Администрирование Linux
Лекция 7
Пользователи

Иртегов Д.В.

Новосибирский гос. Университет

2014

Пользователи в Unix

- Каждый пользователь имеет Uid
  - User IDentificator
  - Небольшое целое число
  - В современных юниксах 32-битное
  - Процесс, работающий из-под root (uid=0) может сделать setgid/initgroups на произвольные gid и setuid(2) на произвольный Uid, в том числе и не зарегистрированные
- Зарегистрированные пользователи также имеют
  - Имя
  - Основную группу и до 16 других групп
  - Пароль и другие атрибуты аутентификации
  - Домашний каталог
  - .-файлы (настройки программ)
- Группа имеет Gid. Вложенных групп не бывает

Пользователь root

- Uid=0
- Может все
  - Стать любым пользователем
    - su(1) без пароля,
    - setuid(2) без ограничений
  - Работать с файлами, не обращая внимания на права
  - Менять права и хозяина любым файлам
  - Посылать сигналы любым процессам
  - Слушать на сокетах <1024
  - Загружать/выгружать модули ядра и менять настройки ядра (в том числе устанавливать rootkit)

Что работает из-под root

- Сервисы, принимающие аутентификацию
  - sshd, login, gdm, ftpd
- Сервисы, запускающие процессы от имени других пользователей
  - cron, at, sendmail/postfix (.forward)
- Setuid-утилиты для смены идентичности
  - su, sudo, suexec
- Администрирование системы
  - rpm/yum, useradd/usermod, fdisk, lvm, telinit, …
- Некоторые сервисы при старте
  - Apache должен слушать на порту 80
- Некоторые странные вещи
  - ping – принимать и отправлять ICMP может только root

БД учетных записей

- Управляется подсистемами PAM и NSS
  - (будем изучать далее)
- В простейшем случае, состоит из текстовых файлов
  - /etc/passwd,
  - /etc/shadow
  - /etc/group

Демонстрация

- Содержимое файлов /etc/passwd, /etc/shadow и /etc/group на тестовых виртуалках и на живой системе
- Обратите внимание, что CentOS/RHeL создают для каждого пользователя группу с тем же именем и gid=uid, и назначают ее основной
- Обратите также внимание, что для большинства сервисов заведены отдельные uid (принцип минимума привилегий)

Пароли

- Хэши паролей лежат в /etc/shadow
- Форматы хэша:
  - Традиционный Unix (reverse DES, ASCII-only, 8 символов)
  - Reverse Blowfish (некоторые дистрибутивы)
  - MD5 (по умолчанию в CentOS)
  - SHA-256, SHA-512
- Формат пароля и его размещение (shadow или passwd) задаются в /etc/pam.d/system-auth

Почему /etc/shadow

- Можно размещать пароли в /etc/passwd
- (настоятельно НЕ рекомендуется)
- /etc/passwd world readable
- Имея хэши паролей, можно подобрать пароли словарной атакой
  - ограничения на число попыток не действуют
  - DES и MD5 уязвимы для brute force
- Утечка хэшей практически эквивалентна утечке самих паролей
- Кроме того, /etc/shadow обеспечивает политики
  - срок жизни пароля,
  - принудительная смена
  - и т.д.

Создание пользователей

- Прямое редактирование /etc/passwd, /etc/shadow
  - НЕ рекомендуется
  - PAM/NSS кэшируют содержимое
  - могут не увидеть изменения или вовсе сдуреть
  - Имеет смысл из single user (сбросить пароль рута)
- useradd, usermod, userdel
  - useradd -c “comment” -m -d dir -s shell \
    -g gid -G gid,gid,gid -u uid login
- groupadd, groupmod, groupdel
- passwd
- id

Упражнение

- Создать пользователя
- Создать группу
- Добавить пользователя в группу
- Установить пользователю пароль
- Зайти этим пользователем

Упражнение
(do not try this at home)

- Добавить себя в группу wheel
- Перелогиниться
- Перед следующими пунктами, убедитесь, что помните пароль root!
- Раскомментировать в /etc/sudoers строку
  # %wheel ALL (ALL) = ALL
- Убрать (закомментировать) свое имя
- Убедиться, что вы по прежнему можете делать sudo

Политики управления паролем

- passwd -l – заблокировать
  - заблокируется только парольная аутентификация, вход по ключу ssh останется разрешен!
  - Полностью заблокировать – надо еще в /etc/ssh/sshd_config написать DenyUsers или DenyGroups и включить в эту группу
- -u – разблокировать
- -d – очистить пароль (sshd может не понравиться)
- -e – устаревание пароля (принудительная смена)
- --maxdays, --mindays, --warndays

