Docker下无代理实时病毒查杀技术

徐荣维(elemeta)

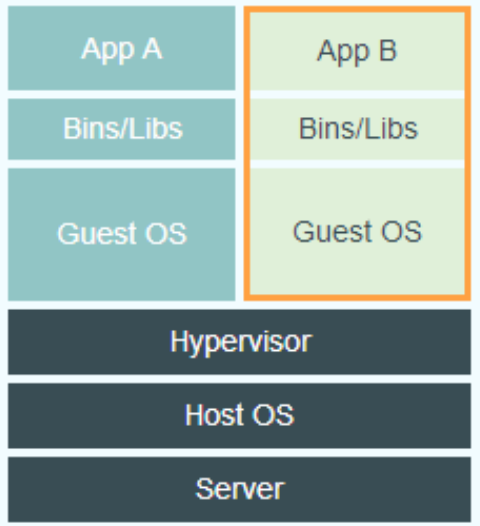
椒图科技【天择实验室】

提要

1. Docker自身安全概述
2. 无代理查杀的可行性分析
3. 无代理查杀的实现

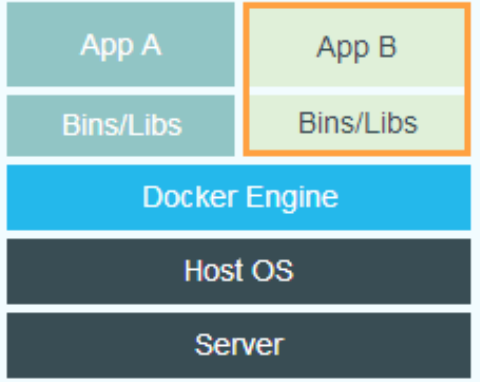
Docker

- 启动速度快
- 资源利用率高
- 性能开销小
- 统一运维和开发环境
- 使应用的打包与部署自动化



Virtual Machines

Each virtualized application includes not only the application - which may be only 10s of MB - and the necessary binaries and libraries, but also an entire guest operating system - which may weigh 10s of GB.



Docker

The Docker Engine container comprises just the application and its dependencies. It runs as an isolated process in userspace on the host operating system, sharing the kernel with other containers. Thus, it enjoys the resource isolation and allocation benefits of VMs but is much more portable and efficient.

