HOWTO zu Sudo (substitute user do)

(C) 2005-2023 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>
OSTC Source Training and Consulting GmbH
http://www.ostc.de

$Id: sudo-HOWTO.txt,v 1.22 2025/02/23 20:14:55 tsbirn Exp $

Dieses Dokument beschreibt die Konfiguration und den Einsatz des "su"-Ersatzes "sudo" (substitute user do).

INHALTSVERZEICHNIS

1) Einführung
2) Konfiguration
2.1) Aliase
3) Kommandoaufruf per "sudo"
4) Kommandozeilen-Optionen
5) Hinweise
6) Konfigurations-Optionen (Defaults)
6.1) Beispiel für aktive Options-Einstellungen

---

1) Einführung

"sudo" (substitute user do) verleiht Benutzern auf einem Rechner für EIN Kommando "root"-Rechte (oder die eines anderen Benutzers), d.h. es vergibt "Benutzer-Privilegien" (user privilege). Es erlaubt daher auch normalen Benutzern die Ausführung von Systembefehlen und somit den Einsatz von "Unter-Administratoren" neben der "root".

Auf modernen Linux-Systemen gilt folgende Admin-Regelung:

* Die Entscheidung, wer mit "sudo" ein Kommando als "root" ausführen darf, wird über die Zugehörigkeit zu einer Admin-Gruppe gesteuert, üblicherweise heißt diese Admin-Gruppe "sudo", "admin" oder "wheel".

* Der allererste Benutzer, der beim Installieren eines Linux-Systems angelegt wird, hat üblicherweise das Admin-Recht, weil er automatisch der Admin-Gruppe als Mitglied zugeordnet wird.

* Alle anderen Benutzer werden standardmäßig nicht der Admin-Gruppe zugeordnet und können daher "sudo" nicht verwenden um "root"-Recht zu erhalten.

* Zur Authentifizierung bei der Nutzung von "sudo" verwendet jeder authorisierte Benutzer sein EIGENES Benutzerpasswort.

* Der Administrator "root" hat überhaupt KEIN Passwort und eine direkte Anmeldung per SSH an seinem Account ist verboten (aus Sicherheitsgründen). D.h. man muss sich zuerst als normaler Benutzer per SSH an einem Rechner anmelden, und kann anschließend per "sudo" auf den root-Account umschalten.

* Das Kommando "su" kann gesperrt/nicht vorhanden sein, d.h man kann nicht permanent auf den Benutzer "root" umschalten, sondern nur für ein Kommando.

* Alle per "sudo" abgesetzten Befehle werden mit Benutzer-Account, Datum + Uhrzeit mitgeloggt, ebenso "sudo"-Befehle von nicht authorisierten Benutzern.

* Die Konfiguration von "sudo" erfolgt in folgenden Dateien:

```
/etc/sudo.conf     # Basis-Konfiguration von "sudo" (plugins, debug, ...)
/etc/sudoers       # Benutzer-Privilegien für "sudo"
/etc/sudoers.d/*   # Weitere Benutzer-Privilegien für "sudo" aufgeteilt auf Dateien
```

2) Konfiguration

Die "sudo"-Konfigurationsdatei "/etc/sudoers" muss die Zugriffsrechte "r--r----- root root" haben (wird bei jedem Aufruf geprüft!) und heisst:

```
  /etc/sudoers
```

Editiert werden sollte diese Datei nur so (lockt die Datei, startet Vi/Vim bzw. den durch EDITOR bezeichneten Editor und prüft die Syntax beim Speichern):

```
  visudo                          # Startet vi und lockt /etc/sudoers!
  vi /etc/sudoers                 # VORSICHT!
  vi /etc/sudoers.d/*             # VORSICHT!
```

Ein Benutzer-Privileg-Eintrag in "/etc/sudoers" hat folgende Form (die Teile in eckigen Klammern [...] dürfen weggelassen werden):

```
  USER HOST = [(RUNAS_USER [: RUNAS_GROUP])] [NOPASSWD:] [!] CMD[,...]
```

