Laboratorul 7
Se(cret/cure???) Linux

Avertisment. Pentru ca majoritatea sunteți începători va recomand ca acest laborator să îl efectuați într-o mașină virtuală cu Debian Bullseye nu pe stații sau acasă direct.

Vedeți că există ceva numit *man nume-comandă* care vă poate ajuta mai bine decât căutatul pe Internet.
Reamintim că SELinux (Security Enhanced Linux) a fost inițial o implementare pentru a Mandatory Access Control permission system (MAC) la nivelul nucleului (kernel) Linux. Acesta este un instrument foarte puternic dar majoritatea administratorilor de sistem îl evită deoarece presupune o înțelegere completă a sistemului de operare Linux (nu numai simpla administrare). Ca atare aici vom discuta numai niște aspecte introductive legate de utilizarea acestuia. Cred că este clar că este operat la nivel supervisor chiar dacă și utilizatorul mai poate inspecta eventual modifica unele aspecte aflate sub controlul lui (dacă...)

Instalare

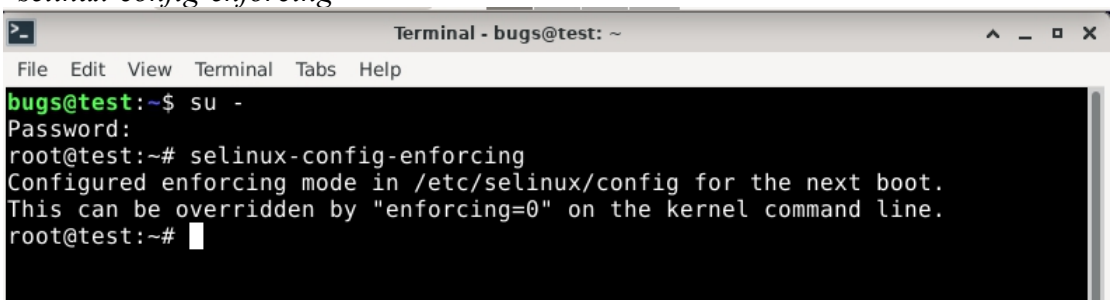
apt update
apt upgrade
apt install policycoreutils selinux-utils selinux-basics

Activarea sistemului

selinux-activate
Este recomandată repornirea mașinii și veți aștepta puțin pentru că va trebui să aplice noile informații (lucrează cu granularitate maximă) apoi mașina se va reporni automat.

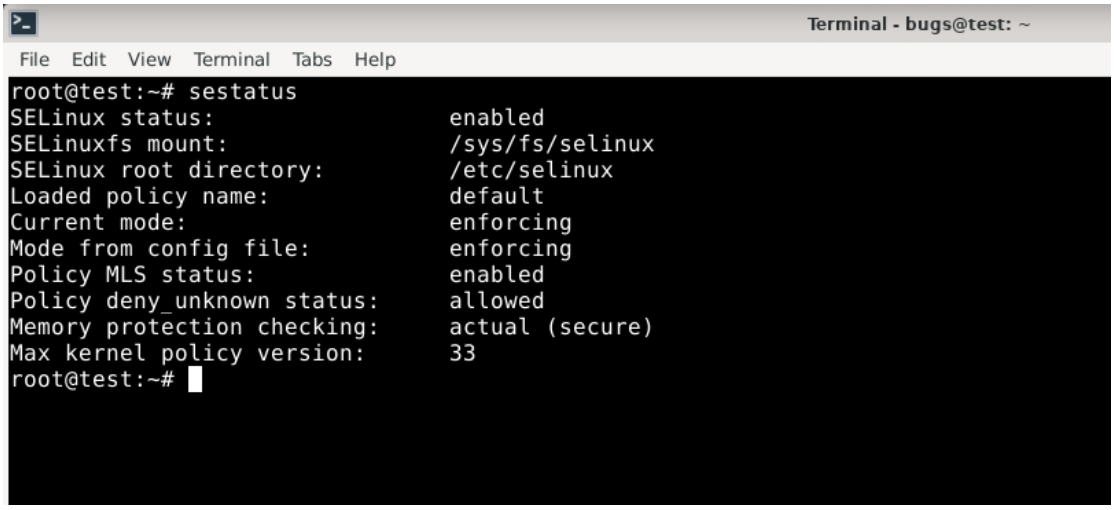
Stabilirea modului de lucru

Acum putem stabili o anumită manieră de a utiliza sistemul, de exemplu activarea completă a abilităților acestuia. Pentru început să vedem care este starea sistemului. Aceasta se realizează cu
selinux-config-enforcing



