Documentation d'installation agent Cortex XDR

Cette procédure est destinée aux OS Linux. Pour l'instant, elle a été testée uniquement sur les distributions Debian, Ubuntu, CentOS et Oracle Linux puisque ce sont celles ci qui constituent la majeur partie de notre parc.

Récupération des fichiers d'installation

La première étape consiste à récupérer les deux fichiers nécessaires depuis notre lecteur réseau à l'aide de la commande `scp` (secure copy protocol), qui utilise le protocole SSH.

Au moment de recuperer les fichiers il faut faire le choix entre `Cortex-X-X-X-XXXXXX_deb.tar.gz` et `Cortex-X-X-X-XXXXXX_rpm.tar.gz`, certaine distribution fonctionne avec dpkg, comme Debian il va alors falloir choisir le *deb, tandis que sur les distribution fonctionnant avec yum / rpm comme CentOS il va falloir choisir l'archive en* _rpm
Ensuite la deuxième archive `cortex-xdr-agent.zip` sera utilisé seulement avec les distributions utilisant rpm.

```
root@[redacted]:~# scp root@[redacted]/APPLICATIONS/XDR/Cortex-9-0-0-141085_deb.tar.gz .
The authenticity of host [redacted] can't be established.
ECDSA key fingerprint is SHA256:IGgXDFB1Ja7Ft4V2ktcZYeGzw7YssoqEpqnqzR8rkxM.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added [redacted] (ECDSA) to the list of known hosts.
root@[redacted]'s password:
Cortex-9-0-0-141085_deb.tar.gz
```

```
[root@[redacted] ~]# scp root@[redacted]/APPLICATIONS/XDR/Cortex-9-0-0-141085_rpm.tar.gz .
root@[redacted] password:
Cortex-9-0-0-141085_rpm.tar.gz
Cortex-9-0-0-141085_rpm.tar.gz
[root@[redacted] ~]# scp root@[redacted]/APPLICATIONS/XDR/cortex-xdr-agent.zip .
root@[redacted]'s password:
cortex-xdr-agent.zip
```

Décompression et configuration des fichiers

Le fichier en .tar.gz fourni par la console admin de Palo Alto que j'ai récupéré sur le lecteur réseau est compressé par défaut, le deuxième est zippé par nous-même. Il faut donc les dézipper à l'aide des commandes `tar -xf` et `unzip`.

```
[root@ ~]# tar -xf Cortex-9-0-0-141085_rpm.tar.gz
[root@ ~]# unzip cortex-xdr-agent.zip
Archive:  cortex-xdr-agent.zip
  inflating: cortex-xdr-agent.asc
```

Le tar.gz contient le paquet en .deb, ainsi qu'un fichier de configuration `cortex.conf`.

Celui-ci devra être modifié pour ajouter une ligne `--proxy-list "ip_du_broker"`. Cette ligne servira à faire communiquer l'agent avec le broker Palo Alto. Ce fichier `cortex.conf` devra être déplacé dans le dossier `panw`, que l'on va créer dans `etc` avec `mkdir -p /etc/panw`.

Cette étape comme précisé plus haut est seulement destiné aux distribution utilisant rpm/yum, le fichier `cortex-xdr-agent.zip` contient un .asc. C'est une clé qui permettra, une fois importée, d'authentifier l'authenticité du paquet à installer, ce qui n'est pas utile quand on installe avec dpkg car ce gestionnaire de paquet n'empêche pas d'installer un paquet non signé.

La commande `rpm --import` va alors ajouter la clé du .asc dans sa bibliothèque de clé.

```
[root@ ~]# rpm --import cortex-xdr-agent.asc
```