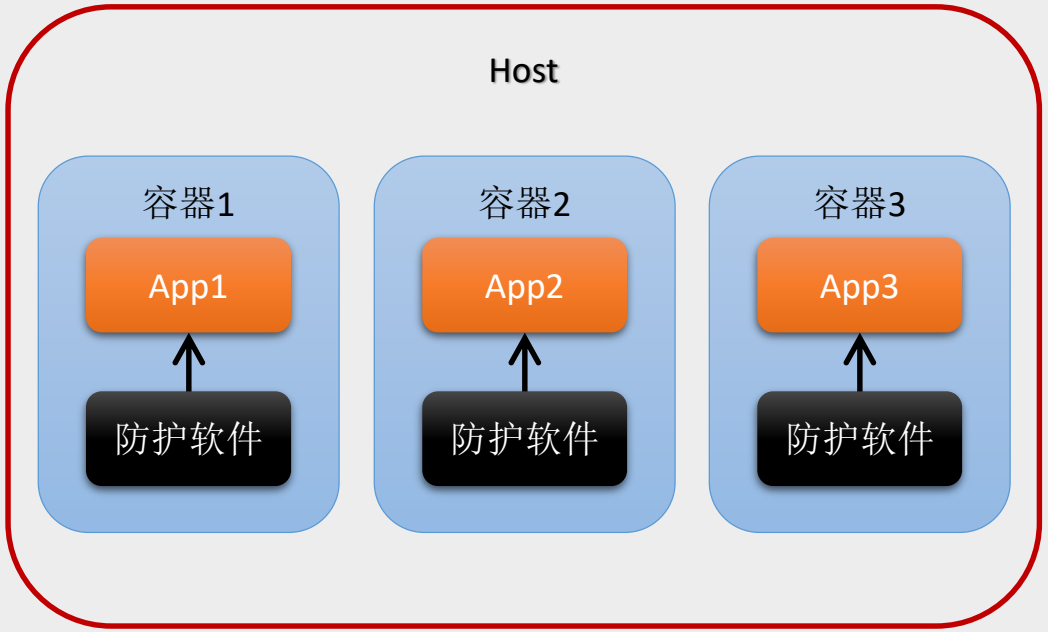
Docker安全

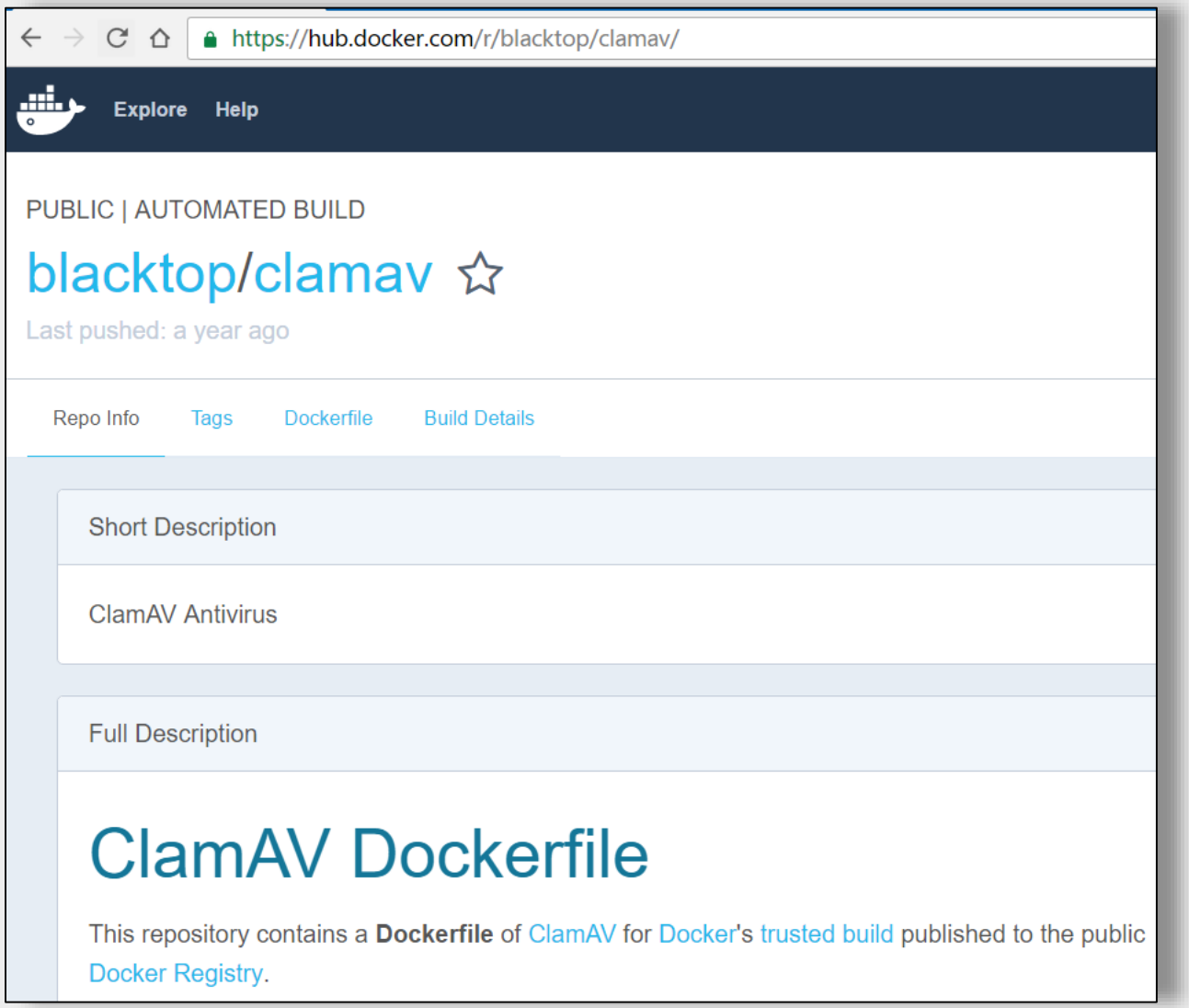
- 不对Host造成影响、不对其他容器造成影响
  - 隔离性(namespaces, cgroups)
  - Docker daemon的攻击面(root权限、REST API)
  - 可信任的镜像
  - Seccomp、SELinux、AppArmor ...
  - https://docker.github.io/engine/security/

容器中的应用程序安全

- 应用程序的安全问题和物理机的情况是一样的
- 依然面临恶意代码(比如：Rootkit、木马、Webshell)的威胁
- 主机防护软件依然有存在的价值！
  - ClamAV
  - Avria
  - 云锁 for Linux
  - AVG for Linux
  - Comodo for Linux

