# 塔防在CipherGateway安全加固方面的思路及实践

炼石网络CipherGateway 白小勇

# 提纲

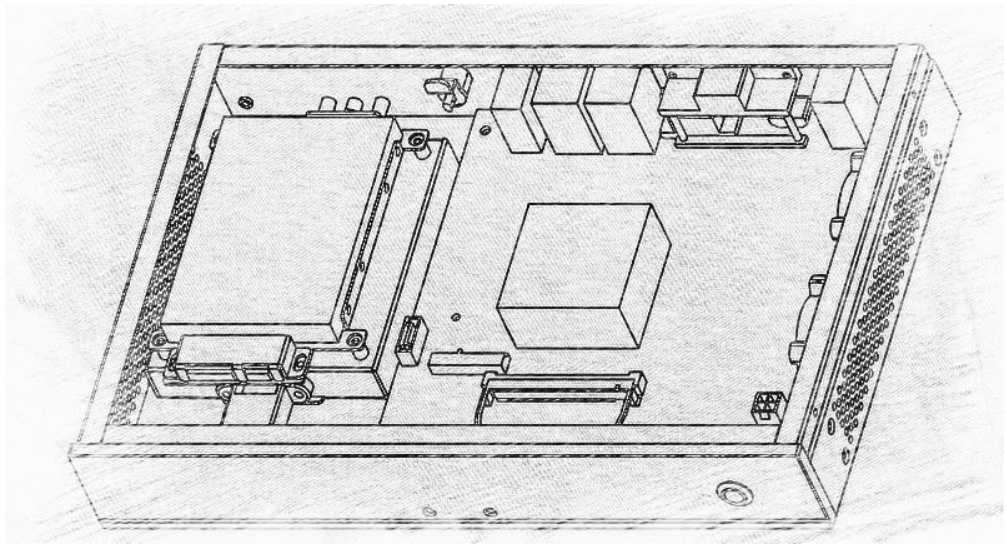
- 我们为什么要塔防？
- 实际案例分析
- 后续工作

# 背景

- CipherGateway让用户放心的上云
- 敏感数据的的明文留在企业内部，只有密文上传至云端

# 为什么CipherGateway产品本身需要塔防?

- CipherGateway Box上留存的数据至关重要
- 对客户整体安全来说，CGBox是重要一环

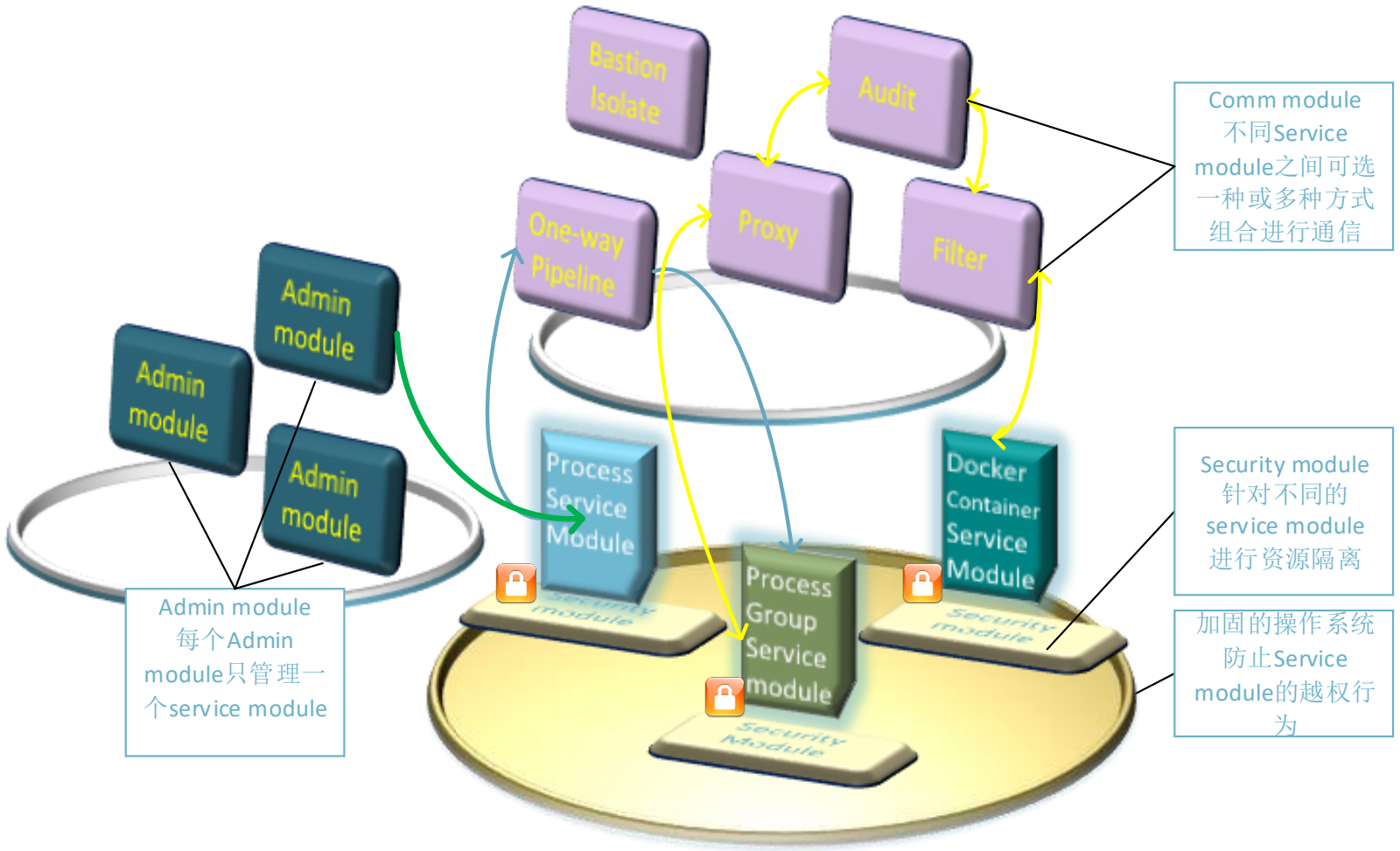


