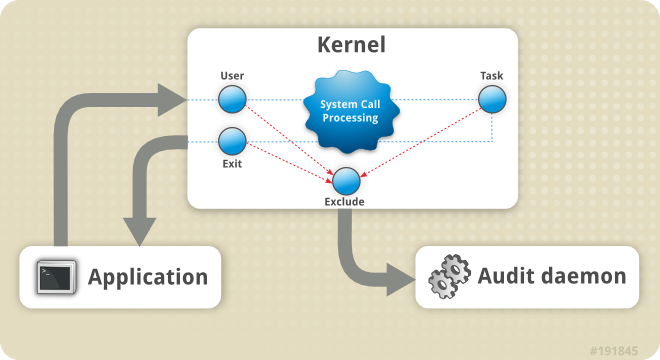
auditd（或 auditd 守护进程）是Linux审计系统中用户空间的一个组件，其负责将审计记录写入磁盘， 架构如下



User ：记录用户空间中产生的事件；它的作用是过滤消息的，内核传递给审计后台进程之前先查询它。

Task：跟踪应用程序的子进程（fork）；当一个任务被创建时，也就是父进程通过 fork 和克隆创建子进程时记录该事件；

Exit：当一个系统调用结束时判断是否记录该调用；

Exclude：删除不合格事件；Exclude 是用来过滤消息的，也就是不想看到的消息可以在这里写规则进行过滤

**相关工具及文件**

**auditctl** : 即时控制审计守护进程的行为的工具，比如如添加规则等等。

**aureport** : 查看和生成审计报告的工具。

**ausearch** : 查找审计事件的工具

**auditspd** : 转发事件通知给其他应用程序，而不是写入到审计日志文件中。

**autrace** : 一个用于跟踪进程的命令。

**/etc/audit/auditd.conf** : auditd工具的配置文件。

**/etc/audit/rules.d/audit.rules** 记录审计规则的文件

**安装**

为了使用审核系统，您必须在系统中安装 audit 软件包。audit packages（audit 及 audit-libs）是默认安装在红帽企业版 Linux 6 中。

作为 root 用户执行以下命令来安装

~]# yum install audit

**配置参数（auditd.conf）**

log\_file 保存审核日志文件的目录（/var/log/audit/） 建议单独分区

max\_log\_file 每个审核日志文件最少的占用空间 默认8 即8M

max\_log\_file\_action 采取何种行动，一旦到达在 max\_log\_file中所设定的极限，则应该设定为 keep\_logs 防止审核日志文件被重写 默认ROTATE（轮询）

space\_left 磁盘中可用空间的大小 此参数必须被设定为一个数字它会给予管理者足够的时间来回应和刷新磁盘空间。 space\_left 价值取决于审核日志文件生成的速度 以MB为单位 如1G 1024 默认75

space\_left\_action 当达到space\_left阈值时要采取的操作 如 emai 或者exec 默认SYSLOG

action\_mail\_acct 向该用户发送auditd电子邮件警告 默认root

admin\_space\_left 可用空间的绝对最小大小 硬限制 默认50

admin\_space\_left\_action 当达到最小限制触发 默认 SUSPEND（延迟）

disk\_full\_action 保存审核日志文件的分区没有可用空间时应触发的动作 默认SUSPEND 或者halt 或者 single

disk\_error\_action 保存在审核日志文件的分区检测到错误时，应该采取行动 默认SUSPEND 或者（syslog、single 或者 halt）

