

sudo

Beginner to Expert in One Hour

Replatform Technologies LLC

August 15, 2007



History

- sudo - short for “superuser do”
- sudo - pronounced: sue due
- 1980 - First available
- 1991 - GPL (Gnu Public License)
- 1996 - BSD License



Define Security

- Configuring a system to allow users do what they need to do and disallow what they do not need to do.



What is sudo?

- Runs commands under the guise of another user in a controlled and configurable fashion.



Why sudo?

- Elevate someone's authority to do things they normally can not do.



Uses for sudo

- Delegation
- Convenience
- Logging



How to use sudo

- preface another command
 - `> sudo kill 3961`
- enter your own password
 - Password: _



Understanding the Demo

(the cast)

- root
 - The all powerful user.
- mrtrust
 - The system administrator when logged his unprivileged account.
- mruser
 - The typical user of a system who does have any special privileges.

Understanding the Demo

• the user name

• the command prompt[†]

```
# mrtrust@mikepb $ id
```

• the command

```
uid=504 (mrtrust) gid=504 (mrtrust)  
groups=504 (mrtrust), 81 (appserveradm),  
79 (appserverusr), 80 (admin)
```

• the significant part of the output

Detailed description: The image shows a terminal window with a red border. The prompt is '# mrtrust@mikepb \$'. The command 'id' is entered. The output shows the user's identity and group memberships. Red arrows and dots point to specific parts of the terminal text with labels: 'the user name' points to 'mrtrust@mikepb', 'the command prompt[†]' points to '\$', 'the command' points to 'id', and 'the significant part of the output' points to '(mrtrust)' in the output line.

[†](custom) to set your prompt: set the PS1 environment variable.

Simple Demo

```
mikepb:~ root# id
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon), 2(kmem), 3(sys), 4(tty), 29
(certusers), 5(operator), 80(admin), 20(staff)
mikepb:~ root# su - mrtrust
# mrtrust@mikepb $ id
uid=504(mrtrust) gid=504(mrtrust) groups=504(mrtrust), 81(appserveradm), 79(apps
erverusr), 80(admin)
# mrtrust@mikepb $ sudo id
Password:
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon), 2(kmem), 3(sys), 4(tty), 29
(certusers), 5(operator), 80(admin), 20(staff)
# mrtrust@mikepb $ exit
logout
mikepb:~ root# su - mruser
# mruser@mikepb $ id
uid=503(mruser) gid=503(mruser) groups=503(mruser)
# mruser@mikepb $ sudo id
Password:
mruser is not in the sudoers file. This incident will be reported.
# mruser@mikepb $ exit
logout
mikepb:~ root# █
```

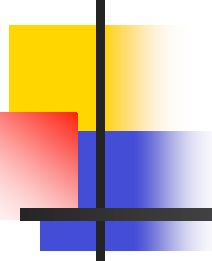


Sudo Principles

- Authenticate
- Authorize
- Restrict
- Log

Authorize Demo

```
# mrtrust@mikepb $ sudo visudo # authorize mruser for id
Password:
# mrtrust@mikepb $ sudo su - mruser
# mruser@mikepb $ sudo id
Password:
uid=0(root) gid=0(wheel) groups=0(wheel), 1(daemon), 2(kmem), 3(sys), 4(tty), 29
(certusers), 5(operator), 80(admin), 20(staff)
# mruser@mikepb $ sudo visudo
Sorry, user mruser is not allowed to execute '/usr/sbin/visudo' as root on mikep
b.
# mruser@mikepb $ exit
logout
# mrtrust@mikepb $ █
```



A deeper look into the
sudoer file.



The most basic sudo authorization rule

User specification (authorization rule):

```
user host=cmnd
```

Example :

```
mruser mikepb=/usr/bin/id
```



Add a twist

User specification:

```
user host=(runas) cmd
```

Example :

```
mruser mikepb =(mrappl) /usr/bin/id
```

Demo of runas rule

```
# mrtrust@mikepb $ sudo visudo # authorize mruser to run id as mrappl
Password:
# mrtrust@mikepb $ sudo su - mruser
# mruser@mikepb $ sudo id
Password:
Sorry, user mruser is not allowed to execute '/usr/bin/id' as root on mikepb.
# mruser@mikepb $ sudo -u mrappl id
uid=505(mrappl) gid=505(mrappl) groups=505(mrappl)
# mruser@mikepb $ exit
logout
# mrtrust@mikepb $ █
```



More about Runas

From Demo:

```
mruser mikepb =(mrapp1) /usr/bin/id
```

Real World:

```
mruser dbserver =(db2inst1) \  
    /home/db2inst1/sqlllib/adm/db2start
```

