sudoers

plugin de stratégie de sécurité de sudo par défaut

Ce plugin détermine les privilèges sudo de l'utilisateur en lisant la stratégie dans /etc/sudoers. sudo consulte le fichier sudo.conf pour déterminer quelle stratégieet plugin charger.

Les arguments du plugins sont les suivants :

ldap_conf=<path> emplacement du fichier ldap.conf alternatif
ldap_secret=<path> emplacement du fichier ldap.secret alternatif
sudoers_file=<path> fichier sudoers alternatif
sudoers_uid=<uid> UID du propiétaire du fichier sudoers
sudoers_gid=<gid> GID du fichier sudoers
sudoers_mode=<mode> mode du fichier sudoers en octal

Authentification Utilisateur

La stratégie de sécurité sudoers nécessite que la plupart des utilisateurs s'authentifient eux-même avant d'utiliser sudo. Un mot de passe n'est pas requis si l'utilisateur invoquant est root, si l'utilisateur cible est le même que l'utilisateur invoquant, ou si la startégie a désactivé l'authentification pour la commande utilisateur ou pour l'utilisateur. À la différence de su, quand sudoers exige une authentification, il valide les accréditifs de l'utilisateur, pas de l'utilisateur cible. Cela peut être changé avec rootpw, targetpw et runaspw.

Si un utilisateur qui n'est pas listé dans la stratégie tente de lancer une commande via sudo, un mail est envoyé aux autorités définies.

Noter qu'aucun mail n'est envoyé si un utilisateur non-autorisé tente de lancer sudo avec -l ou -v sauf en cas d'erreur d'authentification. Cela permet aux utilisateurs de déterminer s'ils sont autorisés ou non à utiliser sudo. Toute tentative d'utiliser sudo est loggé.

Si sudo est lancé par root, et que la variable SUDO_USER est définie, la stratégie sudoers utilise cette valeur pour déterminer qui est l'utilisateur actuel. Cela peut être utilisé par un utilisateur pour logger des commandes via sudo même quand un shell root a été invoqué.

sudoers utilise des fichiers horodatés par utilisateur pour le caching d'accrédifits. Une fois un utilisateur authentifié, un enregistrement est écrit contenant l'uid utilisé pour s'authentifier, l'id de session du terminal et un horodatage. L'utilisateur peut ainsi utiliser sudo sans mot de passe pour une courte période de temps. Par défaut, sudoers utilise un enregistrement séparé pour chaque tty, qui signifie que les sessions login de l'utilisateur sont authentifiés séparément.

Logging

sudoers peut logger toutes les tentatives, réussies ou non et les erreurs dans syslog, un fichier de log, ou les 2. sudoers est également capable de lancer une commande dans un pseudo-tty et logger toutes les entrées/sorties. L'entrée standard, la sortie standard, et l'erreur standard peuvent être loggés même s'ils ne sont pas associés avec un terminal. Le logging d'E/S n'est pas activé par défaut.

Environnement de commande

vu que les variables d'environnement peuvent influencer le comportement des programmes, sudoers fournis un moyen de restreindre les variables de l'environnement utilisateur hérités par la commande à lancer. Il y a 2 manières distincts de gérer ces variables.

Par défaut, l'option env_reset est activée, les commandes sont donc lancées dans un environnement minimal. Dans les systèmes Linux sans environnement PAM, l'environnement est initialisé avec le contenu de /etc/environment. Le nouvel environnement contient TERM, PATH, HOME, MAIL, SHELL, LOGNAME, USER, USERNAME et les variables SUDO_* en plus des variables de processus invoquant permis par env_check et env_keep. C'est une liste blanche de variables d'environnement. Les variables commençant pas () sont supprimées sauf si le nom et la valeur sont matché par env_keep ou env_check, vu qu'elles sont interprétées comme des fonctions par d'anciennes versions de bash

Si, cependant, l'option env_reset est désactivé, toute variable non explicitement refusée par env_check et env_delete sont héritées depuis le processus invoquant. Dans ce cas, env_check et env_delete deviennet comme une blacklist. Les variable commençant pas () sont toujours supprimées. Vu qu'il n'est pas possible de blacklister toutes les variables d'environnement potentiellement dangereuses, le mode env_reset est préférable.

Par défaut, les variables d'environnement sont matchés par leur nom. Cependant, si le motif inclus le signe '=', les noms des variables et leur valeur sont matchés.

La liste complète des variables d'environnement que sudo autorise ou refuse est contenue dans la sortie de sudo -V quand lancé en root.

Dans les systèmes qui supportent PAM où le module pam_env est activé pour sudo, les variables dans l'environnement PAM peuvent être fusionnées dans l'environnement. Si une variable dans l'environnement PAM est déja présente, la valeur est remplacée seulement si la variable n'était pas préservées par sudoers. quand env_reset est activé, les variables préservées depuis l'environnement de l'utilisateur invoquant par la liste env_keep a précédence sur l'environnement PAM. Quand env_reset est désactivé, les variables de l'environnement utilisateur ont précédence sur l'environnement PAM sauf en cas de match de la liste env_delete.

Un cas spécial, si -i est spécifié, sudoers initialise l'environnement sans regarder la valeur de env_reset. DISPLAY, PATH et TERM ne sont pas changés ; HOME, MAIL, SHELL, USER et LOGNAME sont basés sur l'utilisateur cible. Dans les systèmes Linux sans PAM, le contenu de /etc/environment est également inclus.