# 提纲

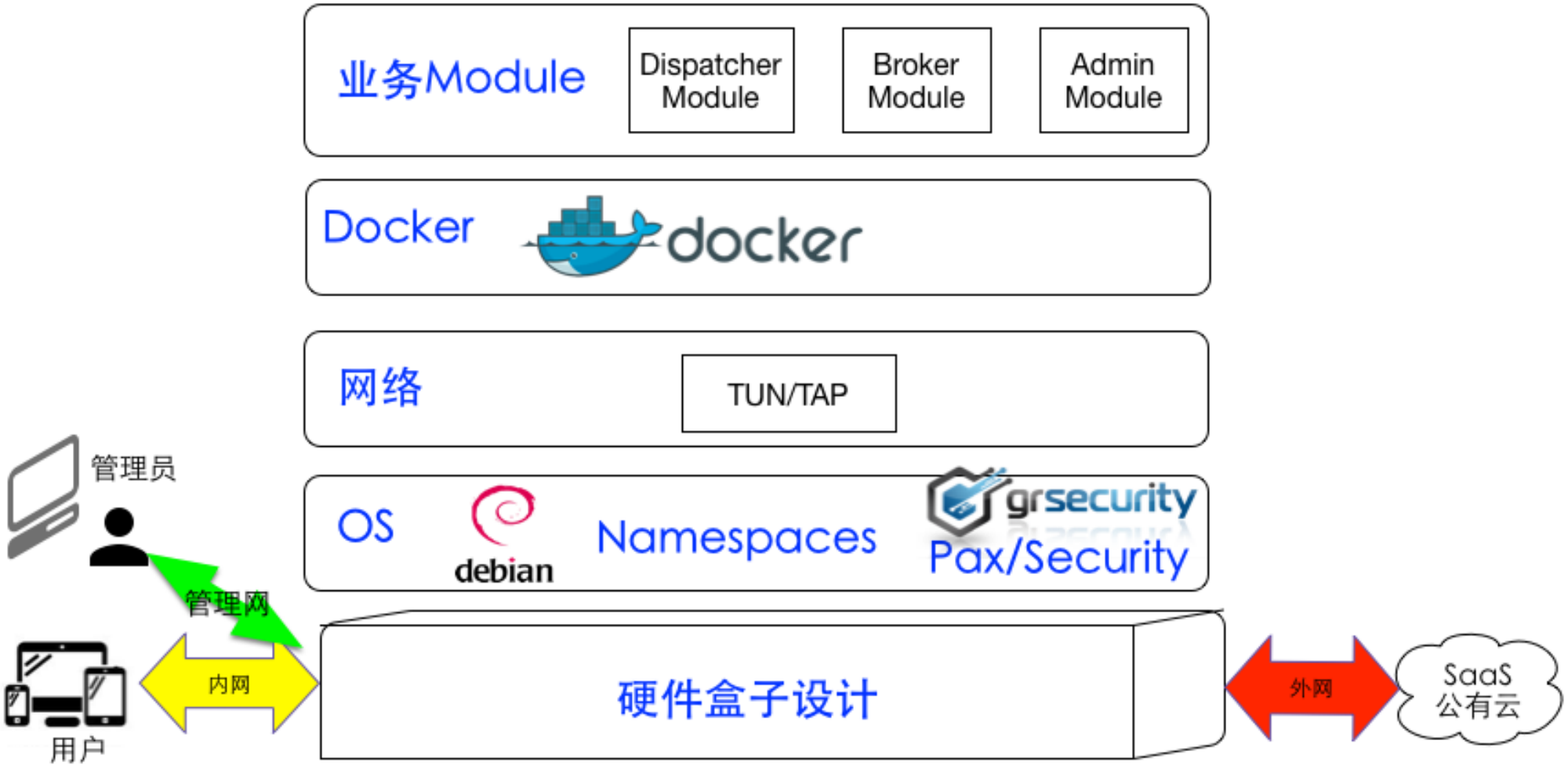
- 我们为什么要塔防？
- 实际案例分析
- 后续工作

# 抽象概念—业务Module

- 转发型Module
- 处理型Module
- 管理型Module

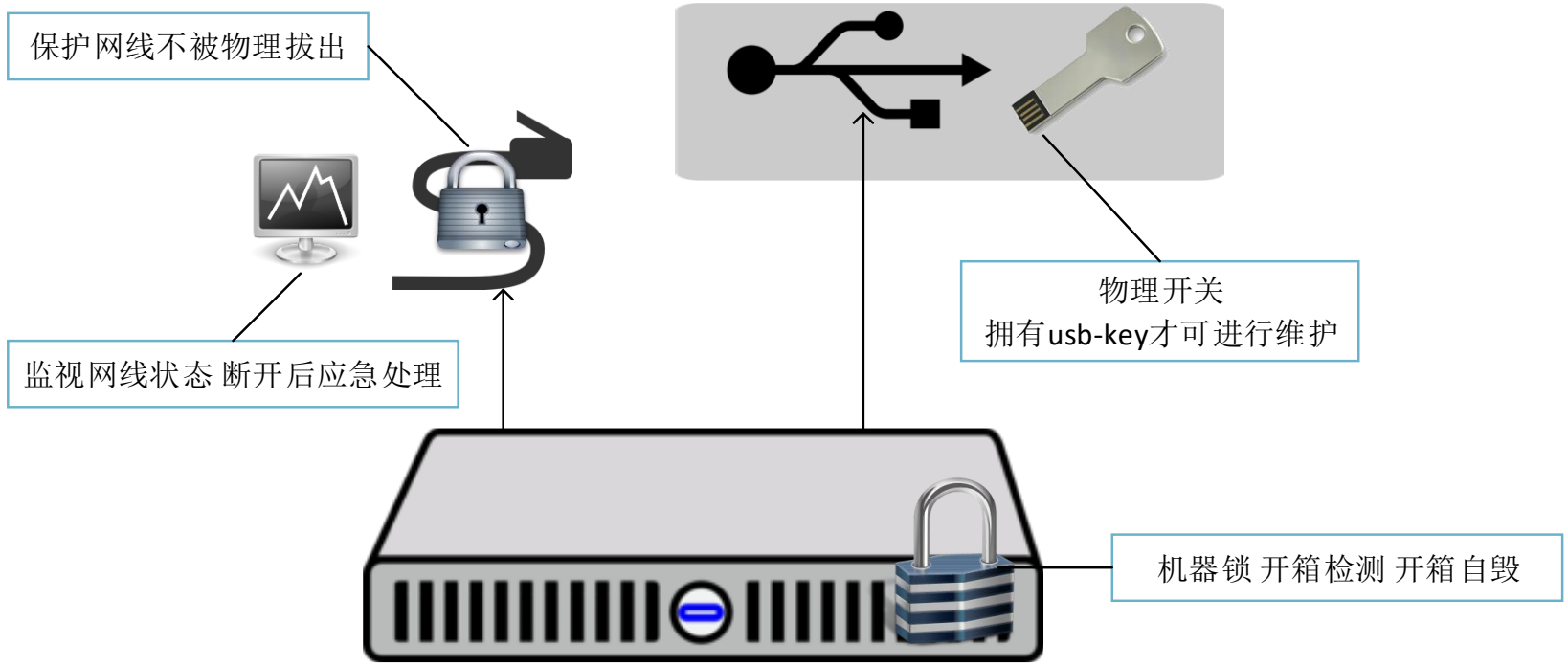


# 安全加固点概要图


业务Module
Dispatcher Module
Broker Module
Admin Module
Docker
docker
网络
TUN/TAP
OS
debian
Namespaces
grsecurity
Pax/Security
管理员
管理网
内网
用户
硬件盒子设计
外网
SaaS
公有云