Observăm că ne trimite la fișierul principal de configurare pentru eventuale configurări suplimentare (manuale). Trebuie restartat din nou sistemul. Acum putem in sfârșit verifica status-ul sistemului

sestatus



Dacă simțiți uneori nevoia renunțării la securitatea oferită o cale imediată poate fi cu un
nano /etc/selinux/config
Unde comentați
SELINUX=enforcing
Și îl înlocuiți cu
SELINUX=disabled
După care restartați sistemul
Reamintesc că pot exista trei stări - *disabled* - nu face nimic, *permissive* (latră dar nu mușcă) adică raportează încălcările apărute si *enforced* când își face treaba. De exemplu dacă vreau rapid să-l comut în permisiv
setenforce 0
apoi
getenforce
pentru a vedea starea sistemului.
La curs am pomenit de unele din modulele mai importante utilizare de SeLinux deci pentru a vedea modulele curente putem utiliza
semodule -l

Activarea funcționalităților extinse de urmărire(jurnalizare) a execuției în Linux

apt-get install auditd audispd-plugins
Apoi se crează o configurare primara prin editarea /etc/audit/rules.d/audit.rules

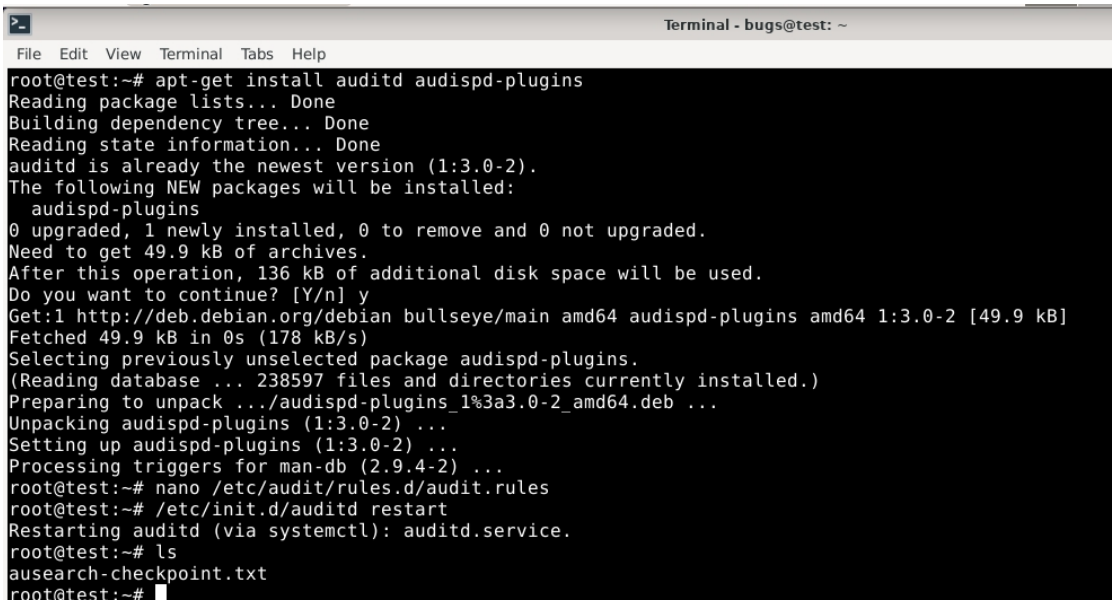
Se poate porni și modifica conform necesităților exemplul de mai jos

```
## Auditd configuration
## Configuration hint:
:
## -w watch file system object, -p sets [r|w|x|a]
## -a add rule: exit, upon syscall exit
## Remove any existing rules
-D
## Having a large buffer ensures we avoid dropping logs
-b 8192
## Failure Mode
## Possible values are 0 (silent), 1 (printk, print a failure message),
## and 2 (panic, halt the system).
-f 1
## Audit the audit logs, and execution of auditd reporting tools
-w /var/log/audit/ -k auditlog
-w /etc/audit/ -p wa -k auditconfig
-w /etc/libaudit.conf -p wa -k auditconfig
-w /etc/audisp/ -p wa -k audispconfig
-w /sbin/auditctl -p x -k audittools
```

```
-w /sbin/auditd -p x -k audittools
-a always,exit -F dir=/var/log/audit/ -F perm=r -F auid>=1000 -F auid!=unset -k audittools
-a always,exit -F path=/usr/sbin/ausearch -F perm=x -k audittools
-a always,exit -F path=/usr/sbin/aureport -F perm=x -k audittools
-a always,exit -F path=/usr/sbin/aulast -F perm=x -k audittools
-a always,exit -F path=/usr/sbin/aulastlogin -F perm=x -k audittools
-a always,exit -F path=/usr/sbin/auvirt -F perm=x -k audittools
## Log all process execution: disabled here by default due to load
# -a exit,always -S execve -k cmd
## Log all process executions by root
-a exit,always -F arch=b64 -F euid=0 -S execve -k rootcmd
-a exit,always -F arch=b32 -F euid=0 -S execve -k rootcmd
## Identify creation of filesystem nodes and file system mounts
-a exit,always -F arch=b32 -S mknod -S mknodat -k specialfiles
-a exit,always -F arch=b64 -S mknod -S mknodat -k specialfiles
-a exit,always -F arch=b32 -S mount -S umount -S umount2 -k mount
-a exit,always -F arch=b64 -S mount -S umount2 -k mount
## Clock and time zone changes
-a exit,always -F arch=b32 -S adjtimex -S settimeofday -S clock_settime -k time
-a exit,always -F arch=b64 -S adjtimex -S settimeofday -S clock_settime -k time
