学习目标

1. 能说出selinux的主要作用
2. 能够通过selinux排错思路进行简单的selinux排错
3. 能够使用tripwire实现入侵后检测

SElinux

selinux介绍

> **SElinux是什么?**
>
> 答: SELinux全称是(Security-Enhanced Linux)，是美国国家安全局对强制访问控制系统（MAC,Mandatory Access Control)的一个实现，目的在于明确的指明某个**进程可以访问哪些资源(文件、网络端口等)**。强制访问控制系统的用途在于增强系统抵御 0-Day 攻击(利用尚未公开的漏洞实现的攻击行为)的能力，所以**它不是网络防火墙或 ACL 的替代品，在用途上也不重复**。
>
> **SElinux怎么实现安全性增强?**
>
> 答: SElinux把进程能访问的资源的**权限最小化**。比如现在有一个nginx搭建的web服务器，但出现了漏洞，使得远程用户可以访问系统上的敏感文件（如/etc/passwd,/etc/shadow这种)，现在还没有出现新的补丁来修复此漏洞。这时如果是开启了selinux的话，selinux是会拒绝nginx进程访问/etc/passwd文件的。
>
> **SElinux的缺点?**
>
> 答: 最主要的缺点就是对于新手来说太难用了,因为selinux的限制太多了。每一个进程访问不同的文件都有不同的权限，就好象一个大公司，每一个人（不同的职位，不同的部门，工龄的不同）对公司里的每一样财产（桌子，椅子，厕所，电脑，打印机等）都拥有不同的权限。如果你是一个刚入职的员工，不可能记住所有，所以可能一不小心就做出越权的事了。但只要我们掌握得当的方法，SElinux我们还是可以玩得动的。

`-rw-r--r--.` 最后一位点号的作用

```
为什么有些文件在九位权限位后有.符号，而有些文件却没有;如下
# ll -z /etc/yp.conf
-rw-r--r--. root root system_u:object_r:net_conf_t:s0
 /etc/yp.conf
# ll -z /etc/fstab
-rw-r--r-- root root ?
/etc/fstab
```

```
这就说明操作系统以前是打开selinux的（其实现在centos6,centos7安装完系统就默认打开了selinux）。
打开了selinux，那么就会为系统里的所有文件分配不同的安全上下文(context，就是类似system_u:object_r:net_conf_t:s0)。而九位权限位后面没有.符号的文件,说明现在操作系统关闭了selinux，并且对此文件进行过修改。
```

selinux的状态查看和启停

查看selinux是否开启

```
# getenforce    --查看selinux是在哪种级别
# sestatus      --可以查看selinux现在的状态
```

selinux在线切换模式

```
# setenforce 1    --临时马上生效，切换成enforcing
# setenforce 0    --切换成permissive
--disabled和这两种模式不能在线切换，需要改配置文件重启系统
```

使用配置文件开启或关闭selinux(**永久开启或关闭selinux,改完配置文件后要重启生效**)

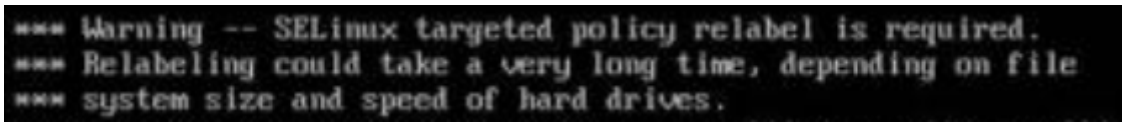
```
# vim /etc/selinux/config

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
```

```
#       enforcing - SELinux security policy is enforced.
--强制级别，违反策略就不允许
#       permissive - SELinux prints warnings instead of
enforcing.  --允许级别，违反发警告
#       disabled - SELinux is fully disabled.
--直接关闭selinux
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only
selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted

--selinux类型，也就是策略policy现在分：
    targeted   主要是对网络服务进行保护,类似阉割版（这是我们最主要
使用的selinux类型)
    minimum    更进一步的阉割版
    mls        对整个系统进行多级别保护,类似完全版
```

```
把SELINUX=disabled改成SELINUX=enforcing，然后reboot系统
如果你原来一直是disabled的情况用了比较久的时间，现在第一次开启
selinux，系统重新启动时，会把所有relable（也就是全部标记context)
这个relable会要等待一段时间（几分钟时间文件大小和硬盘速度决定,如下图
所示)
```



context简单解析

```
# ll -Z /etc/fstab
-rw-r--r--. root root system_u:object_r:etc_t:s0
/etc/fstab

system_u:object_r:etc_t:s0

身份（user)，角色(role)，类型(type)  s0(保护等级,也就是MLS用的,
我们这里是targeted类型,不用这种)

# yum install setools-console  -y