Finalement, restricted_env_file et env_files sont appliqués, si présents. Les variables dans restricted_env_file sont appliqués en premier, et sont sujets aux même restrictions que l'environnement de l'utilisateur invoquant. Les variables env_file sont appliquées en dernier et ne sont pas sujettes à ces restrictions. Dans tous les cas, les variables présentes dans les fichiers ne sont définis à leur valeur spécifiées que si elles ne sont pas en conflit avec une variable d'environnement existante

Format du fichier sudoers

Le fichier sudoers est composé de 2 types d'entrées : les alias et les utilisateurs. Quand plusieurs entrées correspondent à un utilisateur, elles sont appliquées dans l'ordre. En cas de multiple match, le dernier est utilisé.

Syntaxe EBNF

EBNF contient les opérateurs suivants :

- **?** Signifie que le symbole ou groupe de symbole précédent est optionnel, c'est à dire qu'il peut apparaître une fois ou pas du tout
- ***** Signifie que le symbole ou groupe de symbole précédent peut apparaître 0 ou plusieurs fois
- **+** Signifie que le symbole ou groupe de symbole précédent peut apparaître une ou plusieurs fois

Aliases

Il y a 4 types d'alias : User_Alias, Runas_Alias, Host_Alias et Cmnd_Alias :

```
Alias ::= 'User_Alias' User_Alias (':' User_Alias)* |
 'Runas_Alias' Runas_Alias (':' Runas_Alias)* |
 'Host_Alias' Host_Alias (':' Host_Alias)* |
 'Cmnd_Alias' Cmnd_Alias (':' Cmnd_Alias)*

User_Alias ::= NAME '=' User_List

Runas_Alias ::= NAME '=' Runas_List

Host_Alias ::= NAME '=' Host_List

Cmnd_Alias ::= NAME '=' Cmnd_List

NAME ::= [A-Z]([A-Z][0-9]_)*
```

Une liste d'utilisateur consiste de noms d'utilisateurs, UID (préfixé avec '#'), groupes système et GID (préfixés avec '%' et '%#'), netgroups (préfixés par '+'), groupes non-unix (préfixés avec '% :' et '% :#') et de User_Alias. Chaque liste peut être préfixée avec '!'. les " permettent d'éviter le besoin d'échapper les caractères spéciaux, sinon ils doivent être spécifiés en hexa (ex : \x20)

```
User_List ::= User |
  User ',' User_List

User ::= '!'* user name |
  '!'* #uid |
  '!'* %group |
  '!'* %#gid |
  '!'* +netgroup |
  '!'* %:nonunix_group |
  '!'* %:#nonunix_gid |
  '!'* User_Alias
```

Une liste Runas_Alias est similaire à une User_List.

```
Runas_List ::= Runas_Member |
  Runas_Member ',' Runas_List

Runas_Member ::= '!'* user name |
  '!'* #uid |
  '!'* %group |
  '!'* %#gid |
  '!'* %:nonunix_group |
  '!'* %:#nonunix_gid |
  '!'* +netgroup |
  '!'* Runas_Alias
```

Une liste d'hôtes est faite d'un ou plusieurs noms d'hôte, adresses IP, réseaux, +netgroups, et d'autres alias.

```
Host_List ::= Host |
  Host ',' Host_List

Host ::= '!'* host name |
  '!'* ip_addr |
  '!'* network(/netmask)? |
  '!'* +netgroup |
  '!'* Host_Alias
```

Une liste de commande est une liste d'un ou plusieurs nom de commandes, répertoires, et autres alias. Un nom de commande est un nom complet de fichier qui peut inclure des wildcard shell. Un simple nom de fichier permet de lancer une commande avec des arguments. Cependand, il est possible de spécifier les arguments (incluant des wildcard), ou "" pour indiquer que la commande peut seulement être lancée sans arguments. Un répertoire est un chemin complet se terminant par '/', et l'utilisateur pout exécuter tout fichier dans ce répertoire, mais pas dans les sous-répertoires

Si un Cmnd est associé avec des arguments de ligne de commande, les arguments dans Cmnd doivent matcher exactement avec ceux donnés par l'utilisateur. Noter que les caractères suivants doivent être échappés : ',', ' :', '=', '\'. La commande intégrée sudoedit est utilisée pour permettre à un utilisateur de lancer sudo avec -e. Il peut prendre des arguments. Noter que sudoedit est une commande intégrée dans sudo et doit être spécifié dans le fichier sudoers sans chemin.

```
digest ::= [A-Fa-f0-9]+ |
  [[A-Za-z0-9+/=]+

Digest_Spec ::= "sha224" ':' digest |
  "sha256" ':' digest |
  "sha384" ':' digest |
  "sha512" ':' digest

Cmnd_List ::= Cmnd |
  Cmnd ',' Cmnd_List
command name ::= file name |
  file name args |
  file name '""'

Cmnd ::= Digest_Spec? '!'* command name |
  '!'* directory |
  '!'* "sudoedit" |
  '!'* Cmnd_Alias
```

Si un nom de commande est préfixé avec un Hash, la commande ne match que si le hash SHA2 est vérifié. Les formats suivants sont supportés : sha224, sha256, et sha512. La chaîne peut être spécifiée en hexa ou en base64.

Default

Certaines options de configuration ont une valeur par défaut qui peut être changé en temps réel via une ou plusieurs lignes Default_Entry. Cela peut affecter tous les utilisateurs dans tous les hôtes, tous les utilisateurs sur un hôte spécifique, un utilisateur spécifique, une commande spécifique, ou des commandes lancées par un utilisateur spécifiques.

```
Default_Type ::= 'Defaults' |
  'Defaults' '@' Host_List |
  'Defaults' ':' User_List |
  'Defaults' '!' Cmnd_List |
  'Defaults' '>' Runas_List

Default_Entry ::= Default_Type Parameter_List

Parameter_List ::= Parameter |
  Parameter ',' Parameter_List

Parameter ::= Parameter '=' Value |
  Parameter '+=' Value |
  Parameter '-=' Value |
  '!'* Parameter
```

+= et -= sont utilisés pour ajouter ou supprimer des éléments d'une liste. Les entrées sont parsée dans l'ordre suivant : generic, host, user,

et runas, command. S'il y a plusieurs paramètres Default de même type, le dernier match est utilisé. Les paramètres Defaults suivant sont parsé avant tous les autres vu qu'ils affectent les entrées suivantes : fqdn, group_plugin, runas_default, sudoers_locale.

