alexal3.com /

Linux Capabilities простыми словами

⋮ 02.09.2021

Capabilities (привилегии, возможности, полномочия) — это атрибуты ядра, которые дают некоторые привилегии рута (root) процессам или исполняемым файлам. Например, право изменить UID процесса на 0 (UID root'а) или право монтировать/размонтировать файловые системы.

ВНИМАНИЕ! Использование Capabilities чревато риском взлома операционной системы. Пользуйтесь ими на свой страх и риск и тщательно протестируйте в тестовой среде.

Если запустить файл, который имеет полномочия, то созданный процесс их унаследует.

Список Capabilities (неполный)

- `CAP_AUDIT_CONTROL` — включение/выключение аудита ядра;
- `CAP_AUDIT_WRITE` — создание логов аудита ядра;
- `CAP_BLOCK_SUSPEND` — блокировка приостановки работы системы;
- `CAP_CHOWN` — произвольное изменение UID и GID файлов;
- `CAP_DAC_OVERRIDE` — обход проверки разрешений чтения, записи и выполнения файлов;
- `CAP_DAC_READ_SEARCH` — обход проверки разрешений чтения и выполнения файлов и директорий;
- `CAP_FOWNER` — обход проверок разрешений для процессов, чей UID должен соответствовать UID файла;
- `CAP_FSETID` — не удалять SUID и SGID изменяемого файла;
- `CAP_IPC_LOCK` — блокировка оперативной памяти;
- `CAP_IPC_OWNER` — обход проверок разрешений для операций с объектами System V IPC;
- `CAP_KILL` — отправка сигналов чужим процессам;
- `CAP_MAC_ADMIN` — изменение конфигурации MAC;
- `CAP_MAC_OVERRIDE` — изменение MAC (Mandatory Access Control);
- `CAP_MKNOD` — создание специальных файлов, используя системный вызов mknod();
- `CAP_NET_ADMIN` — выполнение различных сетевых операций;
- `CAP_NET_BIND_SERVICE` — связывание сокетов с портами ниже 1024;
- `CAP_NET_RAW` — использование сокетов RAW и PACKET;
- `CAP_PERFMON` — использование систем мониторинга;
- `CAP_SETFCAP` — установка произвольных возможности для файла;
- `CAP_SETGID` — изменение GID;
- `CAP_SETPCAP` — передача или удаление любого вашего Capability процессу с любым PID;
- `CAP_SETUID` — изменение UID;

- CAP_SYS_ADMIN — монтирование и размонтирование файловых систем;
- CAP_SYS_BOOT — перезагрузка, Ctrl-Alt-Del, загрузка нового ядра;
- CAP_SYS_CHROOT — вызов chroot();
- CAP_SYS_MODULE — установка модулей ядра;
- CAP_SYS_NICE — увеличение приоритета процессов;
- CAP_SYS_PACCT — включение/выключение учета процессов;
- CAP_SYS_PTRACE — отладка (debug) процессов;
- CAP_SYS_RAWIO — выполнение операций ввода/вывода порта;
- CAP_SYS_RESOURCE — изменение лимитов системных ресурсов, например, дисковых квот;
- CAP_SYSLOG — выполнение привилегированных операций системного вызова syslog();
- CAP_SYS_TIME — установка времени системных часов и часов реального времени;
- CAP_SYS_TTY_CONFIG — управление терминалом.

Виды Capabilities

Полномочия могут быть нескольких видов — «permitted» (доступные), «inheritable» (наследуемые), «effective» (текущие, эффективные) и «ambient» (наружные). Главное запомнить, что как правило, на файлы устанавливаются виды «permitted» и «effective» (см. Управление Capabilities ниже).

Управление Capabilities

Для управления привилегиями нужно установить пакет libcap2-bin в Debian и Ubuntu или libcap в CentOS, Fedora и Arch.

Чтение

У файла:

```
root@ubuntu:~# getcap /путь/к/файлу
/путь/к/файлу = cap_setuid+ep
```

Здесь ep означает «permitted» и «effective«.

```
root@ubuntu:~# getcap /путь/к/файлу
```

Здесь отсутствие вывода означает, что файл не содержит привилегий.

Поиск файлов с Capabilities по всей системе:

```
root@ubuntu:~# getcap -r / 2>/dev/null
```

Здесь:

- -r — рекурсивный поиск;

- `/` — корневой каталог;
- `2>/dev/null` — не выводить ошибки; используется, т.к. виртуальные файловые системы (например, /proc) не поддерживают эти операции.

Присвоение

```
setcap "CAP_SETGID+ep" /путь/к/файлу
```

Удаление

```
setcap -r /путь/к/файлу
```

Заключение

В заключение хочу попросить тех, кто хорошо разбирается в Capabilities, дать в комментариях описание остальных привилегий простыми словами. Будем дополнять список.