Как кикнуть пользователя или его процессы с сервера GNU\Linux?

Category: GNU/Linux,kill,ps,who,Безопасность
2026-04-08

«**Кикнуть**» — сленговое слово, которое означает «*удалить, выгнать или исключить кого-либо*». Происходит от английского **to kick**, что переводится как «*пнуть*».

Изначально термин использовался в контексте общения в социальных сетях и мессенджерах, а также в компьютерных играх, когда игрока, нарушающего правила, исключали из команды или из игры в целом.

Введение.

Как администратор систем, вы обязаны знать все, что происходит с этими серверами, и что нужно делать, чтобы обеспечить их надежность и безопасность.

Но как узнать, кто вошел в систему на этих серверах **GNU/Linux** и что они делают?

Для этого пригодится системная утилита **who**.

Подробнее о ней у меня написано здесь: *«Утилита who. Информации о пользователях, которые подключены к системе.»*

Кто на сервере?

Посмотрим какие пользователи подключены к серверу:

```
$ sudo who
```

Ответ:

```
[hamster@fedora ~]$ who
hamster  pts/2        2026-04-08 12:10 (100.67.99.11)
hamster  seat0        2026-04-08 11:09
hamster  tty2         2026-04-08 11:09
[hamster@fedora ~]$
```

Как видно удалённо на сервер зашел пользователь hamster с **ip-адреса** 100.67.99.11 под **псевдотерминалом** pts/2 в системе.

Что значит псевдотерминал pts/2?

Буквы pts/2 на хосте — это обозначение **псевдотерминала** под номером 2 в системе **GNU\Linux**.

Псевдотерминал (**PTY**, **псевдо-TTY**) — это виртуальное устройство, которое имитирует аппаратный терминал, но не подключено напрямую к физическому устройству. Оно используется для взаимодействия приложений с пользовательским интерфейсом, например, в эмуляторах терминала, при удалённом доступе через **SSH** или для мультиплексирования терминалов.

Каждый псевдотерминал в системе получает уникальный числовой идентификатор. В каталоге /dev/pts создаётся файл с именем в виде /dev/pts/X, где X — номер псевдотерминала. Например, pts/2 — это файл псевдотерминала под номером 2.

Как это работает:

- Когда процесс открывает файл псевдотерминального устройства для чтения или записи, данные, которые процесс записывает в псевдотерминал, появляются на терминале, связанном с этим псевдотерминалом.
- Когда процесс открывает псевдотерминал для чтения, он получает входные данные от терминала.

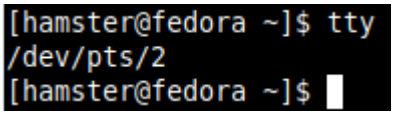
Некоторые примеры использования псевдотерминалов:

- эмуляторы терминала (например, **xterm**);
- программы для удалённого доступа (**sshd**);
- мультиплексоры терминалов (screen, **tmux**).

С помощью команды **tty** можно определить номер терминального устройства, связанного с текущей оболочкой.

$ tty

Ответ:



Да, это наше подключение. Предлагаю, для примера, кикнуть с сервера самих себя, то есть пользователя hamster.

Список процессов пользователей.

Посмотрим список всех процессов, которые инициированы пользователем hamster:

```
$ sudo ps -ef | grep hamster
```

Выходные данные вышеупомянутой команды будут перечислять все PID всех процессов, связанных с пользователем hamster.

Сделаем выборку процессов связанных с **ssh**:

```
$ sudo ps -ef | grep hamster | grep ssh
```

Ответ:

```
[hamster@fedora ~]$ sudo ps -ef | grep hamster | grep ssh
root        8153    1259  0 12:10 ?        00:00:00 sshd-session: hamster [priv]
hamster     8157    8153  0 12:10 ?        00:00:00 sshd-session: hamster@pts/2
hamster     9655    8158  0 13:01 pts/2    00:00:00 grep --color=auto ssh
[hamster@fedora ~]$
```

Как вы можете видеть нужный нам ssh **PID** – это 8157. Рядом с ним пояснение, что это sshd-session: hamster@pts/2.

Уничтожение процессов пользователя.

Уничтожить процесс пользователя с сессией соединения.

Выполним следующую команду, чтобы завершить сеанс sshd-session: hamster@pts/2:

```
$ sudo kill <PID>
```

```
$ sudo kill 8157
```

Если процесс не завершается, используйте принудительное завершение:

```
$ sudo kill -9 <PID>
```

Пользователя hamster единоразово кикнет с сервера, то есть его сессия и терминал закроются и единоразово потеряется доступ на сервер по этому соединению. Позднее или еще раз сразу, это не помешает ему переподключиться к серверу снова.

Учитывая, что в выводе утилиты **who** также был виден **ip-адрес**, с которого вошел пользователь **hamster**, то можно, чтобы заблокировать этот адрес от доступа к серверу (при необходимости).

```
hamster  pts/2        2026-04-08 12:10 (100.67.99.11)
```

Часто данный способ можно использовать еще для уничтожения зависших процессов пользователя, только без гарантии сохранности данных, которые они обрабатывают.

Уничтожить все процессы пользователя.

Например, можно принудительно уничтожить все процессы пользователя zabbix:

```
$ sudo ps -ef | grep zabbix
```

Ответ:

```
root@pve:~# sudo ps -ef | grep zabbix
zabbix    4316       1  0 Mar17 ?        00:59:39 /usr/sbin/zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf
root     1496455 1496385  0 13:15 pts/0    00:00:00 grep zabbix
root@pve:~#
```

```
$ sudo killall -9 -u zabbix
```

Ответ:

```
root@pve:~# sudo killall -9 -u zabbix
root@pve:~# sudo ps -ef | grep zabbix
root     1496884 1496385  0 13:17 pts/0    00:00:00 grep zabbix
root@pve:~#
```

Чтобы запустить заново, например, **Zabbix Agent 2**

```
$ sudo systemctl daemon-reload
$ sudo systemctl restart zabbix-agent2
```

```
$ sudo ps -ef | grep zabbix
```

Ответ:

```
root@pve:~# sudo systemctl daemon-reload
root@pve:~# sudo systemctl restart zabbix-agent2
root@pve:~# sudo ps -ef | grep zabbix
zabbix   1497212       1  0 13:18 ?        00:00:00 /usr/sbin/zabbix_agent2 -c /etc/zabbix/zabbix_agent2.conf
root     1497282 1496385  0 13:18 pts/0    00:00:00 grep zabbix
root@pve:~#
```

Оригиналы источников информации.

1. dzen.ru — «Выгнать за нарушение: что означает «кикнуть» и когда уместно это слово.»
2. itsecforu.ru — «Как узнать, кто заходил на ваш сервер Linux.»
3. ru.console-linux.com — «Как завершить зависшие или нежелательные пользовательские сеансы в Linux.»
4. baeldung.com — «What Are the /dev/pts Files Used for?»