Сегодня в программе

1. Управление доступом и привилегии суперпозльзователя.
2. Управление процессами.
3. Файловая система.

Содержание

Стандартная модель контроля доступа

Основные правила

1. Решения по управлению доступом зависят от того, какой пользователь пытается выполнить операцию или в некоторых случаях от членства этого пользователя в группе UNIX.
2. У объектов есть владельцы. Владельцы имеют широкий контроль над своими объектами.
3. Пользователь является владельцем создаваемых им объектов.
4. Специальная учетная запись пользователя под названием *root* может действовать как владелец любого объекта.
5. Только пользователь *root* может выполнять особо важные административные операции.

Контроль доступа к файловой системе

Группы пользователей: хранятся в файле /etc/group.
Команда определения владельца файла: ls -l.

Учетная запись суперпользователя root

root

Запись всемогущего административного пользователя UNIX.
Идентификатор root (UID): 0 (можно поменять имя пользователя или создать доп.пользователей с UID=0, но это плохая идея).
Примеры привилегированных операций:

- Создание файлов устройств.
- Установка системных часов.
- Повышение лимитов использования ресурсов и приоритетов процессов.
- Изменение имени хоста системы.
- Настройка сетевых интерфейсов.
- Открытие привилегированных сетевых портов (с номерами меньше 1024).
- Выключение системы.

Пример: программа login.

Установка флагов setuid и setgid

setuid и setgid: биты разрешения, изменяющие текущий идентификатор UID или GID результирующего процесса на UID или GID файла, содержащего образ программы (а не на UID и GID пользователя, который ввел команду).
passwd: задание пользовательского пароля (пользователи могут изменять только собственные пароли, но пользователь root может изменять любой пароль).
Могут быть проблемы!!! Минимизация проблем: сократить количество программ, запускаемых с этим флагом.

Управление учетной записью root

Вход в учетную запись root

В Ubuntu прямой вход под root запрещен по умолчанию. Причины:
1. Учетные записи root не содержат информацию о том, какие операции выполнялись с правами root.
2. Сценарий входа в систему не содержит записей о том, кто на самом деле выполняет данную работу.

Команда su: замена идентификатора пользователя

su имя_пользователя: зайти под другим пользователем.
su - : оболочка переходит в режим регистрации.

sudo: ограниченный вариант команды su

Программа sudo принимает в качестве аргумента командную строку, выполняемую с правами root (или правами любого другого ограниченного пользователя).
Файл /etc/sudoers: в нем перечислены люди, имеющий разрешение использовать программу sudo и команды, которые им можно запускать на каждом хосте. Если предлагаемая команда разрешена, программа sudo запрашивает собственный пароль пользователя и выполняет команду.

Управление учетной записью root

Пример конфигурации sudoers

```
# Define aliases for machines in CS & Physics departments
Host_Alias  CS = tigger, anchor, piper, moet, sigi
Host_Alias  PHYSICS = eprince, pprince, icarus

# Define collections of commands
Cmnd_Alias  DUMP = /sbin/dump, /sbin/restore
Cmnd_Alias  WATCHDOG = /usr/local/bin/watchdog
Cmnd_Alias  SHELLS = /bin/sh, /bin/dash, /bin/bash

# Permissions
mark, ed    PHYSICS = ALL
herb        CS = /usr/sbin/tcpdump : PHYSICS = (operator) DUMP
lynda       ALL = (ALL) ALL, !SHELLS
%wheel      ALL, !PHYSICS = NOPASSWD: WATCHDOG
```

Управление учетной записью root

Преимущества и недостатки sudo

Преимущества sudo:

- Благодаря ведению журнала команд значительно улучшается учет.
- Пользователи могут выполнять определенные задачи, не имея ограниченных прав root.
- Реальный пароль root может быть известен только ограниченному кругу лиц.
- Использование программы sudo быстрее, чем su или вход в систему с правами root.
- Поддерживается канонический список всех пользователей с правами root.
- Уменьшается вероятность того, что корневая оболочка останется без присмотра.
- Один файл может управлять доступом ко всей сети.

Управление учетной записью root

Преимущества и недостатки sudo

Недостатки sudo:

- Любое нарушение безопасности личной учетной записи sudoer может быть эквивалентно нарушению самой учетной записи root.
- Регистрацию команд в sudo можно обойти с помощью временного выхода в оболочку из разрешенной программы.