Spécification d'utilisateur

```
User_Spec ::= User_List Host_List '=' Cmnd_Spec_List \
  (':' Host_List '=' Cmnd_Spec_List)*

Cmnd_Spec_List ::= Cmnd_Spec |
  Cmnd_Spec ',' Cmnd_Spec_List

Cmnd_Spec ::= Runas_Spec? Option_Spec* Tag_Spec* Cmnd

Runas_Spec ::= '(' Runas_List? (':' Runas_List)? ')'

Option_Spec ::= (SELinux_Spec | Solaris_Priv_Spec | Date_Spec | Timeout_Spec)

SELinux_Spec ::= ('ROLE=role' | 'TYPE=type')

Solaris_Priv_Spec ::= ('PRIVS=privset' | 'LIMITPRIVS=privset')

Date_Spec ::= ('NOTBEFORE=timestamp' | 'NOTAFTER=timestamp')

Timeout_Spec ::= 'TIMEOUT=timeout'

Tag_Spec ::= ('EXEC:' | 'NOEXEC:' | 'FOLLOW:' | 'NOFOLLOW' |
  'LOG_INPUT:' | 'NOLOG_INPUT:' | 'LOG_OUTPUT:' |
  'NOLOG_OUTPUT:' | 'MAIL:' | 'NOMAIL:' | 'PASSWD:' |
  'NOPASSWD:' | 'SETENV:' | 'NOSETENV:')
```

Une spécification d'utilisateur détermine quelles commandes un utilisateur peut lancer (et sous quel utilisateur). dans les hôte spécifiés. Par défaut, les commandes sont lancées en root. La structure de base est **who where = (as_whom] what**

Runas_Spec

un Runas_Spec détermine l'utilisateur et/ou le groupe sous laquelle une commande peut être lancée. un Runas_Spec pleinement spécifié consiste de 2 Runas_List séparés par ' :' et entre parenthèses. Le premier Runas_List indique sous quels utilisateurs la commande peut être lancée (comme avec sudo -u). Le second définis une liste de groupes qui peuvent être spécifié via sudo -g. Si les 2 Runas_List sont spécifié, la commande peut être lancée avec n'importe quelle combinaison d'utilisateurs et groupes listés dans leur Runas_list respectifs. Si seul le premier est listé, la commande peut être lancée par n'importe quel utilisateur dans la liste mais -g ne peut pas être spécifié. Si le premier Runas_List est vide, mais le second est spécifié, tous les utilisateurs peuvent lancer la commande avec le groupe définis à un listé. Si les 2 sont vides, la commande ne peut être lancée que sous l'utilisateur invoquant. Si Runas_Spec n'est pas spécifiée, la commande peut être lancée en root et aucun groupe ne peut être spécifié.

l'utilisateur dbg peut lancer ces 3 commandes, mais seulememt comme operator :
dgb boulder = (operator) /bin/ls, /bin/kill, /usr/bin/lprm
Il est également possible de remplacer Runas_Spec dans une entrée :
dgb boulder = (operator) /bin/ls, (root) /bin/kill, /usr/bin/lprm
On peut l'étendre peut permettre à dbg de lancer /bin/ls avec soit l'utilisateur soit le groupe à operator :
dgb boulder = (operator : operator) /bin/ls, (root) /bin/kill, /usr/bin/lprm
Noter que bien que la portion groupe permet à l'utilisateur de lancer la commande avec ce groupe, il ne force pas l'utilisateur à le faire. Si aucun groupe n'est spécifié, la commande sera lancée avec le groupe listé dans l'entrée de la base de mot de passe de l'utilisateur. Exemple

de ce qui est permis par l'entrée ci-dessus :
sudo -u operator /bin/ls
sudo -u operator -g operator /bin/ls
sudo -g operator /bin/ls
Dans l'exemple suivant, l'utilisateur tcm peut lancer des commandes qui accèdent à un périphérique modem avec le groupe dialer
tcm boulder = (:dialer) /usr/bin/tip, /usr/bin/cu, /usr/local/bin/minicom
Seul le groupe est définis, la commande est lancée avec l'utilisateur tcm
sudo -g dialer /usr/bin/cu
Plusieurs utilisateurs et groupeg peuvent être présents dans un Runas_Spec, auquel cas l'utilisateur peut sélectionner une combinaison d'utilisateurs et groupes via -u et -g
alan ALL = (root, bin : operator, system) ALL
alan peut lancer une commande soit en root, soit bin, optionnellement en définissant le groupe operator ou system

Option_Spec

un Cmnd peut avoir 0 ou plusieurs options. En fonction du système, les options peuvent consister de rôle/types SELinux, ou timeouts. Une fois une option définis pour une Cmnd, les Cmnd suivantes dans Cmnd_Spec_List héritent de cette option, sauf remplacé par une autre option.

les règles sudoers peuvent spécifier une date de début et de fin via les paramètres NOTBEFORE et NOTAFTER. L'horodatage peut être spécifié en temps généralisé, rfc4517 (yyymmddHHMSSZ). Il est également possible de spécifier un offset en heures et minutes UTC. Par exemple '-0500' correspond à l'heure Eastern Standard aux US.

