【编者的话】在利用Docker承载大家的关键性任务应用程序时，我们必须对五项重要安全问题加以关注。

通过阅读网上帖子以及浏览相关新闻，大家可能会产生一种先入为主的印象，即Docker天生安全性薄弱且尚不足以被直接引入生产环境。不过实际情况是，虽然我们需要对容器安全性加以高度关注，但只要使用得当，其完全可以成为一套远优于单独使用虚拟机或者裸机的安全、高效生产系统。

要安全地使用Docker方案，大家首先需要了解其面对的潜在安全问题，并掌握能够对基于容器之系统加以切实保护的各类主要工具与技术。

大家还需要随时牢记以下五个问题，并在利用Docker承载关键性任务应用程序的整个流程当中秉承这种谨慎的态度。

内核漏洞

与虚拟机系统不同，全部容器及其主机使用的都是同一套共享内核，因此该内核中存在的任何安全漏洞都有可能造成巨大影响。如果某套容器系统导致内核崩溃，那么这反过来又会造成整台主机上的全部容器毁于一旦。在虚拟机当中，情况则要好得多：攻击者必须借道虚拟机内核与虚拟机管理程序之后，才有可能真正接触到主机内核。

拒绝服务攻击

所有容器都共享同样的内核资源。如果某套容器能够以独占方式访问某些资源——包括内存以及用户ID等其它更为抽象化的资源——那么与其处于同一台主机上的其它容器则很可能因资源匮乏而无法正常运转。这正是拒绝服务攻击（简称DoS）的产生原理，即合法用户无法对部分或者全部系统进行访问。

容器突破

能够访问某一容器的攻击者在原则上应该无法借此访问到其它容器或者主机。在默认情况下，用户并不具备命名空间，因此游离于容器之外的任何进程都将在主机之上获得与容器内相同的执行权限; 而如果大家在容器内拥有root权限，那么在主机上亦将具备root身份。这意味着大家需要对这种潜在的权限提升攻击做好准备——这类攻击意味着用户往往通过应用程序代码中需要配合额外权限的bug实现权限提升，从而使攻击者获得root或者其它级别的访问与操纵能力。考虑到容器技术目前仍处于早期发展阶段，因此我们在规划自己的安全体系时，必须要将这种容器突破状况考虑在内。

含毒镜像

那么我们要如何判断自己使用的镜像是否安全、是否存在篡改或者其宣称的来源是否可靠？如果攻击者诱导大家运行由其精心设计的镜像，那么各位的主机与数据都将处于威胁之下。同样的，大家还需要确保自己运行的镜像为最新版本，且其中不包含任何存在已知安全漏洞的软件版本。

违规之秘

当容器面向某数据库或者服务发起访问时，其往往需要某种秘密因素加以配合，例如API密钥或者用户名加密码。能够获取这些秘密因素的攻击者自然会将触手伸向对应服务。这类问题在微服务架构当中往往更为严重，因为在此类环境内各容器会频繁中止与启动，因此受到的威胁远高于一般而言运行周期更长且数据较少的虚拟机系统。我们今天并不会就此类问题的解决办法展开探讨，感兴趣的朋友不妨查阅《Using Docker》（O'Reilly出版社，2015年）中的"Deployment"章节了解相关内容。

以上提到的问题当然还不够全面，不过已经值得大家认真加以考量。如果各位希望了解更多与其解决方案相关的信息，不妨浏览由Adrian Mouat撰写的《Docker Security (https://www.oreilly.com/ideas/docker-security?intcmp=il-webops-free-article-lgen_five_security_concerns_when_using_docker)》一文（需要免费注册）。

原文链接：Five security concerns when using Docker (https://www.oreilly.com/ideas/five-security-concerns-when-using-docker)