存储安全

1. 挂载宿主机的敏感目录到容器中，从而造成容器可以对host敏感文件进行任意操作
2. 以下这些敏感目录及文件没有进行隔离：

/proc/fs、 文件系统信息

/proc/meminfo、 RAM使用的相关信息

/proc/cpuinfo CPU的相关信息

/proc/sysrq-trigger、 使用echo命令来写这个文件的时候，远程root用户可以执行大多数的系统请求关键命令，就好像在本地终端执行一样

/proc/irq、 中断请求设备信息

/proc/bus、 系统总线(Bus)信息

1. 现在还无法设置volume的容量，一旦容器疯狂创建文件，将会耗尽主机的资源，实现dos攻击
2. 假设一个Web容器将网站挂载主机的某一目录下，当其它容器也挂载这个目录或其父目录时，其他容器就可以对Web容器的网站进行任意操作了
3. 跨Docker主机存储时，若远程Storage Provide被劫持，则存储的敏感数据直接被盗取

Docker 为容器提供了两种存放数据的资源：

1. 由 storage driver 管理的镜像层和容器层
2. Data Volume

Storage dirver：

容器由最上面一个可写的容器层，以及若干只读的镜像层组成，容器的数据就存放在这些层中。这样的分层结构最大的特性是 Copy-on-Write：

1. 新数据会直接存放在最上面的容器层。

2. 修改现有数据会先从镜像层将数据复制到容器层，修改后的数据直接保存在容器层中，镜像层保持不变。

3. 如果多个层中有命名相同的文件，用户只能看到最上面那层中的文件。

分层结构使镜像和容器的创建、共享以及分发变得非常高效，而这些都要归功于 Docker storage driver。正是 storage driver 实现了多层数据的堆叠并为用户提供一个单一的合并之后的统一视图。

Docker 支持多种 storage driver，有 AUFS、Device Mapper、Btrfs、OverlayFS、VFS 和 ZFS。它们都能实现分层的架构，同时又有各自的特性。对于 Docker 用户来说，具体选择使用哪个 storage driver 是一个难题，因为：

1. 没有哪个 driver 能够适应所有的场景。
2. driver 本身在快速发展和迭代。

Ubuntu 用的 AUFS，底层文件系统是 extfs，各层数据存放在 /var/lib/docker/aufs。Redhat/CentOS 的默认 driver 是 Device Mapper，SUSE 则是 Btrfs。

Data volume：

Data Volume 本质上是 Docker Host 文件系统中的目录或文件，能够直接被 mount 到容器的文件系统中。Data Volume 有以下特点：

1. Data Volume 是目录或文件，而非没有格式化的磁盘（块设备）。
2. 容器可以读写 volume 中的数据。
3. volume 数据可以被永久的保存，即使使用它的容器已经销毁。

因为 volume 实际上是 docker host 文件系统的一部分，所以 volume 的容量取决于文件系统当前未使用的空间，目前还没有方法设置 volume 的容量。在具体的使用上，docker 提供了两种类型的 volume：bind mount 和 docker managed volume。

bind mount：

假如挂载 <host path>:<container path>，若container path目录已存在，则原有数据会被隐藏起来，取而代之的时host path

docker managed volume：

docker managed volume 与 [bind mount](http://mp.weixin.qq.com/s?__biz=MzIwMTM5MjUwMg==&mid=2653587711&idx=1&sn=271538bc7b5a8c0684d629604271189a&chksm=8d3080e6ba4709f0ba0335379ca6c4a270a1811de404144ff653a807930f13b3cb8bd934a5de&scene=21#wechat_redirect) 在使用上的最大区别是不需要指定 mount 源，指明 mount point 就行了