flush 刷新动作 默认INCREMENTAL\_ASYNC （增量式异步） 或者sync、 data

启动

systemctl start auditd

systemctl enable auditd

**定义审核规则**

审核系统根据一组规则运行，这组规则定义了日志文件中所获取的内容。有三种类型的审核规则可以详细说明

控制规则 — 允许审核系统的行为和它的一些被修改的配置。

文件系统规则 — 也被称为文件监视，允许审核进入特定文件或者目录。

系统调用规则 — 允许记录任何指定程序所做的系统调用

所有与审核服务交互的命令以及审核日志文件都需要 root 特权。作为 root 用户确保您执行这些命令

auditctl 命令允许您控制审核系统的基本功能并且限定规则来决定哪些审核项目要记录。

命令格式

auditctl [选项] filter,action -S syscall -F condition -k label

-b 在 Kernel 中设定最大数量的已存在的审核缓冲区，例如auditctl -b 8192 默认-b 320

-f 当追踪重要错误时设定所要完成的行动，例如：

auditctl -f 2 以上配置触发 kernel 恐慌以防重要错误。

0 在审计失败的情况下，什么也不要做。

1 在审计失败的情况下，将消息写入内核日志。

2 在审计失败的情况下，会引起内核恐慌

**大多数操作系统默认将故障标志设置为' ' 1 ' '最大化系统可用性，同时仍然会引起警报**

-e启动或者禁用审核系统或者锁定它的配置 例如：auditctl -e 2 以上命令锁定审核配置

0 这可以用来临时禁用审计

1 它将启用审计

2 锁定审核配置 （只有重新启动机器才能更改配置）

-r 设定每秒生成信息的速率，例如 auditctl -r 0 以上配置在生成信息方面不设定限制速率

-s 报告审核系统状态，例如 auditctl -s

enabled 1

failure 1

pid 4713

rate\_limit 0

backlog\_limit 320

lost 0

backlog 0

loginuid\_immutable 0 unlocked

-l 列出所有当前加载的审核规则，例如auditctl -l

No rules

-D 删除所有当前加载的审核规则

**定义文件系统规则**

定义文件系统规则，使用以下语法：

auditctl -w path\_to\_file -p permissions -k key\_name

path\_to\_file 是审核过的文件或者目录：

permissions 是被记录的权限：

r — 读取文件或者目录。

w — 写入文件或者目录。

x — 运行文件或者目录。

a — 改变在文件或者目录中的属性。

key\_name 是可选字符串，可帮助您判定哪个规则或者哪组规则生成特定的日志项。

**文件系统规则**

为了定义所有的输写访问权限以及在 /etc/passwd 文件中每个属性更改的规则，执行以下命令：

auditctl -w /etc/passwd -p wa -k passwd\_changes

请注意以下字符串 -k 选项是任意的。

为了定义记录所有输写访问权限，以及在 /etc/selinux/ 目录中所有文件属性更改的规则，执行以下命令：

auditctl -w /etc/selinux/ -p wa -k selinux\_changes

为了定义可以记录执行 /sbin/insmod 命令的规则，在 Linux Kernel 中插入模块，执行以下命令：

auditctl -w /sbin/insmod -p x -k module\_insertion

**定义系统调用规则**

监控系统调用可能会引起高负荷的日志活动，这会让内核承受更大的负荷。所以要慎重衡量哪些系统调用需要放到 audit.rules 中。如果审计的是目录的话，只能对该目录本身的属性进行审计。如果想审计下面的文件，需要一一列出

为了定义系统调用规则，使用以下语法：

auditctl -a action,filter -S system\_call -F 输入栏=value -k key\_name

-a 添加一条系统调用监控规则

-S 显示需要监测的系统调用的名称



**系统调用规则**

为了定义创造日志项 的规则，每次通过程序使用系统调用 adjtimex 或者 settimeofday。当系统使用 64 位架构，请执行以下命令：

~]# auditctl -a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time\_change

为了定义创造日志项的规则，每次由 ID 是 500 或更大的系统用户删除或者重命名文件时，使用（-F auid!=4294967295 选项排除没有设定登录 UID的用户），执行以下命令：

~]# auditctl -a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F auid!=4294967295 -k delete

