

Ansinelli Yohann, Ralite Justin, Mathéo Balazuc

# AUDITD - CHAINSAW

---

## I/ Installation Auditd

Dans un premier temps on va venir installer le paquet avec la commande :

```
sudo apt-get install auditd
```

On peut maintenant regarder le fichier "auditd.conf" et on peut voir que les logs iront dans le fichier "/var/log/audit/audit.log"

```
GNU nano 7.2 /etc/audit/auditd.conf
#
# This file controls the configuration of the audit daemon
#
local_events = yes
write_logs = yes
log_file = /var/log/audit/audit.log
log_group = adm
log_format = ENRICHED
flush = INCREMENTAL_ASYNC
freq = 50
max_log_file = 8
num_logs = 5
priority_boost = 4
name_format = NONE
##name = mydomain
max_log_file_action = ROTATE
space_left = 75
space_left_action = SYSLOG
verify_email = yes
action_mail_acct = root
admin_space_left = 50
admin_space_left_action = SUSPEND
disk_full_action = SUSPEND
disk_error_action = SUSPEND
```

Pour activer auditd :

```
root@WECC:/etc/audit/rules.d# sudo systemctl start auditd
root@WECC:/etc/audit/rules.d# sudo systemctl enable auditd
Synchronizing state of auditd.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable auditd
```

```

root@WECC:/etc/audit/rules.d# sudo auditctl -e 1
enabled 1
failure 1
pid 209756
rate_limit 0
backlog_limit 8192
lost 0
backlog 0
backlog_wait_time 60000
backlog_wait_time_actual 0

```

Dans cette commande "auditctl -e 1" indique que le système doit supprimer automatiquement les plus anciennes entrées d'audit lorsque l'espace disque alloué pour le stockage des journaux d'audit est plein. Cela permet de gérer la saturation du système d'audit en éliminant les entrées les plus anciennes pour faire de la place aux nouvelles.

```

root@WECC:/etc# sudo auditctl -w /etc/passwd -p wxrxa -k beck_passwd

```

La commande "auditctl -w /etc/passwd -p wxrxa -k beck\_passwd" configure le système d'audit pour surveiller le fichier /etc/passwd et enregistrer les événements d'audit lorsqu'il est accédé en écriture, lecture, exécution ou lorsque ses attributs sont modifiés. Ces événements seront associés à la clé "beck\_passwd" pour une identification facile.

```
ausearch -f /etc/passwd
```

La commande "ausearch -f /etc/passwd" recherche donc les journaux d'audit pour tous les événements liés au fichier /etc/passwd. Cela pourrait inclure des informations sur qui a accédé au fichier, quand cela s'est produit :

```

time->Thu Dec  7 18:19:01 2023
type=PROCTITLE msg=audit(1701969541.054:1248): proctitle=7375646F006175736561726368002D66002F6574632F706173737764
type=PATH msg=audit(1701969541.054:1248): item=0 name="/etc/passwd" inode=2361133 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL co
ootid=0
type=CWD msg=audit(1701969541.054:1248): cwd="/etc"
type=SYSCALL msg=audit(1701969541.054:1248): arch=c000003e syscall=257 success=yes exit=13 a0=ffffff9c a1=7f3323d2cbe1 a2=80000 a3=0 items=1 ppid=65
0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts4 ses=3 comm="sudo" exe="/usr/bin/sudo" subj=unconfined key="beck_passwd"

```

```
ausearch -k beck_passwd
```

La commande ausearch -k beck\_passwd recherche donc les journaux d'audit pour tous les événements qui ont été étiquetés avec la clé "beck\_passwd" :

```

----
time->Thu Dec  7 18:27:06 2023
type=PROCTITLE msg=audit(1701970026.177:1727): proctitle=7375646F006175736561726368002D66002F6574632F706173737764
type=PATH msg=audit(1701970026.177:1727): item=0 name="/etc/passwd" inode=2361133 dev=08:01 mode=0100644 ouid=0 ogid=0 rdev=00:00 nametype=NORMAL co
ootid=0
type=CWD msg=audit(1701970026.177:1727): cwd="/etc"
type=SYSCALL msg=audit(1701970026.177:1727): arch=c000003e syscall=257 success=yes exit=13 a0=ffffff9c a1=7fb767fbcbe1 a2=80000 a3=0 items=1 ppid=65
0 suid=0 fsuid=0 egid=0 sgid=0 fsgid=0 tty=pts4 ses=3 comm="sudo" exe="/usr/bin/sudo" subj=unconfined key="beck_passwd"