Best Practice:

```
Runas_Alias DB2 = db2inst1, db2inst2, db2inst3  
mruser dbserver =(DB2) \  
    /home/db2inst[1-3]/sqlllib/adm/db2start
```

Alternative:

```
mruser dbserver =(db2adm) \  
    /home/db2inst[1-3]/sqlllib/adm/db2start
```



Bad Security

```
mruser mikepb=(ALL,!root)/usr/bin/id
```



More about aliases

Recall what a “user specification” is:

```
user host=(runas) cmnd
```

There is an alias for each component:

User_Alias

Host_Alias

Runas_Alias

Cmnd_Alias



What aliases have in common

- The elements are simply comma separated.
 - `User_Alias EMPLOYEES=alice,barry,chriss`
- The elements can be an item or another alias.
 - `User_Alias DBTEST=dave,eugene,EMPLOYEES`
- The elements can be negated with a !.
 - `User_Alias DBPROD=DBTEST,!dave`
- The ALL alias.



Alias specific features

- Specify a unix group name by prefacing with a %.
 - `User_Alias APTESTERS=alice,%testers`
 - (works for `Runas_Alias` too)
- Specify a netgroup name by prefacing with a +.
 - `User_Alias APTESTERS=alice,+testers`
 - (works for `Runas_Alias` and `Host_Alias` too)
- Specify a numeric uid with #.
 - `Runas_Alias DB2INST=#752`



Host_Alias is a different Animal

Host_Alias can specify a machine by host name, ip address, and ip address with a mask (dotted decimal & CIDR).

```
Host_Alias MYMACH = \  
mikepb, \  
192.168.1.104, \  
192.168.1.0/255.255.255.0, \  
192.168.1.0/24
```

User Alias and groups (%)

- mrtrust can run anything:
 - Fragment from sudoers (sudo configuration file)
 - %admin ALL=(ALL) ALL
 - Groups mrtrusted belongs to:
 - # mrtrust@mikepb \$ id
uid=504 (mrtrust) gid=504 (mrtrust)
groups=504 (mrtrust), 81 (appserveradm),
79 (appserverusr), 80 (admin)



Runas Alias and groups (%)

- mruser becomes mister-a-little-bit-trusted:
 - `mruser ALL=(%staff) /usr/bin/ki.`



Odds and Ends

`# Comment lines begin with crosshatch.`

`Continued lines \`

`End with backslash.`



Recipe #1: Edit sudo config

Problem:

Two users editing the sudo configuration file at the same time.

Solution:

Use the `visudo` command like this:

```
sudo visudo
```

Or login as root and just use: `visudo`

Bonus:

`visudo` does syntax edit checks too.



Recipe #2: Colorize vim session

1. Install these files:

1. `$VIM/syntax/sudoers.vim`
 2. `$VIM/ftplugin/sudoers.vim`
- } google is your friend

2. Edit this file:

1. `$VIM/filetype.vim`, add this line:
2. `au BufNewFile,BufRead /etc/sudoers,sudoers.tmp setf sudoers`

3. While editing, issue this command:

1. `:syn on`
2. or add `'syn on'` to your `.vimrc` file



Defaults

- “Defaults” is a keyword in sudo.
- Sets a value for an internal variable.
- Last value wins.
- Can be set globally or
 - by User
 - by Host
 - by Runas user
 - by Cmnd (Version 1.7 of sudo)



Recipe #3: Unlocked Terminals

Problem:

Users who do not lock their terminals when they step away from their workstation.

Solution:

Add this line to the sudoers file:

```
Defaults:mruser timestamp_timeout=0
```

Require Password Demo

```
# mruser@mikepb $ sudo -u mrappl id
Password:
uid=505(mrappl) gid=505(mrappl) groups=505(mrappl)
# mruser@mikepb $ sudo -u mrappl id
uid=505(mrappl) gid=505(mrappl) groups=505(mrappl)
# mruser@mikepb $ exit
logout
# mrtrust@mikepb $ sudo visudo # require a password for mruser
Password:
# mrtrust@mikepb $ sudo su - mruser
# mruser@mikepb $ sudo -u mrappl id
Password:
uid=505(mrappl) gid=505(mrappl) groups=505(mrappl)
# mruser@mikepb $ sudo -u mrappl id
Password:
uid=505(mrappl) gid=505(mrappl) groups=505(mrappl)
# mruser@mikepb $ █
```



Recipe #4: Plug Security Hole

Problem:

Users who log in at multiple workstations.