Dies bedeutet, dass der Benutzer USER auf dem Rechner HOST das Kommando CMD ausführen darf (bzw. mit "!" nicht ausführen darf). CMD läuft dann unter der Kennung "root" (oder "RUNAS" wenn diese in Klammern angegeben ist).

```
  USER HOST = CMD                 # Benutzer USER auf Rechner HOST darf Kommando CMD
ausführen
  USER HOST = CMD1, !CMD2         # Benutzer USER auf Rechner HOST darf Kommando CMD1
ausführen, aber NICHT CMD2
  USER HOST = CMD1, CMD2, CMD3    # Benutzer USER auf Rechner HOST darf Kommandos
CMD1, CMD2, CMD3 ausführen
  USER HOST = NOPASSWD: CMD       # Benutzer USER auf Rechner HOST darf Kommando CMD
ohne Passworteingabe ausführen
  USER HOST = (USER2) CMD         # Benutzer USER auf Rechner HOST darf Kommando CMD
als Benutzer USER2 ausführen
  USER HOST = (USER2:GROUP2) CMD  # Benutzer USER auf Rechner HOST darf Kommando CMD
als Benutzer USER2 + Gruppe GROUP2 ausführen
```

Für HOST, RUNAS und CMD ist jeweils das Schlüsselwort "ALL" möglich, falls dieses Element nicht eingeschränkt werden soll. D.h. mit folgendem Privileg darf der Benutzer USER auf jedem HOST alle Kommandos als beliebiger RUNAS-Benutzer ausführen:

```
  USER ALL = ALL                # Nur "root" als RUNAS-User erlaubt
  USER ALL = (ALL) ALL          # RUNAS-User beliebig
  USER ALL = (ALL : ALL) ALL    # RUNAS-User und RUNAS-Gruppe beliebig
```

"!" vor CMD verbietet das Ausführen dieses Kommando per "sudo":

Normalerweise wird der Benutzer (aus Sicherheitsgründen) beim 1. Aufruf nach einem Passwort gefragt (welches ist einstellbar), mit NOPASSWD: kann dies aber abgeschalten werden (d.h. er wird nicht nach einem Passwort gefragt).

Das CMD ist grundsätzlich mit ABSOLUTEM Pfad anzugeben, mehrere CMDs sind durch "," zu trennen.

Den Namen des eigenen Rechners HOST erfährt man normalerweise aus dem Prompt, mit dem Kommando "hostname" kann er auch ausgegeben werden.

Als Standard-Benutzerprivileg ("user privilege") ist normalerweise von vornherein eingetragen, dass "root" auf allen Rechner alles darf:

```
  # Defaults targetpw       # Auskommentieren --> Aufrufer-Passwort nötig
  root ALL = (ALL) ALL      # root darf alles mit "sudo" (;-)
```

Ebenso ist als Standard-Gruppenprivileg ("group privilege") normalerweise eingetragen, dass alle Mitglieder der Gruppe "wheel", "admin" oder "system" auf

allen Rechner alles dürfen (Zeichen "%" leitet einen Gruppennamen ein):

```
  %wheel ALL = (ALL) ALL    # Mitglieder der Gruppe "wheel" dürfen alles
  %admin ALL = (ALL) ALL    # Mitglieder der Gruppe "admin" dürfen alles
  %adm ALL = (ALL) ALL      # Mitglieder der Gruppe "adm" dürfen alles
  %sudo ALL = (ALL) ALL     # Mitglieder der Gruppe "sudo" dürfen alles
  %system ALL = (ALL) ALL   # Mitglieder der Gruppe "system" dürfen alles
```

Beispiel für weitere Einträge (Benutzer "tom" darf auf Rechner "r99" montieren und demontieren und Rechner "r1" herunterfahren (dort ohne Passwortabfrage!):

```
  tom  r99 = /sbin/mount /media/cdrom, /sbin/umount /media/cdrom
  tom  r1  = NOPASSWD: /sbin/shutdown -h now
```

* Der Platzhalter "ALL" steht jeweils "für alle" Benutzer, Rechner, Kommandos (und ist GROSS zu schreiben).

2.1) Aliase

Als Komponenten USER, HOST, CMD und RUNAS in einem Privilege sind auch sogenannte "Aliase" (oder Makros) verwendbar, in denen sich jeweils mehrere Objekte unter einem Namen zusammenfassen lassen. Hiermit ist eine einfachere Verwaltung und Erweiterung der Berechtigungen möglich:

```
  User_Alias  NAME = USER1, USER1, ...
  Host_Alias  NAME = HOST1, HOST2, ...
  Cmnd_Alias  NAME = CMND1, CMND2, ...
  Runas_Alias NAME = USER1, USER2, ...
```

Beispiele für Aliase:

```
  User_Alias OPERATOR  = hans, rick, cat, dog
  User_Alias ADMIN     = hans, tom
  Host_Alias HOME      = 192.168.1.0/24
  Host_Alias WEBSERVER = 192.168.9.0/24
  Cmnd_Alias PING      = /bin/ping
  Cmnd_Alias SU        = /bin/su
  Cmnd_Alias NEUSTART  = /sbin/reboot, /sbin/shutdown -r *, /sbin/init 6
```