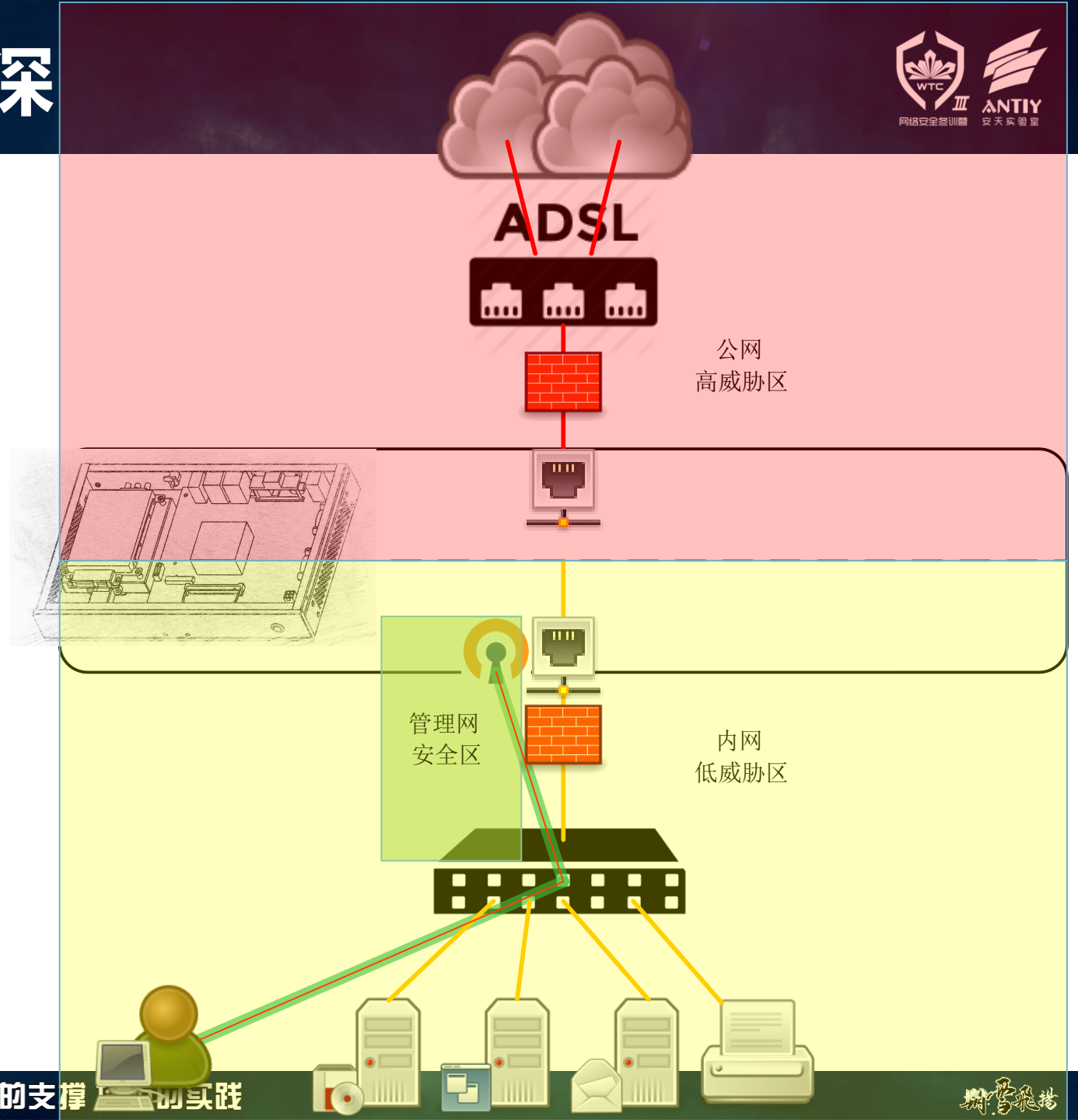
# CGBox硬件设计

- 开箱保护
- 网口保护
- 监视状态
- 物理开关
- 考虑进场运维



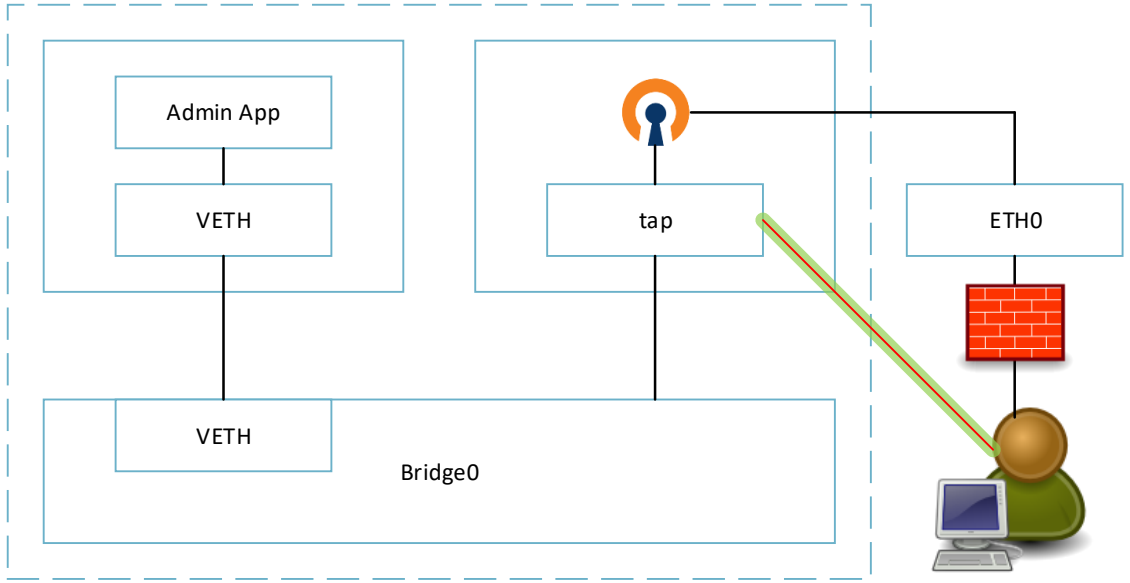
# 网络安全部署的防御纵深

- 红 公网 严重威胁
- 黄 内网 低威胁
- 绿 管理网 保证安全



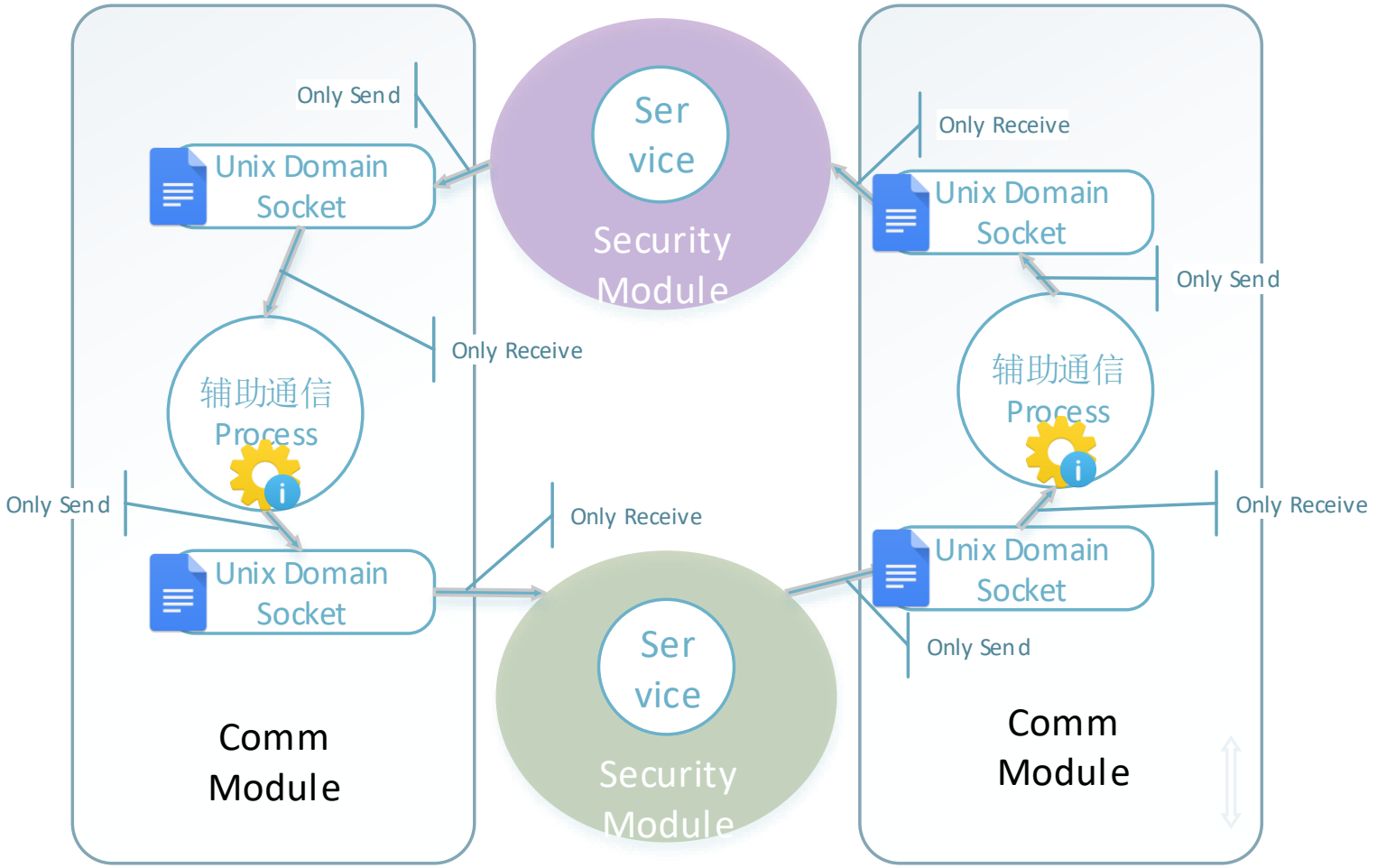
# 网络部署细节

- Openvpn模拟管理网
- 关闭kernel forward
- Iptables手工白名单转发
- Namespace绑定虚拟网卡



