Audit

Audit ((https://github.com/linux-audit) està format per un mòdul oficial del kernel i per un conjunt de comandes d'usuari que recullen la informació sobre els events i crides al sistema detectades per aquest mòdul i l'emmagatzemen per tal de poder estudiar-les (a posteriori o en temps real). Per tant, el podem fer servir per analitzar el que està passant al nostre sistema en gran detall i, a partir d'aquí, prendre les mesures oportunes en relació a la seguretat d'aquest.

> NOTA: Si es volgués (cosa gens recomanable) desactivar el mòdul Audit, la manera més senzilla seria indicar el paràmetre del kernel audit=0 (normalment a la secció adient de l'arxiu de configuració del gestor d'arranc utilitzat)

La informació que Audit és capaç de recollir és àmplia, tota basada en la interceptació de determinades crides al sistema: pot monitoritzar accesos i canvis de continguts i/o permisos en fitxers i carpetes concretes (com ara, per exemple, /etc/shadow), el comportament de sistemes d'autenticació (com ara els intents d'inici de sessió -fallits i/o exitosos- via GDM, TTY, SSH, Kerberos i altres), els intents de connexions de xarxa, les comandes que pugui haver executat un determinat usuari, els intents d'execució o parada d'aplicacions i serveis o de crides al sistema concretes (en aquest cas, per exemple, es pot observar si s'han cridat a les funcions *settimeofday()* o *clock_adjtime()* per deduïr si s'ha volgut canviar la data del sistema, etc), etc.

Cal tenir en compte que hi ha determinats events (com per exemple els de tipus 1100-1299, 1326, 1328, 1331 o majors...veure més avall) sempre són recollits per defecte per Audit sense haver de configurar res. Aquests events (entre els quals es troben, per exemple, els de registre d'inici i tancament de sessions d'usuari) són d'obligada existència per complir amb tots els estàndars gubernamentals de seguretat. A partir d'aquí, si es volen recollir encara més events (o si no volem registrar algun dels "per defecte", cosa gens recomanable), caldrà indicar-los explícitament a Audit mitjançant la configuració de "regles" que estudiarem de seguida.

Els programes d'usuari que podem "acoplar com a sondes" al mòdul Audit del kernel per recollir els events i crides detectades (tant per les regles que haguem definit com per ser d'algun tipus recollit per defecte) són vàries (i és millor que només es faci servir una per no "molestar-se" entre sí):

1.-Tenim el socket "systemd-journald-audit.socket". Podem definir un "socket" Systemd com una espècie de servei Systemd que està permanentment funcionant però que només utilitza la CPU quan rep alguna entrada (en aquest cas, un event/crida interceptada pel kernel) i seguidament torna a passar a "stand-by". A la pràctica, es gestiona com un servei: *systemctl {start | stop | status | enable | disable } systemd-journald-audit.socket* , etc. Aquest socket el que fa és reenviar al Journald tot allò que rebi del mòdul Audit del kernel. Per tant, el Journald serà el nostre lloc de consulta.

Per filtrar els missatges provinents del mòdul Audit, es pot executar la comanda *journalctl _TRANSPORT=audit* (afegint *-f* si es vol en temps real). El paràmetre _TRANSPORT el que indica és de quina manera ha arribat el missatge al Journald...hi ha vàries maneres: la nativa utilitzada pels programes moderns ("journal"), la clàssica utilitzada pels programes més antics ("syslog"), els missatges pròpiament del kernel ("kernel"), etc. Per veure tots els transports possibles reconeguts per la versió de Journald del nostre sistema es pot executar *journalctl --field _TRANSPORT*

Si es volen filtrar missatges provinents del mòdul Audit que es corresponen a algun event en concret, els missatges gravats al Journald disposen del camp _AUDIT_TYPE_NAME que podem utilitzar per això ja que aquest camp indica el tipus d'event detectat i pot valer alguns dels següents valors llistats https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-audit_record_types Per exemple, la comanda *journalctl _AUDIT_TYPE_NAME=USER_LOGIN _TRANSPORT=audit* mostraria tots els intents (fallits i reixits) d'inici de sessió, ja sigui a través de GDM, SSH, TTY, etc (no recull les accions de *su* o *sudo* perquè es corresponen a altres events) També existeix el camp *_AUDIT_TYPE* però val un codi numèric de l'event en qüestió que cal conèixer (els valors 1100-1299, 1326, 1328, 1331...que parlàvem en paràgrafs anteriors).