Installation du paquet

Enfin, on installe le paquet avec `dpkg -i` ou `yum install` en fonction de la distribution.

```
root@[redacted]:~# dpkg -i cortex-9.0.0.141085.deb
(Lecture de la base de données... 49532 fichiers et répertoires déjà installés.)
Préparation du dépaquetage de cortex-9.0.0.141085.deb ...
Active kernel LSM: lockdown,capability,yama,apparmor,tomoyo
[ 1] Checking prerequisites
Verifying Debian 11 (dpkg) packages:
  * openssl ... OK
  * ca-certificates ... OK
Done
Dépaquetage de cortex-agent (9.0.0.141085) sur (9.0.0.141085) ...
Paramétrage de cortex-agent (9.0.0.141085) ...
Active kernel LSM: lockdown,capability,yama,apparmor,tomoyo
[ 1] Upgrading Cortex XDR [9.0.0.141085] at /opt/traps
[ 2] Stopping daemons
              Name       PID          User                Status              Command
               pmd       N/A           N/A               STOPPED              N/A
              clad       N/A           N/A               STOPPED              N/A
              dypd       N/A           N/A               STOPPED              N/A
              spmd       N/A           N/A               STOPPED              N/A
              lted       N/A           N/A               STOPPED              N/A
              pyxd       N/A           N/A               STOPPED              N/A
              cned       N/A           N/A               STOPPED              N/A
              piud       N/A           N/A               STOPPED              N/A
Done
[ 3] Removing old agent's files
Done
[ 4] Backing up current agent's configuration
Done
Using system libraries
Done
[ 5] Creating runtime directory
Done
       n'est pas un exécutable dynamique
[ 6] Updating AppArmor policies
Done
[ 7] Verifying packet filtering tool prerequisite
Done
[ 8] Defining Cortex XDR local services (systemd)
Done
[ 9] Creating/Verifying Cortex XDR auxiliary user
Done
```

```
[root@[redacted] ~]# yum install cortex-9.0.0.141085.rpm
Modules complémentaires chargés : fastestmirror
Examen de cortex-9.0.0.141085.rpm : cortex-agent-9.0.0.141085-1.x86_64
Sélection de cortex-9.0.0.141085.rpm pour installation
Résolution des dépendances
--> Lancement de la transaction de test
---> Le paquet cortex-agent.x86_64 0:9.0.0.141085-1 sera installé
--> Résolution des dépendances terminée

Dépendances résolues

================================================================================
 Package                Architecture          Version                 Dépôt                Taille
================================================================================
Installation :
 cortex-agent           x86_64                9.0.0.141085-1          /cortex-9.0.0.141085  77 M

Résumé de la transaction
================================================================================
Installation   1 Paquet

Taille totale  : 77 M
Taille d'installation : 77 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
SELinux status: Disabled
```

> à noter que le paquet **selinux-policy-devel** (paquet concernant la sécurité), doit être installé au préalable si ce n'est pas déjà fait car il est requis pour l'installation de Cortex, et il n'était pas présent sur certaine machines

Verifications de l'installation

Je vérifie qu'il est actif et installé avec `dpkg -l` qui va vérifier les paquets installés sur la machine. `grep cortex` va filtrer la réponse avec uniquement ceux qui contiennent cortex dans leur nom.