```

On va venir maintenant modifier le fichier "audit.rules" pour y ajouter la ligne "-w /etc/passwd -p wxrxa -k beck\_passwd"

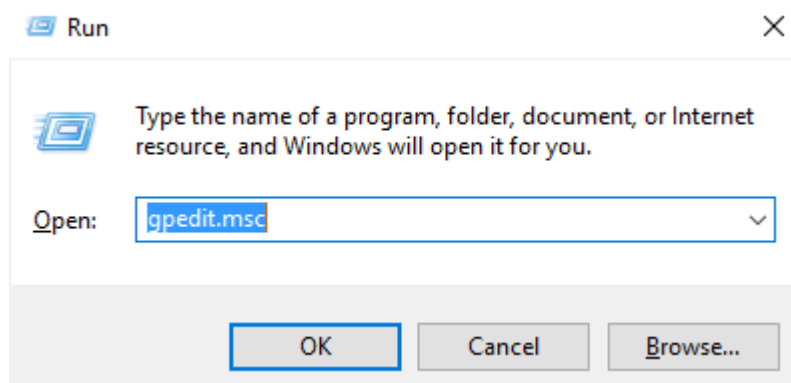
Ajouter cette ligne dans le fichier "audit.rules" signifie que le système d'audit surveillera le fichier /etc/passwd pour des événements spécifiques tels que l'écriture, la lecture, l'exécution et la modification d'attributs, et ces événements seront étiquetés avec la clé "beck\_passwd".

```
systemctl restart auditd
```

