Supervision

I-Ajouter une machine hôte à la supervision

Sur votre serveur Linux à surveiller…

La procédure est un peu plus longue. Il faut d'abord installer le daemon NRPE et les plugins Nagios (qui vont être lancés localement par le daemon NRPE):

Sous Ubuntu/Debian:

```
# sudo apt-get install nagios-nrpe-server
# sudo apt-get install nagios-plugins
# sudo apt-get install xinetd
#apt-get install check-mk-agent
```

Puis éditer le fichier /etc/nagios/nrpe.cfg pour modifier la ligne suivante:

```
...
allowed_hosts = Mettre ici l'adresse IP de votre serveur Nagios
...
```

// Vérifier si le service nagios-nrpe-server est en fonctionne avec la commande ci-dessous :

```
root@racktables-server:/etc/nagios# systemctl status nagios-nrpe-server.service
● nagios-nrpe-server.service - LSB: Start/Stop the Nagios remote plugin execution daemon
   Loaded: loaded (/etc/init.d/nagios-nrpe-server)
   Active: active (running) since mar. 2017-07-04 12:01:59 CEST; 1 weeks 1 days ago
  Process: 4558 ExecStop=/etc/init.d/nagios-nrpe-server stop (code=exited, status=0/SUCCESS)
  Process: 4564 ExecStart=/etc/init.d/nagios-nrpe-server start (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nagios-nrpe-server.service
           └─4567 /usr/sbin/nrpe -c /etc/nagios/nrpe.cfg -d
```

On automatise le lancement du daemon au démarrage du serveur avec la commande:

```
root@racktables-server:/etc/nagios# chkconfig --add nagios-nrpe-server
        nagios-nrpe-server      0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

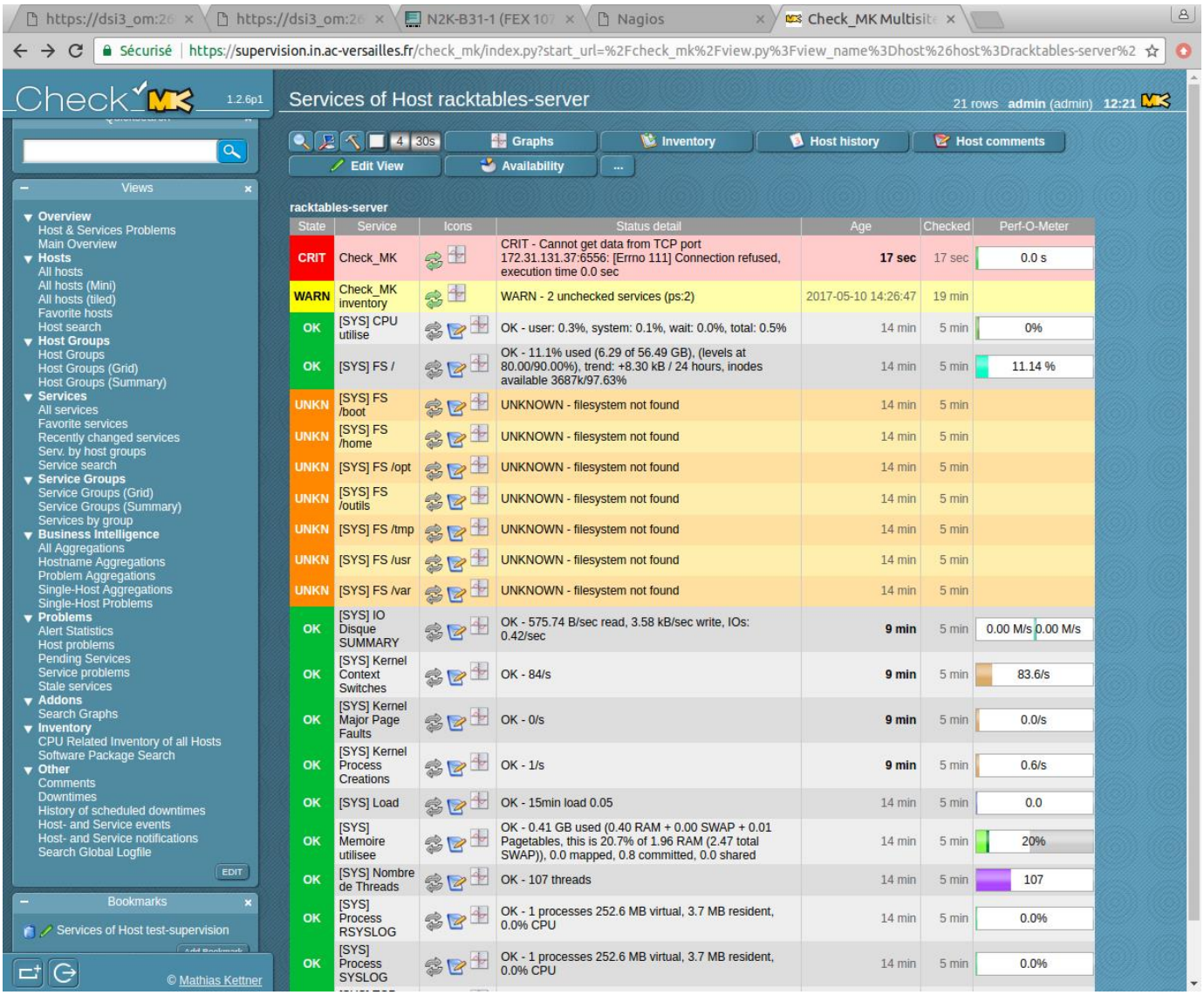
On ajoute une règle pour autoriser le Firewall IPtable à laisser passer les requêtes NRPE (à adapter selon vos règles):