Beispiele für die Nutzung von Aliasen in Benutzer-Privilegien:

```
  OPERATOR HOME      = PING, NEUSTART
  OPERATOR WEBSERVER = ALL, !SU
  ADMIN    WEBSERVER = SU
```

* Als Benutzer sind auch Gruppen mit einem vorangestellten "%"-Zeichen erlaubt:

```
    User_Alias NORMAL_USERS = %users
```

* Als Rechner sind auch Netzgruppen mit einem vorangestellten "+"-Zeichen erlaubt:

```
    Host_Alias SUN = +suns
```

3) Kommandoaufruf per "sudo"

Der Aufruf eines Kommandos unter der Kontrolle von "sudo" erfolgt durch:

```
  sudo /PFAD/ZU/KMDO [OPTIONEN]        # Meist absoluter Pfad notwendig!
```

Beim 1. Aufruf muss man in der Regel sein eigenes Passwort eingeben. Man erhält dann ein "Ticket", das 5 Minuten gültig ist und muss beim erneuten Aufruf dieses Kommandos kein Passwort mehr eingeben. Beispiel:

```
  sudo /sbin/yast


4) Kommandozeilen-Optionen
--------------------------

* -e (--edit) dient dazu, eine oder mehrere Dateien zu editieren. Startet den
  Editor nicht per "sudo" sondern mit den Rechten des Benutzers (um Shell-Escape
  als "root" zu verhindern). Statt dessen werden temporäre Kopien angelegt
  (gehören aufrufendem Benutzer), und der in den Umgebungsvariablen SUDO_EDITOR,
  VISUAL und EDITOR (in dieser Reihenfolge durchsucht) hinterlegte Editor mit
  den Dateien gestartet. Nach dem Verlassen des Editors werden die temporären
  Dateien wieder in die Originaldateien zurückgeschrieben:

    sudo -e DATEI...
    sudoedit DATEI...

* -l (--list) listet dem Benutzer auf, welche Kommandos er mit "sudo" ausführen
  darf:

    sudo -l

* -U (--other-user) erlaubt das Auflisten der erlaubten Kommandos für beliebigen
  Benutzer USER (nur "root" + Benutzer mit dem Privileg "ALL"):

    sudo -U USER -l

* -V (--version) gibt die Sudo-Version aus:

    sudo -V

* -b (--background) führt ein Kommand in Hintergrund aus
  ("&" funktioniert im Zusammenhang mit "sudo" leider nicht):

    sudo -b "find /etc -print > /tmp/liste"
    sudo sh -c "find /etc -print > /tmp/liste &"   # alternativ

* -k (--reset-timestamp) und -K (--remove-timestamp) entfernt die temporär
  zwischengespeicherten Anmeldedaten, sodass beim nächsten sudo-Aufruf wieder
  das Passwort einzugeben ist:

    sudo -k
    sudo -K


5) Hinweise
-----------
* ACHTUNG: Ein Syntaxfehler in "/etc/sudoers" --> "sudo" geht nicht mehr!

* Unbedingt das Aussperren beim Editieren von "/etc/sudoers" verhindern, indem
  eine weitere root-Anmeldung geöffnet wird. Wenn Datei "/etc/sudoers" fertig
  editiert ist, erst testen ob "sudo" noch funktionsfähig ist und dannach
  die zusätzliche root-Anmeldung wieder schließen.

* Per "sudo" als "root" oder ein Benutzer USER anmelden (ohne root-Passwort!):

    sudo su          # root + aktuelle Umgebung
    sudo su -        # root + root-Umgebung (analog echtem Login)
    sudo su USER     # USER + aktuelle Umgebung
    sudo su - USER   # USER + USER-Umgebung (analog echtem Login)

* "sudo" ist kein Dienst, Datei "/etc/sudoers" wird bei jedem Aufruf gelesen.

* "/usr/bin/sudo" muss Set-UID-Recht gesetzt haben.

* Datei "/etc/sudoers" muss Zugriffsrechte "-r--r-----" für Benutzer "root" +
```

Gruppe "root" haben.