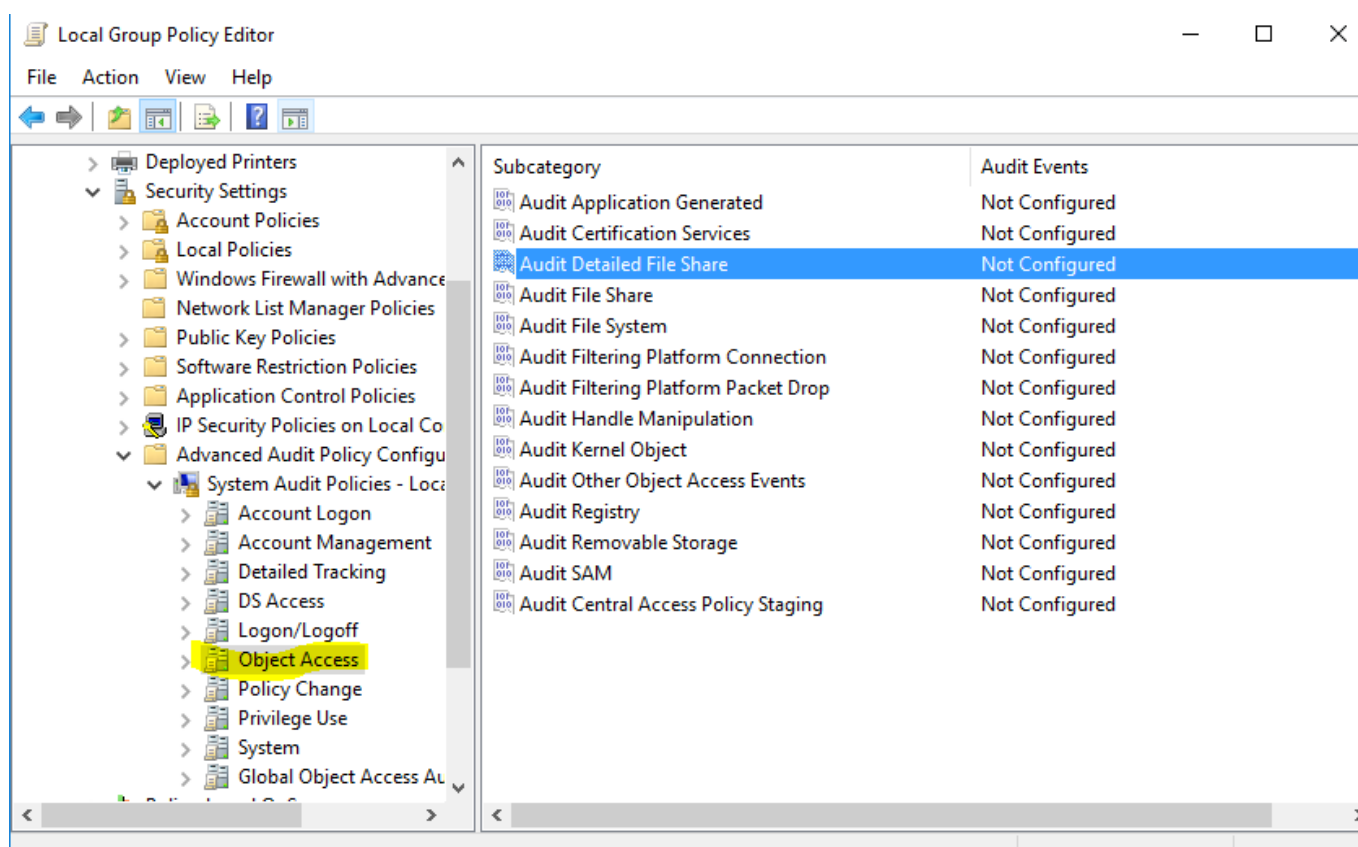
```
cat /var/log/audit/audit.log
```

anssi

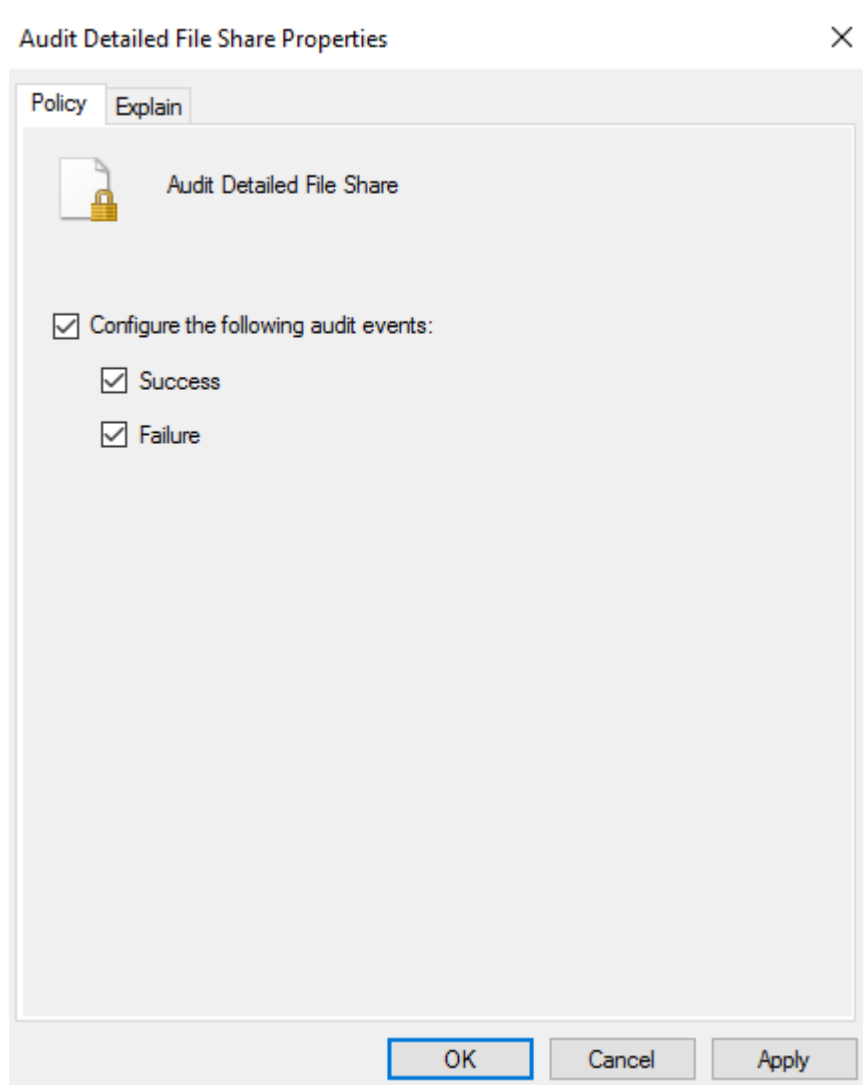
Dans cette partie on va venir activer l'audit sur les postes du GOAD. Dans un premier temps on va faire la combinaison : Windows + R et on va venir taper "gpedit.msc" :