Solution:

Add this line to the sudoers file for a particular user:

```
Defaults:mruser tty_tickets
```

or this for all users:

```
Defaults tty_tickets
```

Recipe #5: Clarify Password Prompt

Problem:

Password prompt is not clear as to which password to type

Solution:

Add this line to the sudoers file:

```
Defaults passprompt="%u@%h Password:"
```

For mruser, the prompt would change to this:

```
mruser@mikepb Password:
```

That makes it clear that mruser should type his own password.



Defaults

- “Defaults” is a keyword in sudo.
- Sets a value for an internal variable.
- Last value wins.
- Can be set globally or
 - by **User**
 - by **Host**
 - by **Runas** user
 - by **Cmnd** (sudo version 1.7)



Defaults

- Globally

- `Defaults tty_tickets`

- By User

- `Defaults:mruser timestamp_timeout=0`

- By Host

- `Defaults@webserver !logfile,syslog=authpriv`

- By Runas User

- `Defaults>mrappl passprompt="%U@%h Password", target`





Recipe #6: Log root user

Problem:

Users who you do not trust need to run 'wide open'.

No Perfect Solution:

- Properly isolate the machine on your network as though it is a foreign computer.
- Install rootsh
 - Readily available on Internet and package managers
- Add this rule to the sudoers file:
 - `mruser foreignhost = /opt/local/bin/rootsh`
- Tell mruser to do this to get a root shell:
 - `sudo rootsh`
- Monitor mruser's root activity in syslog



Recipe #7: Limit root use

Problem:

A script runs as root when a lesser privilege would do.

Solution:

Change the script to run at a lesser privilege or change the resource to require a lesser privilege.

Example:

```
mrtrust apserver = (root) chown -R mrback /backup
```

```
mruser apserver = (root) su - mrback
```



Recipe #8: Isolate Privilege

Problem:

A big script needs so many different privileges that you are tempted to run it as root.

Solution:

Isolate the privileged part of the task.

Example:

```
mruser mikepb = (apache) cat /var/log/httpd/access_log
```

```
mruser mikepb = (postgres)/Users/postgres/bin/writelogrow
```



Tags

- Syntax
 - User Host=(Runas)Tag:Cmnd
- Easy because there are only 4:
 - EXEC
 - NOEXEC
 - PASSWORD
 - NOPASSWORD



Recipe #9: Disable password

Problem:

sudo needs to run from cron job, so there is no user to type a password

Solution:

Use tag `NOPASSWD:` in the user specification for the command in question

Example:

```
mruser mikepb=(mrapp1)NOPASSWD:/usr/bin/id
```



The lecture

```
# mruser@mikepb $ sudo ls
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
mruser@mikepb Password:
```



Recipe #10: Change lecture

Annoyance:

sudo lectures users and it causes more confusion than obedience.

Solution:

Use default no lecture to avoid the lecture or change the lecture.

Example:

```
Defaults !lecture
```

Alternative:

```
Defaults lecture file=/etc/mvlecturefile
```



Logging

Logging is turned on with any or all of these:

Defaults logfile=/var/log/special_sudolog

Defaults mailto=root

Defaults syslog=authpriv

and turned off with any or all of these:

Defaults !logfile

Defaults !mailto

Defaults !syslog

Logging Options

	successful	unsuccessful	bad password	no user	no host	no permission
logfile	M	M				
syslog	C	C				
mailto	O	→	O	O	O	O

M - Mandatory
 C - Configurable
 O - Optional



Logfile entries

Jul 25 16:33:33 : mrtrust : TTY=ttyp1 ; PWD=/Users/mrtrust ;
USER=root ; COMMAND=/usr/bin/id

Jul 25 16:34:02 : mrtrust : 3 incorrect password attempts ; TTY=tt
PWD=/Users/mrtrust ; USER=root ; COMMAND=/usr/bin/id

Jul 25 16:34:42 : mrtrust : TTY=ttyp1 ; PWD=/Users/mrtrust ;
USER=root ; COMMAND=/usr/bin/id

Jul 25 16:36:32 : mruser : user NOT in sudoers ; TTY=ttyp1 ;
PWD=/Users/mruser ; USER=root ; COMMAND=/usr/bin/id

Jul 25 16:38:21 : mruser : command not allowed ; TTY=ttyp1 ;
PWD=/Users/mruser ; USER=root ; COMMAND=/bin/l



syslog Entries

Jul 25 17:18:00 localhost sudo <Alert>: mrtrusted :

3 incorrect password attempts ; TTY=ttyp2 ; PWD=/Users/mrtrust ;
USER=root ; COMMAND=/usr/bin/id

Jul 25 17:19:27 localhost sudo <Notice>: mrtrusted : TTY=ttyp2 ;

