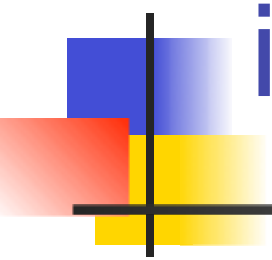
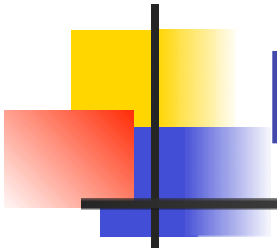


sudo Beginner to Expert in One (and 1/2) Hour

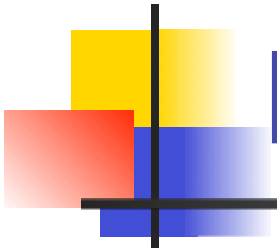
Michael Potter

July 28, 2026



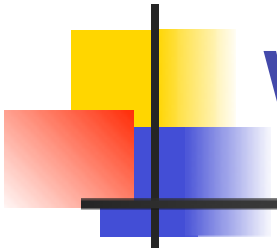
History

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



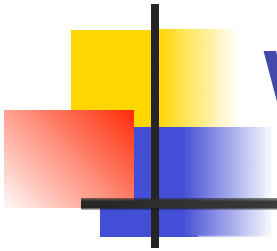
Define Security

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



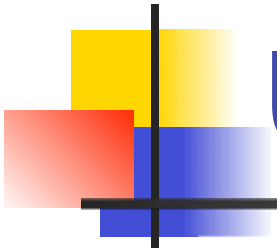
What is sudo?

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



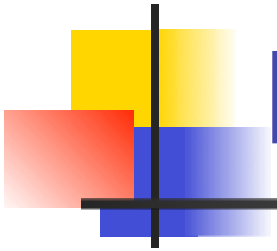
Why sudo?

