讲师：李振良（阿良）

今天课题：《K8s容器运行环境安全加固》

学院官网：www.aliangedu.cn

阿良个人微信

DevOps技术栈公众号

第三章 K8s容器运行环境安全加固

- 最小特权原则（POLP）
- AppArmor 限制容器对资源访问
- Seccomp 限制容器进程系统调用

最小特权原则（POLP）

最小特权原则 (Principle of least privilege，POLP)： 是一种信息安全概念，即为用户提供执行其工作职责所需的最小权限等级或许可。

最小特权原则被广泛认为是网络安全的最佳实践，也是保护高价值数据和资产的特权访问的基本方式。

最小特权原则（POLP）

最小特权原则 (POLP) 重要性：

- **减少网络攻击面：**当今，大多数高级攻击都依赖于利用特权凭证。通过限制超级用户和管理员权限，最小权限执行有助于减少总体网络攻击面。
- **阻止恶意软件的传播：** 通过在服务器或者在应用系统上执行最小权限，恶意软件攻击（例如SQL注入攻击）将很难提权来增加访问权限并横向移动破坏其他软件、设备。
- **有助于简化合规性和审核**：许多内部政策和法规要求都要求组织对特权帐户实施最小权限原则，以防止对关键业务系统的恶意破坏。最小权限执行可以帮助组织证明对特权活动的完整审核跟踪的合规性。

最小特权原则（POLP）

在团队中实施最小特权原则 (POLP)：

- 在所有服务器、业务系统中，审核整个环境以查找特权帐户（例如SSH账号、管理后台账号、跳板机账号）；
- 减少不必要的管理员权限，并确保所有用户和工具执行工作时所需的权限；
- 定期更改管理员账号密码；
- 监控管理员账号操作行为，告警通知异常活动。

AppArmor限制容器对资源访问

AppArmor（Application Armor） 是一个 Linux 内核安全模块，可用于限制主机操作系统上运行的进程的功能。每个进程都可以拥有自己的安全配置文件。安全配置文件用来允许或禁止特定功能，例如网络访问、文件读/写/执行权限等。

Linux发行版内置：Ubuntu、Debian

AppArmor限制容器对资源访问

Apparmor两种工作模式：

- Enforcement（强制模式）：在这种模式下，配置文件里列出的限制条件都会得到执行，并且对于违反这些限制条件的程序会进行日志记录。
- Complain（投诉模式）：在这种模式下，配置文件里的限制条件不会得到执行，Apparmor只是对程序的行为进行记录。一般用于调试。

AppArmor限制容器对资源访问

常用命令：

- apparmor_status：查看AppArmor配置文件的当前状态的
- apparmor_parser：将AppArmor配置文件加载到内核中
    - apparmor_parser <profile># 加载到内核中
    - apparmor_parser -r <profile># 重新加载配置
    - apparmor_parser -R <profile># 删除配置
- aa-complain：将AppArmor配置文件设置为投诉模式，需要安装apparmor-utils软件包
- aa-enforce：将AppArmor配置文件设置为强制模式，需要安装apparmor-utils软件包

AppArmor限制容器对资源访问

K8s使用AppArmor的先决条件：

- K8s版本v1.4+，检查是否支持：kubectl describe node |grep AppArmor
- Linux内核已启用AppArmor，查看 cat /sys/module/apparmor/parameters/enabled
- 容器运行时需要支持AppArmor，目前Docker已支持

AppArmor限制容器对资源访问

AppArmor 目前处于测试阶段，因此在注解中指定AppArmor策略配置文件。

示例：

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-apparmor
  annotations:
    container.apparmor.security.beta.kubernetes.io/<container_name>: localhost/<profile_ref>
...
```

- <container_name> Pod中容器名称
- <profile_ref> Pod所在宿主机上策略名，默认目录/etc/apparmor.d

AppArmor限制容器对资源访问

案例：容器文件系统访问限制

步骤：

1、将自定义策略配置文件保存到/etc/apparmor.d/

2、加载配置文件到内核：apparmor_parser <profile>

3、Pod注解指定策略配置名

AppArmor限制容器对资源访问

示例：限制容器对目录或者文件的访问