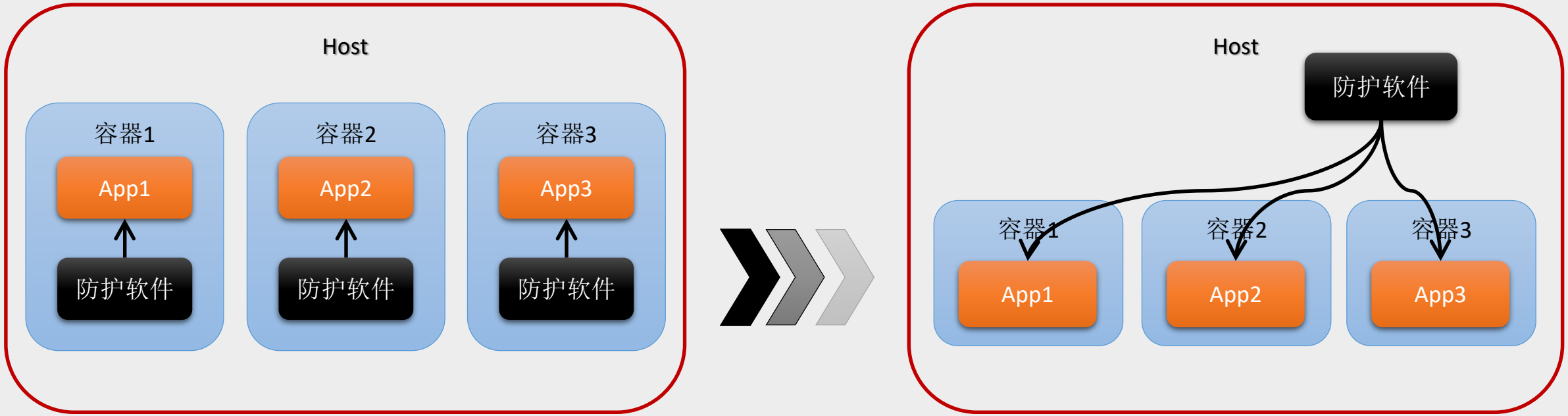
目前的防护软件部署方式





我的想法 – 无代理

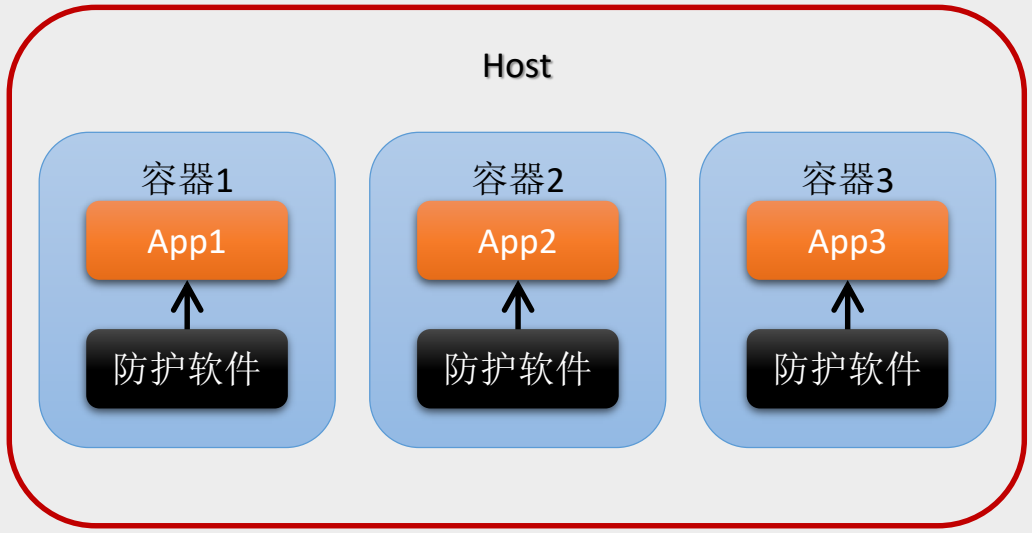
能否只在Host上安装防护软件，
就能够实现对容器内的防护？



无代理部署的优势

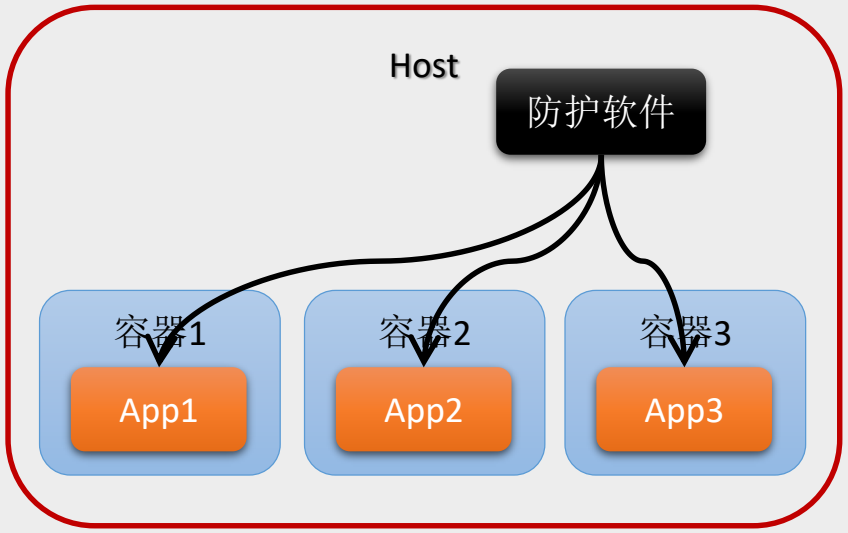
容器内部署方案

- N个进程
- N份磁盘空间
- N份内存使用量
- 消耗CPU资源多
- 升级时需要重新制作镜像



无代理部署方案

- 1个进程
- 1份磁盘空间
- 1份内存使用量
- 消耗CPU资源少
- 升级时不更改镜像



DO IT !