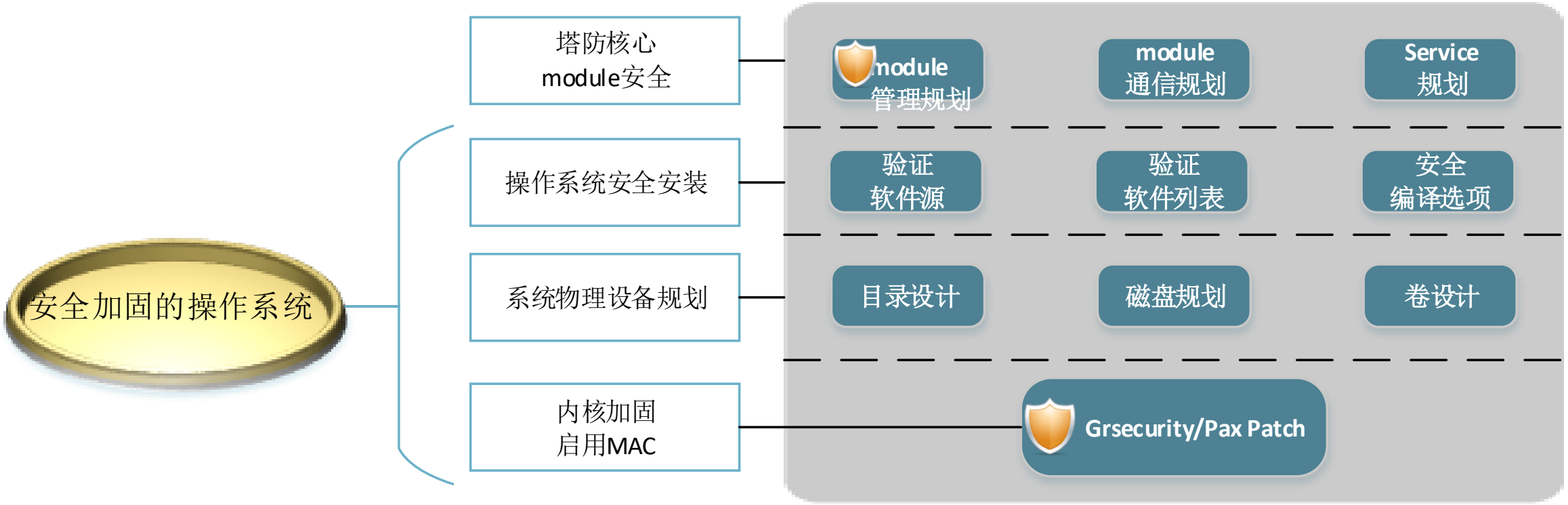
# 业务Module之间的通讯

- Bastion Isolate
- One-Way Pipeline
- Filter
  - WAF
- Audit



# OS的安全构建

- 安全加固的操作系统

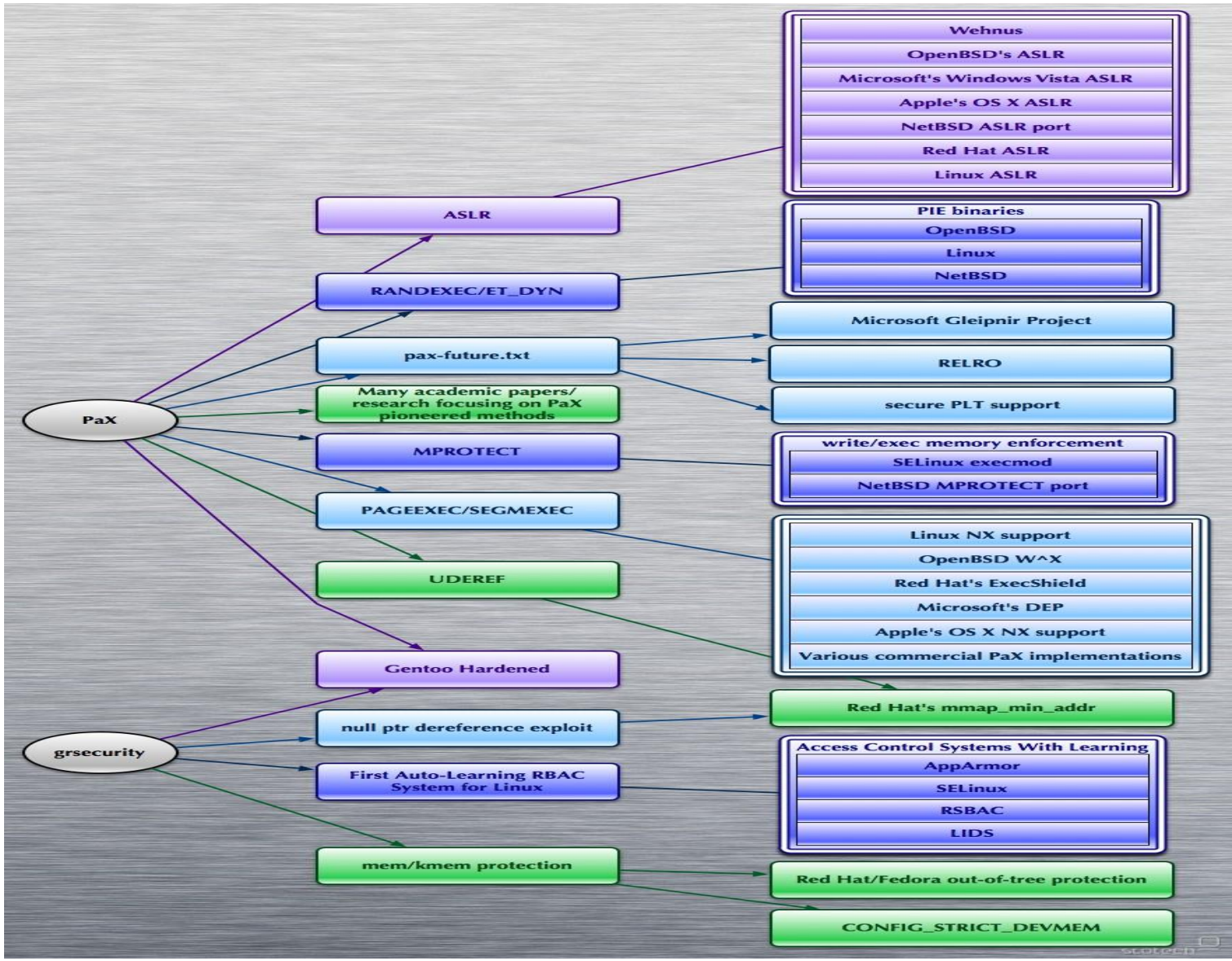


# PaX简介

- 针对linux kernel的一个加固版本的补丁
- 让linux内核的内存页受限于最小权限原则
  - Executable space protections
  - Address space layout randomization
  - Binary markings

# Grsecurity

- PaX
- RBAC
- Chroot restrictions
- Auditing



# 构建安全加固Linux的一个实践步骤

- 1、最小化安装Debian8 GNU/Linux
  - 适当发行包
  - System Utils裁剪
  - 自己的软件源
- 2、安装PaX/Grsecurity加固
  - Grsecurity配置项要结合业务Module的要求
- 3、使用STIG验证
  - 由DISA为了IT安全态势给DoD(美国国防部）提供的一套防御指南
  - stig-4-debian
  - check.sh => STIG-Checking-2015-??-??.log

