker exec etc.

https://github.com/harche/runvm
• RunV runtime showcased that VMs can execute container workloads for better isolation
• Clear Containers do excellent optimizations to get blazing fast speeds for isolated workload execution
• Runvm POC proves that cloud-init approach provides an option to use 'off-the-shelf' pre-existing methods to configure VMs and launch workloads
• Maybe there should be a convergence in future by taking all the good things from all these different approaches trying to solve the same problem
Thank You