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



Uses for sudo

- Delegation
- Convenience
- Logging



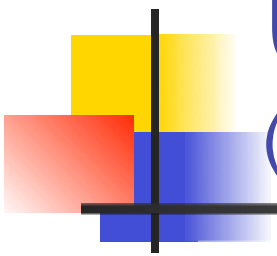
How to use sudo

- preface another command

```
> sudo kill 3961
```

- enter your own password

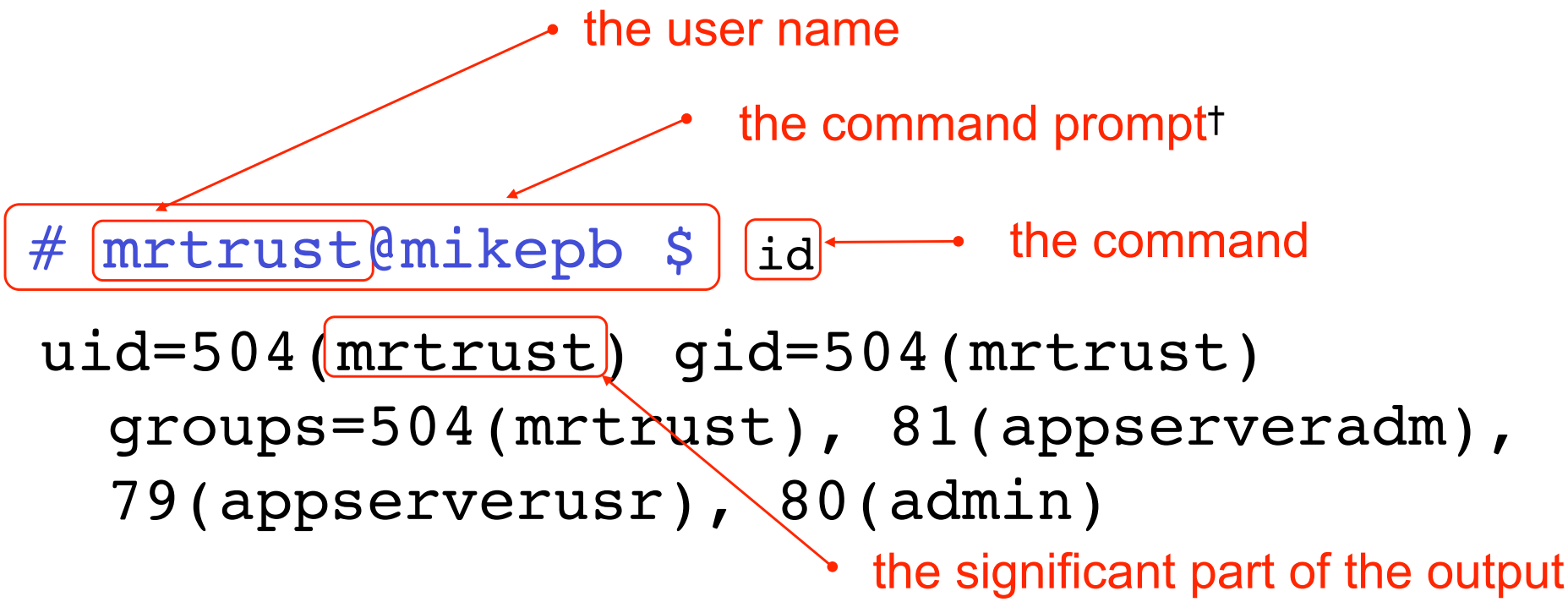
```
—Password: _
```



Understanding the Demo
(the cast)

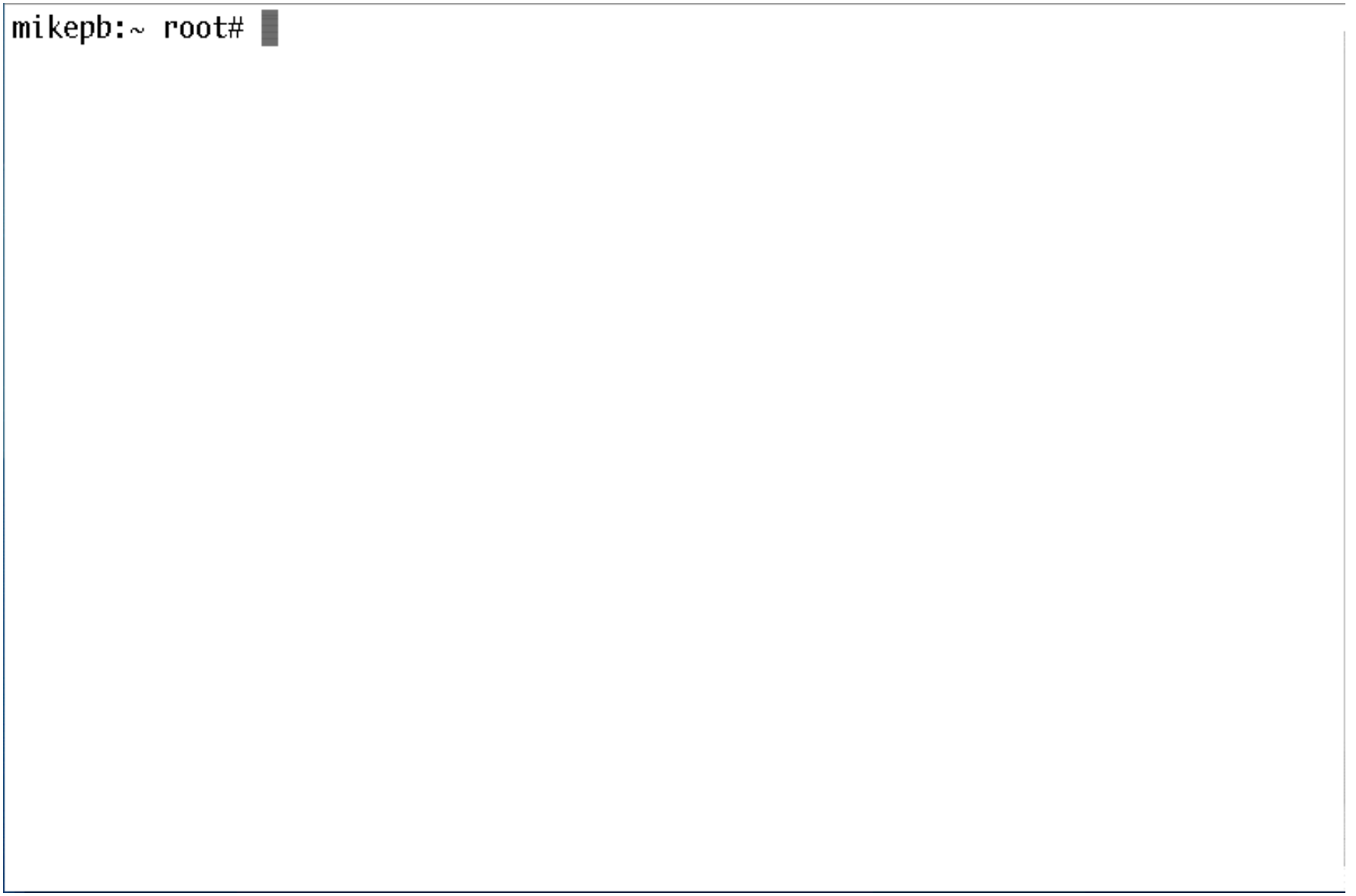
- root
  - The all powerful user.
- mrtrust
  - The system administrator when logged into his unprivileged account.
- mruser
  - The typical user of a system whom does not have any special privileges.