Une commande peut avoir un timeout. S'il expire avant que la commande ne se soit terminées, la commande est terminées. Le timeout peut être spécifié en jours, heures, minutes et secondes (ex : 7d8h30m10s, 14d, 8h30m, 600s, 3600).

Une commande peut avoir 0 ou plusieurs tags. Les valeurs de tag supportés sont : EXEC, NOEXEC, FOLLOW, NOFOLLOW, LOG_INPUT, NOLOG_INPUT, LOG_OUTPUT, NOLOG_OUTPUT, MAIL, NOMAIL, PASSWD, NOPASSWD, SETENV, et NOSETENV. Une fois un tag définis dans une Cmnd, les Cmnd suivants dans la Cmnd_Spec_List héritent de ce tag sauf si remplacé par le tag opposé.

[NO] EXEC autorise à lancer un exécutable lié dynamiquement
[NO] FOLLOW autorise à suivre les liens symboliques pour sudoedit uniquement
[NO] LOG_INPUT Remplace l'option log_input
[NO] LOG_OUTPUT Remplace l'option log_output
[NO] MAIL Remplace la valeur de l'option mail_all_cmnds. N'a pas d'effet avec -l ou -v
[NO] PASSWD Exige que l'utilisateur s'authentifie lui-même avant de lancer une commande
[NO] SETENV Remplace la valeur de l'option SETENV

Wildcard

sudo autorise les wildcard style shell dans les noms d'hôte, noms de chemins, et arguments de ligne de commande dans le fichier sudoers :

***** Matche un jeu de 0 ou plusieurs caractères
? Matche un simple caractère
[...] Matche un des caractères
[!...] Matche tout caractère non listé
\x pour spécifier les caractères '*',' ?','[' et ']'

Inclure d'autres fichiers

Il est possible d'inclure d'autres fichiers sudoers avec les directives #include et #includedir.

Autres caractères spéciaux et mots réservés

'#' est utilisé pour indiquer un commentaire (excepté pour #include et #includedir ou dans le contexte d'un nom d'utilisateur suivi par un ou plusieurs chiffres).

Le mot réservé ALL est un alias intégré qui implique qu'un match réussi toujours.

' !' peut être utilisé comme opérateur NOT dans une liste.

Options sudoers

sudo peut être modifié par des lignes Default_Entry :

always_query_group_plugin (bool) si un group_plugin est configuré, l'utilise pour résoudre les groupe sous la forme %group tant que ce n'est pas également un groupe système. Normalement, seules les groupes sous la forme %#group sont passé au group_plugin

always_set_home (bool) activé, sudo définis HOME au répertoire home de l'utilisateur cible

authenticate (bool) définis, les utilisateurs doivent s'authentifier eux-même avant de lancer des commandes

closefrom_override (bool) définis, l'utilisateur peut utiliser sudo -C qui remplace le point de départ par défaut auquel sudo commence à fermer les descripteurs de fichier.

compress_io (bool) définis, si sudo est configuré pour logger l'entrée ou la sortie d'une commande, les logs I/O sont compressés avec zlib.

exec_background (bool) par défaut, sudo lance une commande en foreground tant que sudo est lui-même en foreground. à on et la commande lancé dans un pty (dû au logging I/O ou du flag use_pty), la commande est lancées en tâche de fond.

env_editor (bool) Définis, visudo utilise la valeur de EDITOR ou VISUAL comme liste d'éditeur par défaut

env_reset (bool) définis, lance la commande dans un environnement minimal

fast_glob (bool) utilise la fonction fnmatch(3) au lieu de glob(3) pour le globbing, ce qui est plus rapide, mais n'est pas capable de matcher les chemins relatifs

fqdn (bool) place les noms ftdn dans le fichiers sudoers quand le nom d'hôte local ne contient pas de nom de domaine.

ignore_audit_errors (bool) autorise à lancer les commandes même si sudoers ne peut écrire dans le log d'audit

ignore_dot (bool) définis, ignore "." ou "" dans la variable PATH

ignore_iolog_errors (bool) autorise à lancer des commandes même si sudoers ne peut écrire dans le fichier de log E/S

ignore_logfile_errors (bool) autorise à lancer des commandes même si sudoers ne peut écrire dans le fichier de log

ignore_local_sudoers (bool) Si définis via LDAP, ne parse pas /etc/sudoers

ignore_unknown_defaults (bool) ne produit pas d'alerte si une entrée Defaut inconnue est rencontrée, ou une options sudoOptions dans LDAP.

insults (bool) insulte les utilisateurs quand ils entrent un mot de passe incorrect

log_host (bool) Définis, le nom d'hôte est loggé

log_input (bool) Définis, log l'entrée utilisateur

log_output (bool) Définis, log la sortie envoyée à l'écran

log_year (bool) Définis, log l'année au format yyyy

long_otp_prompt (bool) En validant un schéma OTP, un prompt à 2 lignes est utilisé pour simplifier le copier/coller du challenge.

mail_all_cmnds (bool) Envoie un mail à chaque tentative d'exécution d'une commande (excepté avec -l ou -v et une authentification réussie)

mail_always (bool) envoie un mail à chaque fois qu'un utilisateur lance sudo.

mail_badpass (bool) Envoie un mail en cas de mot de passe incorrect

mail_no_host (bool) définis, envoie un mail si l'utilisateur invoquant existe dans le fichier sudoers mais n'est pas autorisé à lancer les commandes dans l'hôte courant.