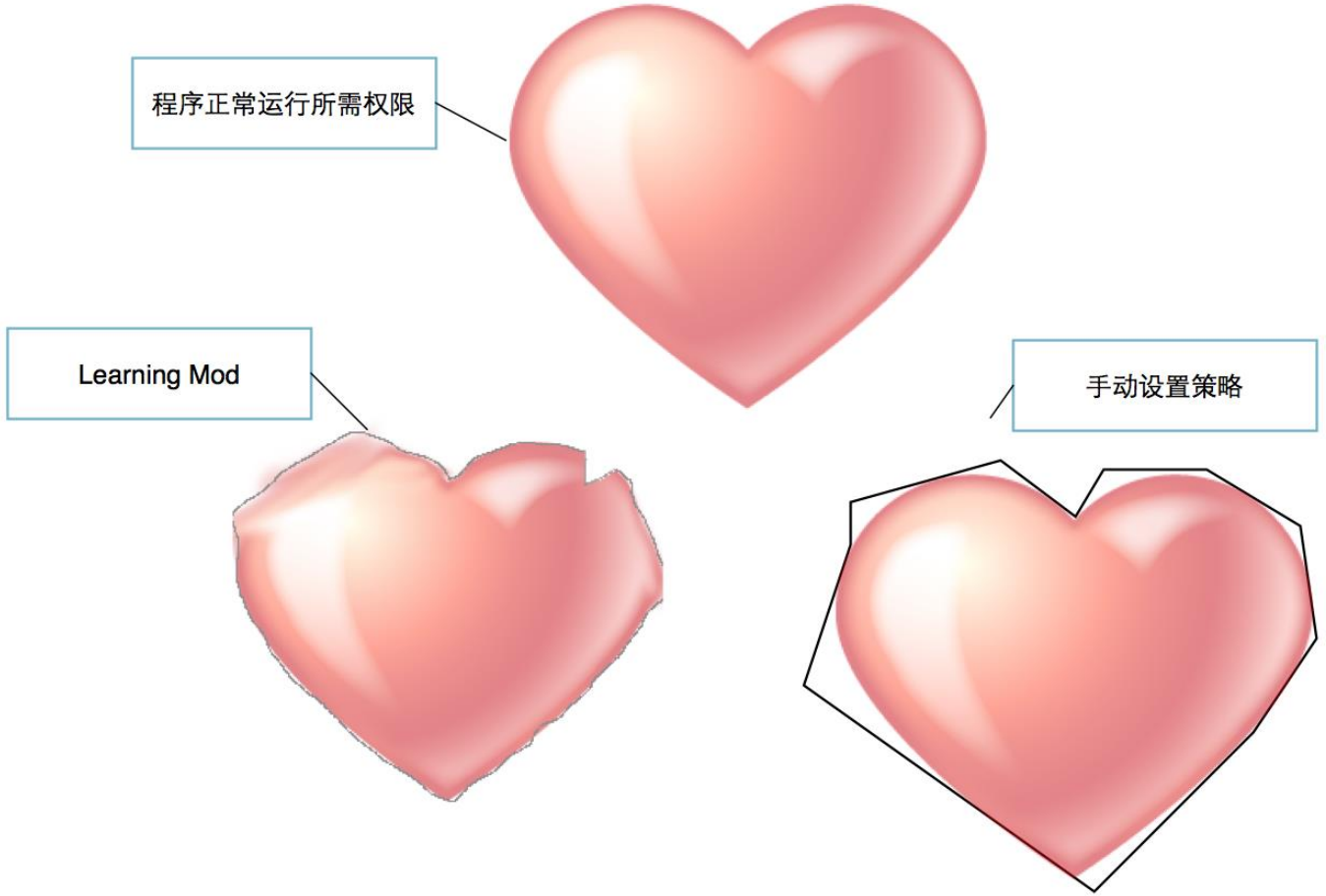
# 构建安全加固的OS实践步骤

## 4、结合业务完善物理机RBAC系统，达到最小权限要求

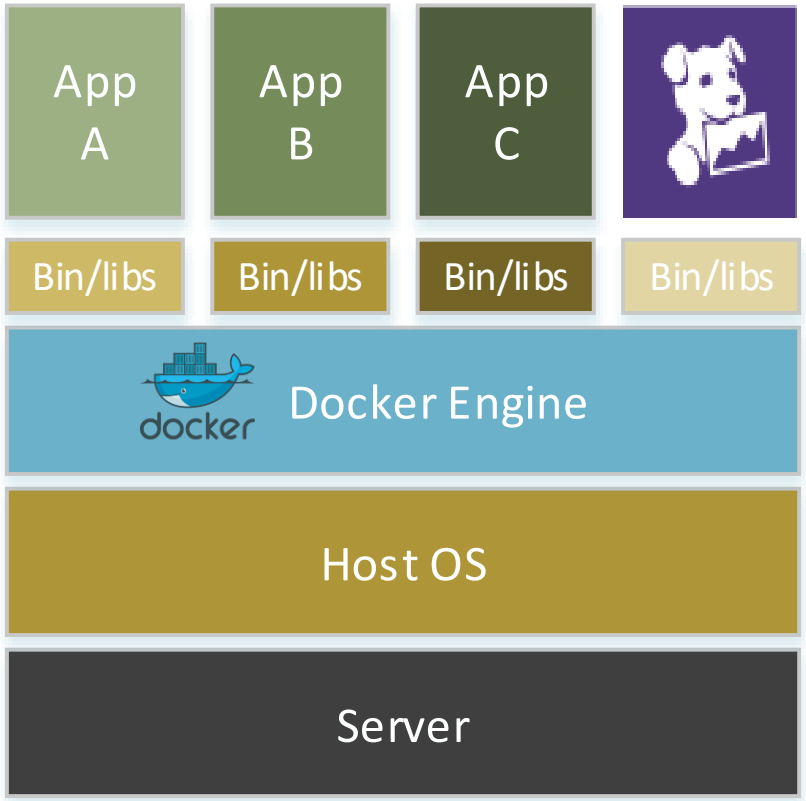
- Role（角色）：如“role admin sA“，“role default ”，”role root uG“，表明有三个角色，分别是admin , default, root
- Subject（主体），每个subject中首先定义了一个可执行程序（注意，一个subject中只定义一个可执行程序，要对多个可执行程序进行定义的话，就需要多个subject），或者更准确一点，定义了一个运行在系统中的进程。然后后面跟着一系列的object，用来规定当前这个进程的权限。每个角色定义的后面都可以跟一个或多个subject
- Object（客体）：每个subject中都会有若干的object，表示每个进程都有若干个操作对象，这些操作对象一般来说都是一些目录，文件等等，用来规定当前这个subject中的进程对这些文件拥有哪些权限
- Gradm Learning