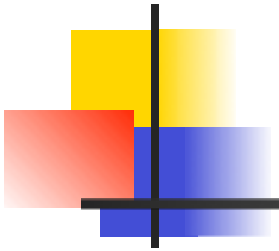
Understanding the Demo



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

Simple Demo





sudo Principles

- Authenticate
- Authorize
- Restrict
- Log

Authorize Demo

A deeper look into the sudoer file.

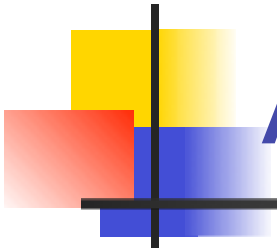
The most basic sudo authorization rule

From the Demo：

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

User specification (authorization rule):

```
user host=cmnd
```



Add a twist

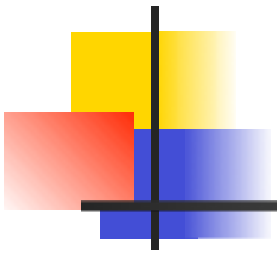
User specification:

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

Example：

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

Demo of runas rule



More about Runas

From Demo:

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

Real World:

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

Alternative:

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

Best Practice:

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

Bonus RunAs Feature for 1.7

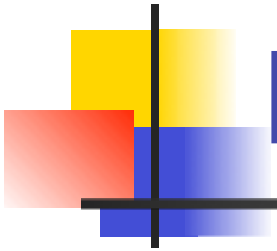
muser ALL=(mruser) /usr/bin/id

mruser@mikepb$ sudo -u mruser -g everyone id -gr

12

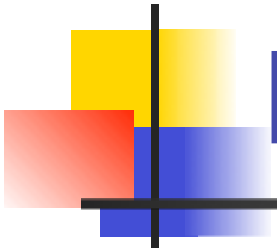
mruser@mikepb$ grep everyone /etc/group

everyone:*:12:



Bad Security

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



More about aliases

Recall what a “user specification” is:

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

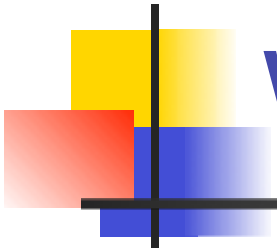
There is an alias for each component:

User_Alias

Host_Alias

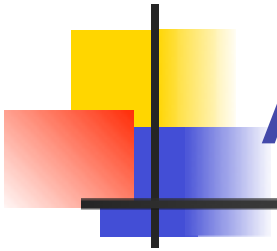
Runas_Alias

Cmnd_Alias



What aliases have in common

- The elements are simply comma separated.
  - `User_Alias EMPLOYEES=alice,barry,chris`
- 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.