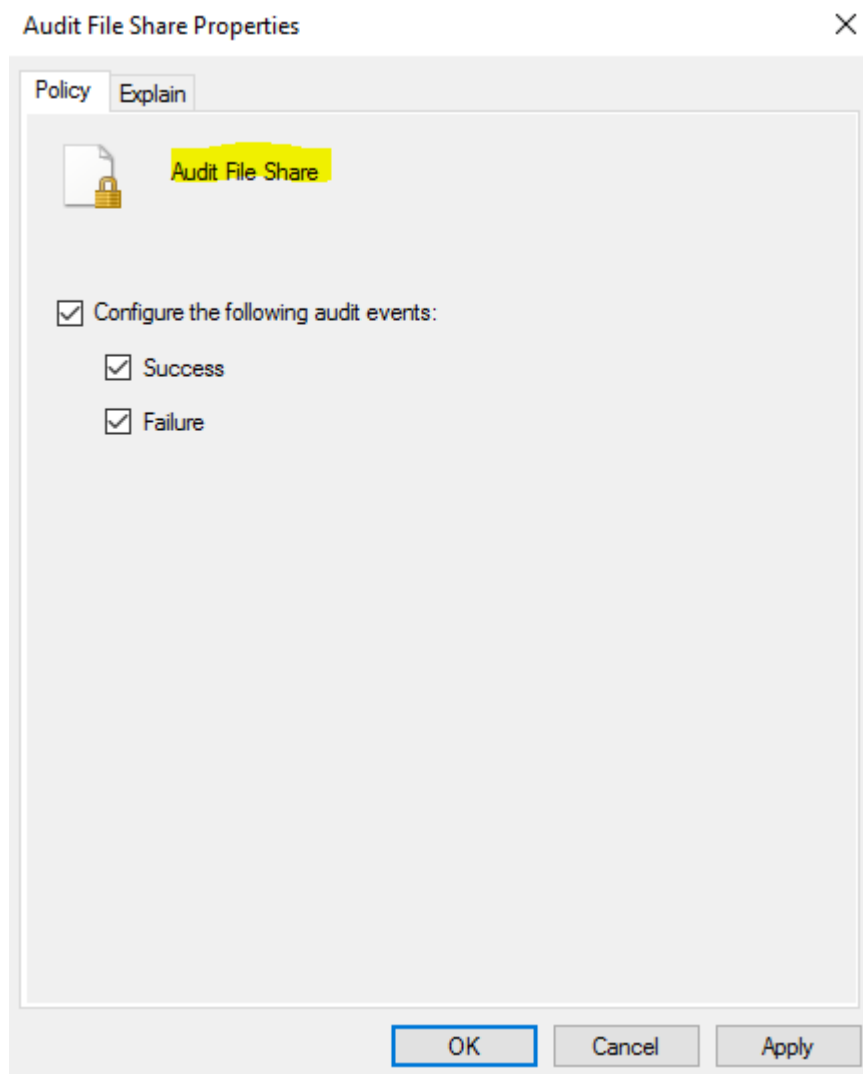
Ensuite on va suivre le chemin : "Windows Settings → Security Settings → Advanced Audit Policy Configuration → System Audit Policies → Object Access " et on va venir ouvrir le fichier "Audit Detailed File Share" :