Systemd-journald-audit no proporciona cap forma d'indicar regles personalitzades per registrar els events que volguem (més enllà dels que ja registra per defecte obligatòriament per les normatives gubernamentals). Per això necessitaríem les eines client proporcionades pel paquet explicat a continuació.

2.-Tenim el paquet "auditd" (a Ubuntu) i "audit" (a Fedora) que instal.la un servei concret (*systemctl start auditd* o *systemctl start audit*, respectivament) i unes quantes eines client. Si es farà servir aquest servei llavors és millor deshabilitar completament el "socket" systemd-journald-audit.socket (així, *systemctl mask systemd-journald-audit.socket*). Aquest servei guarda tots els events i crides recollits del mòdul Audit en un arxiu de disc (per defecte, "/var/log/audit/audit.log") en comptes de en el Journald i té "/etc/audit/auditd.conf" com a fitxer de configuració, el qual inclou les següents opcions (entre d'altres; per veure'n més, consulteu *man auditd.conf*).

log_file : the file audit subsystem logs will be saved to
log_format : the format logs will be saved in
freq : the maximum number of records that can be saved in the buffer
flush : the mode for synchronizing the buffer with the disk ("none" = do nothing; "incremental" = transfers data from the buffer to the hard disk at the frequency set in the freq parameter; "data" = constant synchronization of data; "sync" = synchronize both data and metadata files when writing to the disk)
max_log_file : the maximum log file size in megabytes
max_log_file_action : the action to be taken when the maximum log file size is reached ("keep_logs" = prevents Audit log files from being overwritten)
space_left : the minimum amount of free disk space in megabytes; this amount will trigger the following parameter
space_left_admin : the action to be taken when the minimum allotted disk space is left ("ignore" = do nothing; "syslog" = issues a warning to syslog; "email" = sends notification via email; "suspend" = stops writing records to the disk; "single" = switch to single-user mode; "halt" = shuts down the system)
disk_full_action : the action to be taken when the disk is full (this parameter can take the same values as space_left_admin). A similar option is disk_error_action

Per seleccionar quins events/crides en concret seran guardades al fitxer "audit.log" (a més dels events concrets automàticament registrats degut al que ja hem comentat de les normatives gubernamentals) s'han de crear "regles". Això es pot fer "al vol" mitjançant la comanda *auditctl* (que ve dins del mateix paquet amb el què s'ha instal.lat el servei auditd). Algunes de les opcions d'aquesta comanda són:

auditctl -l : mostra totes les regles personalitzades carregades actualment

auditctl -D : esborra l'efecte de totes les regles personalitzades carregades actualment

auditctl -a {task|exit|exclude},{always|never} [-S nomCrida] [...] [-F filtre] [...] : afegeix una regla per monitoritzar crides al sistema

NOTA: "task" indica que la regla monitoritzarà els events relacionats amb la creació de nous processos (no es fa servir gaire), "exit" indica que la regla monitoritzarà tots els events que passen en executar-se (i finalitzar) la crida al sistema en qüestió (normalment això és el que ens interessarà quasi sempre), "exclude" indica quines crides al sistema no es volen monitoritzar explícitament (aquí podríem indicar que no volem registrar algun dels events automàticament registrats per normatives gubernamentals)

NOTA: "always" indica que l'event especificat es vol guardar al log i "never" indica que no (en el cas de fer servir el tipus "exclude" dóna igual el valor que s'indiqui perquè sempre s'entendrà que és "never")

NOTA: El paràmetre -S indica el nom de la crida al sistema que es vol monitoritzar (*openat, unkink,rmdir,* etc). Es poden indicar vàries crides, cadascuna en un paràmetre -S diferent. Si no s'indica cap, llavors el filtre (si és que es vol fer servir algun) vindrà donat pel/s paràmetre/s -F. Exemples de filtres pel paràmetre -F són:

-F path=/ruta/carpeta(ofitxer) : observa tot el que tingui a veure amb el fitxer o carpeta indicat
-F dir=/ruta/carpeta : observa tot el que tingui a veure amb la carpeta indicada de forma recursiva