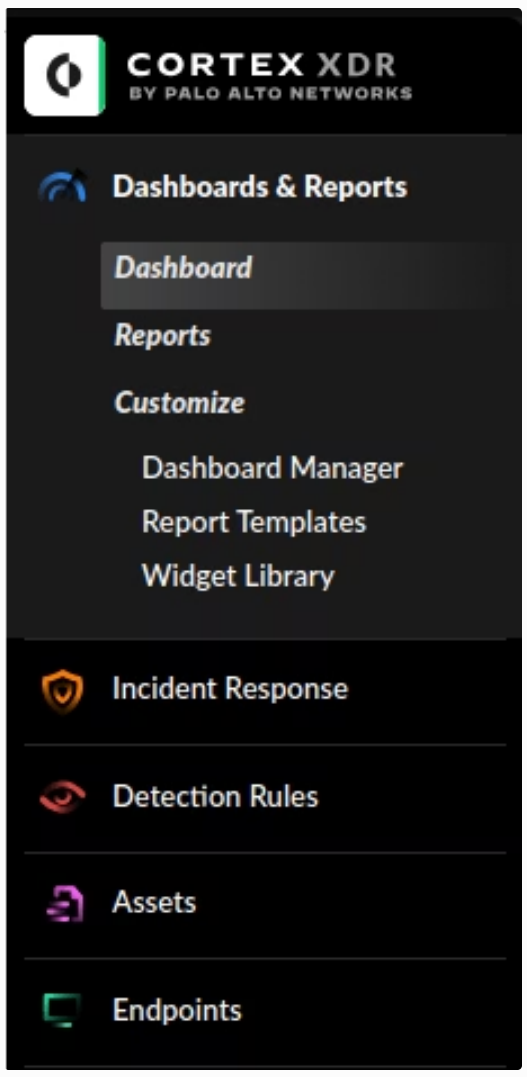
```
root@[redacted]:~# dpkg -l | grep cortex
ii  cortex-agent                  9.0.0.141085                amd64        Palo Alto Networks Cortex XDR(tm) endpoint security agent
root@[redacted]:~# systemctl status traps_pmd.service
● traps_pmd.service - Palo Alto Networks Cortex XDR Agent(tm) daemon
     Loaded: loaded (/etc/systemd/system/traps_pmd.service; enabled; vendor preset: enabled)
     Active: active (running) since Tue 2026-02-17 11:08:19 CET; 25s ago
   Main PID: 177087 (pmd)
      Tasks: 110 (limit: 4657)
     Memory: 683.7M
        CPU: 12.139s
     CGroup: /system.slice/traps_pmd.service
             ├─177087 /opt/traps/bin/pmd
             ├─177211 /opt/traps/python/payload/pyxd -config /opt/traps/python/scripts/service_main.json -type 2
             ├─177294 /opt/traps/analyzerd/clad -n clad -c 338:requests -- --log-level 7 --max-worker-count 10
             ├─177298 /opt/traps/bin/dypdng -a -- 342
             └─177300 /opt/traps/analyzerd/sandboxd -n Yara -c 344:requests

févr. 17 11:08:19 [redacted] systemd[1]: Started Palo Alto Networks Cortex XDR Agent(tm) daemon.
```

Ensuite, `systemctl status traps_pmd` va nous afficher l'état du service ainsi que quelques lignes de log pour s'assurer qu'il n'y a pas de problème. C'est la commande à retenir entre les deux puisqu'elle fonctionne sur toutes les distributions à l'inverse de dpkg.

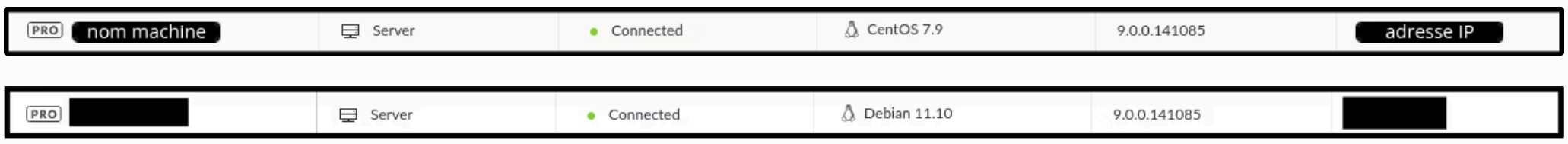
Console Palo Alto

il faut pour finir vérifier sur la console en interface web de Palo Alto que le serveur remonte bien, sans erreur et avec les bonnes informations.



sur l'interface, on a plusieurs espace manipuler notre cortex,
je ne peux pas trop en montrer mais l'interface est très complète, on peut avoir une vu d'ensemble sur le service avec differentes data depuis le **dashboard**,
la categorie **incident response** va nous montrer les differents incidents ainsi que les actions du cortex à propos de celle ci.
L'espace **detection rules** permet de consulter et configurer à notre guise les regles qui vont declencher les alertes.
Assets correspond globalement l'inventaire de tout ce qui est identifié par l'XDR.
Endpoints est une section dédiée spécifiquement à la gestion des agents installés sur les postes de travail et serveurs (état de santé, version de l'agent, déploiement).

Donc pour vérifier qu'une machine remonte bien dans la console Cortex, il suffit de chercher son nom ou IP dans la section **Endpoints** puis **All Endpoints**



Problèmes Rencontré

Un problème majeur rencontré est au niveau des logs, ce services produit une assez grosse quantité de log d'autant plus quand c'est une machine très utilisé, je l'avais installé sur une machine qui n'avait plus beaucoup d'espace disque, et les logs avaient rempli l'espace restant, cela à conduit a des problèmes de fonctionnement du service applicatif de la machine, et nous avons donc été contraint de stopper l'agent XDR temporairement.