黑客上传恶意代码

数据流

Webshell

Web网页

httpd进程

创建文件

Linux文件系统

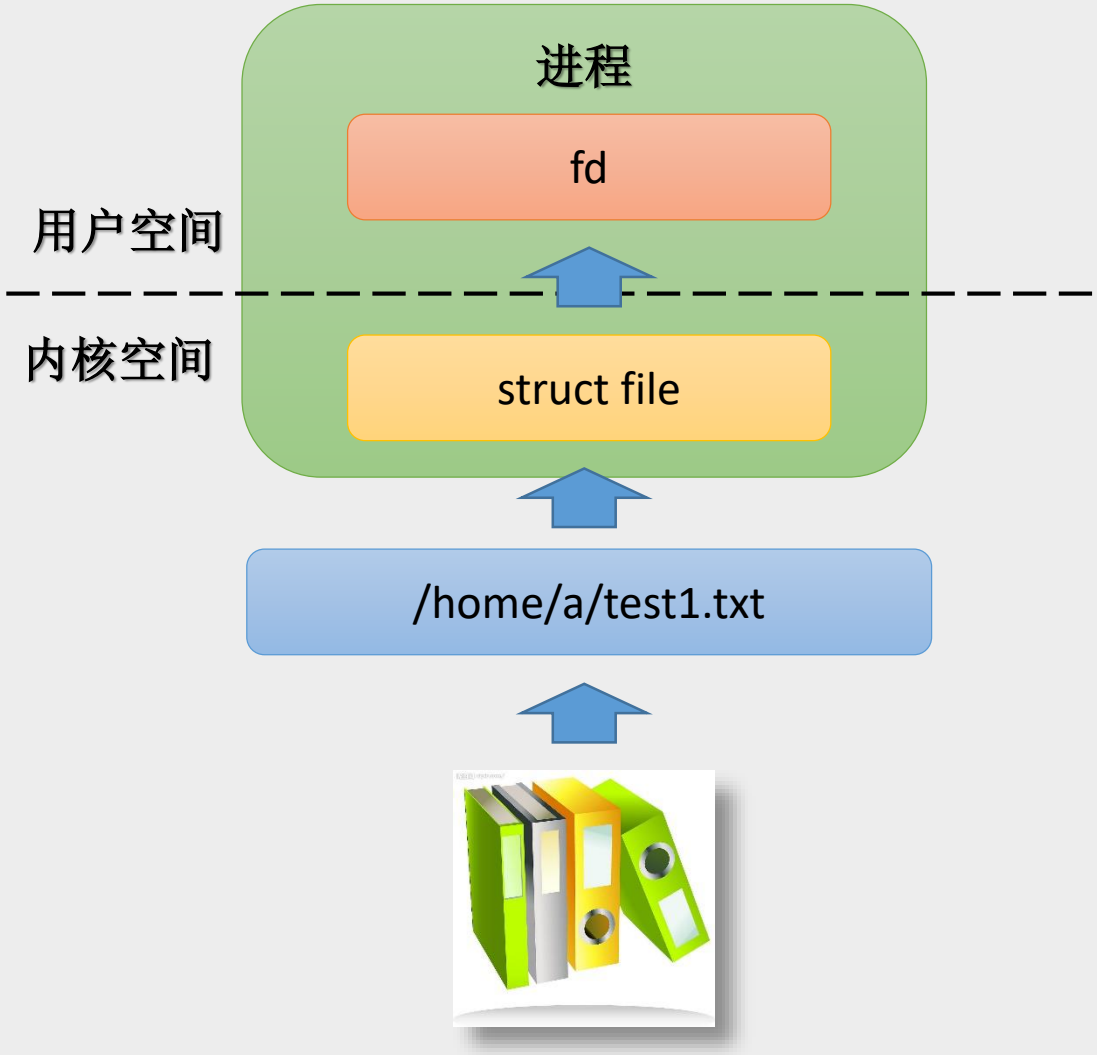
需要解决的两个问题

Host中的主机防护软件

1. **如何拦截容器内的文件操作**
2. **如何读取容器内的文件**

文件系统内核对象

```
open("/home/a/test1.txt", O_CREAT|O_RDWR, ...) = fd;
```



文件描述符 (fd)

文件内核对象 (struct file)

名字空间隔离

挂载点不同

路径内核对象 (struct path)