PWD=/Users/mrtrust ; USER=root ; COMMAND=/usr/bin/id

Jul 25 17:20:42 localhost sudo <Alert>: mruser : command not allowed

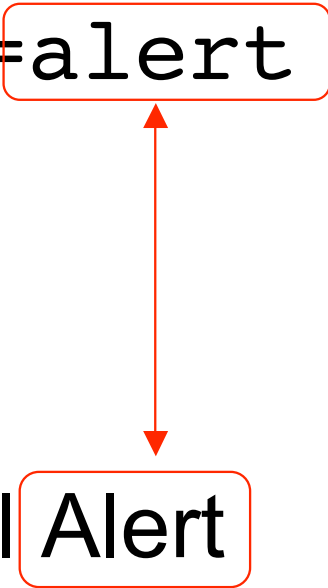
TTY=ttyp2 ; PWD=/Users/mruser ; USER=root ; COMMAND=/bin/l

Jul 25 17:21:53 localhost sudo <Alert>: mruser : user NOT in sudoers

TTY=ttyp2 ; PWD=/Users/mruser ; USER=root ; COMMAND=/bin/l



Recipe #11: syslog

- Put these in the sudoers file:
 - Defaults syslog=local2
 - Defaults syslog_goodpri=notice
 - Defaults syslog_badpri=alert
 - Extract log entries with these commands:
 - syslog -k Sender sudo
 - syslog -k Sender sudo -k Level Alert
- 



Writing your own scripts

Know your input:

- Read from files
- Passed on the command line
- Interaction with users
- Extracted from environment variables



Environment variables

- Defined by sudo
 - SUDO_USER, SUDO_UID, SUDO_GID
- Cleared by sudo
 - LIBPATH, LD_LIBRARY_PATH,
see 'sudo sudo -V' for complete list
- Set by sudo
 - USER
- Dangerous
 - PATH

```
#!/bin/bash

set -o noclobber # avoid overwrite of files
set -o errexit   # exit immediately upon error
#set -o pipefail # detect errors in piped commands

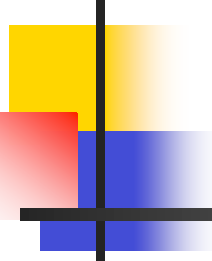
unset PATH # to force good habits in rest of script...

if [[ $SUDO_USER != "mrtrust" ]]; then
    echo "$SUDO_COMMAND only run by mrtrusted: use sudo visudo to
    exit 1
fi

if [[ $USER != "mrappl" ]]; then
    echo "$SUDO_COMMAND only run as mrappl: use sudo visudo to fi
    exit 1
fi

/usr/bin/id

exit 0
```



A

HUGE

Hole

script permissions

Check like this:

```
# mrtrust@mikepb $ ls -l /usr/bin/id
```

```
-r-xr-xr-x 3 root wheel 18452 Aug 21 2005 /usr/bin
```

- The owner should be root or the Runas user.
- The permissions should not include w (write)
 - (except possibly for owner)
- fix with chmod and chown
- check directory too



Recipe #12: Edit apache confi

Problem:

mruser needs to edit the apache configuration file with an interactive editor.

Solution:

Configure the sudoers file with this command:

```
mruser ALL=(apache) sudoedit  
    /etc/httpd/conf/httpd.conf
```

Tell mruser to use this command:

```
sudo -e -u apache /etc/httpd/conf/httpd.conf
```



What are the Alternatives?

- su: `su root -c "command"`
- su -: `su - root -c "command"`
- ssh: `ssh root@hostname "command"`
- rsh: `rsh hostname -l root "command"`
- sudo: `sudo -u root "command"`
- rootsu: login as root using su
- rootcon: login as root using console
- setuid: execute a setuid program or script
root could be any target user.

Alternatives Grid

	su	ssh	rsh	sudo	root su	root con	se
password	target	target /key	target	user	root	root	no
can avoid passwd	no	yes	yes	yes	no	no	n/
scrub env	no	no	no	yes	no	no	ye
.profile	optl	yes	no	no	optl	yes	no
global profile	optl	yes	no	no	yes	yes	no
restrict cmds	no	yes	no	yes	no	no	ye
real user	target	target	target	target	root	root	us
tty owner	user	target	target	user	user	root	us

test for yourself configurations change behavior



Learn more

- Man sudo
- Man sudoers
- Man visudo
- <http://www.sudo.ws/>
- <http://en.wikipedia.org/wiki/Sudo>
- Ask me to help!
 - +1 877 247 6887
 - I am always reading my email:
info@replatformtech.com