准备 2 台虚拟机，配置如下：

vm1（eth1:192.168.2.20），vm2(eth1:192.168.2.30)

vm1 创建账户 netadm,softadm,uesradm,procadm;为所有的账户设置初始密码 123456

vm1 创建账户 devel1,devel2,devel3,test1,test2,test3;为所有账户设置初始密码 passwd

vm1 设置 test3 的账户过期时间为 2019-12-12.

vm1 使用 passwd 命令临时锁定 devel3 账户

vm1 为/etc/resolv.conf 文件添加 i 锁定属性，为/etc/hosts 添加 a 仅可追加属性

vm1 用 test1 用户登陆系统，使用 su 命令切换为 test2 账户在 tmp 下创建一个文件(非交互模式)

vm1 使用 root 登陆系统，设置 sudo 权限，要求如下：

vm1 让 netadm 能以 root 的身份执行网络管理的任务（参考 sudo 命令别名）

vm1 让 softadm 能以 root 的身份执行软件管理的任务

vm1 让 useradm 能以 root 的身份执行账户管理的任务（不能修改 root 密码）

vm1 让 procadm 能以 root 的身份执行进程管理任务（如杀死进程）

vm1 设置虚拟机 ssh 配置，禁止 root 远程本机，设置 sshd 黑名单，禁止 test3 从任何主机远程本机

vm1 真实主机创建一对 ssh 密钥，让真实机可以无密码远程虚拟机，观察密钥在虚拟机中的位置

在 vm1 主机使用 gpg 软件对/etc/rc.d/rc.local 文件进行对称加密，并将加密文件传给 vm2

在 vm2 对主机 vm1 传来的加密文件进行解密

在 vm1 上使用 gpg 创建非对称密钥对，并将公钥到处传给 vm2

在 vm2 主机将 vm1 传过来的公钥导入，并使用公钥对/etc/sysctl.conf 文件加密，并将加密文件传给 vm1，在 vm1 主机使用自己的私钥解密该文件

在 vm1 主机使用私钥给文件/etc/sysctl.conf 文件签名，在 vm2 主机验证签名

使用 aide 软件对/bin/和/sbin/目录进行入侵检测

在 vm2 上安装 nginx,vsftpd,mariadb,mariadb-server,并启动所有对应的服务

在 vm1 上使用 nmap 扫描 vm2 主机的所有 TCP 服务

在 vm2 上配置 nginx 用户认证，并使用 tcpdump 抓取 80 端口相关的数据包，注意默认抓取的是第一个网卡的数据，抓取其他网卡可以使用-i 选项

在 vm1 上使用 firefox 访问 vm2 的页面，输入账户与密码，到 vm2 观察数据包

在 vm2 主机，使用 limit 模块对 nginx 限制并发，限制并发数量为 10，burst 为 10

在 vm2 主机，设置 nginx 拒绝所有非 POST 或 GET 的请求

在 vm2 主机，设置 nginx 防止 buffer 数据溢出

在 vm2 主机登陆 mariadb 服务器，创建一个可以从远程登陆的数据库账户

在 vm2 主机使用 tcpdump 对 3306 进行抓包，在 vm1 连接 vm2 的数据库，进行查询操作，回到 vm2 观察抓取的数据包信息

在 vm2 安装 tomcat，并以 tomcat 身份降级启动 tomcat 服务

在 vm2 主机执行如下命令：

```
mkdir -p /root/{source1.0,surce2.0}/test/
echo “hehe” > /root/source1.0/test.conf
echo “haha” > /root/source2.0/test.conf
echo “hello” > /root/source1.0/test/hello.sh
```

```
echo “hello world” > /root/source2.0/test/hello.sh
cp /bin/find    /root/source1.0/
cp /bin/find    /root/source2.0/
echo “xyz” >> /root/source2.0/find
```

在 vm2 主机对/root/source1.0 和/root/source2.0 生成补丁文件，并使用 patch 工具对 source1 目录下的所有代码打补丁