Содержание

Управление процессами

Идентификатор процесса PID

Ядро назначает каждому процессу уникальный идентификатор.
Идентификаторы PID присваиваются по порядку по мере созданию процессов.

Идентификатор родительского процесса PPID

Чтобы создать новый процесс: существующий процесс должен клонировать сам себя (клон может заменить выполняемую программу другой).
PPID (Parent Pricess ID): идентификатор родительского процесса.

Идентификатор пользователя UID и текущий идентификатор пользователя EUID

UID (User ID) — идентификатор пользователя, создавшего данный процесс.
EUID (Effective User ID) — текущий пользовательский идентификатор процесса, предназначенный для того, чтобы определить, к каким ресурсам и файлам у процесса есть право доступа в данный момент.
GID и EGID: аналогично.

Жизненный цикл процесса

Создание процесса

Системный вызов fork: клонирование самого себя (дочернему возращает 0, родительскому PID). После завершения системного вызова fork дочерний процесс обычно запускает новую программу с помощью одного из системных вызовов семейства exec.

Сигналы

Запросы на прерывание, реализуемые на уровне процессов. Когда поступает сигнал, возможен один из двух вариантов:
1. Если процесс назначил сигналу подпрограмму обработки, то после вызова ей предоставляется информация о контексте, в котором был сгенерирован сигнал.
2. В противном случае ядро выполняет от имени процесса действия, заданные по умолчанию.

Жизненный цикл процесса

Основные сигналы

№[6]	Имя	Описание	Реакция по умолчанию	Перехватывается?	Блокируется?	Дамп памяти?
1	HUP	Отбой	Завершение	Да	Да	Нет
2	INT	Прерывание	Завершение	Да	Да	Нет
3	QUIT	Выход	Завершение	Да	Да	Да
9	KILL	Уничтожение	Завершение	Нет	Нет	Нет
10	BUS	Ошибка на шине	Завершение	Да	Да	Да
11	SEGV	Ошибка сегментации	Завершение	Да	Да	Да
15	TERM	Запрос на завершение	Завершение	Да	Да	Нет
17	STOP	Остановка	Остановка	Нет	Нет	Нет
18	TSTP	Сигнал остановки, посылаемый с клавиатуры	Остановка	Да	Да	Нет
19	CONT	Продолжение после остановки	Игнорируется	Да	Нет	Нет
28	WINCH	Изменение окна	Игнорируется	Да	Да	Нет
30	USR1	Определяется пользователем	Завершение	Да	Да	Нет
31	USR2	Определяется пользователем	Завершение	Да	Да	Нет

Управление процессами

Команда kill:отправка сигналов

kill [-сигнал] pid
kill -9 pid: уничтожить процесс pid гарантированно.
killall: уничтожить процессы, заданные именем.
killall httpd: уничтожить все процессы веб-сервера.
pkill -u ben: поиск процессов, заданных именами (или другими атрибутами), и посылает найденным процессам сигнал.

Управление процессами

Команда ps:текущий контроль процессов

ps aux: получить список всех процессов, выполняющихся в системе.

ключ a: вывести все процессы, ключ x: увидеть даже процессы, отсоединенные от терминала, ключ u: фильтрование по имени или идентификатору пользователя, который запустил программу.

Поле	Содержимое
USER	Имя владельца процесса
PID	Идентификатор процесса
%CPU	Доля времени центрального процессора (в процентах), выделенная процессу
%MEM	Часть реальной памяти (в процентах), используемая процессом
VSZ	Виртуальный размер процесса
RSS	Размер резидентного набора (количество страниц памяти)
TTY	Идентификатор управляющего терминала
STAT	Текущий статус процесса: R — выполняется, D — ожидает записи на диск, S — неактивен (< 20 с), T — приостановлен, Z — зомби. Дополнительные флаги: W — процесс выгружен на диск, < — процесс имеет повышенный приоритет, N — процесс имеет пониженный приоритет, L — некоторые страницы блокированы в ядре, s — процесс является лидером сеанса.
TIME	Количество времени центрального процессора, затраченное на выполнение процесса
COMMAND	Имя и аргументы команды[a]

[a]Программы могут модифицировать эту информацию, так что она не всегда точно представляет реальную командную строку.