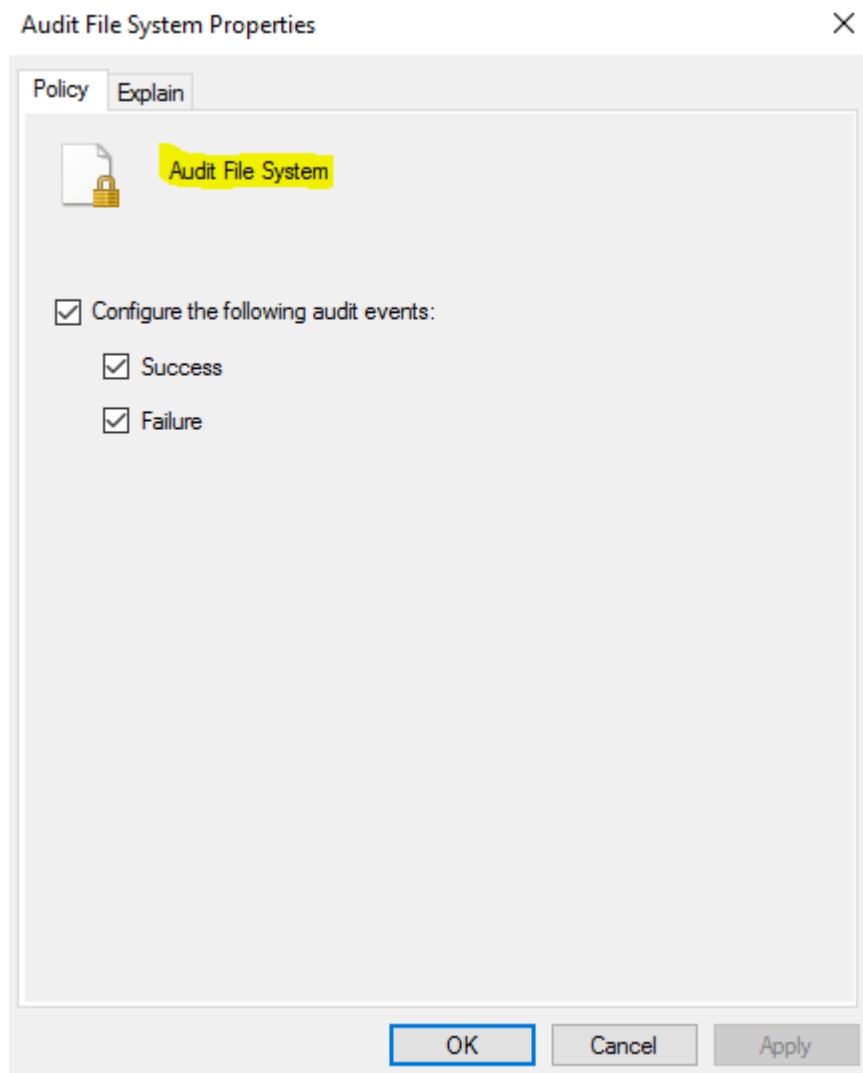


On active "Configure the following audit events" et on vient cocher "Success" et "Failure" :

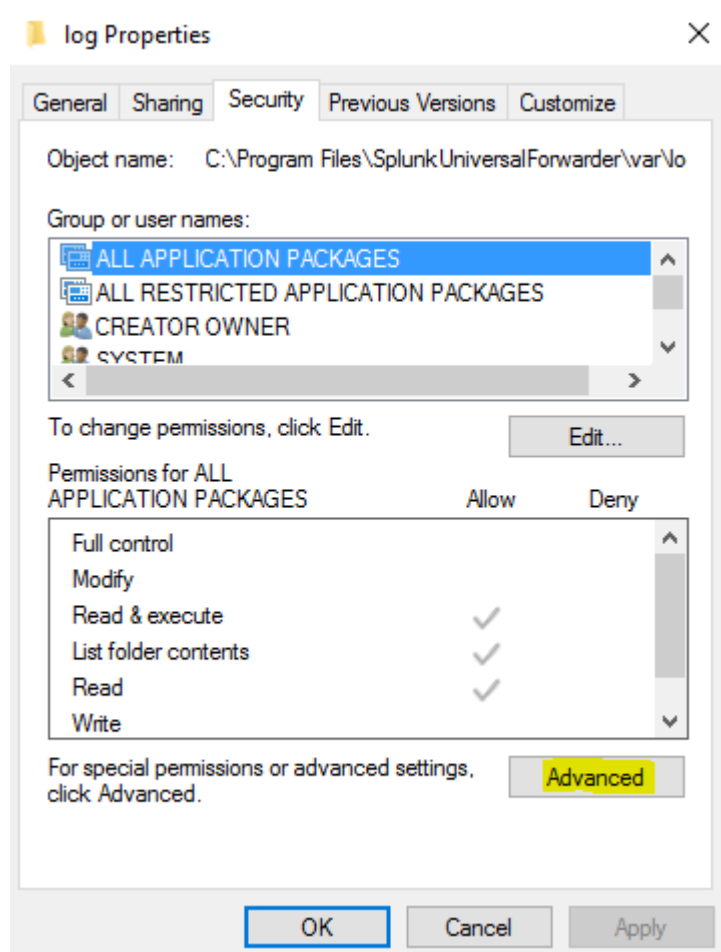


On va faire de même pour les fichiers "Audit File Share" et "Audit File System" :

