Docker 安全配置规范

文档号：SSE_SEC_APP_DOCKER

2016 年 7 月

文档状态	[√] 初稿 [√] 评审通过 [√] 修改 [ ] 发布 [ ]作废	**文档标识**	Docker 安全配置规范
		当前版本	0.2
		作者	孙增
		部门/厂商	技术公司规划部
		完成日期	

文档版本历史表

版本号	作者	操作	日期	文档说明
0.1	孙增	创建	2016-5-19	初稿
0.2	孙增	修订	2016-7-7	修订稿：考虑到实际使用的情形，将第 13 项涉及容器间流量限和第17项配置TLS证书项，以及第 18、19、20 项涉及证书文件权限的配置项的要求有“必须”改为“建议”。

文档审阅记录表

版本号	审核人	审核日期	说明

文档批准记录表

版本号	批准人	审核日期	说明

目的

本文档规定了上交所技术有限责任公司所维护管理的 Docker 应用服务器的主机应当遵循的安全性设置基线，满足系统安全策略的要求。本文档旨在指导系统管理人员或安全检查人员进行 Docker 应用服务器的安全合规性检查和配置。

适用范围

本配置基线标准的使用者包括：服务器系统管理员、应用管理员、安全管理员。本配置基线标准适用的范围包括： 上交所技术有限责任公司所维护管理的面向互联网的 Docker 应用服务器系统。

对于内部使用的 Docker 应用系统，仅作参考，不做强制要求。

适用版本

Docker1.11.0 及以下

实施

本标准发布之日起生效。

例外条款

欲申请本规范的例外条款(如业务确实需要等导致的配置冲突)，申请人必须准备书面申请文件，说明业务需求和原因，送交技术公司规划部进行审批和备案。

版本维护

技术公司规划部根据反馈和建议进行规范的跟踪、维护和改进。

反馈建议至：孙增

邮箱：zsun@sse.com.cn

一、 宿主机配置要求.....1
1、 为容器创建独立的分区.....1
2、 使用已更新过的 Linux 内核.....1
3、 只允许受信任的用户控制 Docker deamon.....2
4、 审计 Docker deamon.....3
5、 审计/var/lib/docker.....3
6、 审计/etc/docker.....4
7、 审计 docker.service.....5
8、 审计 docker.socket.....6
9、 审计/etc/default/docker.....6
10、 审计/etc/docker/daemon.json.....7
11、 审计/usr/bin/docker-containerd.....8
12、 审计/usr/bin/docker-runc.....9
二、 Docker deamon 配置要求.....10
13、 限制容器之间的网络流量.....10
14、 允许 Docker 更改 iptables.....10
15、 使用安全的镜像库.....11
16、 不使用 aufs 存储驱动.....12
17、 为 Docker daemon 配置 TLS 认证.....12
三、 Docker deamon 证书文件权限设置.....13
18、 设置验证镜像库证书文件的权限.....13
19、 设置验证 TLS CA 证书文件的权限.....14
20、 设置验证 Docker 服务器证书文件的权限.....15

Docker 安全配置规范

注：本文档主要基于Docker1.11.0及以下版本，文档初稿参考CIS Docker 1.11.0 Benchmark编写。

一、 宿主机配置要求

1、 为容器创建独立的分区

配置项编号	SSE_SEC_APP_DOCKER_01
配置项说明	所有的容器包括它们的数据和元数据均存储在/var/lib/docker 目录下，默认情况下基于可用性考虑，/var/lib/docker 被挂载在/ or /var 下。这个目录下的存储空间容易很快被占用满，导致 Docker 容器无法使用。因此，最好是创建用于 Docker 文件的单独的分区（逻辑卷）。
配置项实施	可以利用 Logical Volume Manager (LVM)来创建分区。
配置项要求	必须
审计方法	使用命令：grep /var/lib/docker /etc/fstab，查看/var/lib/docker 挂载点的分区细节。
备注	http://www.projectatomic.io/docs/docker-storage-recommendation