```
root@racktables-server:/etc/xinetd.d# iptables -A INPUT -p tcp -m tcp --dport 5666 -j ACCEPT

root@racktables-server:/etc/xinetd.d# iptables -L
        Chain INPUT (policy ACCEPT)
        target     prot opt source               destination
        ACCEPT     tcp  --  anywhere             anywhere            tcp dpt:nrpe

        Chain FORWARD (policy ACCEPT)
        target     prot opt source               destination

        Chain OUTPUT (policy ACCEPT)
        target     prot opt source               destination
```

// Vérifier si la machine écoute bien dans le port 5666 avec la commande ci-dessous :

```
root@racktables-server:/etc/xinetd.d# netstat -tulpn
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale        Adresse distante       Etat        PID/Program name
tcp        0      0 0.0.0.0:22            0.0.0.0:*              LISTEN      6551/sshd
tcp        0      0 127.0.0.1:25          0.0.0.0:*              LISTEN      15119/exim4
tcp        0      0 0.0.0.0:6556          0.0.0.0:*              LISTEN      13149/xinetd
tcp        0      0 0.0.0.0:5666          0.0.0.0:*              LISTEN      13341/nrpe
tcp        0      0 0.0.0.0:111           0.0.0.0:*              LISTEN      1808/rpcbind
tcp6       0      0 :::22                 :::*                   LISTEN      6551/sshd
tcp6       0      0 ::1:25                :::*                   LISTEN      15119/exim4
tcp6       0      0 :::443                :::*                   LISTEN      4648/apache2
tcp6       0      0 :::5666               :::*                   LISTEN      13341/nrpe
```

Etat de la machine racktables-server dans la supervision avant les manipes:



Maintenant, lancez le service nagios-nrpe avec la commande suivante:

systemctl restart nagios-nrpe-server.service
systemctl restart xinetd.service

Dans certains distributions, il faut activer le script de lancement de xinetd. Sinon le service xinetd ne sera pas lancé automatiquement lors du prochain reboot.
On RedHat, SUSE and FEDORA this is done by:

root@racktables-server:/# **chkconfig xinetd on**

```
root@racktables-server:/# netstat -ltn
Connexions Internet actives (seulement serveurs)
Proto Recv-Q Send-Q Adresse locale        Adresse distante      Etat
tcp        0      0 0.0.0.0:22            0.0.0.0:*             LISTEN
tcp        0      0 127.0.0.1:25          0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:6556          0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:5666          0.0.0.0:*             LISTEN
tcp        0      0 0.0.0.0:111           0.0.0.0:*             LISTEN
tcp6       0      0 :::22                 :::*                  LISTEN
tcp6       0      0 ::1:25                :::*                  LISTEN
tcp6       0      0 :::443                :::*                  LISTEN
tcp6       0      0 :::5666               :::*                  LISTEN
tcp6       0      0 :::3306               :::*                  LISTEN
tcp6       0      0 :::111                :::*                  LISTEN
tcp6       0      0 :::80                 :::*                  LISTEN
```

Lorsqu'on essaye de se connecter à la machine client avec le check-mk, la sortie depuis l'agent sur le port 6556 est vide :