inode

系统全局

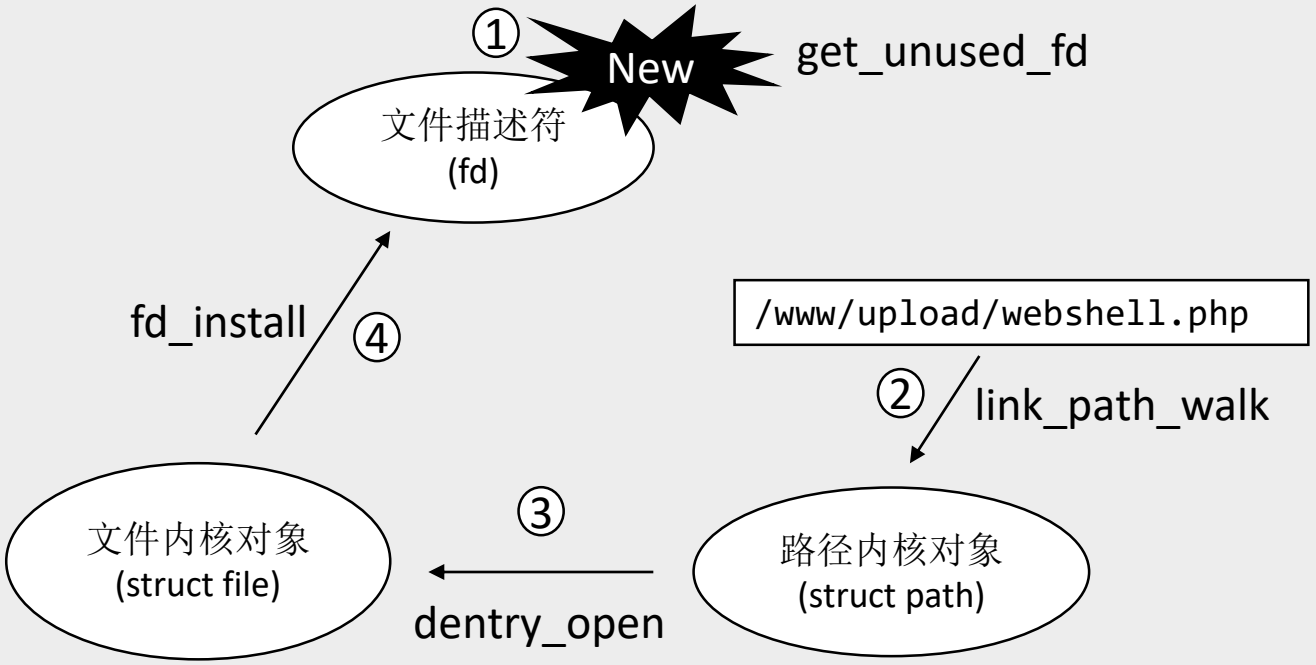
如何创建一个文件

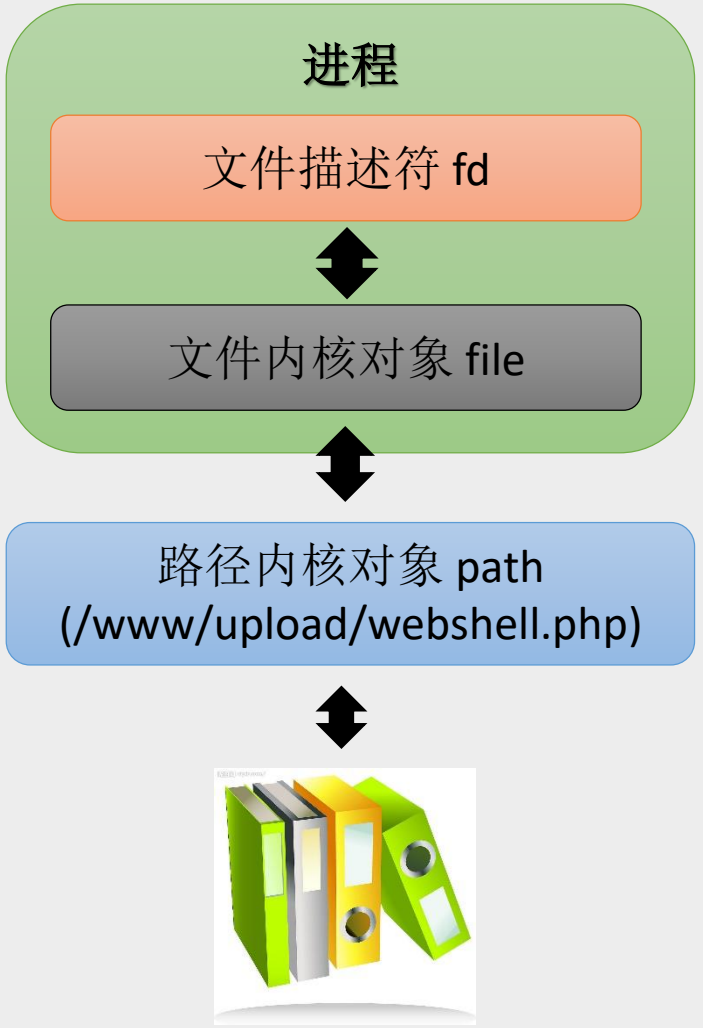
1. 使用open()创建文件，并得到文件描述符
2. 使用write()将内容写到文件中
3. 使用close()关闭文件描述符

```
open(“/www/upload/webshell.php”, O_CREAT|O_RDWR, ...) = fd;