mail_no_perms (bool) Définis, envoie un mail si l'utilisateur invoquant est autorisé à utiliser sudo, mais n'a pas l'autorisation de lancer la commande

mail_no_user (bool) Définis, envoie un mail si l'utilisateur invoquant n'est pas dans le fichier sudoers

match_group_by_gid (bool) par défaut, sudoers recherche chaque groupe dont l'utilisateur est membre par son GID pour déterminer le nom du groupe. Dans les systèmes ou la recherche des groupes est longue, cette option évite de résoudre les GID en noms de groupe

netgroup_tuple (bool) Définis, la recherche de netgroup est effectuée en utilisant les triplets netgroup

noexec (bool) Définis, toutes les commande lancées via sudo fonctionne comme avec le tag NOEXEC

pam_session (bool) Créé une nouvelle session PAM pour la commande à lancer

pam_setcred (bool) tente d'obtenir les accréditifs de l'utilisateur cible

passprompt_override (bool) Demande le mot de passe spécifié par passprompt est seulement utilisé si le prompt fournis par PAM match la chaîne "Password :". si definis, passprompt est toujours utilisé.

path_info (bool) informe quand la commande ne peut pas être trouvée dans le PATH.

preserve_groups (bool) par défaut, sudo initialise le vecteur de groupe à la liste des groupes de l'utilisateur cible. Définis, le vecteur de groupe de l'utilisateur n'est pas altéré. Le GID réel et effectif continuent de matcher l'utilisateur cible

pwfeedback (bool) Fournis un retour visuel en tapant un mot de passe

requiretty (bool) définis, sudo n'est lancé que si l'utilisateur est loggé dans un vrai tty.

root_sudo (bool) Définis, root est autorisé à lancer sudo. Désactivé, empêche sudoedit

rootpw (bool) Définis, sudo demande le mot de passe root au lieu du mot de passe de l'utilisateur invoquant

runaspw (bool) Définis, sudo demande le mot de passe de l'utilisateur définis par runas_default au lieu du mot de passe de l'utilisateur invoquant

set_home (bool) Définis et sudo invoqué avec -s, HOME est définis au home de l'utilisateur cible

set_logname (bool) Définis LOGNAME. Cependant pour certains programmes (incluant RCS) utilisent LOGNAME pour déterminer la vrai identité de l'utilisateur.

set_utmp (bool) Activé, sudo créé une entrée dans utmp ou utmpx quand un pseudo tty est alloué.

setenv (bool) Autorise l'utilisateur à désactiver l'option env_reset avec l'option -E

shell_noargs (bool) Définis et sudo est invoqué sans arguments, agit comme avec -s : lance un shell en root

stay_setuid (bool) Normalement, quand sudo exécute une commande, les UID réel et effectifs sont définis à l'utilisateur cible. Cette option laisse l'UID réel à l'utilisateur invoquant

sudoedit_checkdir (bool) Définis, sudoedit vérifie tous les composants répertoire du chemin à éditer pour vérifier s'il est en écriture par l'utilisateur. Les liens symboliques ne sont pas suivis et sudoedit refuse d'éditer un fichier localisé dans un répertoire accessible en écriture. Non effectif pour root.

sudoedit_follow (bool) Suit les liens symboliques en ouvrant les fichiers.

targetpw (bool) Définis, demande le mot de passe de l'utilisateur spécifié par -u au lieu du mot de passe de l'utilisateur invoquant

tty_tickets (bool) Définis, les utilisateurs doivent s'authentifier pour chaque tty

umask_override (bool) Définis, sudo définis le umask spécifié dans le fichier sudoers sans modification

use_netgroups (bool) Définis, les netgroups peuvent être utilisé à la place d'un utilisateur ou hôte

use_pty (bool) Définis, sudo lance la commande dans un pseudo-tty même si aucun plugin I/O n'est définis

user_command_timeouts (bool) Définis, l'utilisateur peut spécifier un timeout sur la ligne de commande

utmp_runas (bool) Définis, sudo stocke le nom de l'utilisateur runas en mettant à jours utmp au lieur du nom de l'utilisateur invoquant

visiblepw (bool) par défaut, sudo refuse de s'exécuter si l'utilisateur doit entrer un mot de passe mais qu'il n'est pas possible de désactiver l'écho dans le terminal. Cela permet de lancer des commande comme "ssh somehost sudo ls" vu que ssh n'alloue pas de tty par défaut en lançant une commande.

closefrom (int) Avant d'exécuter une commande, sudo ferme tous descripteur de fichier ouvert autre que 0-2. Spécifie un fd à partir duquel fermer les fd.

command_timeout (int) Délai maximum pour l'exécution d'une commande.

maxseq (int) Numéro de séquence maximum qui est substitué pour l'échappement "%{seq}" dans le fichier de log I/O.

passwd_tries (int) Nombre maximum de tentative pour entrer en mot de passe

syslog_maxlen (int) Par défaut, sudoers créé des messages de log jusqu'à 980 octets. Pour éviter de tronquer les messages, sudoers split les message supérieurs à syslog_maxlen.

loglinelen (int) Nombre de caractères par ligne de log. n'a pas d'effet pour syslog.

timestamp_timeout (int) Nombre de minutes avant de redemander un mot de passe.

umask (int) umask pour la création des fichier

badpass_message (string) Message affiché si un utilisateur entre un mot de passe incorrect

editor (string) Liste d'éditeurs autorisé pour visudo.

iolog_dir (string) Répertoire parent où construire le nom de chemin pour le répertoire de log I/O. Seulement utilisé si log_input et log_output sont activés. Les échappements suivants sont permis :

%{seq} Étendu pour augmenter le numéro de séquence en base 36

%{user} Étendu au login de l'utilisateur invoquant

%{group} Étendu au GID réel de l'utilisateur invoquant

%{runas_user} Étendu au login de l'utilisateur qui exécute la commande

%{runas_group} Étendu au groupe qui exécute la commande

%{hostname} Étendu au nom de l'hôte local sans le nom de domaine

%{command} Étendu au nom de la commande lancée

iolog_file (string) Chemin du fichier de log, relatif à iolog_dir pour les logs I/O. Accepte les séquences d'échappement de iolog_dir

iolog_flush (string) Définis, sudo vide les données de log sur disque après chaque écriture au lieu de le même en tampon. Permet de voir les logs en temps réel.