```

```
seinfo - SELinux policy query tool
-a   列出所有相关的属性
-u   列出selinux的所有user
-t   列出selinux的所有type
-r   列出selinux的所有role
-b   列出所有布尔值(getsebool -a可以列出这些布尔值当前的属性)
```

selinux对服务的限制

实例1

selinux对httpd进程访问文件的限制

以httpd服务为例（准备一台虚拟机，打开selinux)

第1步: 安装软件，并启服务

```
# yum install httpd httpd-devel -y
# systemctl start httpd
# systemctl enable httpd
```

第2步: 准备实验需要的文件

```
准备第一个文件
# echo "main page" > /var/www/html/index.html

准备第二个文件
# ll -Z /etc/passwd
-rw-r--r--. root root system_u:object_r:passwd_file_t:s0
/etc/passwd
# cp /etc/passwd /var/www/html/     --注意这里是cp

准备第三个文件
# touch /root/123
# ll -Z /root/123
-rw-r--r--. root root
unconfined_u:object_r:admin_home_t:s0 /root/123
# mv /root/123 /var/www/html/       --注意这里是mv

# ll /var/www/html/ -Zd
```

```
16  drwxr-xr-x. root root
    system_u:object_r:httpd_sys_content_t:s0 /var/www/html/
```

第3步: 客户端用浏览器访问测试，会发现123这个文件不能访问，另两个文件都可以访问

```
1  # ll /var/www/html/ -z
2  -rw-r--r--. root root
   unconfined_u:object_r:admin_home_t:s0 123
3  -rw-r--r--. root root
   unconfined_u:object_r:httpd_sys_content_t:s0 index.html
4  -rw-r--r--. root root
   unconfined_u:object_r:httpd_sys_content_t:s0 passwd
5  现在可以发现在apache默认家目录下创建的文件和cp过来的文件都会和目录的安全上下文一样；但mv的就保留原来的安全上下文（或者使用cp -a命令也可以保留）
6
7  用另一台客户端使用firefox来访问，主页和passwd文件可以被访问，但123文件不能被访问；其实就是http程序的http_t对123文件的admin_home_t没有访问权限
```

第4步: 解决

```
1  chcon    (change  context)
2      -t  修改类型域
3      -R  递归
4      -r  角色
5      -u  身份
6      --reference=示例文件    把安全上下文改成和示例文件一样
7
8  所以如果让客户端能够访问123文件，就用下面的命令来修改
9  # chcon -t httpd_sys_content_t /var/www/html/123    --改一个文件
10 # chcon -t httpd_sys_content_t  -R /var/www/html/   --递归改整个目录下的所有文件和子目录
```

实例2

selinux对httpd进程访问新家目录的限制

第1步: 修改httpd配置文件，换一个新的家目录

```
# mkdir /www
# vim /etc/httpd/conf/httpd.conf    --把家目录改成/www
119 DocumentRoot "/www"
131 <Directory "/www">
# systemctl restart httpd
```

第2步: 对比原家目录和现家目录的context

```
# ll /var/www/html/ -dZ
drwxr-xr-x  root root
system_u:object_r:httpd_sys_content_t /var/www/html/
# ll /www/ -dZ
drwxr-xr-x  root root root:object_r:default_t
 /www/
```

第3步: 新建主页文件，客户端访问不了

```
# echo "www main page" > /www/index.html    --在现家目录里建立一个主页
# ll -Z /www/index.html     --查看主页信息，是继承了上级目录的属性
-rw-r--r--  root root root:object_r:default_t
 /www/index.html

客户端访问，却访问不到里面的网页文件，因为网页文件权限属性没改
```

第4步: 解决

```
加-R参数把整个目录及其目录内文件都修改context
# chcon -R -t httpd_sys_content_t /www/
或者使用--reference指定按照某文件的标准来修改context
# chcon -R --reference=/var/www/html/ /www/
```