write(fd, “<?php  ... ?>”, length) = length;
close(fd);
```

文件操作的内核实现

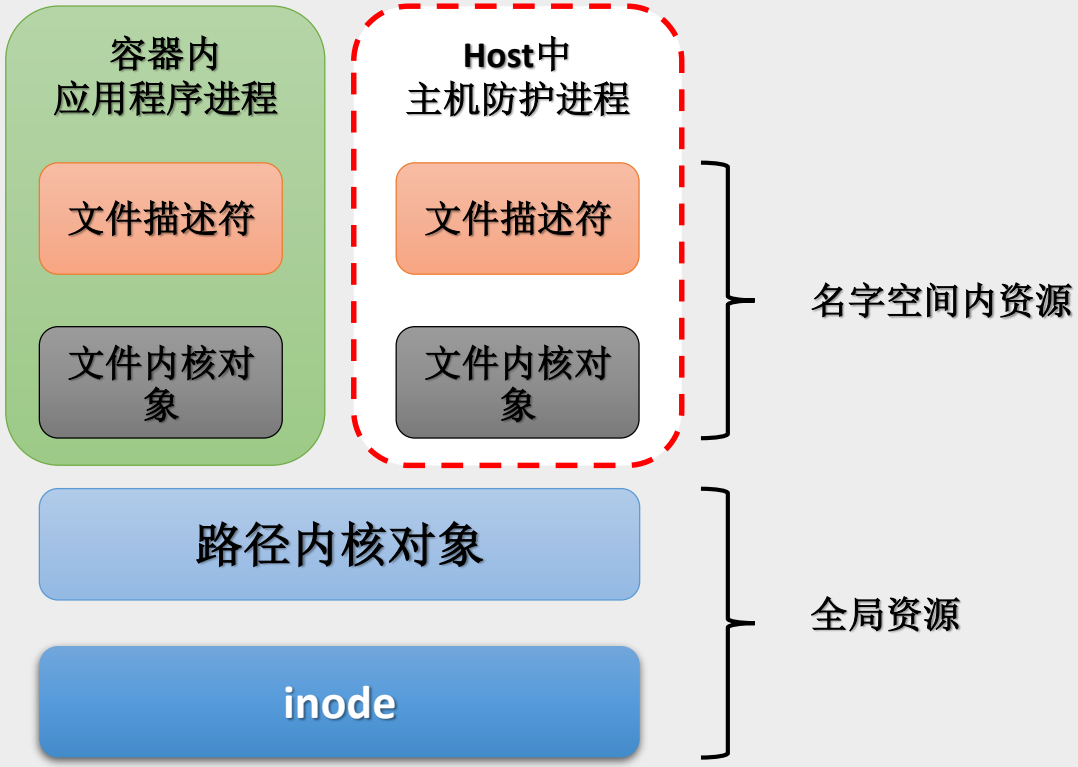
```
open("/www/upload/webshell.php", O_CREAT|O_RDWR, ...) = fd;
```