-w /etc/localtime -p wa -k localtime
## Updates to cron
-w /etc/cron.allow -p wa -k cron
-w /etc/cron.deny -p wa -k cron
-w /etc/cron.d/ -p wa -k cron
-w /etc/cron.daily/ -p wa -k cron
-w /etc/cron.hourly/ -p wa -k cron
-w /etc/cron.monthly/ -p wa -k cron
-w /etc/cron.weekly/ -p wa -k cron
-w /etc/crontab -p wa -k cron
-w /var/spool/cron/crontabs/ -k cron
## Credential/user/group/login changes
-w /etc/group -p wa -k etcgroup
-w /etc/passwd -p wa -k etcpasswd
-w /etc/gshadow -k etcgroup
-w /etc/shadow -k etcpasswd
-w /etc/security/opasswd -k opasswd
-w /usr/bin/passwd -p x -k passwd_modification
-w /usr/sbin/groupadd -p x -k group_modification
-w /usr/sbin/groupmod -p x -k group_modification
-w /usr/sbin/addgroup -p x -k group_modification
-w /usr/sbin/useradd -p x -k user_modification
-w /usr/sbin/usermod -p x -k user_modification
-w /usr/sbin/adduser -p x -k user_modification
-w /etc/login.defs -p wa -k login
-w /etc/securetty -p wa -k login
-w /var/log/faillog -p wa -k login
-w /var/log/lastlog -p wa -k login
-w /var/log/tallylog -p wa -k login
## Changes to network configurations
-w /etc/hosts -p wa -k hosts
-w /etc/network/ -p wa -k network
## system startup scripts
-w /etc/inittab -p wa -k init
-w /etc/init.d/ -p wa -k init
-w /etc/init/ -p wa -k init
## library search paths
-w /etc/ld.so.conf -p wa -k libpath
## kernel parameters
-w /etc/sysctl.conf -p wa -k sysctl
## modprobe configuration
-w /etc/modprobe.conf -p wa -k modprobe
## pam configuration
```

```
-w /etc/pam.d/ -p wa -k pam
-w /etc/security/limits.conf -p wa -k pam
-w /etc/security/pam_env.conf -p wa -k pam
-w /etc/security/namespace.conf -p wa -k pam
-w /etc/security/namespace.init -p wa -k pam
## system configuration changes
-w /etc/ssh/sshd_config -k sshd
-a exit,always -F arch=b32 -S sethostname -k hostname
-a exit,always -F arch=b64 -S sethostname -k hostname
-w /etc/issue -p wa -k etcissue
-w /etc/issue.net -p wa -k etcissue
## Capture all failures to access on critical elements
-a exit,always -F arch=b64 -S open -F dir=/etc -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/bin -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/sbin -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/usr/bin -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/usr/sbin -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/var -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/home -F success=0 -k unauthedfileacess
-a exit,always -F arch=b64 -S open -F dir=/srv -F success=0 -k unauthedfileacess
## Monitor for use of process ID change (switching accounts) applications
-w /bin/su -p x -k priv_esc
-w /usr/bin/sudo -p x -k priv_esc
-w /etc/sudoers -p rw -k priv_esc
## Monitor usage of commands to change power state
-w /sbin/shutdown -p x -k power
-w /sbin/poweroff -p x -k power
-w /sbin/reboot -p x -k power
-w /sbin/halt -p x -k power
## Do not allow configuration changes
## -e 0 disables, -e 1 enables, -e 2 locks configuration until reboot
-e 1
```

