相关参考

<https://www.bilibili.com/read/cv15553799/?spm_id_from=333.999.collection.opus.click>

容器安全并不是简单在某一环做出相应的安全配置就可以的，我们需要考虑四个主要方面，一是内核的内在安全性及其对命名空间和cgroup的支持，二是Docker 守护进程本身的攻击面，三是容器配置文件中的漏洞，四是内核的“强化”安全特性以及它们如何与容器交互。所以我们必须从基础操作系统环境、容器服务、容器镜像以及业务与运维开发人员分别入手，才能提高容器的安全性，减少被攻击的可能性。

Docker 服务为了防止黑客在控制容器后能够对宿主机进行攻击，提供了三个主要的隔离机制，其分别是Namespace 机制、Capabilities 机制和 CGroups 机制。如果攻击对宿主机产生了影响，就说明入侵者已经突破了Docker服务的保护，这就是容器安全中常说的Docker容器逃逸。

默认情况下，Docker 启动的容器被严格限制只允许使用内核的一部分能力。并且Docker采用白名单机制，禁用必需功能之外的其它权限

为了加强安全，容器可以禁用一些没必要的权限。

完全禁止任何 mount 操作；

禁止直接访问本地主机的套接字；

禁止访问一些文件系统的操作，比如创建新的设备、修改文件属性等；

禁止模块加载。

这样就算攻击者在容器中取得了 root 权限，也不能获得本地主机的较高权限，能进行的破坏也有限

从3大机制方面理解下docker基础安全防御

然后主要从漏洞复现去思考怎么解决

给定14个漏洞都是非法文件操作，所以从文件操作方面加强

以下为一些可进行操作的环节

参考<https://www.bilibili.com/read/cv15554240/?spm_id_from=333.999.collection.opus.click>

1.主机安全配置

1.1 更新docker到最新版本

1.2 为容器创建一个单独的分区

1.3 只有受信任的用户才能控制docker守护进程

1.4 审计docker守护进程

1.5 审计docker相关的文件和目录

2.docker守护进程配置

2.1 限制默认网桥上容器之间的网络流量

2.2 设置日志级别为info

2.3 允许 docker 更改iptables

2.4 不使用不安全的镜像仓库

2.5 建议不使用aufs存储驱动程序

2.6 docker守护进程配置TLS身份认证

2.7 配置合适的 ulimit 资源控制

2.8 启用用户命名空间

2.9 使用默认cgroup

2.10 启用docker客户端命令的授权

2.11 配置集中和远程日志记录

2.12 禁用docker resitry v1版本支持

2.13 启用实时恢复

2.14 禁用 userland 代理

2.15 限制容器获取新的权限

3.docker 守护进程文件配置

3.1 设置 docker.service 文件所属和权限

3.2 设置docker.socket文件所属和权限

3.3 设置/etc/docker目录所有权为root:root

3.4 设置仓库证书文件所有权为root：root

3.5 设置TLS CA证书文件所有权为root:root

3.6 设置docker服务器证书文件所有权为root:root

3.7 设置docker服务器证书密钥文件所有权为root：root

3.8 设置/var/run/docker.sock文件所有权为root:docker

3.9 设置daemon.json文件所有权为root：root

3.10 设置 /etc/default/docker 文件所有权为 root:root

4.容器镜像和构建文件

4.1 创建容器的用户

4.2 容器使用可信的基础镜像

4.3 容器中不安装没有必要的软件包

4.4 扫描镜像漏洞并且构建包含安全补丁的镜像

4.5 启用docker内容信任

4.6 将HEALTHCHECK说明添加到容器镜像

4.7 不在dockerfile中单独使用更新命令

4.8 镜像中删除setuid和setgid权限

4.9 在dockerfile中使用copy而不是add

4.10 涉密信息不存储在dockerfile

4.11 仅安装已经验证的软件包

4.12 容器内部项目指定运行用户

5.容器运行时保护

5.1 设置SElinux安全选项

5.2 linux内核特性在容器内受限

5.3 不使用特权容器

5.4 敏感的主机系统目录不要挂载在容器上

5.5 SSH 不在容器中运行

5.6 特权端口禁止映射到容器内

5.7 只映射必要的端口

5.8 不共享主机的网络命名空间

5.9 确保容器的内存使用合理

5.10 正确设置容器上的CPU优先级

5.11 确保进入容器的流量绑定到特定的网卡

5.12 容器重启策略on-failure设置为5

5.13 确保主机的进程命名空间不共享

5.14 主机的IPC命令空间不共享

5.15 主机设备不直接共享给容器

5.16 设置默认的ulimit配置（在需要时）

5.17 设置主机的UTS命令空间不共享

5.18 不要使用docker的默认网桥docker0