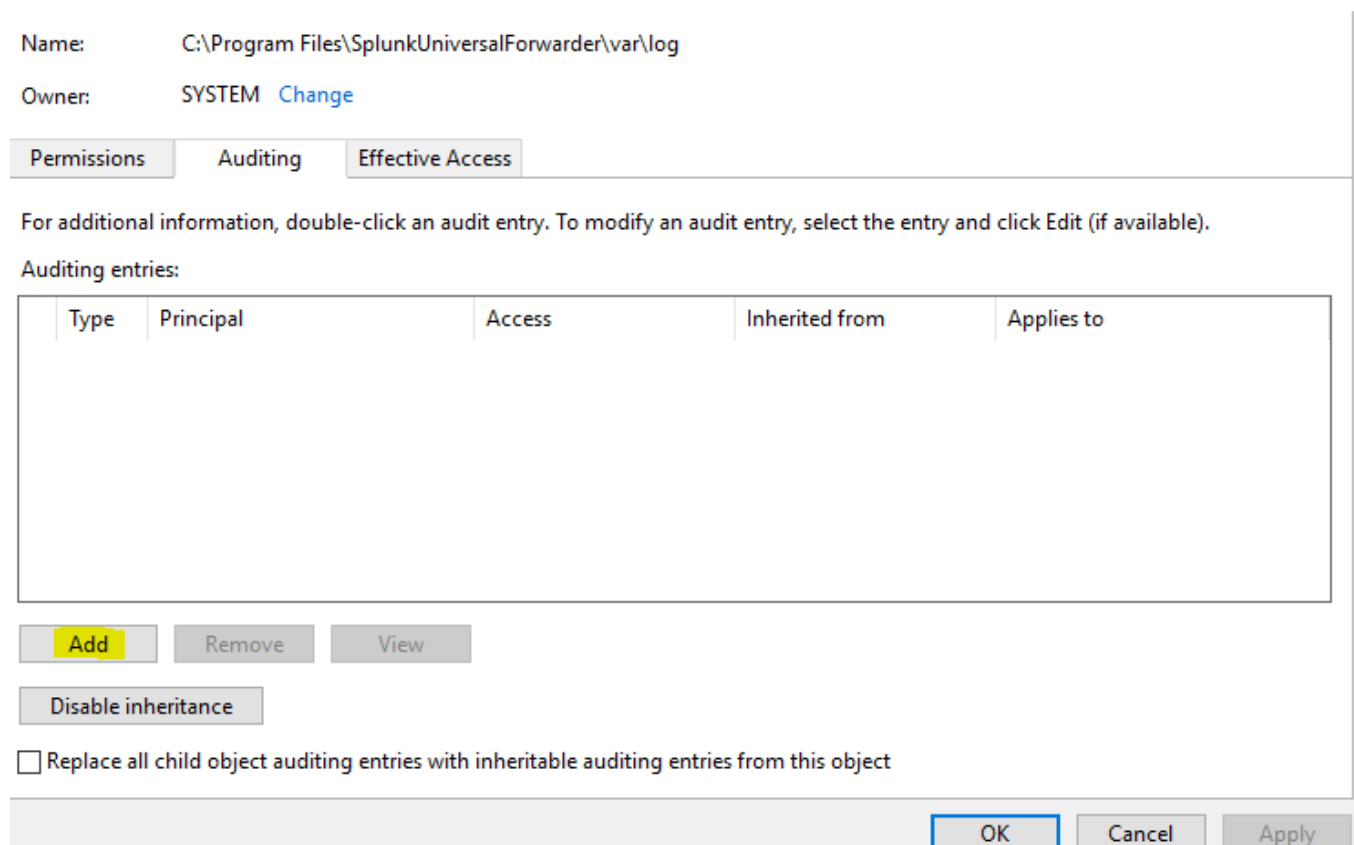




Ensuite on va venir choisir le fichier pour lequel on veut donner l'accès à l'audit. Pour ma part j'ai choisi le fichier "Log" de Splunk, on clique droit puis sur "propriétés" et on se rend dans "Security" et "Advanced" :