```
bdiderot@REC160083:~$ sudo check_mk -d 172.31.131.37
Problem contacting agent: Empty output from agent at TCP port 6556
```

La machine client écoute bien sur le port 6556… Ceci est vérifié avec la commande netstat -an | grep 6556

- ***Vérification de check_mk en local***

1/- ///On checke en local si le check-mk sort bien les infos … Pour ceci, on lance le client directement dans la machine hôte et vérifiez si on récupère les données…

root@racktables-server:/usr/bin# check_mk_agent

2/- lancer le client en local avec localhost :

root@racktables-server:/# telnet localhost 6556

3/- lancer le client en local avec son fqdn :

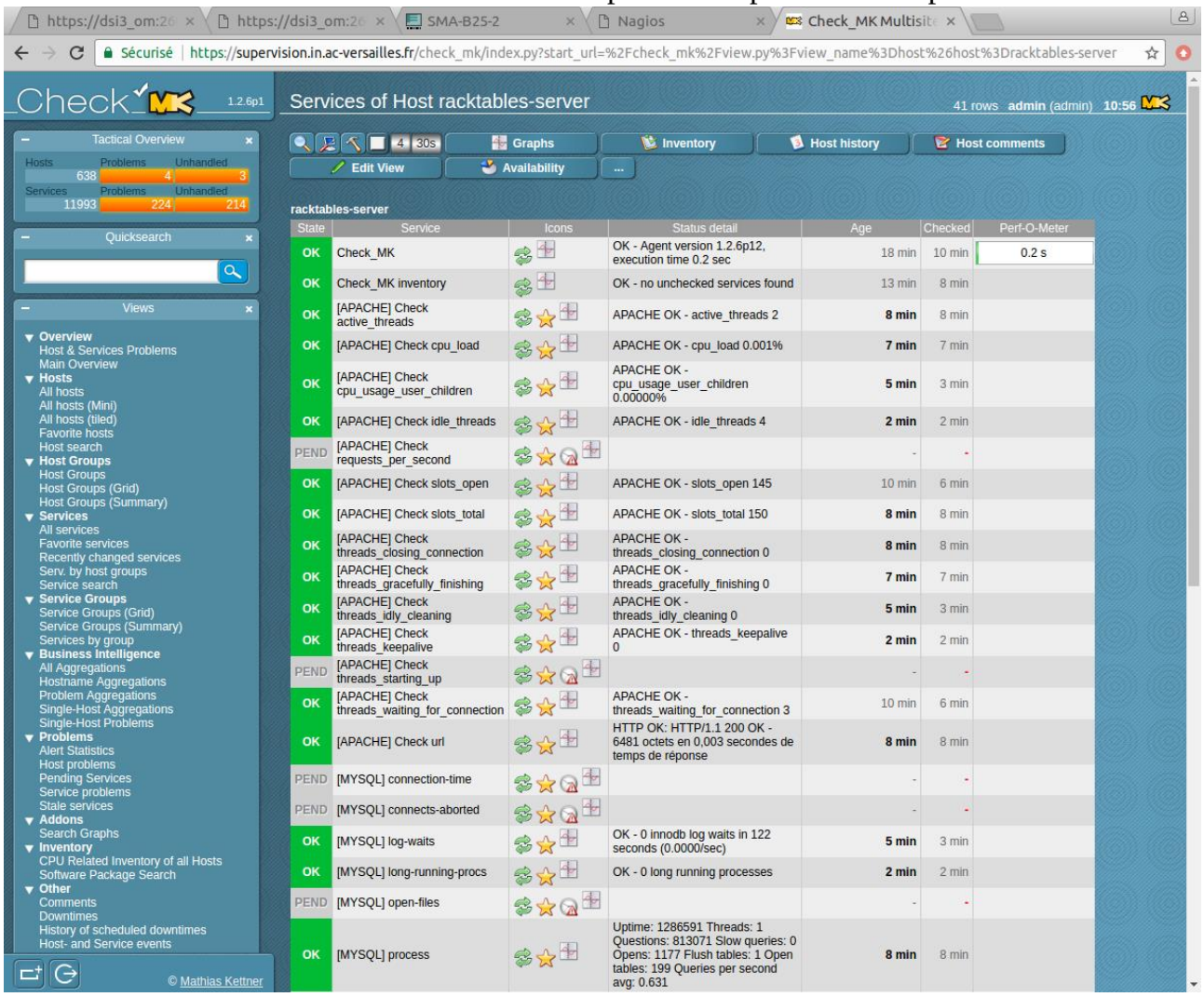
root@racktables-server:/# telnet racktables-server 6556

- On autorise le serveur supervision à accéder au machine client pour récupérer les données. Pour ceci, il faut ajouter le serveur dans le fichier /etc/hosts.allow du machine client :