使用系统调用语法来定义文件系统也是有可能的。对于 -w /etc/shadow -p wa 文件系统规则来说，以下命令为模拟的系统调用创造了规则：

~]# auditctl -a always,exit -F path=/etc/shadow -F perm=wa

/etc/audit/audit.rules 文件中定义持久的审核规则和控制

为了定义在重新启动时可以一直有效的审核规则，您必须把它们包含在 /etc/audit/rules.d/audit.rules 文件中

auditctl 指令可以被用来读取来自指定文件的规则，使用 -R 选项，例如：

~]# auditctl -R /usr/share/doc/audit-version/stig.rules

**定义控制规则**

文件可以只包括以下的控制规则，修改审核系统的行为： -b、-D、-e、-f、或者 -r

在 audit.rules中控制规则

# Delete all previous rules

-D

# Set buffer size

-b 8192

# Make the configuration immutable -- reboot is required to change audit rules

-e 2

# Panic when a failure occurs

-f 2

# Generate at most 100 audit messages per second

-r 100

在 audit.rules 中的文件系统和系统调用规则

-w /etc/passwd -p wa -k passwd\_changes

-w /etc/selinux/ -p wa -k selinux\_changes

-w /sbin/insmod -p x -k module\_insertion

-a always,exit -F arch=b64 -S adjtimex -S settimeofday -k time\_change

-a always,exit -S unlink -S unlinkat -S rename -S renameat -F auid>=500 -F auid!=4294967295 -k delete

**理解审核日志文件**

默认情况下，在 /var/log/audit/audit.log 文件中的审核系统储存日志项；如果启用日志旋转，就可以旋转储存在同一目录中的 audit.log 文件

以下的审核规则记录了每次读取或者修改 /etc/ssh/sshd\_config 文件的尝试：

-w /etc/ssh/sshd\_config -p warx -k sshd\_config

如果 auditd 守护程序在运行，就需在审核日志文件中运行以下命令创造新事件：

~]# cat /etc/ssh/sshd\_config

在 audit.log 文件中的事件如下所示：

type=SYSCALL msg=audit(1364481363.243:24287): arch=c000003e syscall=2 success=no exit=-13 a0=7fffd19c5592 a1=0 a2=7fffd19c4b50 a3=a items=1 ppid=2686 pid=3538 auid=500 uid=500 gid=500 euid=500 suid=500 fsuid=500 egid=500 sgid=500 fsgid=500 tty=pts0 ses=1 comm="cat" exe="/bin/cat" subj=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023 key="sshd\_config"

type=CWD msg=audit(1364481363.243:24287): cwd="/home/shadowman"

type=PATH msg=audit(1364481363.243:24287): item=0 name="/etc/ssh/sshd\_config" inode=409248 dev=fd:00 mode=0100600 ouid=0 ogid=0 rdev=00:00 obj=system\_u:object\_r:etc\_t:s0

以上事件由三个记录组成（每个以 type= 密码作为开始），共享相同的时间戳和编号。每个记录包含好几对 name=value ，由空格或者逗号分开。以下是关于以上事件的详细分析

type=SYSCALL

type 输入栏包含这类记录。在这个例子中， SYSCALL 数值详细说明连接到 Kernel 的 系统调用触发了这个记录。

为了列出所有可能的类型值和它们的解释，请参考〈[“审核记录类型”](https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/7/html/security_guide/sec-Audit_Record_Types)〉。

msg=audit(1364481363.243:24287):

msg 输入栏记录：

audit(time\_stamp:ID) 表格中记录的时间戳和特殊 ID。如果多种记录生成为相同审核事 件的一部分，那么它们可以共享相同的时间戳和 ID。

Kernel 或者用户空间应用提供不同的事件特定 name=value 组。

arch=c000003e

arch 输入栏包括关于系统CPU架构的信息。值 c000003e 是使用 16 进制表示法编码。 当使用 ausearch 命令搜寻审核记录时, 使用 -i 或者 --interpret 选项自动转化为 16 进制值可供人读取的对等语。 c000003e 值被解释为 x86\_64。

syscall=2

syscall 输入栏记录了传输给 Kernel 的输入栏类型。值 2 可以与在 /usr/include/asm/unistd\_64.h 文件中可供人读取的对等语相匹配。 在这种情况下，2 是 open 系统调用。 请注意 ausyscall 实用程序允许您把系统调用数字转换成可供人读取 的对等语。 使用 ausyscall --dump 命令来展示所有的系统调用和它们的号码。