实例3

selinux对httpd进程监听端口的限制

```
httpd默认只监听80和443这两个端口,如果你改为监听12000，则启动为报错

# rpm -qf `which semanage`
policycoreutils-python-2.5-8.el7.x86_64

```

```
# yum install policycoreutils-python

# semanage port -l |grep http         ——查看http相关支持端口
http_cache_port_t              tcp      3128, 8080, 8118, 8123, 10001-10010
http_cache_port_t              udp      3130
http_port_t                    tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988
pegasus_https_port_t           tcp      5989

解决方法：
# semanage port -a -t http_port_t -p tcp 12000
# semanage port -l |grep http_port_t    --再次去查，就可以看到有12000这个端口了
http_port_t                    tcp      12000, 80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t            tcp      5988

再把httpd的端口改成12000启动，就OK了
```

由上面的三个例子,我们可以感受到selinux是如何增强服务安全性的（我们使用的是targeted类型,所以主要是保证服务安全)。那么当需求和selinux的规则产生冲突时,那么做法一般就是改规则,如果规则改不了则只能改需求。

selinux排错工具及排错思路

安装相关软件包，并确认审计日志服务开启

```
# yum install selinux\* setroubleshoot\* policy\*   -y
# systemctl status auditd.service       --确认审计日志服务开启
```

排错日志及排错工具

```
此日志里有selinux相关的信息，但是不容易看懂
# cat /var/log/audit/audit.log

以比较容易懂的方式查看selinux的日志，并给出解决的提示,全打开日志时间太久
# sealert -a /var/log/audit/audit.log

打开图形排错工具，简单方便(但需要图形支持)
# sealert
```

selinux排错思路总结

1. 如果服务访问不了,或者是服务里的文件访问不了等问题；可以先尝试`setenforce 0`改成permissive模式；如果改了还不能访问,那么肯定就不是selinux的问题了;如果改了就可以访问,那么可以确认是selinux的问题.
2. 如果是selinux的问题，那么就访问一下让他报错，然后通过查日志，或者图形工具来查看解决方法

```
# sealert -a /var/log/audit/audit.log
# sealert
```

3. 根本日志里提供的信息来进行排错
4. 如果仍然不能解决，请自行百度，谷歌
5. 解决后，做好工作笔记，以后就直接把selinux问题解决

实例4

使用selinux排错思路实现普通用户在samba家目录上传下载

selinux开启环境下,实现让普通用户登录samba,可以在自己的家目录上传下载

第1步: 在服务器端安装samba,并启动

```
server# yum install samba-* -y
server# systemctl start smb.service
server# useradd abc
server# smbpasswd -a abc              这一步才是设置abc登录samba的密码
New SMB password:
Retype new SMB password:
Added user abc.
```

第2步: 用普通用户abc登录samba服务器自己的家目录后，ls查看不到里面的内容，也下载不了

```
client# smbclient //10.1.1.11/abc  -U abc
Enter abc's password:
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.6.9-164.el6]
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
smb: \> get 123          --这里最好get 123下载一下，否则samba服务器那边使用sealert出不来提示信息
NT_STATUS_ACCESS_DENIED opening remote file \123
smb: \> put /etc/fstab fstab     --客户端的/etc/fstab上传到samba服务器取名仍然叫fstab，但也被selinux禁止
NT_STATUS_ACCESS_DENIED opening remote file \fstab
```

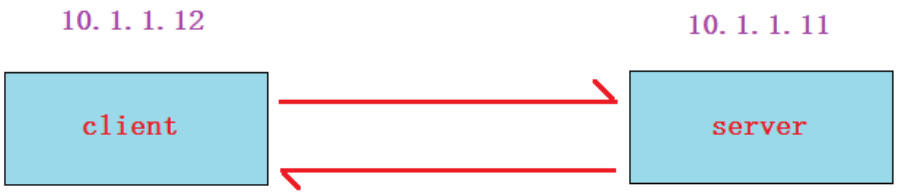
第3步: 通过sealert工具提示知道解决方法为

```
server# setsebool -P samba_enable_home_dirs 1
```

实例5

使用selinux排错思路实现普通用户在samba自定义目录上传下载

selinux开启环境下, 普通用户在samba自定义目录里拥有上传，下载权限



第1步: 在服务端修改samba配置，并重启服务

```
server# vim /etc/samba/smb.conf
[test]
        comment = test
        path = /test
        public = no
        writable = yes

server# systemctl restart smb.service

server# mkdir /test
server# chmod 757 /test
server# touch /test/123  --touch一个文件做下载测试
```