-F perm=aw : observa tot el que tingui a veure amb canvi d'atributs en fitxers o carpetes ("a"), lectures de fitxers o carpetes ("r"), escriptures ("w") o execucions ("x")
-F auid>=1001 : observa tot el que tingui a veure amb el UID (o nom d'usuari, també) indicat
-F msgtype!=CWD : observa tot el que no tingui a veure amb els missatge del tipus indicat (on "tipus" pot valer qualsevol dels valors _AUDIT_TYPE_NAME vistos amb "systemd-journald-audit.socket")
-F exe=/ruta/programa : observa tot el que tingui a veure amb el programa indicat
-F exit=ERROR : observa tot el que tingui a veure amb el valor de retorn indicat de crides al sistema
Per saber més filtres possibles (*pid, inode, filetype, gid, success,* etc), consultar la pàgina del manual de *auditctl*

NOTA: Es pot afegir el paràmetre *-k etiqueta* per afegir un identificador a l'event detectat vinculant-lo a la regla en qüestió. Per realitzar búsquedes és còmode.

auditctl -w /ruta/fitxer/o/carpeta [-p aw]: afegeix una regla per monitoritzar l'accés (r), escriptura (w), execució (x) o canvi d'atributs (a) del fitxer o carpeta indicat. Si no s'indica el paràmetre -p s'entén que es volen monitoritzar totes les accions (r,w,x i a). Aquest tipus de regles són més específiques que les genèriques que monitoritzen crides al sistema perquè fan servir una API pròpia del kernel especialitzada en aquests tipus d'events anomenada Inotify (https://en.wikipedia.org/wiki/Inotify)

NOTA: Es pot afegir el paràmetre *-k etiqueta* per afegir un identificador a l'event detectat vinculant-lo a la regla en qüestió. Per realitzar búsquedes és còmode.

NOTA: Auditd cannot deal with files that do not exist when the rules are created: any file that is added to your system while audit is already running is not watched unless you extend the rule set to watch this new file.

auditctl -d {task|exit|exclude},{always|never} [-S nomCrida] [...] : esborra la regla indicada de monitorització de crida al sistema (cal escriure exactament els mateixos paràmetres que amb *-a* per identificar la regla en qüestió)

auditctl -W /ruta/fitxer/o/carpeta [-p aw] : esborra la regla indicada de monitorització de fitxers o carpetes (cal escriure exactament els mateixos paràmetres que amb *-w* per identificar la regla en qüestió)

auditctl -R /ruta/fitxer.rules : llegeix i carrega les regles indicades al fitxer especificat

NOTA: De les comandes anteriors es pot deduir que, per exemple, executar *auditctl -w /ruta/carpeta -p wa* i executar *auditctl -a always,exit -F dir=/ruta/carpeta -F perm=wa* és equivalent

Cal tenir en compte, però, que les regles definides amb la comanda *auditctl* són temporals, així que si les volem guardar permanentment serà necessari escriure-les dins del fitxer de text "/etc/audit/rules.d/audit.rules" (i reiniciar el servei, el qual s'encarregarà de llegir i carregar automàticament les regles allà indicades). La manera d'indicar les regles en aquest fitxer (una per línia) és la mateixa que l'usada en invocar a la comanda *auditctl*: -a exit,always -S open seria una regla vàlida, per exemple. L'ordre de les regles escrit al fitxer és important perquè a la primera regla que s'hi trobi que quadri amb les característiques de la crida o fitxer/carpeta monitoritzats, s'aplica i no se segueix llegint; és per això que se sol recomanar escriure primer regles concretes i específiques i anar fent-les més generals més endavant.

NOTA: La comanda *auditctl* no només serveix per gestionar les regles sinó també serveix per controlar el propi comportament del servei "auditd" sobreescrivint les opcions de configuració que puguin haver al seu fitxer de configuració gràcies als seus paràmetres *-b, -f, -e {0|1|2}, -r* o *-s* , entre d'altres

NOTA: En realitat, de fitxers de regles n'hi poden haver molts. Cada cop que s'inicia el servei audit aquest executa automàticament un script anomenat "augenrules" que arreplega tots els fitxers escrits dins de la carpeta "rules.d" i els col.lapsa, en ordre a l'arxiu "/etc/audit/audit.rules", que és l'arxiu que realment fa servir el dimoni. L'ordre de les regles és important perquè the kernel evaluates the rules in the order in which they were defined so it's good idea to place the most active rules first in order to speed up evaluation. No obstant, per simplificar, nosaltres només farem servir un fitxer per escriure-hi a dins totes les regles que necessitem.