**success=no**

success 输入栏记录了系统调用是否被成功地记录在特定事件中。在这种情况下，调用 不会成功。

exit=-13

exit 输入栏包含详细说明由系统调用所返回的退出代码的值。在不同的系统调用中，值 各不相同。您可以用以下命令把值解释为可供人读取的对等语：ausearch --interpret --exit -13（假设您的审核日志中包含的事件没有退出代码 -13）。

a0=7fffd19c5592, a1=0, a2=7fffd19c5592, a3=a

a0 到 a3 输入栏记录了前四个参数，在这个事件中使用 16 进制编码系统调用。这些 参数取决于使用的系统调用；它们可以通过 ausearch 实用程序来解释。

items=1

items 输入栏包含事件中路径记录的数量。

ppid=2686

items 输入栏记录了父进程 ID（PPID）。在这个情况下，2686 是 bash 进程的 PPID。

pid=3538

pid 输入栏记录了进程 ID（PID）。在这个情况下，3538 是 cat 进程的 PID。

**auid=500**

auid 输入栏记录了审核用户 ID，这个是 loginuid。这个ID是用户在登录时使用的并且 即使当用户身份改变时，也可以通过每个进程获取该ID。（例如，通过切换用户账户， 使用 su - john命令）。登录时候的账户uid

**uid=500**

执行进程的用户 ID。使用以下指令：ausearch -i --uid UID，用户 ID 就可以被解释为 用户名字。在这个情况下， 500 是 shadowman 的用户 ID。

**gid=500**

执行进程用户的 ID组。

euid=500

进程用户的有效用户 ID。

suid=500

进程的用户的设置用户 ID。

fsuid=500

进程用户的文件系统用户 ID。

egid=500

进程用户的有效群组 ID。

sgid=500

进程用户的设置群组 ID。

fsgid=500

进程的用户的文件系统群组 ID。

**tty=pts0**

进程的终端。

ses=1

进程会话的会话 ID

**comm="cat"**

comm 输入栏记录了命令行的名字，它被用于调用分析进程。在这种情况下 cat 命令 被用来触发审核事件。

**exe="/bin/cat"**

进程的可执行的路径。

subj=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

subj 输入栏记录了 SELinux 语境，运用此语境可以在执行时间中标注分析进程。

**key="sshd\_config"**

key 输入栏记录了所有管理者定义的字符串，它与在审核日志中生成该事件的规则有关。

第二记录

type=CWD

在第二记录中，type 输入栏数值是 CWD — 当前工作目录。这个记录的目的是为了记 录当前进程的位置以防在相关的 PATH 记录中捕捉到相对路 径。运用这个方法可以 重塑绝对路径。

msg=audit(1364481363.243:24287)

msg 输入栏持有与第一记录中的数值相同的时间戳和 ID 值。

cwd="/home/shadowman"

cwd 执行时进入目录的路径，在目录中触发系统调用。

第三记录

type=PATH

在第三记录中， type 输入栏值是 PATH。每个审核事件包含一个 PATH 对于每条路径 种类的记录作为一个参数，传输给系统调用。在审核事件中，只有一条路径 （/etc/ssh/sshd\_config）被用来作为参数。

msg=audit(1364481363.243:24287):

msg 输入栏持有与第一和第二记录中的值相同的时间戳和 ID 值。

item=0

item 输入栏表明在所有项目中，哪个项目在 SYSCALL 类型记录中，参考了当前记录。 这个是数字是以零为基准；值 0 意味着它是第一项。

name="/etc/ssh/sshd\_config"