小结

- 如何拦截容器内文件操作
  - 因为共享内核，内核中可以看到每个容器发生的事情
  - 通过HOOK系统调用即可
- 如何读容器内的文件
  - 模拟open()的实现，复制一份fd和文件内核对象。
  - 查杀进程使用fd读文件即可



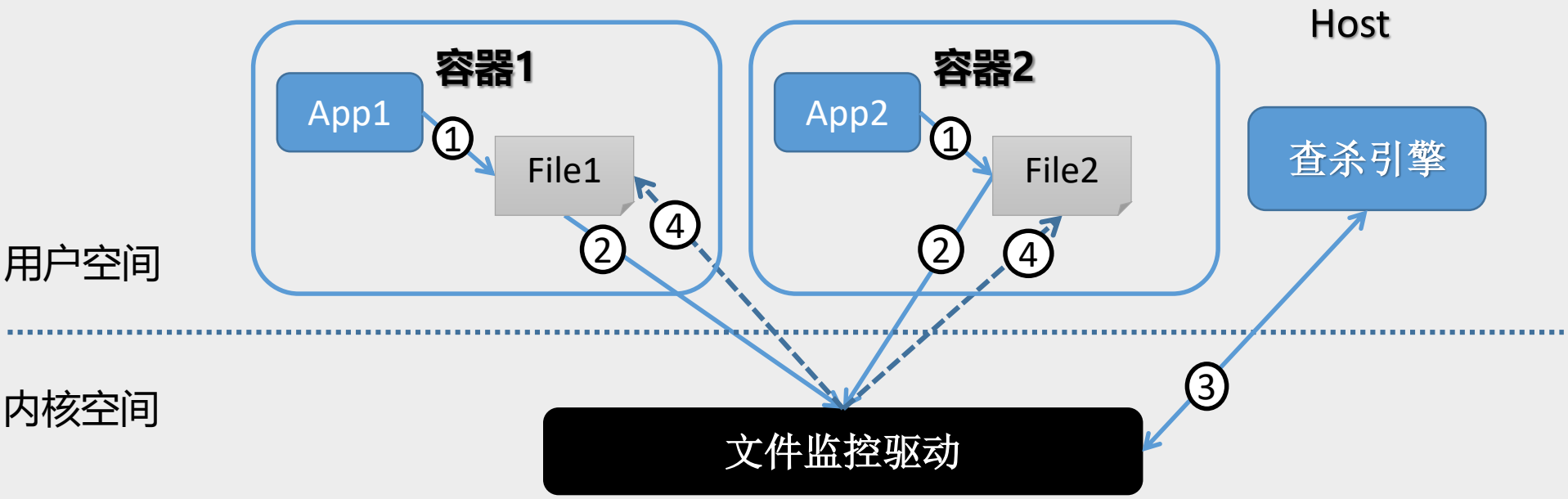
设计

我需要一个文件监控驱动和查杀引擎

文件监控驱动
(Ring0)

