Skillbox

# Наследование прав

**Медведик Давид**

Ведущий инженер-программист в «Positive Technologies»

# Цель урока

Познакомиться с командами наследования прав в Linux.

# В этом уроке мы рассмотрим

- Списки контроля доступа (ACL)
- Команду getfacl
- Команду setfacl
- Команду umask

# Зачем нам нужно это знать?

- Умение настраивать наследование прав
- Уметь работать с расширенными правами доступа

# Списки контроля доступа (ACL)

Списки контроля доступам (ACL) — это расширенные права, которые используются для реализации сложных структур прав доступа.

- Каждый пользователь входит в минимум одну группу
- Группа пользователей может содержать некоторое количество пользователей, но не может содержать или включаться в другие группы
- Группа может быть пустой

# Списки контроля доступа (ACL)

Существуют два типа ACL:

- ACL для доступа
- ACL по умолчанию

# Команда getfacl

**Getfacl** — выводит листинг ACL-прав для указанных объектов.

Примеры использования:

- getfacl * — отобразит права ACL для всех объектов в текущем каталоге
- getfacl test.txt — отобразит ACL для файла test.txt
- getfacl test* — отобразит ACL для всех файлов в текущем каталоге, которые начинаются на test
- getfacl -R * — отобразит ACL для всех объектов текущего каталога

# Команда setfacl

Списки ACL можно задать:

- на уровне пользователей — назначаются ACL конкретным пользователям
- на уровне групп — назначаются ACL конкретным группам
- с помощью маски эффективных прав — ограничение максимальных прав для пользователей и/или групп
- для пользователей, не включённых в группу данного файла

# Команда setfacl

Setfacl **[опции]** {ключ} <список правил> объект.

- [опции] — задаёт дополнительные опции
- {ключ} — задаёт режим работы утилиты
- <список правил> — собственно, сами правила доступа к объекту
- объект — объект, к которому применяется ACL

# Команда setfacl

Часто используемые ключи:

| Ключ | Описание |
|---|---|
| `--set` или `--set file` | Устанавливает новые указанные права ACL, удаляя все существующие |
| `-m` или `-M file` | Модифицирует указанные ACL на объекте. Другие существующие ACL сохраняются |
| `-x` или `-X file` | Удаляет указанные ACL-права с объекта. Стандартные права Unix не изменяются |

# Команда setfacl

Часто используемые опции:

| Опция | Описание |
|---|---|
| **-b** | Удаляет все ACL-права с объекта, сохраняя основные права |
| **-k** | Удаляет с объекта ACL по умолчанию |
| **-d** | Устанавливает ACL по умолчанию на объект |
| **–restore=file** | Восстанавливает ACL-права на объекты из ранее созданного файла с правами |
| **-R** | Рекурсивное назначение прав |

# Команда setfacl

Формирование списка правил:

| Синтаксис | Описание | Пример использования |
|---|---|---|
| `u:<uid>:<perms>` | Назначает ACL для доступа заданному пользователю | Пример:<br>`setfacl -m u:skillbox:rw myfile.odt` |
| `g:<gid>:<perms>` | Назначает ACL для доступа заданной группе | Пример:<br>`setfacl -m g:children:r myfile.odt` |
| `m:<perms>` | Назначает маску эффективных прав | Пример:<br>`setfacl -m m:rx myfile.odt` |
| `o:<perms>` | Назначает ACL для доступа пользователям, не включённым в группу файла | Пример:<br>`setfacl -m o: myfile.odt` |

# Команда umask

**Umask** — это пользовательская маска, которая используется для определения конечных прав доступа.

Umask PERMISSIONS.

# Команда umask

Разрешения на создание файлов по умолчанию: 666 и, для каталогов, 777.

Например, чтобы рассчитать, как uname 022 повлияет на вновь созданные файлы и каталоги, используйте:

- файлы: 666 - 022 = 644
- каталоги: 777 - 022 = 755

Отображение значения маски в символической записи, используя опцию -S:

umask -S u = rwx, g = rx, o = rx

# Выводы

Итак, сегодня мы познакомились с очень важными командами тонкой настройки прав, такими как getfacl, setfacl, umask, с помощью которых можем настроить безопасность системы более грамотно. А также изучили их формат и способы применения.

# Спасибо
# за внимание!