```
vi /etc/apparmor.d/k8s-deny-write

#include <tunables/global>
profile k8s-deny-write flags=(attach_disconnected) {
  #include <abstractions/base>
  file,  # 允许所有文件读写
  deny /bin/** w, # 拒绝所有文件写
  deny /data/www/** w,
}
```

- 第一行：导入依赖，遵循C语言约定
- 第二行：指定策略名
- 第三行：{} 策略块

访问文件权限模式：

字符	描述
r	读
w	写
a	追加
k	文件锁定
l	链接
x	可执行

匹配目录和文件：

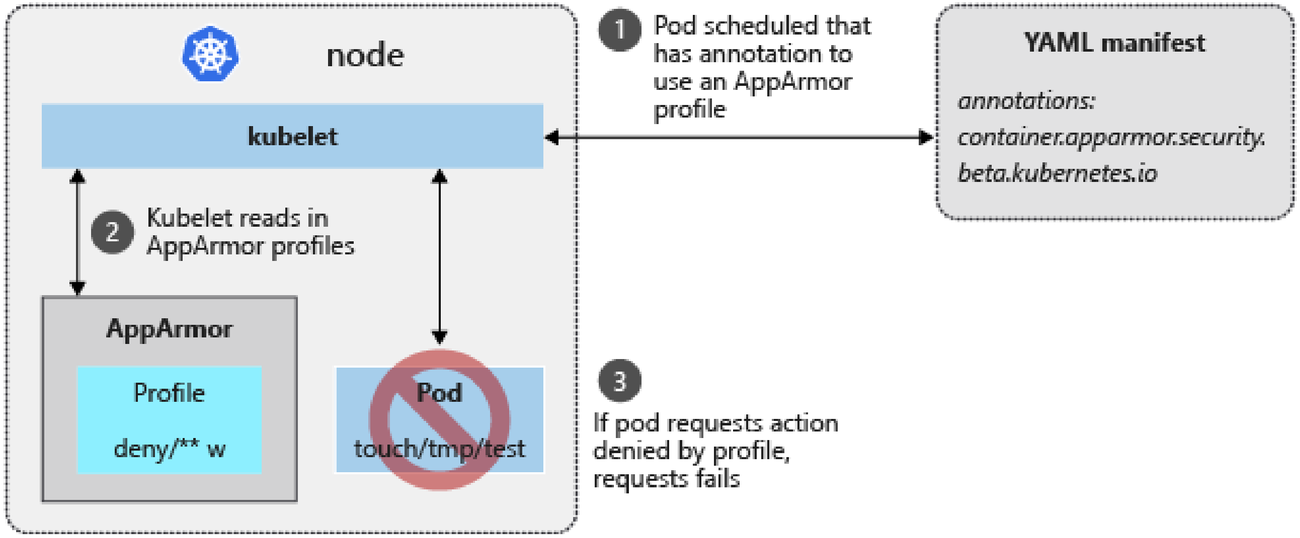
通配符	描述	示例
*	在目录级别匹配零个或多个字符	/dir/* 匹配目录中的任何文件 /dir/a* 匹配目录中以a开头的任意文件 /dir/*.png 匹配目录中以.png结尾的任意文件 /dir/a*/ 匹配/dir里面以a开头的目录 /dir/*a/ 匹配/dir里面以a结尾的目录
**	在多个目录级别匹配零个或多个字符	/dir/** 匹配/dir目录或者/dir目录下任何文件和目录 /dir/**/ 匹配/dir或者/dir下面任何目录
[] [^]	字符串，匹配其中任意字符	/dir/[^.]* 匹配/dir目录中以.之外的任何文件 /dir/**[^/] 匹配/dir目录或者/dir下面的任何目录中的任何文件

AppArmor限制容器对资源访问

示例：限制容器对目录或者文件的访问

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-apparmor
  annotations:
    container.apparmor.security.beta.kubernetes.io/hello: localhost/k8s-deny-write
spec:
  containers:
  - name: hello
    image: busybox
    command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]
```