iolog_group (string) Nom du groupe pour la créations des répertoires et fichiers de log I/O

iolog_mode (string) mode à utiliser pour créer les fichiers de log I/O

iolog_user (string) Propriétaire des répertoires et fichiers de log I/O

lecture_status_dir (string) Répertoire dans lequel sudo stocke les fichiers de status par utilisateur. Défaut : /var/adm/sudo/lectured

mailsub (string) Sujet du mail envoyé. %h étend au nom d'hôte

pam_login_service (string) Nom du service utilisé avec -i. Défaut : sudo

pam_service (string) Nom du service que PAM applique. Défaut : sudo

passprompt (string) Prompt par défaut à utiliser en demandant le mot de passe. Les échappements suivants sont supportés :

%H Nom d'hôte local fqdn

%h Nom d'hôte local

%p utilisateur correspondant au mot de passe demandé

%U Nom de l'utilisateur cible

%u nom de l'utilisateur invoquant

%% le caractère %

role (string) Rôle SELinux par défaut à utiliser en construisant un nouveau contexte de sécurité pour lancer la commande.

runas_default (string) Utilisateur par défaut pour lancer les commande.

sudoers_locale (string) Locale à utiliser pour parser le fichier sudoers, les commandes de logging, et l'envoie de mail.

timestampdir (string) Répertoire dans lequel sudo stocke ses fichiers d'horodatage.

timestampowner (string) Propriétaire du répertoire de status de lecture, d'horodatage et tous les fichiers qui y sont stockés.

type (string) Type SELinux par défaut pour construire un nouveau contexte SELinux

env_file (string) Spécifie le chemin complet d'un fichier contenant les variables d'environnement à définir pour le programme à lancer.

exempt_group (string) Les utilisateurs dans ce groupe sont exemptés du mot de path et des exigences PATH.

fdexec (string) Détermine si sudo exécute une commande par son chemin ou par un fd ouvers. always|never exécute ou non par fd, digest_only n'exécute par fd que si la commande a un digest associé.

group_plugin (string) plugin group sudoers avec des arguments optionnels.

lecture (string) Contrôle quand une lecture courte est affichée avec le prompt. always|never|once.

lecture_file (string) Fichier contenant le message à afficher au lieu du message embarqué

listpw (string) Contrôle quand un mot de passe est requis quand un utilisateur lance sudo -l :

all Toutes les entrées de l'utilisateur pour l'hôte courant dans le fichier sudoers doivent avoir le flag NOPASSWD pour éviter d'entrée le mot de passe

always L'utilisateur doit entrer un mot de passe pour utiliser -l

any Au moins une entrée de l'utilisateur dans sudoers doit avoir le flag NOPASSWD pour ne pas entrer de mot de passe

never L'utilisateur ne doit jamais entrer un mot de passe pour utiliser l'option -l

logfile (string) Chemin du fichier de log sudo.

mailerflags (string) flags à utiliser en invoquant mailer Défaut : -t

mailerpath (string) Chemin du programme pour envoyer les mails.

mailfrom (string) Adresse du champ From

mailto (string) Adresse de destination de mail

restricted_env_file (string) Fichier contenant les variables à définir dans l'environnement. Ces variables de manière similaires à l'environnement de l'utilisateur invoquant.

secure_path (string) Chemin de toutes les commandes lancées par sudo

syslog (string) facilité syslog pour les logs

syslog_badpri (string) Priorité sysloge quand l'utilisateur n'est pas autorisé à lancer une commande ou en cas d'echec de l'authentification

syslog_goodpri (string) Priorité syslog à utiliser quand l'utilisateur est autorisé à lancer une commande et que l'authentification a réussie.

verifypw (string) Contrôle quand un mot de passe est requis avec sudo -v :

all Toutes les entrées dans le fichiers sudoers doivent avoir NOPASSWD pour éviter d'entrer un mot de passe.

always L'utilisateur doit toujours entrer sont mot de passe pour utiliser l'option -v

any Au moins une entrée dans le fichier sudoers pour l'utilisateur doit avoir le flag NOPASSWD pour éviter d'entrer un mot de passe

never L'utilisateur n'a jamais besoin d'entrer un mot de passe

env_check (string) Variables d'environnement à supprimer sauf si elles sont sûres. Pour toutes les variables excepté TZ, sûre signifie que la valeur de la variable ne contient pas de '%' ou '/'

env_delete (string) Variables d'environnement à supprimer de l'environnement de l'utilisateur quand env_reset n'est pas effectif.

env_keep (string) Variables d'environnements à préserver de l'environnement de l'utilisateur quand env_reset est effectif

Plugins group

sudoers supporte sa propre interface de plugin pour permettre de rechercher des groupes non-Unix dans une source autre que la base de groupe Unix standard. Celà permet d'implémenter la syntaxe nonunix_group.

Les plugins group sont spécifiés via le paramètres group_plugin. Les plugins group suivants sont installés par défaut :

group_file.so Supporte un fichier de groupes alternatif qui utilise la même syntaxe que /etc/group. Le chemin du fichier doit être spécifié en arguments

system_group.so Supporte la recherche de groupe via getgrnam() et getgrid(). Il peut être utilisé dans le cas où des utilisateurs appartiennent à des groupes non-présents dans le vecteur groupe supplémentaire de l'utilisateur. Ce plugin n'a pas d'options

Format de log

sudoers peut logger des évènements en utilisant syslog(3) ou un simple fichier. Le format est pratiquement identique dans les 2 cas.