Voici le fichier /etc/hosts.allow ci-dessous :

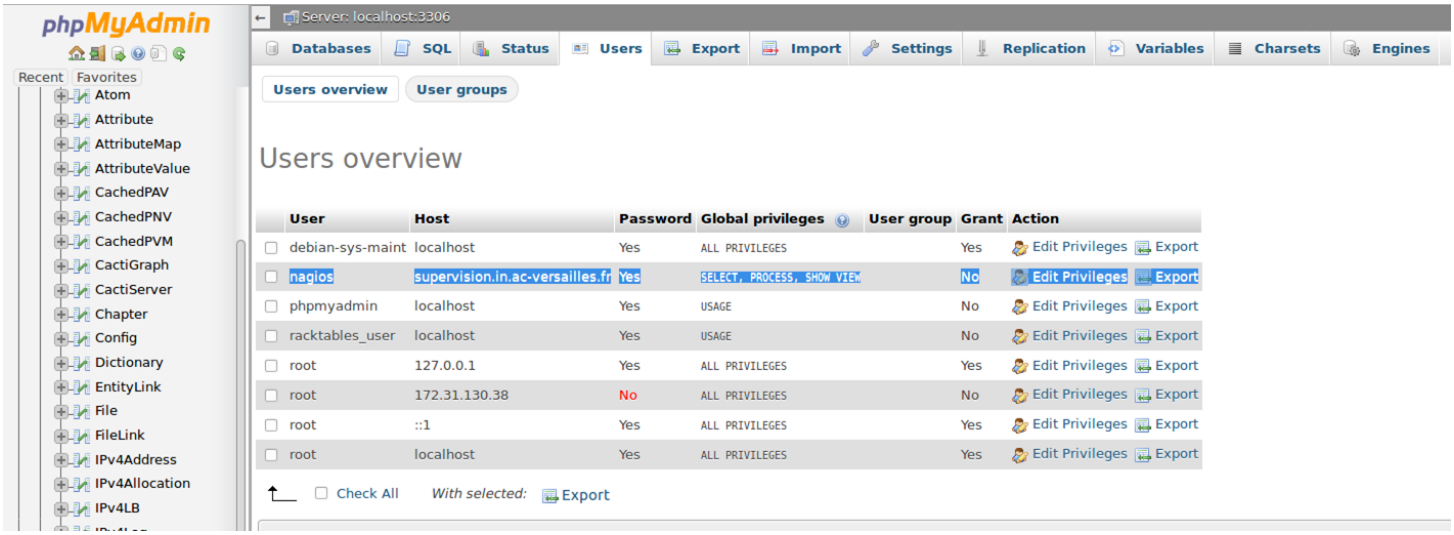
```
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

ALL:172.31.131.135
##@ ip du machine supervision
```

Etat de la machine racktables-server dans la supervision après les manipules :



// créer l'utilisateur nagios ds le mysql pour qu'il accède au bdd et attribuer tous les droits

nagios'@'supervision.in.ac-versailles.fr'



Les commandes nécessaires :

-cmk -II : permet d'inventorier les hôtes et de les configurer automatiquement
-cmk -o : ceci va crée le fichier de configuration nagios pour les hôtes et les services

option : transfert de fichier d'une machine à une autre avec la connexion ssh

scp check-mk-raw-1.2.8p24_0.jessie_amd64.deb root@racktables-server:/home/administrateur