NOTA: Tal com hem dit, "systemd-journald-audit.socket" no proporciona cap mecanisme per indicar les regles personalitzades que es volen registrar però sí reacciona a les regles carregades via *auditctl {-a|-w}* o bé *auditclt -R ...*

La informació recollida a l'arxiu "audit.log" es composa de diferents logs formats cadascun per varis camps com ara "*type*" (que pot valer qualsevol dels mateixos valors _AUDIT_TYPE_NAME que ja vam veure amb "systemd-journald-audit.socket" -és a dir, un dels llistats ací: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-audit_record_types -, entre els quals podem destacar SYSCALL, CWD, PATH, DAEMON_START o USER_LOGIN), "*msg*" (que conté un timestamp i un identificador únic per cada log), "*syscall*" (que conté el nº de crida al sistema concreta detectada; per veure la relació entre nº i nom de la crida es pot executar la comanda *ausyscall --dump*), "*success*" (que pot valer "yes" o "no" segons l'execució de la crida hagi tingut éxit o no), "*exit*" (que conté el valor de retorn de la crida; per interpretar aquest valor en un format "humà" es pot executar *ausearch –interpret --exit -13*), "*a0*" (el primer paràmetre de la crida), "*a1*" (el segon paràmetre de la crida), "*pid*", "*ppid*", "*uid*", "*tty*", "*comm*" (el nom de la comanda/programa d'usuari utilitzada per invocar la crida en qüestió), "*exe*" (la ruta absoluta d'aquesta comanda/programa), "*key*" (el valor indicat amb el paràmetre -k de *auditctl*), etc. Per més informació sobre els camps possibles, consulteu https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/app-Audit_Reference#sec-Audit_Events_Fields

Com que el contingut de "audit.log" és difícil de gestionar directament, una altra aplicació que ve dins del paquet "auditd" és *ausearch* , la qual serveix per buscar dins d'aquest arxiu (o el que s'especifiqui amb el seu paràmetre *-if*) events específics d'una forma sencilla i ràpida sense haver de fer servir *grep, cut* ni res semblant. Alguns exemples són (sempre executada com a root):

> NOTA: A la majoria de comandes *ausearch* també es pot afegir el paràmetre *-i*, el qual mostra diferents valors numèrics (com ara l'UID) en format "humà"

ausearch -m USER_LOGIN -sv no : busca intents d'inici de sessió fallits (amb *-sv yes* serien exitosos). Els possibles valors del paràmetre *-m* són els que pot tenir el camp "type" dels logs generats per Auditd; la llista sencera es pot obtenir executant *ausearch -m* . Tal com hem dit, es poden indicar varis paràmetres i això farà que es busqui tot el conjunt dels events indicats; per exemple, *ausearch -m ADD_USER -m DEL_USER -m ADD_GROUP* buscaria els canvis d'usuaris i grups al sistema

ausearch -m SYSCALL -sv no -ts yesterday -te now : busca totes les crides al sistema fallides des d'ahir fins avui. Els funcionament dels paràmetres *-ts* i *-te* s'explicarà en paràgrafs següents

ausearch -sc openat : busca totes les accions registrades per la crida del sistema indicada

ausearch -x /usr/bin/docker : busca totes les accions registrades per l'executable indicat

ausearch -p 4575 : busca totes les accions registrades pel PID indicat

ausearch -f /etc/fstab : busca totes les accions registrades pel fitxer indicat

ausearch -tm {ttyX|ssh} : busca totes les accions registrades pel terminal indicat

ausearch -ua 500 : busca totes les accions registrades per l'usuari indicat

ausearch -k etiqueta : busca totes les accions registrades per l'etiqueta indicada

ausearch -a idEvent : mostra el missatge de l'event l'identificador del qual s'ha indicat

