

Home

PUBLIC

Stack Overflow

Tags

Users

Jobs

TEAMS

+ Create Team

Ask Question

## Allowing a non-root user to drop cache

I am carrying out performance tests on a system where I need to ensure I am reading data from the disk, and that it is not just cached (say from earlier tests). I read [here](#) that I can drop cache with the command

5

```
echo 3 | sudo tee /proc/sys/vm/drop_caches
```

★

1

However, note that even though my account is an admin account (login peter), it still requires my password. I want to be able to run this in a batch script without the requirement to input a password (as this is obviously manual)

More [research](#) led me to the sudoers file. My plan was to place the above command into a one line script called dropCache, and edit sudoers so that I could run it without entering a password. So I added the line

```
ALL ALL=(ALL)NOPASSWD:/home/peter/dropCache
```

at the end of my sudoers file (using visudo). With my admin account, if I run

```
sudo -l
```

I get

```
(ALL) NOPASSWD: /home/peter/dropCache
```

However, if I run my dropCache script I still get asked for my password

```
./dropCache
[sudo] password for peter:
```

Any help with this would be much appreciated. I am running Ubuntu 12.04

Thanks Peter

[linux](#) [sudoers](#) [clear-cache](#)

asked Nov 30 '12 at 14:15

 [Peter Cogan](#)  
640 ● 1 ● 10 ● 19

### 1 Answer

What I did when I needed this was I wrote a small C program, changed the owner of the compiled file to root, and set the setuid bit.

7

Here is the source code:

✓

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/types.h>

extern void sync(void);

int main(void) {
    if (geteuid() != 0) {
        fprintf(stderr, "flush-cache: Not root\n");
        exit(EXIT_FAILURE);
    }
    printf("Flushing page cache, dentries and inodes...\n");
    // First: the traditional three sync calls. Perhaps not needed?
    // For security reasons, system("sync") is not a good idea.
    sync();
    sync();
    sync();
    FILE* f;
    f = fopen("/proc/sys/vm/drop_caches", "w");
    if (f == NULL) {
        fprintf(stderr, "flush-cache: Couldn't open /proc/sys/vm/drop_caches\n");
        exit(EXIT_FAILURE);
    }
    if (fprintf(f, "3\n") != 2) {
        fprintf(stderr, "flush-cache: Couldn't write 3 to
/proc/sys/vm/drop_caches\n");
        exit(EXIT_FAILURE);
    }
    fclose(f);
    printf("Done flushing.\n");

    return 0;
}
```

edited Jan 21 at 12:10

answered Nov 30 '12 at 14:23



[Thomas Padron-McCarthy](#)  
22.8k ● 5 ● 43 ● 69

Hi Thomas - cool - let me give it a try and get back to you! – [Peter Cogan](#) Nov 30 '12 at 14:24

1 That worked! Thanks Thomas - you deserve your high reputation score – [Peter Cogan](#) Nov 30 '12 at 14:34

This executable is a security risk - you don't set your PATH and run an external command without an absolute path name. You can just call sync() system call directly instead of running the external sync command. – [Amos Shapira](#) Jan 30 '14 at 19:36

@Amos: Yes, I agree. Thanks! – [Thomas Padron-McCarthy](#) Jan 31 '14 at 14:52