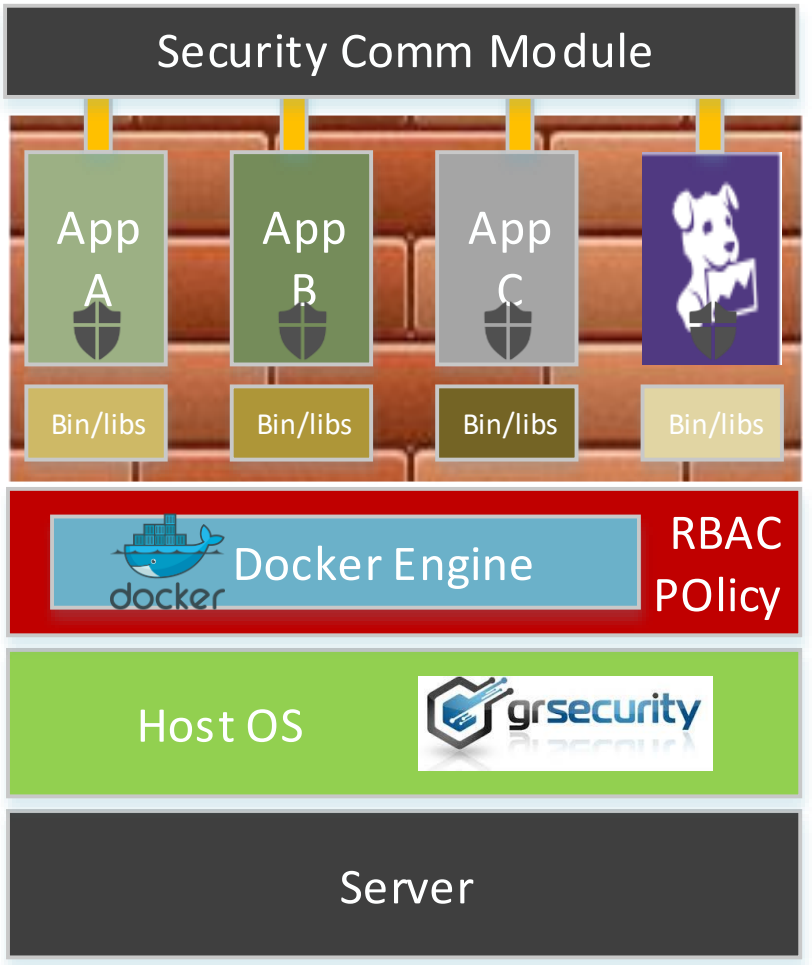
# Grsecurity RBAC策略

- Learning 不够准确
- 自己的白名单机制
- 不一定完美但要行之有效



# 对Docker本身加固



# Docker Namespaces

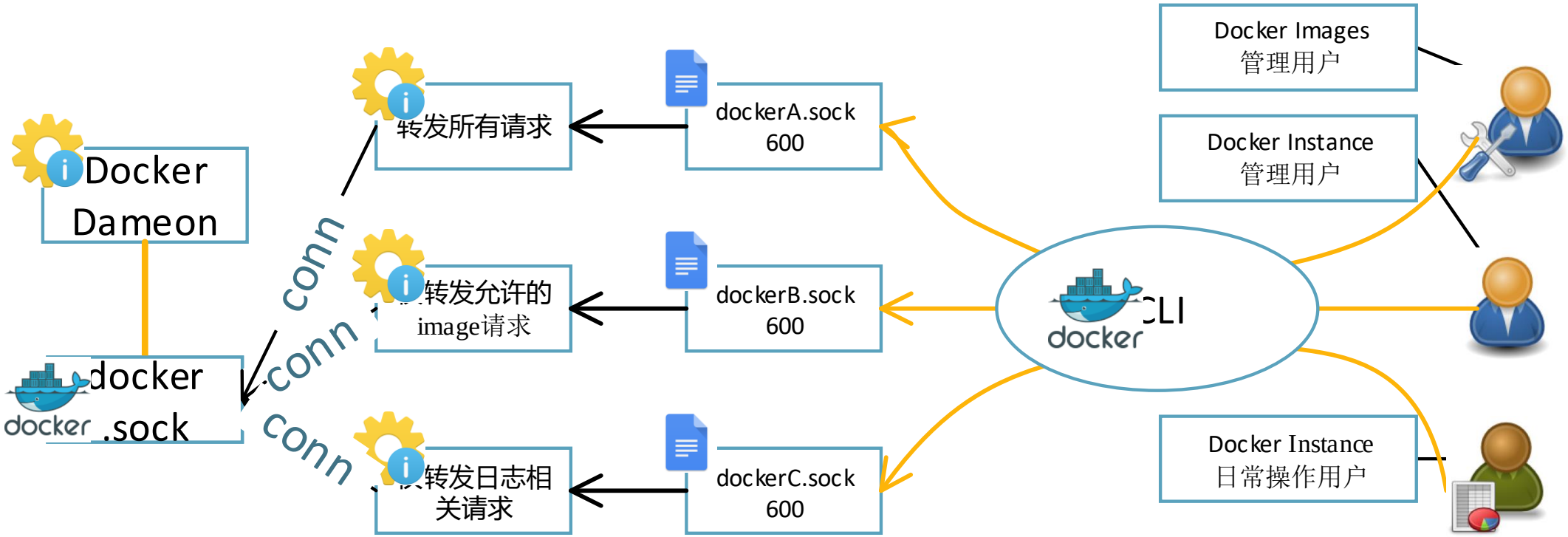
- Namespaces （UTS，IPC，PID，Mount，NetWork）
- 其他夭折的Namespaces方案（Audit,Syslog,Device,Time…）
- User Namespaces (try merge in docker 1.9.0)

# Docker使用方式安全加强

- Dockerfile的安全编写实践
- 每个Docker Image绑定一个对应的RBAC policy文件
- Docker默认的机制并不是安全优先的
  - Docker run -V –P –Link均导致难以加入安全点
  - -V 可以用namespace mount只读
  - -P --link 废除docker本身的forward，手工iptables白名单