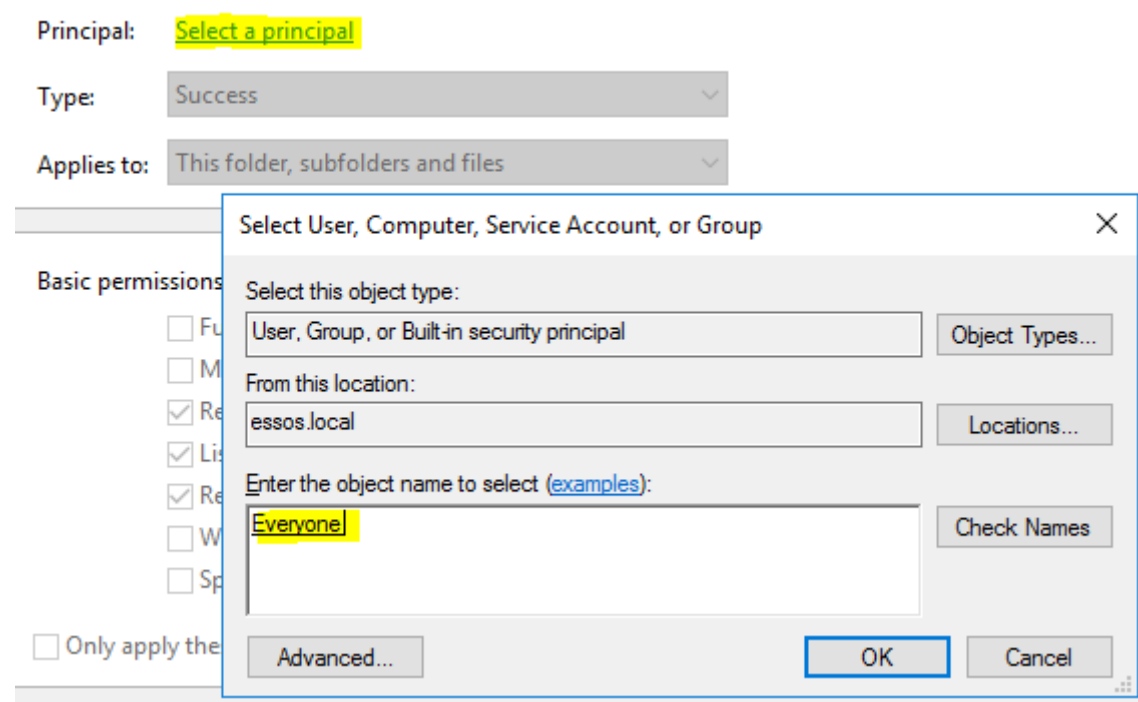


Sur la fenêtre on vient cliquer sur "Add" :

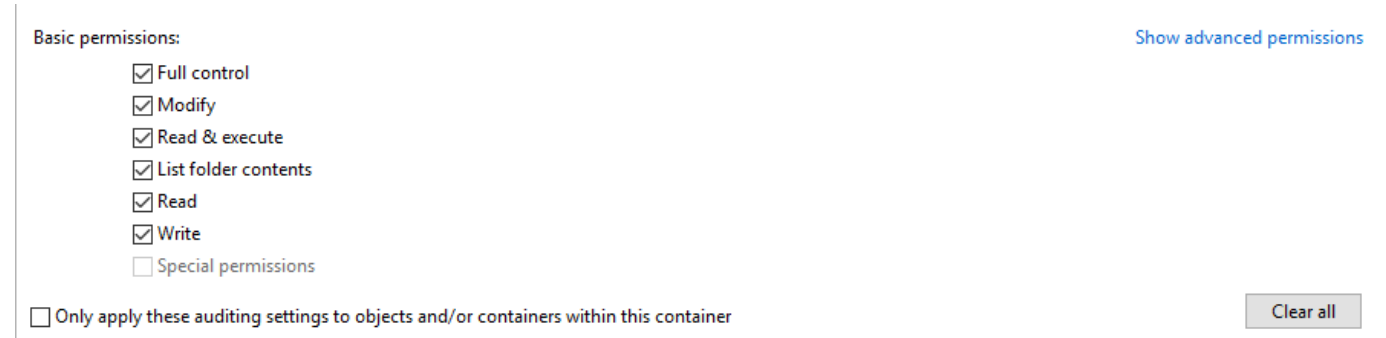


On clique sur "Select a principal" et on va venir ajouter comme objet "Everyone" pour dire Tout le monde :