Пользовательское окружение

- Домашний каталог
- .-файлы
- Переменные среды

Домашний каталог

- Предпоследняя колонка в /etc/passwd
- ~ - ваш домашний каталог
- ~user – домашний каталог user’a
- Если не доступен, графический десктоп зайти не может
- Содержит .-файлы, файлы и каталоги десктопа
- Может содержать личные файлы пользователя (обычно их там и держат)

.-файлы

- Файлы с именем, начинающимся с ‘.’, в Unix считаются скрытыми
  - ls по умолчанию их не показывает
  - ls -a показывает
- Хранят настройки программ
- У некоторых программ это один файл, напр. .vimrc
- У некоторых программ это целые каталоги, например .mozilla

.-файлы шелла

- Исполняются при каждом* запуске шелла
  - * не все и не при каждом, см. далее
- /etc/profile – глобальный файл для всех bourne shell (sh, ksh, bash)
- /etc/bashrc – глобальный файл для bash
- ~/.profile – личный файл для всех bourne shell
- ~/.bashrc – личный файл для bash
- ~/.bash_profile – login bash
- ~/.bash_history – история всех команд bash
- ~/.bash_logout – исполняется при завершении login bash

Среда (Environment)

- Набор переменных shell
- VAR=value
- export VAR – переменная становится доступна остальным программам
- export VAR=value (только в bash)
- VAR=value cmd – запустить команду с VAR равной value, но переменную не менять
  - TZ=America/Los-Angeles date
- $VAR – подставить значение
  пустая строка, если переменная не установлена
  - rm -rf $UNSET_VARIABLE/

Среда (продолжение)

- Некоторые переменные влияют на поведение многих программ
  - PATH (исполняемые файлы)
  - LD_LIBRARY_PATH (поиск .so файлов, редко используется в мирных целях)
  - TZ (часовой пояс, если не установлена, то /etc/localtime)
  - HOME (программы ищут там свои .-файлы)
  - USER (ssh host = ssh $USER@host)
  - DISPLAY (все программы X11)
  - LANG, LC_* (локализация, кодировка)
- Некоторые переменные важны только некоторым программам
  - INCLUDE – c-компилятор
  - PS[1-4] – форматы приглашения bash
  - EDITOR – visudo запускает $EDITOR
  - PAGER – man листает страницы при помощи $PAGER
  - TERM – тип терминала (bash, vi, mc, *tui)

Среда (продолжение)

- Переменные среды наследуются от родителя
- Переменные, установленные в /etc/profile и .-файлах, *обычно* действуют на всю сессию
- Любую переменную можно поменять, но это подействует только на этот процесс и его потомков
- «Переменные среды хранятся в /etc/profile» - неправда
- Они там не хранятся, они там вычисляются

Откуда берутся переменные среды - загрузка

- В RHeL/CentOS 6 нет способа установки переменных среды для init
  - /etc/environment считается устаревшим
- Все сервисы исполняются шелловскими скриптами из /etc/init.d,
- Шелл исполняет /etc/profile,
- Поэтому установленные в /etc/profile переменные действуют на все сервисы

Переменные среды - логин

- Перед графическим логином, GDM устанавливает DISPLAY и LANG/LC_*
- При логине по ssh,
  - перечисленные в /etc/ssh/ssh_config клиента и
  - разрешенные в /etc/ssh/sshd_config сервера.
  - Обычно это TERM, LANG/LC_*, TZ
  - в RHeL TZ почему-то запретили
- В процессе самого логина устанавливаются
  - USER, LOGNAME, HOME, SHELL
- Потом исполняются /etc/profile и .-файлы
  - При графическом логине это тоже происходит!

Демонстрация

- Изменение языка в GDM перед логином
- Показ /etc/profile, /etc/profile.d, /etc/bashrc
- Установка переменных среды в терминальной сессии xterm

Продвинутый синтаксис для работы с переменными

- ${VAR:-word} – если VAR не определена, word (VAR не менять)
- ${VAR:=word} – если VAR не определена, word (VAR=word)
- ${VAR:offset:len} – подстрока
- ${VAR/pattern/string} – замена подстроки
- $((expression)) – арифметическое выражение

Немного про .bash_history

- Сохраняет историю всех команд bash
- history, !123, стрелка вверх
  - history -c – очистить историю
  - history -w, -a – записать историю текущей сессии в .bash_history
    полезно если у вас открыто несколько сессий
- Подстановки
  - !123 – строчка №123
  - !-20 – строчка на 20 строк истории назад
  - !! – то же, что !-1
  - !string – последняя строчка, начинающаяся со string
  - !^ - первый аргумент (второе слово)
  - !$ - последний аргумент
  - Много всего, man bash, /history expansion
  - History -p [много подстановок] – показать результат подстановки, но ничего не исполнять
    полезно перед тем, как попытаетесь сделать что-то сложное, да еще из-под рута

Упражнение

- Выведите все команды в вашей истории, начинающиеся с sudo
  - Не вздумайте их исполнять!
  - Используйте history -p