第2步: 客户端访问(可以登录成功，但是ls看不到内容,上传下载也不可以)

```
client# smbclient  //10.1.1.11/test -U abc
Enter abc's password:
Domain=[MYGROUP] OS=[Unix] Server=[Samba 3.6.9-164.el6]
smb: \> ls
NT_STATUS_ACCESS_DENIED listing \*
smb: \> get 123
NT_STATUS_ACCESS_DENIED opening remote file \123
smb: \> put /etc/inittab inittab
NT_STATUS_ACCESS_DENIED opening remote file \inittab
```

第3步: 通过sealert工具提示知道解决方法为

```
方法一:
server# semanage fcontext -a -t samba_share_t /test
server# restorecon  -v /test

方法二:
server# chcon -R -t samba_share_t /test

方法三:
server# setsebool -P samba_export_all_rw 1

--总结:如果一个服务不能访问；那么一般的检查步骤为:服务器端口-->网络->iptables->服务配置->系统权限->pam->selinux
--在生产环境中，在配置服务时，可以打开selinux，并使用上面的方法让你的服务可以被正常访问，这样就加强了安全（因为别人就算找到你服务的漏洞并破解，他仍然会受系统的selinux的限制）
```

小结: 如果一个服务不能访问；那么一般的检查步骤为:服务器端口 --> 网络 - ->iptables --> 服务配置 - -> 系统权限 --> pam --> selinux 在生产环境中，在配置服务时，可以打开selinux，并使用上面的方法让你的服务可以被正常访问，这样就加强了安全（因为别人就算找到你服务的漏洞并破解，他仍然会受系统的selinux的限制）

课后练习: 在selinux环境允许ftp匿名上传文件

（要求换一个新的ftp家目录/ftp，不用默认的/var/ftp）

```
在selinux环境允许ftp匿名上传文件（要求换一个新的ftp家目录/ftp，不用默认的/var/ftp）

方法一：
1，chmod 757 /ftp
2，要vsftpd的主配置文件里加上下面两个参数anon_upload_enable=YES和anon_mkdir_write_enable=YES
3，setsebool -P allow_ftpd_full_access 1

方法二：
1，chmod 757 /ftp
2，要vsftpd的主配置文件里加上下面两个参数anon_upload_enable=YES和anon_mkdir_write_enable=YES
3，chcon -t ftpd_tmp_t -R /ftp
```

SElinux总结

selinux的主要作用是什么?

答: 利用权限最小化原则，精细地控制服务进程访问系统资源（文件，端口等)的权限，达到防御0-day（破解)攻击的目的。

什么情况适合用selinux?

答: 对安全要求非常高,并且有能力能够解决selinux的限制问题的企业环境。

在selinux开启的环境下,如果客户端访问服务器或服务器的文件不能成功,排错的思路应该如何？请补充!

答: 服务端端口是否开启--》网络是否连通--》防火墙是否禁止--》系统权限--》服务配置--》PAM--》SElinux

入侵检测

入侵检测介绍

入侵检测系统（IDS,Intrusion Detection Systems)是对入侵行为的检测和发现。他通过对计算机网络或计算机系统中若干关键点收集信息并对其进行分析，从中发现网络或系统中是否有违反安全策略的行为和被攻击的迹象。

```
因特网
  |
路由
  |
防火墙
  |
入侵检测系统
  |
内部服务器群
```

入侵后检测 入侵后一般会替换或修改某些文件达到留后门的目的。开源软件的代表: tripwire

入侵中检测 实时检测数据包的行为。开源软件的代表: snort

入侵前检测 可以了解下如阿里云的态势感知。

tripwire

> 问题: 怀疑服务器被攻击后,怎么排查黑客的痕迹?

参考: http://cloud.baidu.com/forum/topic/show?topicId=262295

tripwire介绍

当服务器遭到黑客攻击时，在多数情况下，黑客可能对系统文件等等一些重要的文件进行修改。重要的文件那么多，管理员怎么知道是哪个或哪些被修改了?