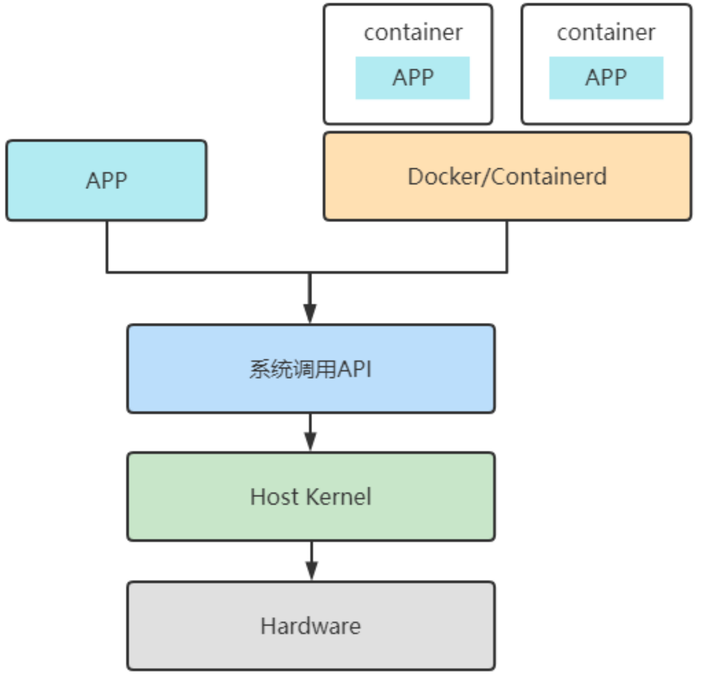
AppArmor限制容器对资源访问



工作流程

Seccomp 限制容器进程系统调用

对于 Linux 来说，用户层一切资源相关操作都需要通过系统调用来完成；系统调用实现技术层次上解耦，内核只关心系统调用API的实现，而不必关心谁调用的。



调用关系图

Seccomp 限制容器进程系统调用

Seccomp（Secure computing mode） 是一个 Linux 内核安全模块，可用于应用进程允许使用的系统调用。
容器实际上是宿主机上运行的一个进程，共享宿主机内核，如果所有容器都具有任何系统调用的能力，那么容器如果被入侵，就很轻松绕过容器隔离更改宿主机系统权限或者进入宿主机。
这就可以使用Seccomp机制限制容器系统调用，有效减少攻击面。

Linux发行版内置：CentOS、Ubuntu

Seccomp 限制容器进程系统调用

Seccomp在Kubernetes 1.3版本引入，在1.19版本成为GA版本，因此K8s中使用Seccomp可以通过以下两种方式：

- 1.19版本之前

```
annotations:
  seccomp.security.alpha.kubernetes.io/pod: "localhost/<profile>"
```

- 1.19版本+

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-seccomp
spec:
  securityContext:
    seccompProfile:
      type: Localhost
      localhostProfile: <profile>  #  Pod所在宿主机上策略文件名，默认目录：
/var/lib/kubelet/seccomp
  containers:
...
```

Seccomp 限制容器进程系统调用

示例：禁止容器使用chmod

```
mkdir /var/lib/kubelet/seccomp
vi /var/lib/kubelet/seccomp/chmod.json
{
    "defaultAction": "SCMP_ACT_ALLOW",
    "syscalls": [
        {
            "names": [
                "chmod"
             ],
            "action": "SCMP_ACT_ERRNO"
        }
    ]
}
```

seccomp基本配置文件包括三个元素：

- defaultAction：在syscalls部分未定义的任何系统调用默认动作为允许
- syscalls
  - names 系统调用名称，可以换行写多个
  - SCMP_ACT_ERRNO 阻止系统调用

Seccomp 限制容器进程系统调用

大多数容器运行时都提供一组允许或不允许的默认系统调用。通过使用 runtime/default 注释 或将 Pod 或容器的安全上下文中的 seccomp 类型设置为 RuntimeDefault，可以轻松地在 Kubernetes 中应用默认值。

Docker默认配置说明：https://docs.docker.com/engine/security/seccomp/

课后作业

1、在工作节点上加载课堂上讲解的apparmor策略文件k8s-deny-write，并在Pod中应用该策略

2、在工作节点上加载课堂上讲解的seccomp文件，禁止容器里使用chmod命令，并在Pod中应用该策略

谢谢

阿良个人微信

DevOps技术栈公众号

阿良教育：www.aliangedu.cn