Les commandes que sudo lance sont loggés en utilisant le format suivant :

```
date hostname progname: username : TTY=ttyname ; PWD=cwd ; USER=runasuser ; GROUP=runasgroup ; TSID=logid ;
ENV=env_vars COMMAND=command
```

date Date et heure à laquelle la commande a été lancée au format "MMM,DD,HH :MM :SS". Via syslog, le format de date est contrôllé par syslog. log_year permet d'inclure l'année

hostname Le nom de l'hôte où sudo a été lancé. Seulement présent via syslog

progname Le nom du programme, généralement sudo ou sudoedit. Seulement présent via syslog

username Nom de login de l'utilisateur invoquant

ttyname Nom court du terminal

cwd Répertoire de travail courant

runasuser Utilisateur sous lequel la commande est lancée

runasgroup Groupe sous lequel la commande est lancée

logid Identifiant de log I/O qui peut être utilisé pour rejouer la sortie des commande (log_input ou log_output doit être activé)

env_vars variables d'environneent spécifiés sur la ligne de commande.

command La commande exécutée

Entrées de log des commandes refusées

Si l'utilisateur n'est pas autorisé à lancer la commande, la raison du refus suit le nom de l'utilisateur. Les raisons possibles sont :

user NOT insudoers L'utilisateur n'est pas listé dans le fichie sudoers

user NOT authorized on host L'utilisateur est listé dans le fichier sudoers mais n'est pas autorisé à lancer les commandes sur cet hôte

command not allowed L'utilisateur est listé dans le fichier sudoers mais n'est pas autorisé à lancer la commande spécifiée

3 incorrect password attemps L'utilisateur a échoué "passwd_trie" fois

a password is required sudo -n a été spécifié, mais un mot de passe est requis

sorry, you are not allowed to set the following environment variables L'utilisateur a spécifié des variables d'environnement sur la ligne de commande qui ne sont pas autorisés par sudoers

Entrées de log d'erreur

Si une erreur se produit, sudoers log un message et, dans la plupart des cas, envoie un message à l'administrateur local via email. Les erreurs possibles incluent :

parse error in /etc/sudoers near line N Erreur dans la configuration

problem with defaults entries Options Defaults inconnus. N'empêche pas sudo de fonctionner

timestamp owner (username) : No such user l'utilisateur spécifié dans timestampowner n'a pas été trouvé dans la base de compte

unable to open/read /etc/sudoers le fichier sudoers n'est pas accessible

unable to open/read /etc/sudoers Le fichier /etc/sudoers n'existe pas

/etc/sudoers is not a regular file Le fichier existe mais n'est pas un fichier régulier ou un lien symbolique

/etc/sudoers is owned by uid N, should be 0 Le fichier n'est pas possédé par root

/etc/sudoers is world writable Le fichier est accessible en écriture par tout le monde. Normalement il devrait être 0440

/etc/sudoers is owned by gid N, should be 1 le fichier n'a pas le bon groupe.

unable to open /var/run/sudo/ts/username sudoers n'arrive pas à lire ou créer le fichier d'horodatage pour l'utilisateur. Le répertoire parent doit être 0711

unable to write to /var/run/sudo/ts/username sudoers n'arrive pas à écrire dans le fichier d'horodatage de l'utilisateur

/var/run/sudo/ts is owned by uid X, should be Y Le répertoire ts est possédé par un utilisateur autre que timestampowner.

/var/run/sudo/ts is group writable Le répertoire ts est accéssible en écriture par le groupe. Doit être 0700

Notes sur syslog

Par défaut, sudoers log les messages via syslog. date, hostname et progname sont ajoutés par syslog, et non sudoers. ces informations peuvent donc varier d'un système à l'autre. La taille maximum des messages syslog varie également d'un système à l'autre. syslog_maxlen peut être utilisé pour changer la valeur par défaut de 980octets.

Notes sur le fichier de log

Si logfile est activé, sudoers log dans un fichier local. sudoers utilise un format similaire à syslog, avec quelques différences importantes :

- progname et hostname ne sont pas présents
- si log_year est activé, l'année est incluse
- Les lignes plus longue que loglinelen sont coupées et continées sur la ligne suivantes

Fichier de log d'E/S

Quand le logging I/O est activé, sudo lance la commande dans un pseudo-tty et log toutes les E/S de l'utilisateur dans "iolog_dir" en utilisant un ID de session unique qui set inclus dans les logs, préfixé par "TSID=". Chaque log I/O est stocké dans un répertoire séparé qui contient les lignes suivantes :

log Un fichier text contenant la date à laquelle la commande a été lancée, le nom de l'utilisateur invoquant, le nom de l'utilisateur cible, le nom du groupe cible, le terminal, le nombre de lignes et colonnes du terminal, le répertoire courant où la commande a été lancée et le chemin de la commande avec ses arguments.

timing Un log de la quantité de temps, et le nombre d'octets, entre chaque entrées de log I/O.

ttyin Entrée depuis le tty de l'utilisateur

stdin Entrée depuis un pipe ou un fichier

ttyout Sortie du pseudo-tty

stdout Sortie standard dans un pipe ou redirigé dans un fichier

stderr Erreur standard vers un pipe ou redirigé dans un fichier

Tous les fichiers autre que log sont compressé au format gzip sauf si compress_io est désactivé. À cause du tampon, il n'est pas possible d'afficher les logs I/O en temps réel sauf si iolog_flush est utilisé.

Vu que le log E/S de chaque session utilisateur est stocké dans un répertoire séparé, les utilitaires de rotation de log traditionnels ne peuvent pas être utilisés pour limiter le nombre de logs E/S. La manière la plus simple est d'utiliser l'option maxseq. Une fois la sequence atteinte, il est réinitialisé à 0 est sudoers tronque et réutilise les logs E/S existants.