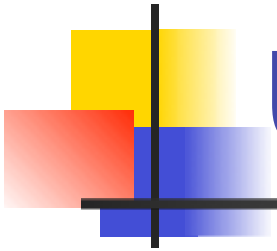
Alias specific features

- 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`
  - `(only works for Runas_Alias)`

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
```



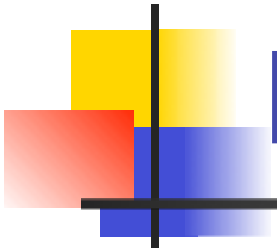
User Alias and groups (%)

- 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)`

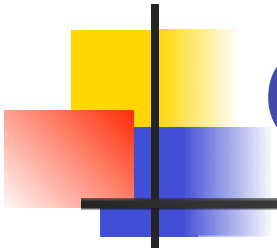
      `groups=504(mrtrust), 81(appserveradm),`

      `79(appserverusr), 80(admin)`



Runas Alias and groups (%)

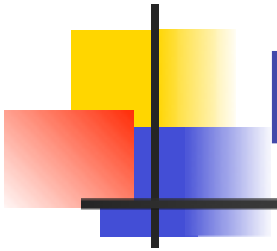
- mruser becomes mister-a-little-bit-trusted:
  - `mruser  ALL=(%staff) /usr/bin/kill`



Odds and Ends

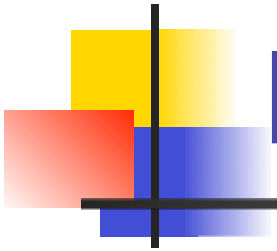
```
# Comment lines begin with crosshatch.


Continued lines \
End with backslash.
```



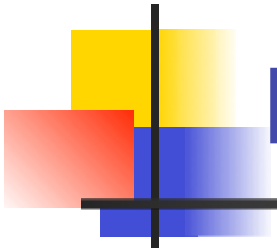
Passing Parameters

- /usr/bin/id
  - can run id with any argument.
- /usr/bin/id -gr
  - can run id with ONLY -gr
- /usr/bin/id ""
  - can only run id without arguments
- /bin/ls *fnmatch*
  - *fnmatch* is POSIX filename match library



Passing Parameters

mruser mikepb=/usr/bin/cat /var/log/kernel*.log*



Recipe #1: Edit sudo config

Problem:

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

Solution:

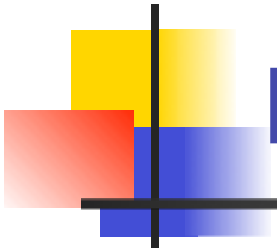
Use the `visudo` command like this:

```
sudo visudo
```

Or login as root and just use: `visudo`

Bonus:

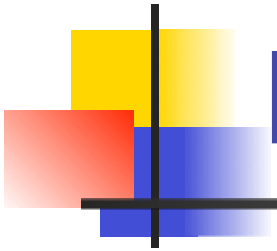
visudo does syntax edit checks too.



Recipe #2: Colorize vim session

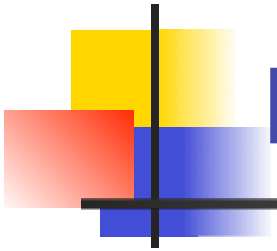
1. Install these files:
    1. `$VIM/syntax/sudoers.vim`
    2. `$VIM/ftplugin/sudoers.vim`

    } 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:

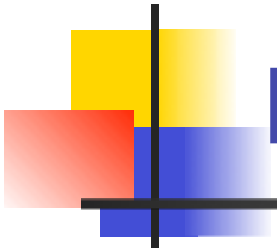
Users who do not lock their terminals when they step away from their workstation.

Solution:

Add this line to the sudoers file:

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

Require Password Demo



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:

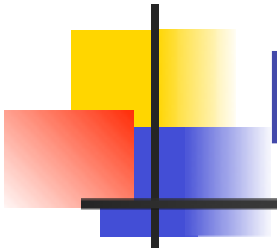
```
Defaults passprompt="%u@%h Password:"
```

For mruser, the prompt would change to this:

```
mruser@mikepb Password:
```

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

Also can be done on the command line with -p.

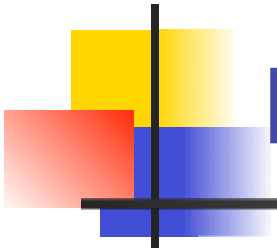


Defaults

- "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", targetpw`



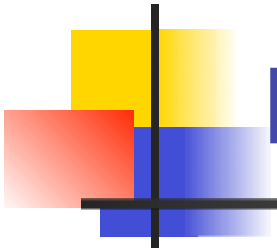
Recipe #6: Log root user

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.

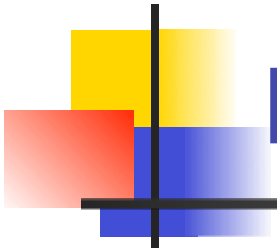
Solution:

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.

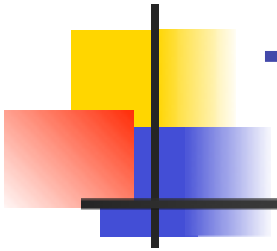
Solution:

Isolate the privileged part of the task.

Example:

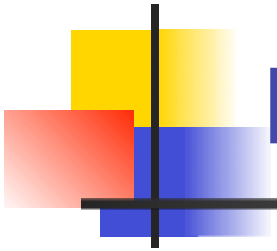
mruser mikepb = (apache) /usr/bin/cat /var/log/httpd/access_log

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



Tags

- Syntax
  - User Host=(Runas)Tag:Cmnd
- Easy because there are only 6:
  - EXEC
  - NOEXEC
  - PASSWORD
  - NOPASSWORD
  - SETENV (new in 1.7)
  - NOSETENV (new in 1.7)



Recipe #9: Disable password

Problem:

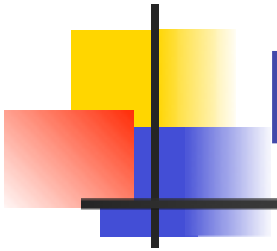
sudo needs to run from cron job, so there is no user to type a password.

Solution:

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

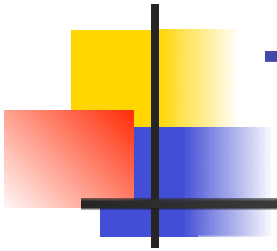
Example:

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



Recipe X: Pass in a variable

Problem:
Some commands need environment variables set.
Solution:
Add SETENV: to the user specification of the command in question. Then add the variable to the sudo command.
Example:
mruser ALL=SETENV:/usr/bin/env
sudo SOMEVAR=SILLY env |grep SOMEVAR

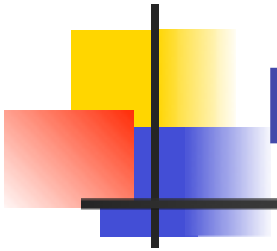


The lecture

```
# 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:

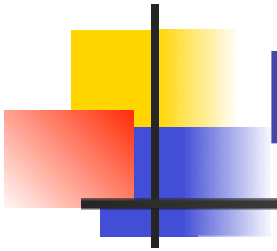
Use default no lecture to avoid the lecture or change the lecture.