如果是单个文件是否被修改，可以使用md5或rpm -Vf命令查看

```
rpm -vf
缺点1:只针对安装的rpm包里的文件做完整性检测,无法对自建的文件做检测
缺点2:只能检测单个文件，你需要写脚本循环去检测多个文件

# rpm -vf /etc/php.ini       --这个文件是rpm版的php的配置文件，第一次使用这个命令，没有任何显示，表示此文件没有做过任何的改变

```

```
# rpm -Vf /etc/php.ini      --对这个文件做微小的改变后，再执行此命令就会有信息了
S.5....T.  c /etc/php.ini

# man rpm
       S file Size differs
       M Mode differs (includes permissions and file type)
       5 digest (formerly MD5 sum) differs
       D Device major/minor number mismatch
       L readLink(2) path mismatch
       U User ownership differs
       G Group ownership differs
       T mTime differs
       P caPabilities differ
```

如果检查多个文件是否被改动了，tripwire这种软件就可以用得上了。这一软件采用的技术核心就是对每个要监控的文件产生一个数字签名，保留下来。当文件现在的数字签名与保留的数字签名不一致时，那么现在这个文件必定被改动过了。

实验

在linux服务器上安装tripwire实现入侵后检测

实验准备:

一台linux虚拟机就OK（实际环境中这台linux服务器最好是在业务连网上线前)

实验步骤:

1. 安装tripwire软件
2. 建立密钥文件（为了保护策略配置文件和tripwire数据库文件）
3. 设计检测策略并修改配置文件
4. 将修改好的配置文件写入数据文件（文本类型写入成data类型)
5. 初始化tripwire数据库文件（按照修改好的策略记录相关文件的数字签名)
6. 测试
7. 以后的策略更新方法

实验过程:

1, 安装tripwire软件

```
# yum install epel-release
# yum install tripwire

# ls /etc/tripwire/
twcfg.txt    --定义tripwire各个相关文件的路径
twpol.txt    --定义检测的对象文件，及违规时采取的行为

# cat /etc/tripwire/twcfg.txt    --查看此文件，有以下相关的文件定义

/var/lib/tripwire/$(hostname).twd    --用于存放生成的数据文件（也就是所有要监控的文件的数字签名的存档）
/var/lib/tripwire/report/            --存放检测的报告
```

2, 建立密钥文件

```
# tripwire-setup-keyfiles

Enter the site keyfile passphrase:
Verify the site keyfile passphrase: --输入你设定的site密码两次
Generating key (this may take several minutes)...Key generation complete.

Enter the local keyfile passphrase:
Verify the local keyfile passphrase:    --输入你设定的local密码两次
Generating key (this may take several minutes)...Key generation complete.

Signing configuration file...        --签名配置文件
Please enter your site passphrase:  --输入刚设定好的site密码
Wrote configuration file: /etc/tripwire/tw.cfg

Signing policy file...           --签名策略文件
Please enter your site passphrase:  --输入刚设定好的site密码
Wrote policy file: /etc/tripwire/tw.pol

# ls /etc/tripwire/     --做完后，就会多产生几个配置文件了
vm1.cluter.com-local.key  site.key  tw.cfg  twcfg.txt
 tw.pol  twpol.txt

```

site-key：用于保护Tripwire配置。除非我们再次生成配置，否则对tripwire配置所做的任何更改都不会生效，我们会提示我们输入site-key密码。（第4步会用到）

local-key：它用于验证tripwire二进制文件。当我们想要更新tripwire系统数据库时，我们需要运行tripwire命令，并且会提示我们输入local-key的密码。(第5步会用到)

3,设计检测策略并配置

```
# man twpolicy  --可以查看策略帮助

#  -  ignore the following properties
#  +  check the following properties
#
#  a  access timestamp (mutually exclusive with +CMSH)
#  b  number of blocks allocated
#  c  inode creation/modification timestamp
#  d  ID of device on which inode resides
#  g  group id of owner
#  i  inode number
#  l  growing files (logfiles for example)
#  m  modification timestamp
#  n  number of links
#  p  permission and file mode bits
#  r  ID of device pointed to by inode (valid only for device objects)
#  s  file size
#  t  file type
#  u  user id of owner
#
#  C  CRC-32 hash
#  H  HAVAL hash
#  M  MD5 hash
#  S  SHA hash

Device          = +pugsdr-intlbamcCMSH ;
Dynamic         = +pinugtd-srlbamcCMSH ;
Growing         = +pinugtdl-srbamcCMSH ;
IgnoreAll       = -pinugtsdrlbamcCMSH ;
IgnoreNone      = +pinugtsdrbamcCMSH-l ;
ReadOnly        = +pinugtsdbmCM-rlacSH ;
Temporary       = +pugt ;