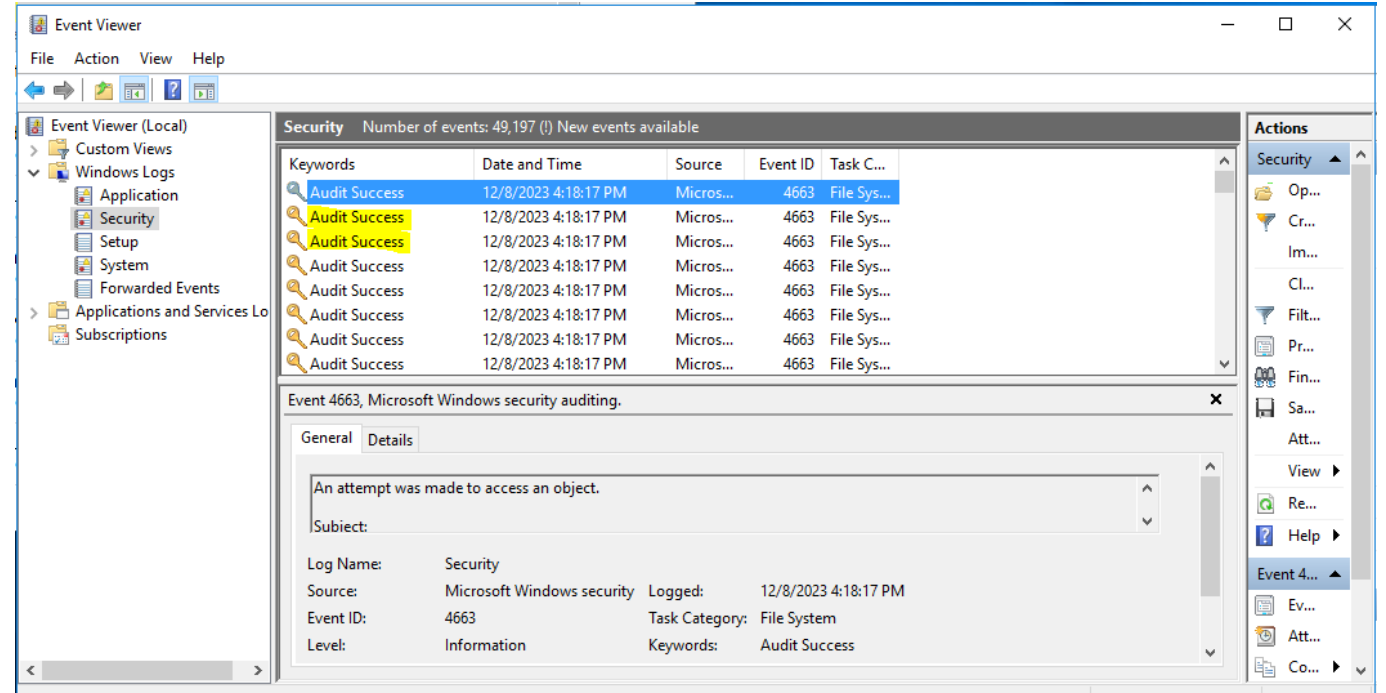




On coche l'ensemble des permissions :



Maintenant quand on va sur l'application **Event viewer** et dans "Windows Logs > Security" on va pouvoir les logs "Audit Success", ce qui signifie que la configuration a marché :



### III/ Installation Chainsaw

Pour installer chainsaw on va venir faire la succession de commande suivante :

```
git clone https://github.com/countercept/chainsaw.git
cd chainsaw
cargo build --release
cd target/release
sudo cp ./chainsaw /usr/local/bin
```

On vient ensuite créer notre directory :

```
mkdir chainsaw_workdir
cd chainsaw_workdir
git clone https://github.com/countercept/chainsaw.git
git clone https://github.com/SigmaHQ/sigma.git
```

Pour les fichiers evtx à chasser on va venir utiliser le fichier que j'ai obtenu avec les logs sysmon sur Splunk. Vous pouvez retrouver le fichier dans : Installation\_Siem/SPUNK/sysmon\_log.evtx

On va pouvoir maintenant lancer une chasse globale des attaques et on va mettre la sortie dans un fichier **chainsaw\_hunt\_global.out** :

```
chainsaw hunt /home/test/Téléchargements/sysmon_log.evtx -s sigma/ --
mapping chainsaw/mappings/sigma-event-logs-all.yml >
chainsaw_hunt_global.out
```

### IV/ Installation Hayabusa

Pour installer Hayabusa il faut exécuter les commandes suivantes :

```
sudo apt install musl-tools libssl-dev
rustup install stable-x86_64-unknown-linux-musl
rustup target add x86_64-unknown-linux-musl
git clone https://github.com/Yamato-Security/hayabusa.git --recursive
cd hayabusa
cargo build --release --target=x86_64-unknown-linux-musl
cp ./target/x86_64-unknown-linux-musl/release/hayabusa
/sur/local/bin/hayabusa
chmod +x /usr/local/bin/hayabusa
```

On va maintenant utiliser hayabusa pour obtenir la répartition des différents "events" par ID avec la commande :

```
hayabusa eid-metrics -f /home/test/Téléchargements/sysmon_log.evtx >  
hayabusa_sysmon.out
```

Evtx File Path: /home/test/Téléchargements/sysmon\_log.evtx

Total Event Records: 933

First Timestamp: 2023-12-07 09:07:31.307 +01:00

Last Timestamp: 2023-12-07 10:06:10.587 +01:00

Total	%	Channel	ID	Event
703	75.3%	Sysmon	1	Process Creation
136	14.6%	Sysmon	22	DNS Query
56	6.0%	Sysmon	13	Registry Value Set
26	2.8%	Sysmon	11	File Creation or Overwrite
6	0.6%	Sysmon	15	Alternate Data Stream Created
3	0.3%	Sysmon	5	Process Terminated
1	0.1%	Sysmon	16	Sysmon Service Configuration Changed
1	0.1%	Sysmon	8	Remote Thread Created (Possible Code Injection)
1	0.1%	Sysmon	4	Sysmon Service State Changed

Elapsed time: 00:00:04.415