També disposem de la comanda *aureport*, que permet obtenir informes sobre els events recollits. Es pot executar tal qual (sempre com a root) per obtenir un resum global o bé amb els següents paràmetres per obtenir informació més específica. Per exemple:

aureport -ts 23/03/2019 00:00:00 -te 31/03/2019 00:00:00 : mostra un resum de tots els events detectats durant la setmana indicada. Si no s'indica data es considera el dia d'avui; si no s'indica hora es considera les 00:00:00. El format vàlid de la data ve donat per la localització del sistema. També es poden indicar com a valors tant de *-ts* com de *-te* les paraules "*boot*", "*today*", "*yesterday*", "*this-week*", "*this-month*", "*this-year*" i "*recent*" (que vol dir "des de fa 10 minuts"). Aquests paràmetres es podran afegir a qualssevol dels descrits a continuació

NOTA: A la majoria de comandes *aureport* també es pot afegir el paràmetre *-i*, el qual mostra diferents valors numèrics (com ara l'UID) en format "humà"

NOTA: A moltes comandes *aureport* es pot afegir també el paràmetre *--summary* per obtenir informació més sintetitzada sobre l'aspecte en qüestió indicat

NOTA: A moltes comandes *aureport* es pot afegir també els paràmetres *--success* o *--failed* per obtenir només informació dels events existosos o fallits sobre l'aspecte en qüestió indicat

aureport -e : mostra la llista dels events detectats

aureport -s : mostra la llista de les crides al sistema detectades

aureport -x : mostra la llista dels events detectats relacionats amb executables
aureport -p: mostra la llista dels events detectats relacionats amb processos

aureport -f: mostra la llista dels events detectats relacionats amb fitxers

aureport -n : mostra comportaments anòmals del sistema

aureport -l : mostra un resum dels logins de tots els usuaris
aureport -au : mostra un resum dels intents d'autenticació
aureport -u : mostra un resum dels events relacionats amb usuaris

NOTA: Per a què *aureport* generi un informe a partir de la sortida obtinguda per *ausearch* (a través d'una canonada), aquesta comanda *ausearch* ha de tenir el paràmetre *-r* per formatejar la seva sortida convenientment

Si es vol analitzar amb *aureport* un altre fitxer que no sigui "/var/log/audit/audit.log" (per exemple, si és un fitxer obtingut d'una altra màquina diferent o bé és un fitxer de log rotat), cal indicar-lo amb el paràmetre *-if fitxer* En aquest sentit, per saber quin període de temps està emmagatzement en els diferents fitxers de log rotats es pot executar *aureport -t*

EXERCICIS:

0.-a) ¿Què mostra la comanda *last -a*? ¿I *last -s 2019-01-01 00:00:00 -t 2019-02-01 00:00:00*? ¿I *last pepito*? ¿I *last tty2*? ¿I *sudo lastb*? ¿D'on obtenen la informació que mostren (mira la seva pàgina del manual)? ¿Què mostra, en canvi, la comanda *w*?

b) ¿Quina/es comanda/es executaries per tallar inmediatament la connexió si veiessis a la sortida de la comanda *w* que hi ha algú connectat a través de SSH al teu sistema? PISTA: Mira com funciona el paràmetre -u de la comanda *pkill* (o també de *killall*)

1.-a) Instal.la el paquet "audit" (a Fedora) o "auditd" (a Ubuntu) i comprova que el servei corresponent estigui funcionant. Converteix-te en root i desactiva SELinux si treballes amb Fedora (recorda que això es fa temporalment executant *setenforce permissive*)

b) Comprova amb *auditctl -l* que no tinguis cap regla personalitzada definida (i si fos el cas, esborrar-les amb *auditctl -D*). Executa ara la comanda *auditcl -w /sbin/modprobe -p x -k PEPE* (a Ubuntu) o *auditctl -w /usr/sbin/modprobe -p x -k PEPE* (a Fedora). ¿Per a què creus que serveix la comanda anterior, i més concretament en l'àmbit de la seguretat (llegeix la nota de sota)? Comprova amb *auditctl -l* que ara s'hagi carregat correctament la regla anterior i seguidament executa *rmmod raid0 && modprobe raid0* (a Ubuntu) o *rmmod video && modprobe video* (a Fedora). Observa quines són les darreres línies que han aparegut dins del fitxer "/var/log/audit/auditd.log"...hauries de veure el registre de l'execució de la comanda *modprobe*. ¿La veus? SPOILER: no. La resposta, al següent apartat!