Activarea noii configurații se face cu
/etc/init.d/auditd restart



Dacă se dorește analiza jurnalelor de execuție (log-uri) rezultate cel mai simplu este sa ne uitam direct în audit.log
cat /var/log/audit.log sau
tail -f /var/log/audit.log

Aceasta este necesară deoarece SELinux va încerca să utilizeze subsistemele de urmărire ale Linuxului iar în caz contrar va utiliza numai sistemele implicite ceea ce nu este recomandat.
Analiza execuției în Linux poate fi realizată și fără SeLinux deoarece AVC-urile acestuia sunt jurnalizate implicit în audit.log unde se mai pot găsi și alte mesaje despre diverse evenimente de exemplu

type=SYSCALL msg=audit(...) : arch=x86_64 syscall=socket success=no exit=EACCES(Permission denied) a0=inet a1=SOCK_DGRAM a2=icmp a3=0x7fffac013050 items=0 ppid=2685 pid=17292 auid=admin uid=root gid=root euid=root suid=root fsuid=root egid=root sgid=root fsgid=root tty=tty1 ses=1 comm=ping exe=/bin/ping subj=sysadm_u:sysadm_r:ping_t:s0-s0:c0.c1023 key=(null)

Pentru a configura fișierul țintă se poate utiliza parametrul *log_file* aflat în */etc/audit/auditd.conf.*

Pentru situația în care se dorește centralizarea jurnalelor de la mai multe mașini se poate fie activa *syslog forwarding* sau plugin-ul *audisp-remote* .

În primul caz sistemul de urmărire este configurat să trimită evenimentele local iar administratorul trebuie să-l configureze explicit pentru ca să trimită aceste evenimente către o destinație externă. Pentru aceasta se editează fișierul */etc/audit/plugins.d/syslog.conf* și îl inițializez pe *active* cu *yes*:

nano /etc/audit/plugins.d/syslog.conf

active = yes

direction = out

path = /sbin/audisp-syslogtype = always

args = LOG_INFO

format = string

Totuși abordarea clasică nu garantează trimiterea și în plus nu oferă posibilități de securizare a comunicației spre deosebire de utilizarea lui *audisp-remote* care va lucra cu un server *auditd*. Pentru a realiza aceasta operație trebuie întâi configurat un demon de audit pe mașina care va centraliza jurnalele de execuție și să-i dăm acces la portul 60. Se poate schimba inclusiv maniera de organizare a informațiilor dintr-un mesaj inclusiv se pot adăuga informații despre mașina de origine. Astfel se va modifica *auditd.conf:*

nano /etc/audit/auditd.conf

tcp_listen_port = 60

log_format = ENRICHED

name_format = HOSTNAME

Apoi pe mașina sursă se va utiliza aceași configurație pentru *auditd.conf* dar fără a stabili și un port pentru ascultare. Apoi se va modifica *audisp-remote.conf* pentru a se putea conecta la mașina centralizatoare

nano /etc/audit/audisp-remote.conf

remote_server = <targethostname>

port = 60

Activarea plugin-ului audisp-remote se face prin modificarea fișierului au-remote.conf

nano /etc/audit/plugins.d/au-remote.conf

active = yes

Reporniți demonul asociat pentru a tine cont de noua configurație.