```

```

/usr/sbin/siggen    +pinugtsdbmCM-rlacSH ;
```

```
--以twpol.txt下面这一段为例来说明它这个默认策略文件定义的意思
(
  rulename = "Tripwire Binaries",
  severity = $(SIG_HI)
)
{
  $(TWBIN)/siggen                       -> $(SEC_BIN) ;
  $(TWBIN)/tripwire                     -> $(SEC_BIN) ;
  $(TWBIN)/twadmin                      -> $(SEC_BIN) ;
  $(TWBIN)/twprint                      -> $(SEC_BIN) ;  --这四个文件都要被以SEC_BIN的检测等级来检测
}

--然后又找到有这句变量定义
SEC_BIN        = $(ReadOnly) ;

--又有下面这句的定义（rpm版里要man twpolicy去找),这些字母就代表了需要检测的属性
ReadOnly       = +pinugtsdbmCM-rlacSH ;

所以$(TWBIN)/siggen                       -> $(SEC_BIN) ;这一句配置的意思就是
/usr/sbin/siggen    +pinugtsdbmCM-rlacSH
```

```
现在/etc/tripwire/twpol.txt已经默认定义了非常多的文件,但现在这里有一个问题:它默认定义的文件或命令，你的系统上不一定有，所以你要注释掉它。它默认注释的文件或命令，你的系统上却有，所以你要打开注释.
比如:
下面这一句,它默认是定义的，但我系统上没有，所以需要前面加#号来注释
/sbin/busybox                         -> $(SEC_CRIT) ;
下面这一句，默认是注释的，但我系统上有，所以我想去掉前面的注释
#/sbin/lvchange                       -> $(SEC_CRIT) ;


为了实现这两个需求，我们需要使用脚本来处理默认的policy文件（TRANtwpol.sh是我提供给大家的一个shell脚本)
# cp TRANtwpol.sh /etc/tripwire/
# cd /etc/tripwire/
```

```
# cp twpol.txt twpol.txt.bak     --因为这个脚本是用sed直接操作修改这个文件，所以可以先备份一下
# sh TRANtwpol.sh twpol.txt      --处理完后的twpol.txt文件就是实现了上面两个需求的文件


下面再增加一段自定义配置供大家举一反三
(
        rulename = "notes",
        severity = $(SIG_HI)
)
{
        /share/                    -> $(SEC_INVARIANT) (recurse = 0);
        /share/notes/              -> $(SEC_CRIT);
        !/share/notes/program/ ;
}



--这个策略的意思是/share/目录本身（不递归到下级子目录），只检查$(SEC_INVARIANT);/share/notes/及其递归下级所有文件和子目录都检查$(SEC_CRIT);/share/notes/program/及其递归下级所有都不检查任何属性

```

4, 将修改好的配置文件写入数据文件

```
# twadmin -m P twpol.txt     --将修改完毕的文件，编码写入policy file
Please enter your site passphrase:
Wrote policy file: /etc/tripwire/tw.pol

# file /etc/tripwire/tw.pol --把你修改好的策略都写入了这个数据文件里
/etc/tripwire/tw.pol: data
```

5, 初始化tripwire数据库文件

```
# tripwire --init   --通过上一步的数据文件来初始化数据库文件（就相当于是对你所有配置的文件做了数字签名)
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
--这里等待时间较长，几分钟左右
Wrote database file: /var/lib/tripwire/vm1.cluter.com.twd
--数据库文件路径
The database was successfully generated.
```

6, 测试

```
怀疑被入侵后进行手动检测,或者用crontab周期性检测,然后通过查看报告就知道哪些文件被恶意修改过
# tripwire --check                          --对所有定义的文件进行一次检测，速度较慢
# tripwire --check --rule-name notes     --指定只检测的定义的规则名

Wrote report file: /var/lib/tripwire/report/vm1.cluster.com-20160930-101619.twr --检测完的报告，时间格式为系统的年月日-时分秒

通过下面命令查看历史报告，可以看到更详细的信息
# twprint --print-report --twrfile /var/lib/tripwire/report/vm1.cluster.com-20160930-101619.twr

一般为了安全性，在写入数据库之后，把明文的twpol.txt文件给删除或者备份到其它介质
# rm /etc/tripwire/twpol.txt -rf
```