Fichiers

/etc/sudo.conf Fichier de configuration de sudo

/etc/sudoers Fichier de stratégie de sudoers

/etc/group Fichier de groupes locaux

/etc/netgroup Liste des netgroups

/var/log/sudo-io Fichiers de log d'E/S

/var/run/sudo/ts Répertoire contenant les timestamps pour la stratégie de sécurité sudoers

/var/adm/sudo/lectured Répertoire contenant les fichiers de status de lecture pour la stratégie de sécurité sudoers

/etc/environment Environnement initial pour les système sans PAM

Notes de sécurité

- l'opération ' !' ne doit pas être utilisé pour soustraire des commandes depuis AL, vu qu'il n'empêche pas d'exécuter ces commandes via un autre nom.
- fast_glob ne permet pas d'utiliser ' !' correctement quand le chemin inclus du globbing.

Empêcher les échappements du shell

Quand sudo exécute un programme, ce programme est libre de faire ce qu'il veut, incluant de lancer d'autres programmes. Cela peut être un problème de sécurité vu qu'il n'est pas commun pour un programme d'autoriser les échappements du shell, qui laisse un utilisateur bypasser le contrôle d'accès à sudo. Les programmes communs qui permettent les échappement du shell incluent les éditeurs, paginateurs, mails et terminaux. Il y a 2 approches à ce problème :

restrict Éviter de donner accès aux commande qui autorise un utilisateur à lancer des commandes arbitraires. De nombreux éditeurs ont des modes restreints qui désactive les échappements, bien que sudoedit est une meilleur solution aux éditeurs lancés via sudo.

noexec De nombreux systèmes qui supportent les librairies partagées ont la capacité de remplacer les fonctions par défaut en pointant une variable d'environnement (ex LD_PRELOAD) vers une librairie alternative. Dans de tels systèmes, la fonctionnalité noexec de sudo peut être utilisée pour éviter de lancer un programme qui exécute d'autres programmes.

Édition sécurisée

Le plugin sudoers inclus sudoedit qui permet aux utilisateurs d'éditer les fichiers de manière sécurisée avec l'éditeur de leur choix. sudoedit est une commande intégrée et doit être spécifié dans le fichier sudoers dans chemin. Cependant, il peut prendre des arguments de ligne de commande. À la différence des autres commandes sudo, l'éditeur est lancé avec les permissions de l'utilisateur invoquant et avec l'environnement non-modifié.

Les utilisateurs ne devraient jamais avoir les permissions sudoedit pour éditer un fichier qui réside dans un répertoire auquel l'utilisateur a accès en écriture, soit directement, ou via wildcard. Si l'utilisateur à ces droits, il peut remplacer le fichier avec un lien vers un autre fichier. Pour éviter cela, les liens symbolique ne sont pas suivis dans les répertoires en écriture (option sudoedit_checkdir).

Vérification du fichier timestamp

sudoers vérifie le propriétaire de son répertoire d'horodatage et ignore le contenu du répertoire s'il n'est pas possédé par root, ou s'il est accessible en écriture par un utilisateur non-root.

Bien que ce répertoire devrait être effacé au reboot, tous les systèmes ne contiennent pas de répertoire /var/run. Pour éviter de potentielles problèmes sudo ignore les fichiers timestamps antérieur au boot système.

Certains systèmes avec des environnements graphiques permettent à des utilisateurs non-privilégiés de changer l'heure système. Vu que sudoers s'assure de l'horloge système pour la validation des horodatages, il peut être possible dans de tels systèmes qu'un utilisateur qui lance sudo pour une durée supérieur à timestamp_timeout en reculant l'horloge. Pour combattre cela, sudoers utilise une horloge monolitique pour ses horodatages si le système le supporte.

Vu que les fichiers d'horodatage résident dans le système de fichier, ils peuvent survire à une session utilisateur. En résultat, un utilisateur peut être capable de se logger, lancer une commande avec sudo après s'être authentifié, se déconnecter, se reconnecter, et lancer sudo sans s'authentifier tant que l'horodatage est dans les 5 minutes. Quand l'option tth_tickets est activé, l'horodatage inclus le numéro de périphérique du terminal de l'authentification. Cela fournis une granularité par tth mais les horodatages continue à survivre entre les sessions. L'enregistrement d'horodatage inclus également l'ID de session du processus qui a été authentifié en dernier. Cela empêche les processus dans des session terminal différents d'utiliser le même enregistrement d'horodatage. Cela aide également à réduire les chances qu'un utilisateur soit capable de lancer sudo sans entrer un mot de passe en se déconnectant et revenant dans le même terminal.

Debuggage

Le plugin sudoers support un framework de debuggage qui peut aider à suivre ce qu'il fait en interne s'il y a un problème. Cela peut être configuré dans le fichier sudo.conf

sudoers utilise le même format de flag de debug que sudo. "subsystem@priority". Les priorités sont crit, err, warn, notice, diag, info, trace, debug. Les sous-systèmes suivant sont utilisés par sudoers :

alias traitements User_Alias, Runas_Alias, Host_Alias et Cmnd_Alias
all Match tous les sous-systèmes
audit Code d'audite Linux
auth Authentification utilisateur
defaults Paramètres Defaults
env Gestion de l'environnement
ldap sudoers basé sur LDAP
logging Support du logging
match Match des utilisateurs, groupes, hôtes et netgroups dans le fichier sudoers
netif Gestion des interfaces réseaux
nss Gestion nss dans sudoers
parser Parsing de fichier sudoers
perms Définition des permissions
plugin Équivalent de main pour le plugin
pty Code lié au pseudo-tty
rbtree redblack tree interne
sssd Sudoers basé sur sssd
util Fonctions utilitaires