> NOTA: La comanda *rmmod* descarrega un determinat mòdul del kernel de memòria (per tant, el desactiva). La comanda *modprobe* el torna a carregar. Els mòduls indicats al paràgraf anterior són irrellevants, simplement s'ha procurat que la seva descàrrega no afecti al funcionament correcte del sistema.

c) La raó de perquè no es veuen línies relacionades amb la comanda *modprobe* al registre d'Audit és perquè en realitat *modprobe* és un link a l'executable */bin/kmod* (o pots comprovar amb *file $(which modprobe)* o també *ls -l $(which modprobe)*) i les regles Audit no funcionen per links. Sabent això, elimina la regla definida a l'apartat anterior i introdueix una de nova que monitoritzi l'execució de */bin/kmod* Torna a recarregar un mòdul qualsevol del kernel i observa que, ara sí, al final de l'arxiu audit.log apareixen unes quantes línies relacionades amb la comanda *kmod* (pots buscar per l'etiqueta, si t'és més senzill). ¿Quin valor tenen les dades "comm" i "exe"?

2.- a) Digues a priori per a què servirien en relació a la seguretat del sistema (i comprova seguidament que hagis encertat provocant d'alguna manera -digues quina- l'activació de la regla) les següents comandes:

auditctl -w /etc/passwd -p wa
auditctl -a exit,always -F exe=/usr/bin/killall
auditctl -a exit,always -S adjtimex -S settimeofday -S clock_settime
auditctl -a exit,always -S unlink -S unlinkat -S rename -S renameat -F auid>=1000
auditctl -a exit,always -S execve -F auid=0
auditctl -a exit,always -S listen
auditctl -a exit,always -S open -S openat -F exit=EACCES -F exit=EPERM
auditctl -a exclude,always -F msgtype=CWD

b) Digues què pretén el projecte https://github.com/Neo23x0/auditd i explica tres de les seves regles

3.-Executa les següents comandes i digues quina informació et mostren:

aureport -s -i | grep "^[0-9]" | cut -f 4,6 -d " " | sort | uniq
aureport -s -i | grep "^[0-9]" | cut -f 4,5 -d " " | sort | uniq
aureport -s -i –summary

aureport -u -i | grep "^[0-9]" | cut -f 4,7 -d " " | sort | uniq
aureport -u -i –summary

aureport -f -i | grep "^[0-9]" | cut -f 4,8 -d " " | sort | uniq
aureport -f -i

aureport -l -i
aureport -l -i --failed
aureport -l -i –success

aureport -e -i
aureport -au -ts yesterday -te today | grep no
ausearch -ui 1000 -x /bin/rm
ausearch -m DAEMON_START -sv no
ausearch -k PEPE

Els intents (fallits o no) d'inici de sessió (ja sigui des de SSH, TTY, *su*, *sudo*, etc) són registrats automàticament per Auditd. Una manera de trobar aquests registres al log és simplement fixar-se en els missatges de tipus USER_AUTH. Si volguéssim distingir entre els diferents mecanismes d'autenticació (SSH, TTY, etc), podem filtrar pel valor "exe" dels missatges. O si volguéssim distingir entre fallits i no, podem filtrar pel valor "res" dels missatges. També és interessant el camp "acct", que informa del compte d'usuari utilitzat i "hostname", que en el cas d'accessos remots informa de la IP de la màquina client.

4.-Havent llegit el paràgraf blau anterior...

a) ...inicia un servidor SSH a la màquina on estàs realitzant aquests exercicis i intenta-hi entrar des del client de la màquina real escrivint malament la contrasenya de l'usuari del servidor i seguidament escrivint-la bé. Observa quins missatges es guarden al registre d'Audit; fixa't sobre tot en el valor dels camps "type", "exe", "res", "acct" i "hostname". També pots observar la sortida d'*aureport*

b) Obre un terminal virtual a la màquina on estàs realitzant aquests exercicis i intenta iniciar-hi sessió, primer escrivint la contrasenya malament i després bé. Observa quins missatges es guarden al registre d'Audit; fixa't en el valor dels camps "type", "exe", "res" i "acct". També pots observar la sortida d'*aureport*

c) Executa (a la mateixa màquina sempre) una comanda qualsevol amb *sudo* o *su*, primer escrivint la contrasenya demanada malament i després bé. Observa quins missatges es guarden al registre d'Audit; fixa't sobre tot en el valor dels camps "type", "exe", "res" i "acct". També pots observar la sortida d'*aureport*