Tema 1. Instalați și verificați funcționarea SeLinux

Configurarea sistemului local de urmărire pentru SeLinux

După cum am văzut dacă nici o formă suplimentară de urmărire nu a fost activată Linux-ul va depune evenimentele utilizând facilitatea primara din kernel (kern.*) in */var/log/messages*

Dați un *cat* peste el de test

Deoarece de multe ori aceste mesaje sunt pentru administratorul comun este preferabil ca mesajele AVC (Access Vector Cache) ale SeLinux să fie depuse separat de exemplu în */var/log/avc.log*. Pentru aceasta am putea modifica configurația pentru *rsyslog* prin adăugarea în */etc/rsyslog.d* a uneia noi numită *99-selinux.conf*, ca mai jos:

nano /etc/rsyslog.d/99-selinux.conf

:msg, contains, "avc: " -/var/log/avc.log

Efectul se vede după repornirea sistemului. Dacă nu am creat aceasta configurație pot obține rapid ultimele mesaje AVC sau altele înrudite cu dmesg:

dmesg | grep avc | tail

Totuși abordarea nu este recomandată deoarece există categorii de utilizatori cu acces la *dmesg*. Din acest motiv SeLinux nu permite ca utilizatorii normali să acceseze inelul nucleului și ca atare să abia acces la *dmesg*. Dacă totuși se dorește (hmm) se poate seta *user_dmesg* din SELinux ca *on:*

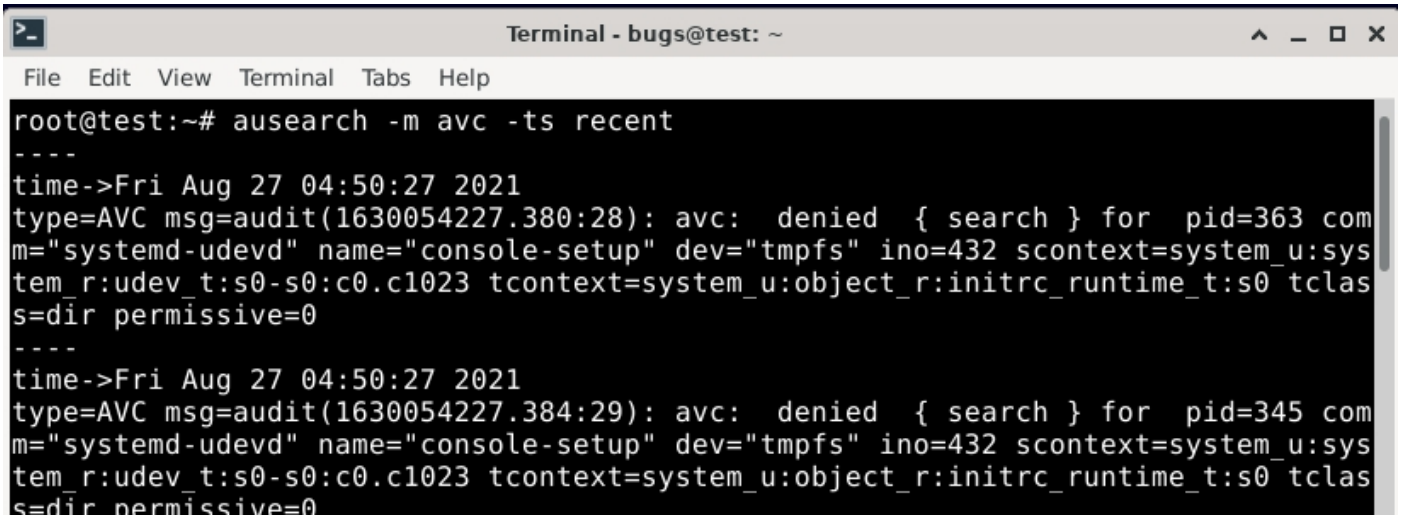
setsebool user_dmesg on

Dacă dorim să vedem direct valorile gestionate de SeLinux:

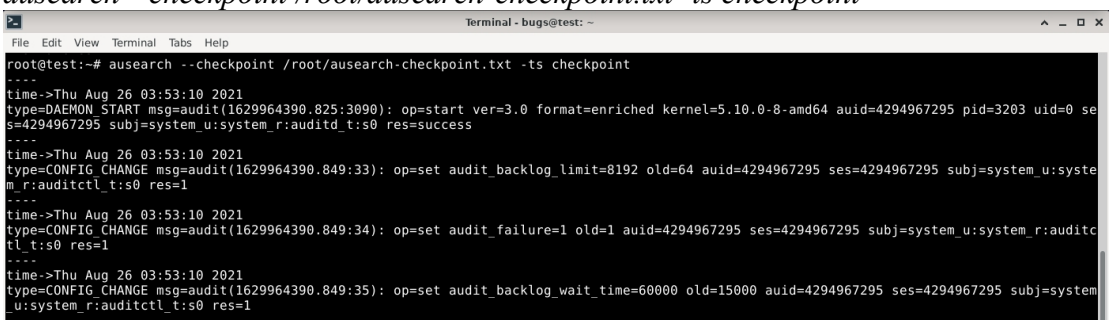
getsebool -a | less
Pentru abordări mai complicate avem *ausearch* care caută în /var/log/audit/audit.log
ausearch -ts recent
Aici în loc de recent pot avea:

- *today*, de la miezul nopții până acum
- *yesterday*, la fel din ziua trecută
- *this-week, this-month,* sau *this-year*, la fel dar cu referire la săptămână, lună și anul curent
- *checkpoint*, tine cont de punctul de verificare - checkpoint - (momentul creării acestuia) la ultima execuție
- *boot*, de când s-a pornit sistemul
- *week-ago*, cu șapte zile în urmă (tot de la miezul nopții)