Управление процессами

Команды мониторинга и изменения приоритетов

top: интерактивный мониторинг процессов. Расширенная версия: htop.
nice и renice: изменение приоритета выполнения.
nice -n 5 /bin/task.sh
renice -5 8829
renice 5 -u boggs

Управление процессами

Файловая система /proc

Псевдофайловая система, в которую ядро помещает большой объем информации о состоянии системы.

Файл	Содержимое
cgroup	Группа управления, которой принадлежит процесс
cmd	Команда или программа, выполняемая процессом
cmdline[a]	Полная командная строка процесса (разделенная нулями)
cwd	Символическая ссылка на текущий каталог процесса
environ	Переменные среды процесса (разделенные нулями)
exe	Символическая ссылка на файл, который должен выполняться

Окончание табл. 4.4

Файл	Содержимое
fd	Подкаталог, содержащий ссылки на дескрипторы каждого открытого файла
fdinfo	Подкаталог, содержащий дополнительную информацию о дескрипторах каждого открытого файла
maps	Информация отображения памяти (сегменты совместного использования, библиотеки и т.п.)
ns	Подкаталог, содержащий ссылку на пространство имен каждого открытого файла
root	Символическая ссылка на корневой каталог процесса (определенный командой chroot)
stat	Информация об общем состоянии процесса (для ее получения лучше всего использовать команду ps)
statm	Информация об использовании памяти

Периодические процессы

Демон cron:команды расписания

Инструмент для запуска команд в соответствии с заданным расписанием.
Файл конфигурации cron: crontab, хранится в /var/spool/cron.

Содержание

Файловая система

Основное назначение: упорядочение хранимых ресурсов системы. Файловая система состоит из следующих основных компонентов:

- пространство имен — методы именования объектов и организации их в виде единой иерархии;
- API — набор системных вызовов для перемещения между объектами и управления ими;
- модель безопасности — схема защиты, сокрытия и совместного использования объектов;
- реализация — программный код, который связывает логические модели с дисковой подсистемой.

Файловая система

Монтирование и демонтирование файловой системы

Команда *mount*: присоединение файловой системы к файловому дереву в точке монтирования.
sudo mount /dev/sda4 /users — подключает к точке монтирования, заданной в виде пути /users, файловую систему, расположенную в дисковом разделе (Устройстве) /dev/sda4.
umount: отсоединить файловую систему. Ключи: -l: "ленивое"демонтирование, -f: принудительное демонтирование.
/etc/fstab: файл, содержащий сведения о файловых системах, монтируемых в системе.
fsck — команда, проверяющая исправность файловой системы;
lsof — утилита, выводящая информацию о том, какие файлы используются тем или иным процессом.
mkfs — утилита, создающая файловую систему на некотором устройстве.
fdisk — утилита для управления дисками (в т.ч. разметкой).
cfdisk — более удобная версия *fdisk*.

Структура файлового дерева

Каталог	Содержимое
/bin	Команды операционной системы ядра
/boot	Ядро и файлы для его загрузки
/compat	Файлы и каталоги в системе FreeBSD, обеспечивающие бинарную совместимость с системой Linux
/dev	Файлы устройств: дисков, принтеров, псевдотерминалов и т.д.
/etc	Важные файлы запуска и конфигурации системы
/home	Стандартные домашние каталоги пользователей
/lib	Библиотеки, совместно используемые библиотеки и команды, применяемые в каталогах /bin и /sbin
/media	Точки монтирования файловых систем на съемных носителях
/mnt	Временные точки монтирования
/opt	Программные пакеты необязательных приложений (которые пока не находят широкого применения)
/proc	Информация о всех выполняющихся процессах
/root	Домашний каталог суперпользователя (часто просто /)
/run	Точки встречи для выполняемых программ (идентификаторы PID, сокеты и т.д.).
/sbin	Команды, необходимые для обеспечения минимальной работоспособности системы[a]
/srv	Поля, зарезервированные для распределения через веб и другие серверы
/sys	Интерфейсы разных ядер (Linux)
/tmp	Временные файлы, которые могут удаляться при перезагрузке
/usr	Иерархия дополнительных файлов и программ
/usr/bin	Большинство команд и исполняемых файлов
/usr/include	Файлы заголовков, предназначенные для компиляции C-программ
/usr/lib	Библиотеки и вспомогательные файлы для стандартных программ
/usr/local	Локальные программы (программы, создаваемые или устанавливаемые локальным пользователем); отражает структуру каталога /usr
/usr/sbin	Менее важные команды системного администрирования
/usr/share	Элементы, общие для различных систем
/usr/share/man	Страницы интерактивной документации
/usr/src	Исходные коды нелокальных программных пакетов (не находит широкого применения)
/usr/tmp	Дополнительный каталог для временных файлов, которые могут сохраняться при перезагрузке