2、 使用已更新过的 Linux 内核

配置项编号	SSE_SEC_APP_DOCKER_02
配置项说明	Docker 在 daemon 模式下对内核有特殊的要求，至少使用 A3.10 以上的 linux 内核。老版本的 linux 内核有很多已知的缺陷，在容器使用时容易造成数据丢失和

	频繁的死机。此外，使用更新的 Linux 内核，能确保在系统上线前较早的发现关键内核缺陷。
配置项实施	根据 Docker 对系统和内核的需求，选择适合的系统和内核。
配置项要求	必须
审计方法	使用命令：uname –r 查看是否使用了新版本的 linux 内核。
备注	1.https://docs.docker.com/installation/binaries/#check-kernel-dependencies 2. https://docs.docker.com/installation/#installation-list

3、 只允许受信任的用户控制 Docker deamon

配置项编号	SSE_SEC_APP_DOCKER_03
配置项说明	目前，Docker deamon 需要 root 权限，用户加入到“docker”组中会充分给予其 root 访问权限。这意味着，你可以启动一个容器并可以将主机上的目录映射到该容器上，通过容器可以毫无限制的更改主机文件系统，也就是说通过该过程可以提升权限。
配置项实施	移除“docker”组中不被信任的用户，同时不要创建涉及宿主机敏感数据的目录映射到容器数据卷。 影响：作为构建和执行容器的普通用户权利将受到限制。
配置项要求	必须
审计方法	使用命令：getent group docker 查看是否不被信任的用户在 Docker 用户组内。
备注	1.https://docs.docker.com/articles/security/#docker-daemon-attack-surface 2.https://www.andreas-jung.com/contents/on-docker-security-docker-group-co

	nsidered-harmful 3.http://www.projectatomic.io/blog/2015/08/why-we-dont-let-non-root-users-run-docker-in-centos-fedora-or-rhel/

4、 审计 Docker deamon

配置项编号	SSE_SEC_APP_DOCKER_04
配置项说明	默认情况下 Docker deamon 不会被审计，应当审计所有 Docker deamon 活动。因为 Docker deamon 使用 root 权限运行，有必要审计其活动。
配置项实施	添加 Docker deamon 规则，例如： 1、 在/etc/audit/audit.rules 文件下添加：-w /usr/bin/docker -k docker 2、 然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	必须
审计方法	使用命令：auditctl -l | grep /usr/bin/docker 可以看到 Docker deamon 一条审计规则。
备注	1.https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

5、 审计/var/lib/docker

配置项编号	SSE_SEC_APP_DOCKER_05
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。/ var / lib/docker 就是这样的一个目录。默认情况下/ var / lib/docker 不被审计，因为它拥

	有所有有关容器的信息，应当进行审计。
配置项实施	为/var/lib/docker 目录增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /var/lib/docker -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	必须
审计方法	使用命令：auditctl -l | grep /var/lib/docker 可以看到/var/lib/docker 的一条审计规则。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

6、 审计/etc/docker

配置项编号	SSE_SEC_APP_DOCKER_06
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。/etc/docker 就是这样的一个目录。默认情况下/etc/docker 不被审计，因为它拥有用于 Docker daemon 和 Docker client 之间 TLS 通信的各种证书和密钥，应当进行审计。
配置项实施	为/etc/docker 目录增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /etc/docker -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	必须
审计方法	

	使用命令：auditctl -l | grep /etc/docker 可以看到/etc/docker 的一条审计规则。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

7、 审计 docker.service

配置项编号	SSE_SEC_APP_DOCKER_07
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。docker.service 就是这样的一个文件。默认情况下 docker.service 不被审计，也可能不被使用。如果 deamon 参数被管理员修改 docker.service 就可能存在，因为该文件拥有 Docker deamon 的各种参数，应当被审计（如适用）。
配置项实施	为 docker.service 文件增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /usr/lib/systemd/system/docker.service -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	建议
审计方法	1、 使用命令找到文件的位置： systemctl show -p FragmentPath docker.service 2、 如果该文件不存在，此建议不适用。如果该文件存在，执行命令验证是否有对应文件的审计规则： auditctl -l | grep docker.service 应看到 docker.service 的审计规则和所在的位置。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