Cea mai simplă filtrare este:
ausearch -m avc -ts recent

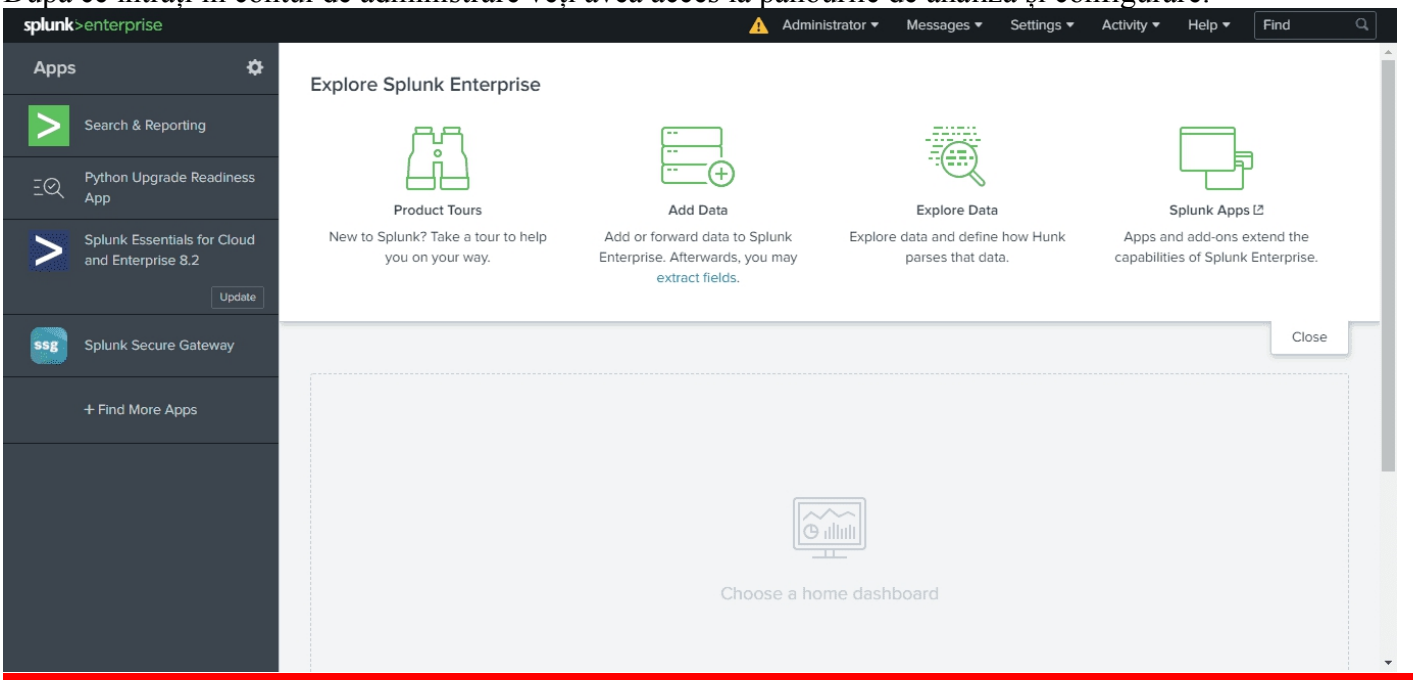


Dacă vreau erorile de accesare a contului
ausearch --message USER_LOGIN --success no --interpret
Pentru a căuta mesaje pentru toate conturile, grupurile și schimbările de rol putem utiliza:
ausearch -m ADD_USER -m DEL_USER -m ADD_GROUP -m USER_CHAUTHTOK -m DEL_GROUP -m CHGRP_ID -m ROLE_ASSIGN -m ROLE_REMOVE -i
Pentru a afișa toate acțiunile unui anumit utilizator pe baza identificatorului acestuia auid (de ex 500) putem folosi:
ausearch -ua 500 -i
Dacă doresc de exemplu afișarea tuturor apelurilor sistem eșuate din ultima zi avem:
ausearch --start yesterday --end now -m SYSCALL -sv no -i
Interesantă este utilizarea lui checkpoint în special atunci când se dorește depanarea eventualelor probleme care apar din configurarea SeLinux deoarece afișează toate mesajele apărute de la ultimul apel al lui ausearch.
ausearch --checkpoint /root/ausearch-checkpoint.txt -ts checkpoint



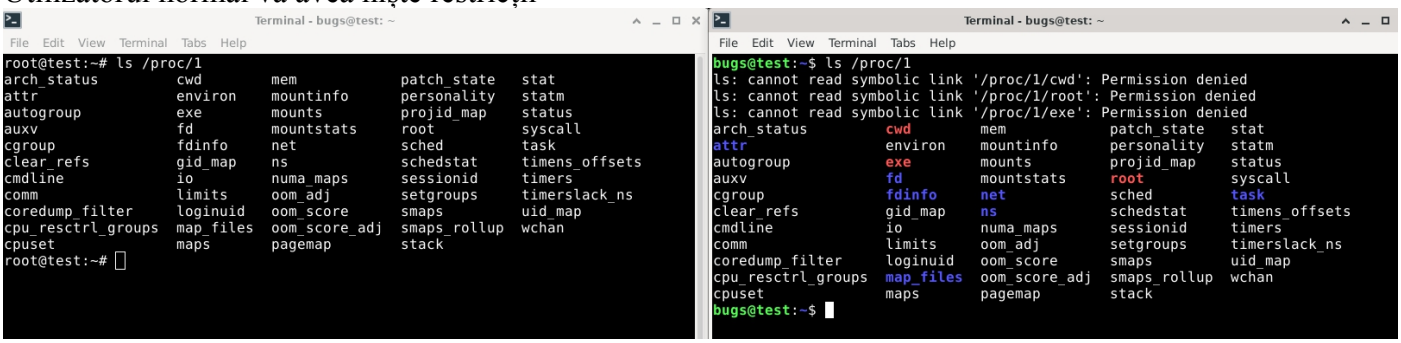
Mergând mai departe se poate utiliza *splunk* din categoria aplicațiilor SIEM
Intâi se aduce pachetul debi de la mama lui de exemplu *splunk-8.2.1-ddff1c41e5cf-linux-2.6-amd64.deb* (vă faceți cont) la *https://www.splunk.com/en_us/goto/Download_4_2*
Apoi instalarea
apt install ./splunk--amd64.deb*