name 输入栏记录了文件或者目录的所有路径，作为参数被传输给系统调用。在这种情 况下，它是 /etc/ssh/sshd\_config 文件。

inode=409248

inode 输入栏包含索引结点数字，与记录在事件中的文件和目录有关。以下命令体现了与 409248 索引结点数字相关的文件和目录：

~]# find / -inum 409248 -print

/etc/ssh/sshd\_config

dev=fd:00

dev 输入栏明确说明了设备的次要和主要 ID，它包含记录在事件中的文件和目录。在 这种情况下，值代表 /dev/fd/0 设备。

mode=0100600

mode 输入栏记录了文件和目录权限，用 16 进制表示法编码。在这种情况下，0100600 可以被解释为 -rw-------，意味着对于 /etc/ssh/sshd\_config 文件，只有 root 用户拥有 读取并且输入权限

ouid=0

ouid 输入栏记录了对象所有者的用户 ID。

ogid=0

ogid 输入栏记录对象拥有者的群组 ID。

rdev=00:00

rdev 输入栏包含记录的设备识别器只用于特殊文件。在这种情况下，正常文件是不用 来作为记录文件的。

obj=system\_u:object\_r:etc\_t:s0

obj 输入栏记录了 SELinux 语境，运用此语境可以在执行时间中标注分析进程。

以上分析过的审核事件是事件所包含的所有可能位置栏的一小部分。为了列出所有事件的位置栏及解释，请参考〈 [“审核事件字段”](https://access.redhat.com/documentation/zh-cn/red_hat_enterprise_linux/7/html/security_guide/app-Audit_Reference" \l "sec-Audit_Events_Fields)〉

**搜索审核日志文件**

ausearch 实用程序允许您为特定事件搜索审核日志文件。默认情况下，ausearch 寻找 /var/log/audit/audit.log 文件。您可以使用 ausearch options -if file\_name命令来详细说明不同的文件。在一个 ausearch 命令中提供多种选择等同于使用 AND 运算符。

因登录失败而搜索 /var/log/audit/audit.log 文件，请使用以下命令。

ausearch --message USER\_LOGIN --success no --interpret

搜索所有的账户，群组，角色变更，请使用以下命令：

~]# ausearch -m ADD\_USER -m DEL\_USER -m ADD\_GROUP -m USER\_CHAUTHTOK -m DEL\_GROUP -m CHGRP\_ID -m ROLE\_ASSIGN -m ROLE\_REMOVE -i

搜索所有的由特定用户所执行的记录操作，使用用户的登录 ID（auid），请使用以下命令：

~]# ausearch -ul 500 -i

搜寻从昨天至今所有的失败的系统调用，请使用以下命令：

~]# ausearch --start yesterday --end now -m SYSCALL -sv no -i

**创建审核报告**

aureport 实用程序允许您生成有关记录在审核日志文件中事件的总结和分栏式报告。默认情况下，查询在 /var/log/audit/ 目录中的所有 audit.log 文件来创建报告。您可以指定不同的文件来运行报告而不使用 aureport options -if file\_name 命令

生成有关过去三天内不包括示例日在内的记录的事件，请使用以下命令：

~]# aureport --start 04/08/2013 00:00:00 --end 04/11/2013 00:00:00

生成所有可执行文件事件的一份报告，请使用以下命令：

~]# aureport -x

生成以上可执行文件事件的总结，请使用以下命令：

~]# aureport -x --summary

生成所有用户失败事件的总结报告，请使用以下命令：

~]# aureport -u --failed --summary -i

生成每个系统用户登录失败的总结报告，请使用以下命令：

~]# aureport --login --summary -i

通过 ausearch 查询搜索 用户 500 所有的文件访问事件生成一份报告，请使用以下命令：

~]# ausearch --start today --loginuid 500 --raw | aureport -f --summary

生成一份报告有关所有被查询的审核文件以及所包含事件的事件范围，使用以下命令：

~]# aureport -t