Структура файлового дерева

Каталог	Содержимое
/var	Системные данные и конфигурационные файлы
/var/adm	Разное: журнальные файлы, записи об инсталляции системы, административные компоненты
/var/log	Системные журнальные файлы
/var/run	Те же функции. что и в каталоге /run, а также символические ссылки
/var/spool	Буферные каталоги для принтеров, электронной почты и т.д.
/var/tmp	Каталог для временного хранения файлов (после перезагрузки файлы не исчезают)

Типы файлов

Определение типа существующего файла: утилита *file*.

Кодирование типов файлов в листинге команды ls

Тип файла	Символ	Создается командой	Удаляется командой
Обычный файл		Редакторы, `cp` и др.	`rm`
Каталог	`d`	`mkdir`	`rmdir, rm -r`
Файл символьного устройства	`c`	`mknod`	`rm`
Файл блочного устройства	`b`	`mknod`	`rm`
Локальный сокет	`s`	`socket(2)`	`rm`
Именованный канал	`p`	`mknod`	`rm`
Символическая ссылка	`l`	`ln -s`	`rm`

ln — создание жесткой ссылки.

Атрибуты файлов

chmod: команда изменения режима доступа.
chmod 740 file.sh — владелец может всё, группа владельца только читать, остальные ничего не могут.

Спецификация	Назначение
`u+w`	Владельцу файла дополнительно дается право записи
`ug=rw,o=r`	Владельцу и членам группы дается право чтения/записи, остальным пользователям — право чтения
`a-x`	У всех пользователей отбирается право выполнения
`ug=srx,o=`	Владельцу и членам группы дается право чтения/выполнения, устанавливаются также биты SUID и SGID; остальным пользователям запрещается доступ к файлу
`g=u`	Членам группы предоставляются такие же права, что и владельцу

Дополнительный бит: бит, которому в коде доступа соответствует восьмеричное значение 1000 (sticky bit). Раньше этот бит запрещал выгрузку программ из памяти. Если его установить сейчас для каталога, то файловая система позволит удалять и переименовать его файлы только владельцу каталога, владельцу файла или суперпользователю.

Атрибуты файлов

Смена владельцев и группы

chown и *chgrp* — смена владельца файла и группы.
sudo chown -R user /folder — рекурсивно сменить владельца на user у папки /folder.
sudo chgrp -R group /folder — рекурсивно сменить группу файла.
sudo chown -R user:group /folder — сменить сразу и владельца, и группу.

Команда umask: задание стандартных прав доступа

umask: изменение стандартного режима доступа к создаваемым файлам.

Восьмеричное число	Двоичное число	Режим доступа	Восьмеричное число	Двоичное число	Режим доступа
0	000	rwx	4	100	-wx
1	001	rw-	5	101	-w-
2	010	r-x	6	110	--x
3	011	r--	7	111	---

umask 027 — предоставляет все права владельцу файла, запрещает читать файл членам группы и не дает никаких прав другим пользователям.
Стандартное значение *umask*: 022.

Задания

- Написать однострочник, который из вывода команды ls -l получает для текущего пользователя (команда whoami) все файлы (не директории), число ссылок на которые больше одного.
- Написать скрипт, получающий список символических ссылок в каталоге.
- Написать однострочник, создающий для всех файлов, начинающихся с D* и расположенных в текущей директории, символические ссылки, располагающиеся в домашнем каталоге пользователя.
- Изучить мануал команды uname и получить из вывода версию ядра операционной системы.
- Написать скрипт, определяющий самую часто используемую команду из history.
- С помощью команды wc найти в текущей папке файл с наибольшим количеством символов.
- Напишите однострочную команду, которая у всех файлов, владельцем которых является root, сменит владельца на текущего пользователя.

Конец темы

Спасибо за внимание!