8、 审计 docker.socket

配置项编号	SSE_SEC_APP_DOCKER_08
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。docker.socket 就是这样的一个文件。默认情况下 docker.socket 不被审计，也可能不被使用。因为该文件拥有 Docker deamon 套接字的各种参数，应当被审计（如适用）。
配置项实施	为 docker.socket 文件增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /usr/lib/systemd/system/docker.socket -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	建议
审计方法	1、 使用命令找到文件的位置： systemctl show -p FragmentPath docker.socket 2、如果该文件不存在，此建议不适用。如果该文件存在，执行命令验证是否有对应文件的审计规则： auditctl -l | grep docker.socket 应看到 docker.service 的审计规则和所在的位置。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

9、 审计/etc/default/docker

配置项编号	SSE_SEC_APP_DOCKER_09
配置项说明	

	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。/etc/default/docker 就是这样的一个文件。默认情况下/etc/default/docker 不被审计，也可能不被使用。因为该文件拥有 Docker deamon 的各种参数，应当被审计（如适用）。
配置项实施	为/etc/default/docker 文件增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /etc/default/docker -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	建议
审计方法	使用命令：auditctl -l | grep /etc/default/docker 查看/etc/default/docker 的审计规则。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html

10、 审计/etc/docker/daemon.json

配置项编号	SSE_SEC_APP_DOCKER_10
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。/etc/docker/daemon.json 就是这样的一个文件。默认情况下/etc/docker/daemon.json 不被审计，也可能不被使用。因为该文件拥有 Docker deamon 的各种参数，应当被审计（如适用）。
配置项实施	为/etc/docker/daemon.json 文件增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /etc/docker/daemon.json -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	建议

审计方法	使用命令：auditctl -l | grep /etc/docker/daemon.json 查看/etc/docker/daemon.json 的审计规则。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html 2. https://docs.docker.com/engine/reference/commandline/daemon/#daemon-configuration-file

11、 审计/usr/bin/docker-containerd

配置项编号	SSE_SEC_APP_DOCKER_11
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。/usr/bin/docker-container 就是这样的一个文件。默认情况下/usr/bin/docker-containerd 不被审计，也可能不被使用。因为目前 Docker deamon 依靠 containerd 派生出（spawn containers）很多容器，应当被审计（如适用）。
配置项实施	为/usr/bin/docker-containerd 文件增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /usr/bin/docker-containerd -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	建议
审计方法	使用命令：auditctl -l | grep /usr/bin/docker-containerd 查看/usr/bin/docker-containerd 的审计规则。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html 2. https://github.com/docker/docker/pull/20662 3. https://containerd.tools/

12、 审计/usr/bin/docker-runc

配置项编号	SSE_SEC_APP_DOCKER_12
配置项说明	Docker deamon 在 root 权限下运行。其行为取决于一些关键的文件和目录。/usr/bin/docker-runc 就是这样的一个文件。默认情况下/usr/bin/docker-runc 不被审计，也可能不被使用。因为目前 Docker deamon 依靠 runC 派生出（spawn containers）很多容器，应当被审计（如适用）。
配置项实施	为/usr/bin/docker-runc 文件增加一条审计规则，例如： 1、在/etc/audit/audit.rules 文件下添加：-w /usr/bin/docker-runc -k docker 2、然后重启审计守护进程：service auditd restart 影响：审计会产生相当大的日志文件。确保定期回滚和归档。此外，创建审计独立分区，以避免填满根文件系统。
配置项要求	建议
审计方法	使用命令：auditctl -l | grep /usr/bin/docker-runc 查看/usr/bin/docker-runc 的审计规则。
备注	1. https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security_Guide/chap-system_auditing.html 2. https://github.com/docker/docker/pull/20662 3. https://containerd.tools/ 4. https://github.com/opencontainers/runc