7, 以后的策略更新方法

```
a),导出正在使用的策略
# twadmin -m p > /etc/tripwire/twpol.txt

b),按照需求对其进行修改

c),再导进去
# twadmin -m P /etc/tripwire/twpol.txt

d),重新生成数据文件
# tripwire --init
```

snort(拓展)

snort介绍

Snort是一款能够对网络上的数据包进行抓包分析（嗅探）,并能根据所定义的规则进行响应及处理的软件。

```
              外网
               |
              路由
               |
              防火墙
               |
              snort
               |
     web   mail  dns    ftp


snort软件包:
daq-2.0.6-1.centos7.x86_64.rpm          --依赖包
snort-2.9.9.0-1.centos7.x86_64.rpm      --主程序包
snortrules-snapshot-2990.tar.gz         --官方的免费入侵检测规则文件（也有收费的版本)
官方pdf格式安装配置文档:
snort-centos6x-7x-rev1.pdf
```

实验

在linux服务器上安装snort实现实时入侵检测

实验准备:

一台linux虚拟机就OK

实验步骤:

1. 安装snort软件
2. 解压规则文件(也就是官方的入侵检测规则文件，有免费的也有收费的），并创建黑，白名单文件
3. 配置snort主配置文件
4. 修改并确认嗅探的网卡名
5. 启动服务间进行自我检测，并解决错误
6. 启动snort服务
7. 测试

实验过程:

```
1,安装snort软件
# rpm -ivh daq-2.0.6-1.centos7.x86_64.rpm
# yum install snort-2.9.9.0-1.centos7.x86_64.rpm   --直接接rpm包的路径，会自动在你配置好的yum源里找依赖（这个软件有一个依赖包在epel源里)

2,解压规则文件(也就是官方的入侵检测规则文件，有免费的也有收费的），并创建黑，白名单文件
# tar xf snortrules-snapshot-2990.tar.gz -C /etc/snort/
# touch /etc/snort/rules/{white_list.rules,black_list.rules}

3,配置snort主配置文件
# vim /etc/snort/snort.conf
45 ipvar HOME_NET 10.1.1.0/24            --监控的内网网段(也就是你实验的网段)
48 ipvar EXTERNAL_NET !$HOME_NET         --外网网段为非内网

104 var RULE_PATH /etc/snort/rules       --这几句必须全部改成绝对路径，如果用默认的相对路径，其它参数在调用时，相对的目录就不一样了(这一句默认就是绝对路径，但下面几句是相对路径，需要修改)
105 var SO_RULE_PATH /etc/snort/so_rules
106 var PREPROC_RULE_PATH /etc/snort/preproc_rules
113 var WHITE_LIST_PATH /etc/snort/rules
114 var BLACK_LIST_PATH /etc/snort/rules

```

```
4,确认服务启动配置文件里的网卡名称(如果你的网卡名是eth0，那可以直接跳过这步；但如果不是eth0，则需要修改下面两个文件的两个地方)

# vim /etc/rc.d/init.d/snortd
47  INTERFACE="-i eth0"      --不是eth0的对应的改成你的网卡名称
# vim /etc/sysconfig/snort
15 INTERFACE=eth0        --不是eth0的对应的改成你的网卡名称


5,启动服务前，使用下面命令进行自我检测
# snort -T -i eth0 -u snort -g snort -c /etc/snort/snort.conf

报错:
ERROR: /etc/snort/snort.conf(253) Could not stat dynamic module path "/usr/local/lib/snort_dynamicrules": No such file or directory.

解决方法:
# mkdir -p /usr/local/lib/snort_dynamicrules
# chown -R snort:snort /usr/local/lib/snort_dynamicrules
# chmod -R 700 /usr/local/lib/snort_dynamicrules

# snort -T -i eth0 -u snort -g snort -c /etc/snort/snort.conf   --再让snort做自我检测，最后报如下两句，则表示成功

......
......
......
Snort successfully validated the configuration!
Snort exiting

6,启动snort服务
# systemctl start snortd

# ps -ef |grep snort |grep -v grep  --验证有下面的进程，表示启动ok
snort     7482     1  0 16:14 ?        00:00:00 /usr/sbin/snort -A fast -b -d -D -i eth0 -u snort -g snort -c /etc/snort/snort.conf -l /var/log/snort