Example:

```
Defaults !lecture
```

Alternative:

```
Defaults lecture_file=/etc/mylecturefile
```

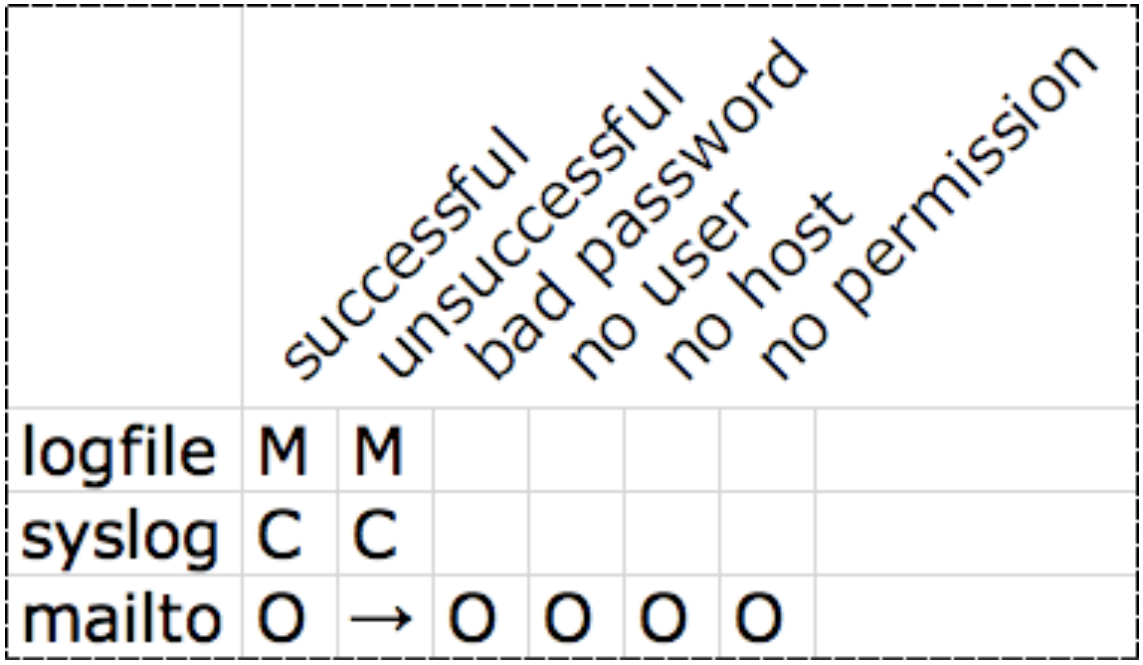


Logging

Logging is turned on with any or all of these:
Defaults logfile=/var/log/special_sudolog
Defaults mailto=root
Defaults syslog=authpriv

and turned of with any or all of these:
Defaults !logfile
Defaults !mailto
Defaults !syslog

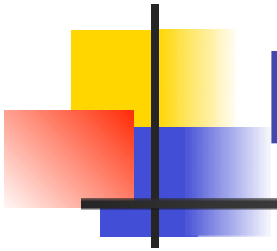
Logging Options



M - Mandatory

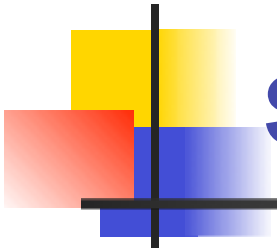
C - Configurable

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=ttyp1 ;
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/ls

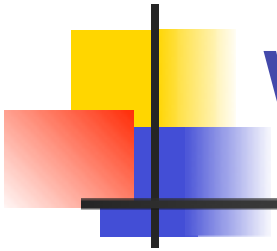


syslog Entries

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/ls
Jul 25 17:21:53 localhost sudo <Alert>: mruser : user NOT in sudoers ;
TTY=ttyp2 ; PWD=/Users/mruser ; USER=root ; COMMAND=/bin/ls

Recipe #11: syslog

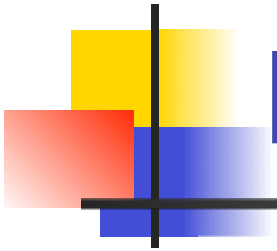
- Put these in the sudoers file:
  - `Defaults syslog=local2`
  - `Defaults syslog_goodpri=notice`
  - `Defaults syslog_badpri=alert`
- Extract log entries with these commands:
  - syslog -k Sender sudo
  - syslog -k Sender sudo -k Level Alert