二、 Docker deamon 配置要求

13、 限制容器之间的网络流量

配置项编号	SSE_SEC_APP_DOCKER_13
配置项说明	默认情况下，同一台主机上所有容器之间的网络流量都被允许。因此，每个容器都具有读取到同一主机容器网络中通信的所有数据包的能力，可能导致容器间的信息泄露。因此如果没有需要，限制所有的容器间的通信，放开有特定连接需要的容器间的通信。
配置项实施	以 daemon 方式运行 Docker，将'--icc=false'作为参数。 影响：跨容器通信将被禁用。同一主机上没有容器将能够与另一个容器进行通信。如果主机上希望容器之间的任何通信。那么它需要使用容器 link 明确定义。
配置项要求	建议
审计方法	使用命令：ps -ef | grep docker 查看'--icc'参数是否为'false'。
备注	1. https://docs.docker.com/v1.8/articles/networking/

14、 允许 Docker 更改 iptables

配置项编号	SSE_SEC_APP_DOCKER_14
配置项说明	在 Linux 内核中 iptables 用于建立，维护和检查 IP 数据包过滤规则表，默认情况下'iptables'被设置成'true'状态，允许更改系统的 iptables 规则。Docker 需要实现自动化网络配置选项就应能够更改系统的 iptables 规则，为了防止出现网络配置错误阻碍容器与外界的正常通信现象，应当允许 Docker deamon 更改 iptables，也会节省每次选择运行容器或修改网络选项时更新的 iptables 的麻烦。

配置项实施	不使用'--iptables=false'参数运行 Docker deamon。 例如：docker daemon --iptables=false
配置项要求	必须
审计方法	使用命令：ps -ef | grep docker 确保'--iptables'不存在或不是'false'。
备注	1. https://docs.docker.com/v1.8/articles/networking/

15、 使用安全的镜像库

配置项编号	SSE_SEC_APP_DOCKER_15
配置项说明	Docker 会考虑镜像库的安全与否，通常认为镜像库是安全的。安全的镜像库通常是 TLS 通信，镜像库的 CA 证书的副本被放置在 Docker 主机的“/etc/docker/certs.d/<registry-name>/”目录下。不安全的镜像库没有任何有效的证书并且不使用 TLS 通信，这样的镜像库可能通过潜在的漏洞篡改生产系统。另外，如果是不安全的镜像库，使用'docker pull', 'docker push', and 'docker search'命令后不会有错误的信息返回，使用者很容易在不被告知有潜在风险的前提下长时间只用该不安全的镜像库。
配置项实施	不能使用不安全的镜像库，例如不能使用以下命令开启 Docker deamon: docker daemon --insecure-registry 10.1.0.0/16 默认情况下，本地私有的镜像库是安全的。
配置项要求	必须
审计命令	执行命令：ps -ef | grep docker 发现任何被使用的不安全镜像库，确保'--insecure-registry'参数没有出现。
备注	1. https://docs.docker.com/registry/insecure/

16、 不使用 aufs 存储驱动

配置项编号	SSE_SEC_APP_DOCKER_16
配置项说明	在 Docker 实例中不要使用 aufs 存储驱动。aufs 存储驱动是最古老的存储驱动程序，它是基于 Linux 内核补丁集是不太可能被合并到主 Linux 内核，会导致一些严重的内核崩溃。aufs 存储驱动不支持使用最新内核的 Linux 发行版。 默认情况下 Docker 使用'devicemapper'作为存储驱动。
配置项实施	不要明确使用 aufs 作为存储驱动，例如：不要用一下命令启动 Docker deamon： docker daemon --storage-driver aufs 影响：aufs 是容器共享可执行文件和共享库的内存时唯一使用的存储驱动程序，如果用相同程序或库运行数千容器时 aufs 可能是有用的。
配置项要求	必须
审计方法	执行下面的命令并验证 aufs 不作为存储驱动程序： docker info | grep -e "^Storage Driver:\s*aufs\s*$" 不返回任何值。
备注	1. http://docs.docker.com/reference/commandline/cli/#daemon-storage-driver-option 2. https://github.com/docker/docker/issues/6047 3.http://muehe.org/posts/switching-docker-from-aufs-to-devicemapper/ 4.http://jpetazzo.github.io/assets/2015-03-05-deep-dive-into-docker-storage-drivers.html#1