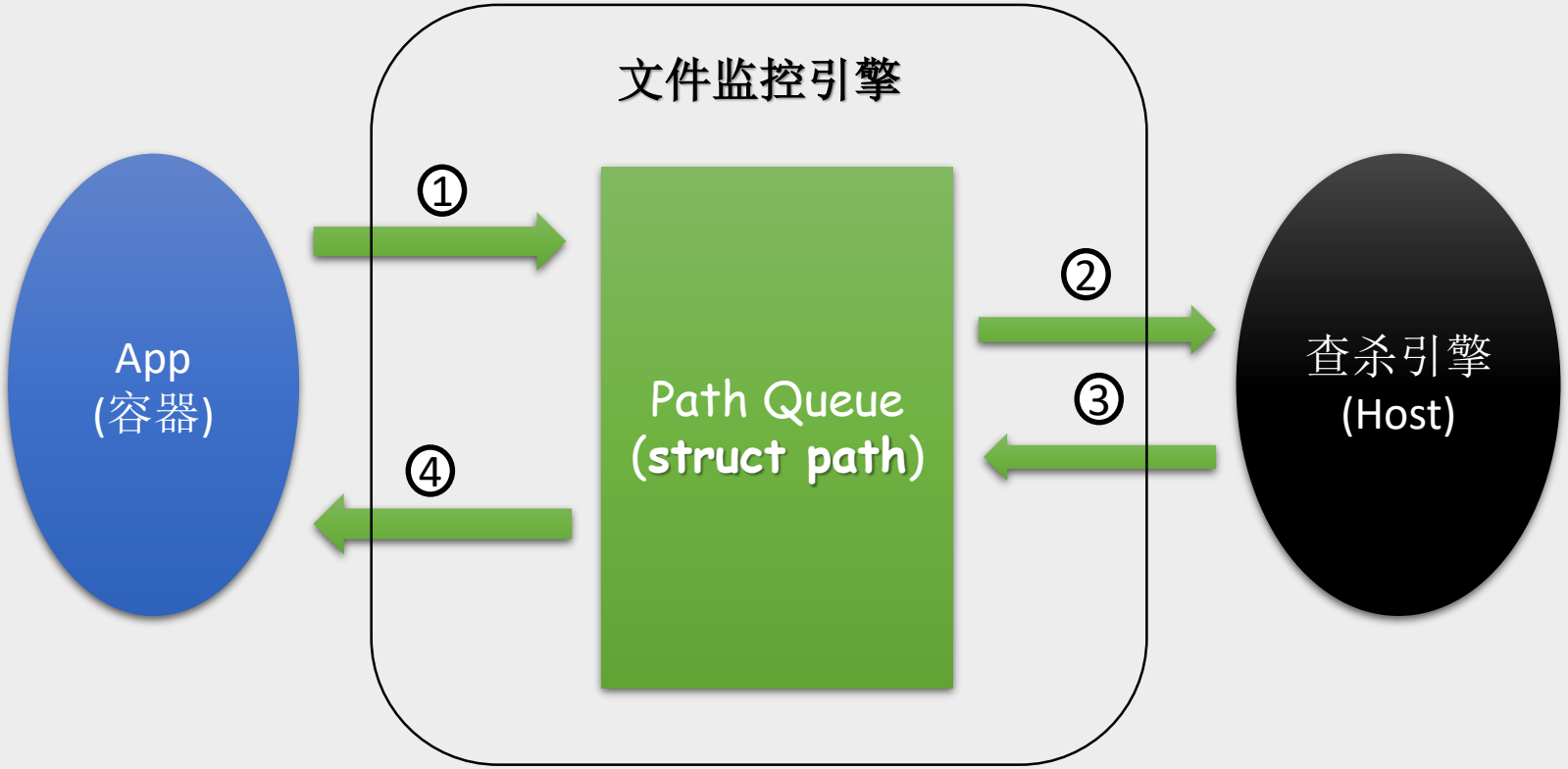
查杀引擎
(Ring3)

工作流程



1. App创建文件
2. Driver捕获容器内创建的文件
3. Driver将文件交给Host中的病毒扫描引擎，并得到扫描结果
4. 结果是Allow则放过该操作；如果Deny则拦截该操作

驱动工作流程



1. 拦截文件操作，将路径内核对象加入到待查杀文件队列
2. 查杀引擎从驱动中取出路径，并待查杀文件的文件描述符
3. 反馈查杀结果给驱动(Allow、Deny)
4. 拦截操作

效果如何


```
insmod ...
./fscanner
```


```
sudo docker run -it centos /bin/bash
echo “<?php echo 'hello world';?>” > 1.php
echo “<?php @eval($_POST['c']);?>“ > 2.php
ls -l
```


```
sudo docker run -it centos /bin/bash
echo “<?php echo 'hello world';?>” > 1.php
echo “<?php @eval($_POST['c']);?>“ > 2.php
ls -l
```

谢谢

徐荣维 (elemeta)
xurw@jowto.com
椒图科技天择实验室