```

```
7,测试
# vim /etc/snort/rules/local.rules  --这是/etc/snort/snort.conf里读取的N条规则中的其中一条，是个空文件，以它做测试，写上下面一条规则（表示任何icmp包都会警告）

alert icmp any any -> any any (msg:"ICMP Testing Rule"; sid:1000001;)

# systemctl restart snortd

然后使用任意IPping本机，会发现/var/log/snort/alert里有警告信息

# snort -A console  -i eth0 -u snort -g snort -c /etc/snort/snort.conf  --或者使用-A console指令启动snort让所有警告信息都直接显示到终端
```

堡垒机(拓展)

参考: http://www.jumpserver.org/

安全大总结

1. **硬件安全**:
   - 机房安全
   - 服务器bios密码，grub密码
   - luks硬盘或分区加密等
2. **密码安全**:
   - 密码复杂度: pam_cracklib.so,pam_pwquality.so
   - 密码定期更新等安全策略: /etc/shadow,/etc/login.defs
3. **权限安全**:
   - 把一些重要的系统文件降权，如把存放有明文密码的文件权限改为600
   - 把不能被修改的文件加上i属性，如chattr +i /etc/passwd
   - 文件acl，selinux等

4. **用户安全**:
    - 非登录用户不要使用/bin/bash，如useradd -r -s /sbin/nologin user1，usermod -s /sbin/nologin user2
    - 登出用户时自动清空命令历史记录，如echo "history -c && rm -rf .bash_history" >> $HOME/.bash_logout
    - 终端不操作一段时间后自动注销，如echo TMOUT=300 >> /etc/profile; source /etc/profile
    - 利用sudo对用户授权，将用户权限分割，如张三只能有网络相关命令操作权限，李四只能管理服务相关操作，王五只能对安装，卸载软件包有相关软件等。
5. **服务安全**:
    - 几乎每个服务都有实现其安全的方式或配置参数，比如ssh服务不使用密码登录，而使用密钥登录
    - 有些服务没有访问控制功能，可以使用防火墙或tcp_wrapper做服务的访问控制
    - 有些服务还能托管到xinetd这种服务下，使用xinetd的参数进行安全控制等
6. **验证安全**:
    - pam可植入式验证模块
    - ldap用户集中化验证等
7. **软件维护安全**:
    - 软件有漏洞或bug需要更新维护
    - 软件安装时检测签名等
8. **审计安全**:
    - 如果某一位工程师删除了某个重要文件，最后都不承认是自己做的，这样就不能实现责任到人，也就是说不能实现有效的监管。所以我们需要审计功能。审计相关的命令和日志非常晦涩难懂，所以有些公司倾向于使用更简单粗爆的方式，如远程连接服务器操作需要录屏(阿里云的堡垒机有此功能)等
9. **网络通讯安全**:
    - iptables，firewalld
    - ip tunnel，VPN
    - ssh安全(密钥连接，堡垒机或跳板机)
    - SSL等
10. **入侵检测**:
    - 入侵后检测(完整性检查)

- 入侵中检测(实时入侵检测)
- 入侵前检测(态势感知)

11. **防攻击**:
    - DDOS攻击（通过修改内核参数优化只能一定程度上缓解攻击,还可以做流量清洗;如果实在顶不住可以用防火墙暂时将可疑IP禁止访问等）
12. **代码安全**:
    - 这方面主要是涉及到开发的，如果代码太烂全是漏洞，运维的不能老做背锅侠吧😃
13. **漏洞扫描和安全评测**:
    - 简单免费的话可以使用类似nmap，lynis这种软件，要求高，并且公司并没有安全团队的话可以花钱请专业团队来评测
14. **操作安全**:
    - 养成良好的习惯，把握细节。如:ctrl+alt+delete会让服务器关闭，rhel6里要禁止ctrl+alt+delete关机 - 把/etc/init/control-alt-delete.conf里的start on control-alt-delete这句给注释掉;rhel7,centos7里注释或删除/usr/lib/systemd/system/ctrl-alt-del.target文件
    - 不要去别人的电脑连接服务器,用自己的电脑.
    - 有职业素养，删库跑路毕竟只能是一句玩笑话