* Jede BERECHTIGTE Benutzung von "sudo" wird in "/var/log/auth.log"
(evtl. auch in "/var/log/messages") protokolliert:

```
Jan  8 09:43:07 charlton sudo:   tsbirn : \
  TTY=ttyp1 ; PWD=/home/tsbirn ; USER=root ; \
  COMMAND=/usr/bin/view /var/log/auth.log
```

* Jede UNBERECHTIGTE Benutzung von sudo wird in "/var/log/auth.log"
(evtl. auch in "/var/log/messages") protokolliert:

```
Jan  8 09:43:15 charlton sudo:   kurs : user NOT in sudoers ; \
  TTY=pts/2 ; PWD=/home/kurs0 ; USER=root ; COMMAND=/bin/ls
```

* Fehlermeldung falls "sudo" nicht erlaubt:

```
USER is not in the sudoers file.  This incident will be reported.
```

* Mit "sudo" als root anmelden:

```
sudo su -   # su ausführen
sudo -i     # -i=interactive
sudo bash   # Shell starten
```

* Der Aufruf von "cd" alleine per "sudo" ist sinnlos, da die von "sudo" gestartete
Shell anschliessend wieder verlassen wird, die Wirkung des cd-Kommandos also
mit dieser Shell vergessen wird:

```
sudo cd /usr/local/protected
```

Sinnvoll dagegen ist die Kombination von "cd" mit weiteren Kommandos:

```
sudo "cd /usr/local/protected; ls -l *.log"         #
sudo -c sh "cd /usr/local/protected; ls -l *.log"   # OK
```

* Umlenkung von/auf Datei ist durch folgenden Trick erreichbar (-c=command):

```
sudo sh -c "find /etc -print > /tmp/liste"
```

* Der Start von per Pipe verbundenen Programmen per sudo ist problemlos:

```
sudo ls /var/log/apache2 | less           # 1. Prozess als "root"
ls /var/log/apache2 | sudo less           # 2. Prozess als "root"
sudo "ls /var/log/apache2 | sudo less"    # Beide Prozesse als "root"
```

* Die Kommandozeilen-Vervollständigung per TAB-Taste verhält sich bei
vorangestelltem "sudo" evtl. nicht vernünftig. Dann das Kommando erst ohne
das führende "sudo" per TAB zusammenbauen und ausführen (macht wg. fehlender
Rechte nichts). Dann das letzte Kommando erneut per "sudo" ausführen:

```
 CMD ...   # CMD ohne sudo (TAB-Comletion funktioniert, Aufruf scheitert)
 sudo !!   # --> sudo CMD ... wird ausgeführt
```

* Wenn man "sudo" etwas bequemer verwenden will, kann man z.B. die eigene
".alias"-Datei um folgende Abkürzungen erweitern:

```
alias y="sudo /sbin/yast"
alias y2="sudo /sbin/yast2"
alias feierabend="sudo /sbin/shutdown -h now"      # Rechner JETZT anhalten
```

* Wenn der "vi" nicht gerade der Lieblingseditor ist, als "root" in der
grafischer Oberfläche anmelden und "/etc/sudoers" mit "gedit" editieren. Oder
den Editor in den Umgebungsvariablen SUDO_EDITOR, EDITOR und VISUAL setzen
(werden in dieser Reihenfolge ausprobiert). Die Config-Variable "editor" in
"/etc/sudoers" listet die prinzipiell erlaubten Editoren auf (absolute Pfade).

* Die Timestamps der Tickets werden in "/var/run/sudo" abgespeichert.

* Auch als root kann man "sudo" gut einsetzen (zeichnet alle Befehle auf).

6) Konfigurations-Optionen (Defaults)