Nu uitați contul și parola de administrator pe care o stabiliți la instalare, puteți să îl lăsați pornit implicit sau nu. Pentru a putea accesa interfața de administrare trebuie să îi dați acces în firewall:
ufw allow 8000
Abia acum se poate accesa interfața. Local la *http://localhost:8000* ea vine și cu posibilitatea de accesare la distanta pe același port.
După ce intrați în contul de administrare veți avea acces la panourile de analiză și configurare:



Tema 2. Jucați-vă un pic pentru ca să vedeți ce poate face (evident cititul documentației este obligatoriu dar .. știu că nu prea aveți chef)

Analiza activităților blocate de SeLinux
Sunt foarte importante deoarece din motive de viteză în faza primară de configurare putem scăpa din vedere configurarea lui pentru anumite activități legitime. Nu se recomanda utilizarea regulii *dontaudit* mai bine rezolvați problema altfel aveți doar un fals sentiment de securitate. Desigur odată sistemul configurat aceste mesaje pot da informații prețioase despre comportamente neașteptate ale aplicațiilor deci sunt importante pentru echipa albastra.
Mesajele despre blocarea activităților sunt emise imediat ce SeLinux-ul efectuează o astfel de blocare
De exemplu pentru comanda:
ls /proc/1
Utilizatorul normal va avea niște restricții



Dacă lansez o inspecție
ls -ldZ /proc/1
Atunci rezultă că
dr-xr-xr-x. 9 root root system_u:system_r:init_t:s0 0 Aug 27 04:50 /proc/1
Puteți să căutați apoi mesajul asociat acestei operații de blocare și să îl analizați.
Dacă luăm un mesaj oarecare:
time->Fri Aug 27 05:08:05 2021 type=AVC msg=audit(1630055285.152:148): avc: denied { search } for pid=1657 comm="systemd-udevd" name="console-setup" dev="tmpfs" ino=432 scontext=system_u:system_r:udev_t:s0-s0:c0.c1023 tcontext=system_u:object_r:initrc_runtime_t:s0 tclass=dir permissive=0

Vedem că el conține o serie de informații după cum urmează:

- Tipul - în acest caz avc
- Acțiunea efectuată de SeLinux - de obicei este *denied* (dar poate fi și *granted* dacă s-au prevazut reguli clare pentru SeLinux)
- Permisiunile (ce a vrut să facă procesul) în cazul nostru *search*
- Identificatorul procesului (*pid=1675*)
- Numele procesului(comenzii) *"systemd-udevd"*
- Numele destinației/resursei asupra căreia se efectuează comanda *name="console-setup"*
- Numele dispozitivului pe care se află resursa anterioară *dev="tmpfs"*
- Numărul *inode* al fișierului/directorului țintă *ino=432*
- Contextul sursei (contextul unde se află plasat procesul - domeniul acestuia) *scontext=system_u:system_r:udev_t:s0-s0:c0.c1023*
- Contextul resursei țintă *tcontext=system_u:object_r:initrc_runtime_t:s0*
- Clasa obiectului - pentru obiectul țintă de exemplu un fișier, soclu, nod, pipe, descriptor de fișier - *tclass=dir*
- Modul permisiv - modul în care se afla SeLinux pentru respectivul domeniu când a fost executată comanda - dacă este 0 atunci era activ altfel era permisiv - *permissive=0*

Evident că cine știe *regex* și *gawc* va aplica direct filtrări peste fișierele de mesaje restul poporului va rămâne cu instrumentele prezentate aici.

Tema 3. propuneți o linie de comandă care să aplice o filtrare peste un astfel de fișier (de ex să-mi afișeze tot ce a blocat pe un anumit port sau orice altceva vă trece prin cap)

Tema pe acasă

Realizați o aplicație bazată pe utilizarea calculului funcțional care să pornească de la un set de reguli primite ca un fișier de intrare și să analizeze continuu jurnalul de securitate al SeLinux generând alerte conform instrucțiunilor din respectivele reguli