17、 为 Docker daemon 配置 TLS 认证

配置项编号	

	SSE_SEC_APP_DOCKER_17
配置项说明	默认情况下 TLS 认证没有被配置，它可以使 Docker deamon 在一个特定的 IP 和端口,或者其他任何 Unix 套接字而不只是默认的 Unix 套接字监听。配置 TLS 认证可以通过 IP 和端口来限制访问 Docker deamon。
配置项实施	按照 Docker 文件或其他参考文献中提到的步骤实施。 影响：需要为 Docker daemon 和 Docker clients 管理和保护好相应的证书和密钥。
配置项要求	建议
审计方法	使用审计命令：ps -ef | grep docker 查看下列参数都应当存在： '--tlsverify' '--tlscacert' '--tlscert' '--tlskey'
备注	1. http://docs.docker.com/articles/https/ 2.http://www.hnwatcher.com/r/1644394/Intro-to-Docker-Swarm-Part-2-Comfiguration-Modes-and-Requirements 3.http://www.blackfinsecurity.com/docker-swarm-with-tls-authentication/

三、 Docker deamon 证书文件权限设置

18、 设置验证镜像库证书文件的权限

配置项编号	SSE_SEC_APP_DOCKER_18
配置项说明	验证镜像库证书文件的权限应为 root 权限所拥有，并将该权限设置为 444 或更加严格。通常在/etc/docker/certs.d/<registry-name>目录下包含镜像库证书文件，为保证该文件的完整性，应设置为被 root 权限所拥有，并设置权限为 444 或更加严格。默认情况下，该项权限拥有者是 root 权限，是由系统或用户特定的 umask 值决定。

配置项实施	chown root:root /etc/docker/certs.d/<registry-name>/* chmod 444 /etc/docker/certs.d/<registry-name>/*
配置项要求	建议
审计方法	stat -c %U:%G /etc/docker/certs.d/* | grep -v root:root chmod 444 /etc/docker/certs.d/<registry-name>/*
备注	1. https://docs.docker.com/articles/certificates/ 2.http://docs.docker.com/reference/commandline/cli/#insecure-registries

19、 设置验证 TLS CA 证书文件的权限

配置项编号	SSE_SEC_APP_DOCKER_19
配置项说明	验证 TLS CA 证书文件的权限应为 root 权限所拥有，并将该权限设置为 444 或更加严格。基于给定的 TLS CA 证书文件可以用于验证 Docker 服务器，该证书文件应当给予保护。为保证该文件的完整性，应设置为被 root 权限所拥有，并设置权限为 444 或更加严格。默认情况下，该项权限拥有者是 root 权限，权限设置不是 444，而是由系统或用户特定的 umask 值决定。
配置项实施	chown root:root <path to TLS CA certificate file> chmod 444 <path to TLS CA certificate file>
配置项要求	建议
审计方法	stat -c %U:%G <path to TLS CA certificate file> | grep -v root:root stat -c %a <path to TLS CA certificate file>
备注	1. https://docs.docker.com/articles/certificates/ 2. http://docs.docker.com/articles/https/

20、 设置验证 Docker 服务器证书文件的权限

配置项编号	SSE_SEC_APP_DOCKER_20
配置项说明	验证 Docker 服务器证书文件应为 root 权限所拥有，并将该权限设置为 444 或更加严格。该证书文件拥有 Docker 服务器证书的私钥，该密钥文件应当给予保护，防止任何篡改或不需要读取。为保证该文件的完整性，应设置为被 root 权限所拥有，并设置权限为 400 或更加严格。默认情况下，该项权限拥有者是 root 权限，权限设置不是 400，而是由系统或用户特定的 umask 值决定。
配置项实施	chown root:root <path to Docker server certificate key file> chmod 400 <path to Docker server certificate key file>
配置项要求	建议
审计方法	stat -c %U:%G <path to Docker server certificate key file> | grep -v root:root stat -c %a <path to Docker server certificate key file>
备注	1. https://docs.docker.com/articles/certificates/ 2. http://docs.docker.com/articles/https/