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 fix"
   exit 1
fi

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

/usr/bin/id

exit 0
```

A

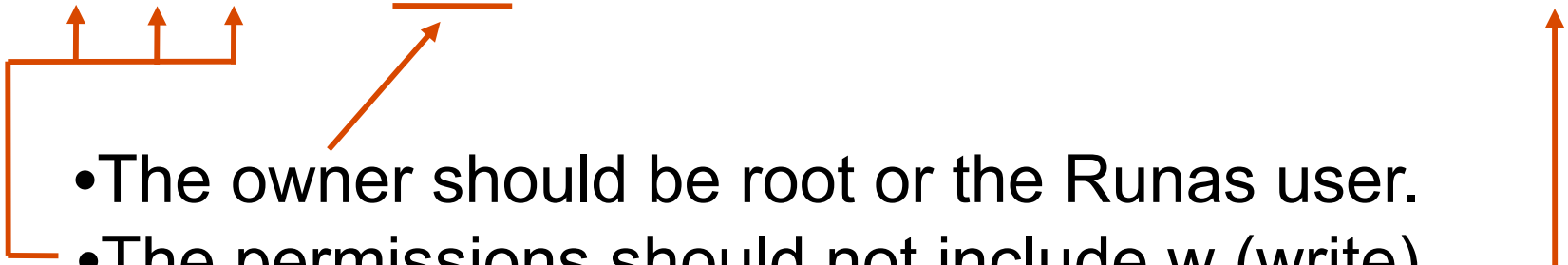
HUGE

Hole

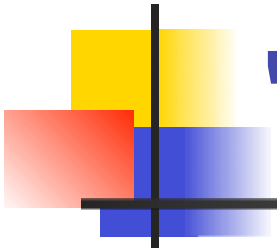
script permissions

Check like this:

```
# mrtrust@mikepb $ ls -l /usr/bin/id
-r-xr-xr-x   3 root  wheel  18452 Aug 21  2005 /usr/bin/id
```



- 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

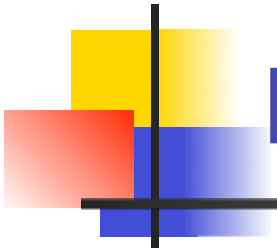


"sudo su -" alternative

sudo su -

sudo -s

sudo -s -u mruser



Recipe #12: Edit apache config

Problem:

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

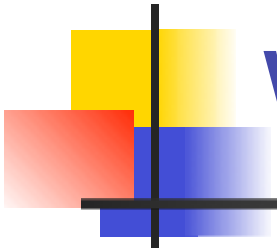
Solution:

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?

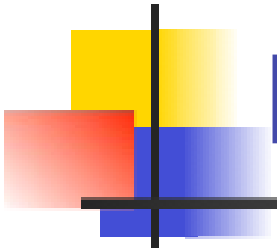
- su: su root -c “command”
- su -: su - root -c “command”
- ssh: ssh root@hostname “command”
- rsh: rsh hostname -l root “command”
- sudo: sudo -u root “command”
- rootsu: login as root using su
- rootcon: login as root using console
- setuid: execute a setuid program or script

root could be any target user.

Alternatives Grid

	su	ssh	rsh	sudo	root su	root con	setuid
password	target	target /key	target	user	root	root	no
can avoid passwd	no	yes	yes	yes	no	no	n/a
scrub env	no	no	no	yes	no	no	yes
.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	yes
real user	target	target	target	target	root	root	user
tty owner	user	target	target	user	user	root	user

test for yourself, configurations change behavior



Learn more

- Man sudo
- Man sudoers
- Man visudo
- http://www.sudo.ws/
- http://en.wikipedia.org/wiki/Sudo
-