Option	Bedeutung
authenticate	Require users to authenticate by default
editor	Path to editor for use by visudo
exempt_group	Users in this group are exempt from password and PATH requirements
fqdn	Require fully-qualified hostnames in sudoers file
ignore_dot	Ignore '.' in $PATH
long_otp_prompt	Put OTP prompt on its own line
path_info	Allow some information gathering to give useful error messages
preserve_groups	Don't initialize group vector to that of target user
requiretty	Only allow user to run sudo if they have a tty
root_sudo	Root may run sudo
runas_default	Default user to run commands as
shell_noargs	If sudo is invoked with no arguments, start a shell
stay_setuid	Only set effective uid to target user, not real uid
umask	Umask to use or 0777 to use user's
use_loginclass	Apply defaults in target user's login class if there is one
env_check	Environment variables to check for sanity
env_delete	Environment variables to remove
env_editor	Visudo will honor EDITOR environment variable
env_keep	Environment variables to preserve
env_reset	Reset environment to a default set of variables
insults	Insult user when they enter an incorrect password
lecture	Lecture user first time they run sudo
log_host	Log hostname in (non-syslog) log

```
file                                |
  | log_year          | Log year in (non-syslog) log
file                                  |
  | logfile           | Path to log
file                                                  |
  | loglinelen        | Length at which to wrap log file lines (0 for no
wrap)             |
  +-------------------
+-----------------------------------------------------------------------+
  | mail_always       | Always send mail when sudo is
run                                 |
  | mail_badpass      | Send mail if user authentication
fails                               |
  | mail_no_host      | Send mail if user is not in sudoers for this
host                  |
  | mail_no_perms     | Send mail if user is not allowed to run a
command                  |
  | mail_no_user      | Send mail if user is not in
sudoers                               |
  | mailerflags       | Flags for mail
program                                              |
  | mailerpath        | Path to mail
program                                                |
  | mailsub           | Subject line for mail
messages                                        |
  | mailto            | Address to send mail
to                                            |
  +-------------------
+-----------------------------------------------------------------------+
  | rootpw            | Prompt for root's password, not
users's                             |
  | runaspw           | Prompt for runas_default user's password, not
users's               |
  | targetpw          | Prompt for target user's password, not
users's                    |
  | passprompt        | Default password
prompt                                          |
  | passwd_timeout    | Password prompt
timeout                                          |
  | passwd_tries      | Number of tries to enter a
password                                |
  | listpw            | When to require a password for 'list'
pseudocommand                |
  | verifypw          | When to require a password for 'verify'
pseudocommand                |
  | badpass_message   | Incorrect password
message                                         |
  +-------------------
+-----------------------------------------------------------------------+
  | always_set_home   | Always set $HOME to target user's home
directory                    |
  | set_home          | Set $HOME to target user when starting a shell with
-s             |
  | set_logname       | Set LOGNAME and USER environment
variables                           |
  +-------------------
+-----------------------------------------------------------------------+
  | syslog            | Syslog facility if syslog is being used for
logging                |
  | syslog_badpri     | Syslog priority to use when user authenticates
unsuccessfully      |
  | syslog_goodpri    | Syslog priority to use when user authenticates
successfully        |
  +-------------------
+-----------------------------------------------------------------------+
  | timestamp_timeout | Authentication timestamp
```

```
timeout                                  |
  | timestampdir      | Path to authentication timestamp
dir                                |
  | timestampowner    | Owner of authentication timestamp
dir                                |
  | tty_tickets       | Use a separate timestamp for each user/tty
combo                 |
  +-------------------
+-----------------------------------------------------------------+
```

6.1) Beispiel für aktive Options-Einstellungen

Ausgabe von "sudo -V" als Benutzer root:

```
  Sudo version 1.6.7p5
  Authentication methods:                                      pam
  Syslog facility if syslog is being used for logging:         auth
  Syslog priority to use when user authenticates successfully: notice
  Syslog priority to use when user authenticates unsuccessfully: alert
  Ignore '.' in $PATH
  Send mail if the user is not in sudoers
  Use a separate timestamp for each user/tty combo
  Lecture user the first time they run sudo
  Require users to authenticate by default
  Root may run sudo
  Set $HOME to the target user when starting a shell with -s
  Allow some information gathering to give useful error messages
  Visudo will honor the EDITOR environment variable
  Set the LOGNAME and USER environment variables
  Length at which to wrap log file lines (0 for no wrap): 80
  Authentication timestamp timeout:                       5 minutes
  Password prompt timeout:                                5 minutes
  Number of tries to enter a password:                    3
  Umask to use or 0777 to use user's:                     022
  Path to mail program:                                   /usr/sbin/sendmail
  Flags for mail program:                                 -t
  Address to send mail to:                                root
  Subject line for mail messages:                         *** SECURITY information for
%h ***
  Incorrect password message:                             Sorry, try again.
  Path to authentication timestamp dir:                   /var/run/sudo
  Default password prompt:                                Password:
  Default user to run commands as:                        root
  Path to the editor for use by visudo:                   /usr/bin/vi
  Environment variables to check for sanity:
          LANGUAGE
          LANG
          LC_*
  Environment variables to remove:
          BASH_ENV
          ENV
          TERMCAP
          TERMPATH
          TERMINFO_DIRS
          TERMINFO
          _RLD*
          LD_*
          PATH_LOCALE
          NLSPATH
          HOSTALIASES
          RES_OPTIONS
          LOCALDOMAIN
          IFS
  When to require a password for 'list' pseudocommand:   any
```

```
  When to require a password for 'verify' pseudocommand: all
  Local IP address and netmask pairs:                    10.0.3.53 / 0xffffff00


--------------------------------------------------------------------------------
```