# Docker管理方案

- 针对Docker的RBAC

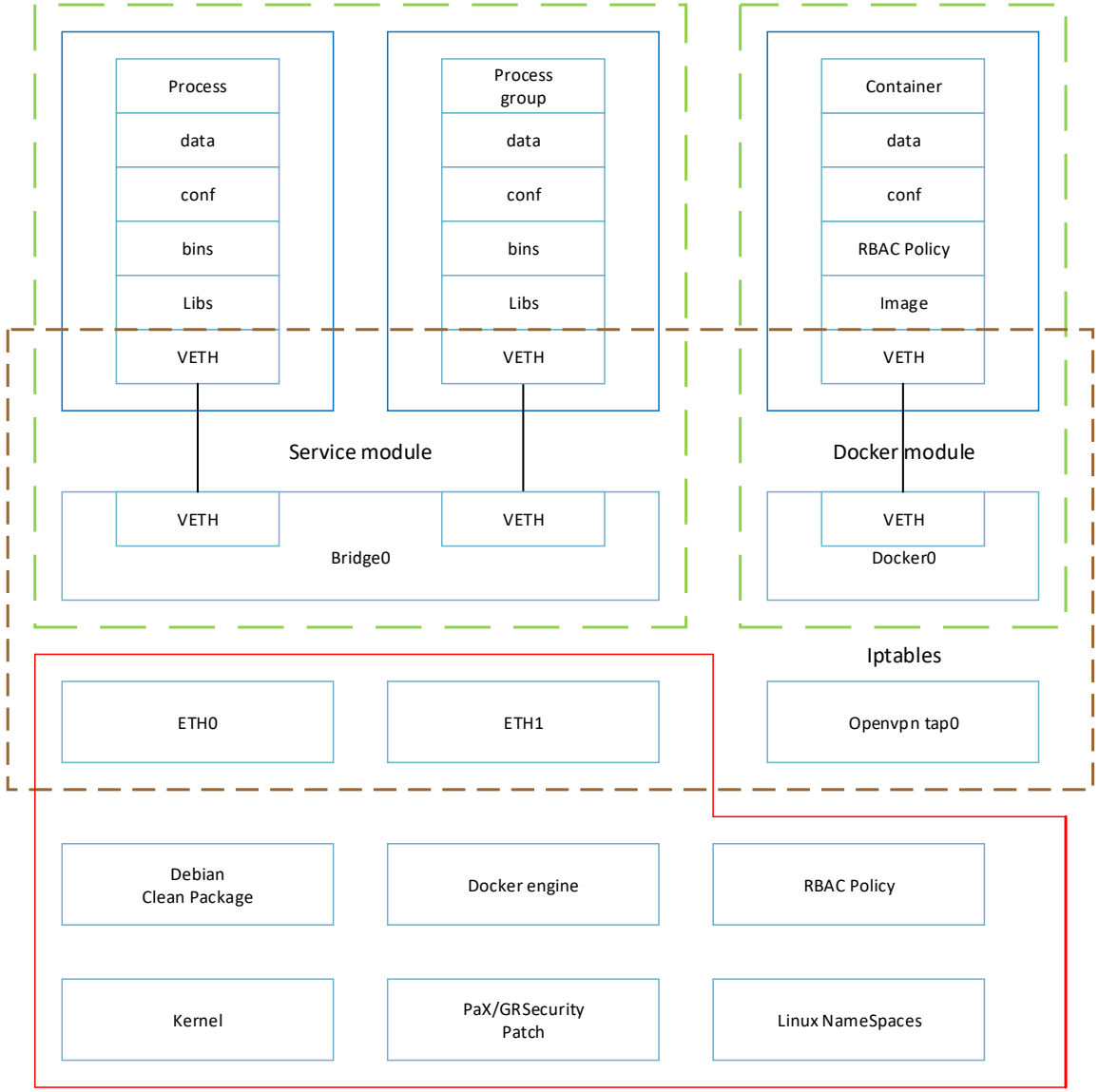


# Linux Namespaces

- Docker之外的NS应用
- Mount
- PID
- IPC
- Network

# 网络视角回顾

- Module
  - Service module
  - Docker module
- Network
- OS

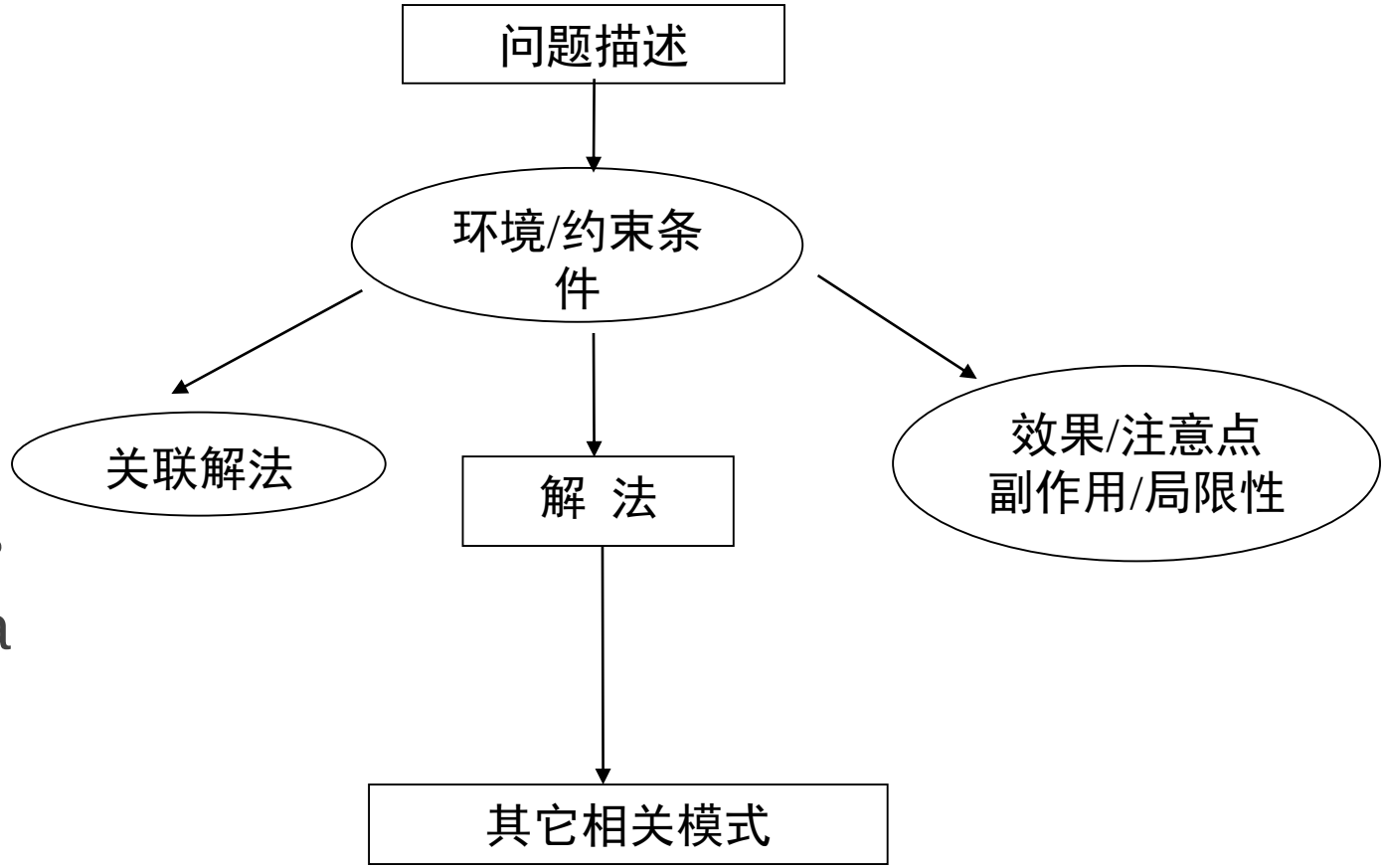


# 提纲

- 我们为什么要塔防？
- 实际案例分析
- 后续工作

# 设计模式发展

- 模式之父Alexander给出的模式经典定义是：每个模式都描述了一个在我们的环境中不断出现的问题，然后描述了该问题的解决方案的核心，通过这种方式，我们可以无数次地使用那些已有的解决方案，无需再重复相同的工作。
- 模式：A pattern is a solution to a problem in a context——模式是在**特定环境**中**解决问题**的一种**方案**。



# GoF 23 种设计模式


- 历史性著作《设计模式：可复用面向对象软件的基础》一书中描述了23种经典面向对象设计模式，创立了模式在软件设计中的地位。
- 由于《设计模式》一书确定了设计模式的地位，通常所说的设计模式隐含地表示“面向对象设计模式”。但这并不意味“设计模式”就等于“面向对象设计模式”。
- 希望一起来提炼塔防设计模式。

# 谢谢大家

THANK YOU FOR YOUR ATTENTION

白小勇

bxy@ciphergateway.com