Changes submitted for bug 1672447
Email sent to:
no one

Bug 1672447 - sudo is resetting the real user id in the process tables (edit)

Save Changes

Status:	NEW (edit)	**Reported:**	2019-02-05 00:06 UTC by thomas.walker.lynch@gmail.com
Alias:	None (edit)	**Modified:**	2019-02-05 11:16 UTC (History)
Product:	Fedora	**CC List:**	□ Add me to CC list 5 users (edit)
Component:	sudo Click to list all components (Show other bugs)	**Ignore Bug Mail:**	□ (never email me about this bug)
Version:	rawhide	**Fixed In Version:**	
Hardware:	All Unspecified	**Doc Type:**	If docs needed, set a value
Priority:	unspecified	**Doc Text:**	If this bug requires documentation, please select an appropriate Doc Type value.
Severity:	low	**Verified:**	None (edit)
Target Milestone:	---	**Clone Of:**	
Assignee:	Daniel Kopeček	**Environment:**	
QA Contact:	Fedora Extras Quality Assurance	**Last Closed:**	
Docs Contact:		**Dependent Products:**	
URL:		**Flags:**	*None yet set* (set flags)
Whiteboard:			
Keywords:			
Personal Tags:			
Depends On:			
Blocks:			

TreeView+ depends on / blocked

Hide advanced fields

Attachments	(Terms of Use)
Add an attachment (proposed patch, testcase, etc.)	

External Trackers

Add External Bug: Tracker Bug ID URL Or paste full URL here +

Groups:

Only users in at least one of the selected groups can view this bug:

Unselecting all groups makes this a more public bug.

Users in the roles selected below can always view this bug:

☑ Reporter
☑ CC List

The assignee , QA contact and Docs contact can always see a bug, and this section does not take effect unless the bug is restricted to at least one group.

thomas.walker.lynch@gmail.com 2019-02-05 00:06:55 UTC Description [reply] [−]

Collapse All Comments
Expand All Comments

Add Comment
Show CC Changes

```
Description of problem:  Note the man page for the pam login module,
https://linux.die.net/man/8/pam_loginuid  It says "You should not use it for
applications like sudo or su as that defeats the purpose by changing the
loginuid to the account they just switched to." And there is good reason for
this, as otherwise there is no reliable method for finding the real user id.
Currently people appear to be using environment variables, but those can be
changed by programs or scripts after sudo is called (but not before, as sudo
resets them).


Version-Release number of selected component (if applicable):
F29


How reproducible:
completely
```

Steps to Reproduce:
1. enter this program:

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

int main(){
  int uid = getuid();
  int euid = geteuid();
  printf("real_id: %u effective_id: %u\n",uid,euid);
  return 0;
}
```

2. compile it:

```
gcc -o real_id real_id.cc
```

3. run it from a shell:

```
> ./real_id
real_id: 49972 effective_id: 49972
```

4. now run it with sudo:

```
> sudo ./real_id
real_id: 0 effective_id: 0
```

Actual results:

```
real_id: 0 effective_id: 0
```

The real_id is reset, and there is no way to reliably get the information back because it is gone from the OS process table.

Expected results:

```
real_id: 49972 effective_id: 0
```

here the processes real id remains the processes real id

Additional info:
When a program runs as root, there is no way to reliably distinguish if it is natively run, or if it has been invoked by a user land shell through sudo. Say for example, program1 calls program2.   Userland calls program 1 via sudo:
sudo program1.   Then program1 calls program2. Program1 might be poorly written where it does not pass its environment to program2, or it might be subtly malicious and change environment variables. program2 might then mistakenly believe it is root native run root program, instead of a userland invoked program. Furthermore, it can be the case, that a user land person sets the variables, for example SUDO_USER=whoever and then calls program2 but not through sudo. If program2 relies on the environment variables it might assume that it is running under sudo when it is not. Perhaps then it makes a mess and crashes when it gets to the part where it really needs to be root.

thomas.walker.lynch@gmail.com 2019-02-05 11:11:58 UTC Comment 1 [reply] [−]

This affects applications that desire to provide a service to a specified user, particularly those that are called by programs that have been called by sudo. The question within such a program will be 'which user?'. As noted above this question can be perhaps be reliably answered by examining the SUDO_USER variable in the target program called by sudo, I'm not really sure as the environment variable passing system was not designed to be bullet proof (?), but demonstrably not in secondary programs that sudo target program calls.

Note, however, setuid root programs do preserve the real id.

```
-r-sr-x---. 1 root     morpheus 18440 Feb  5 11:47 real_id_suid_root
```

Where this is the same program as given in the bug report above, runs as:

```
> ./real_id_suid_root
real_id: 49972 effective_id: 0
```

Which is the expected behavior.

People have suggested instead using getlogin(3) or programs that call it, but that is based on utmp, as the manpage notes, https://linux.die.net/man/3/getlogin, "
Unfortunately, it is often rather easy to fool getlogin()." Others have suggested reading other log files that have similar problems. All of this because the real id is missing. In addition because setuid root programs function as expected, even if such a solution was found, another sudo target program that calls such a program would not know to use this workaround. (Once the real id is lost, it remains lost to all children processes.)

Perhaps one solution is to say that a program run by sudo should never call a service that needs to know who the user who it is providing the service for really is. That would be really tough rule for programmers to verify. In addition, today, because the real id has been reset to user, any such program will assume that it is providing the service for root. That is a bad thing. It has surely already happened that root data has been affected by this.

Additional Comments:

Comment Preview

□ Need additional information from other

Status: NEW

Mark as Duplicate

Save Changes

Format For Printing - XML - Clone This Bug - TreeView+ - Top of page