LECORCHE Hubert
JEANDROZ Sylvain

Groupe 8b
Année 2007-2008

SUPERVISION RESEAU AVEC NAGIOS

RAPPORT DE PROJET

Tuteur: R. Protière

Remerciements

Nous tenons à remercier Mr Protiere, notre tuteur, pour son aide dans la compréhension du projet et du cahier des charges.

Merci également à Mr Merleau et Mr Buche qui nous ont prêté tout le materiel nécessaire à la réalisation de ce projet.

Et enfin merci à l'ensemble du corps enseignant pour l'encadrement des projets de deuxième année, qui nous permettent de découvrir la gestion de projets,le travail en groupe et les moyens de communications mis en œuvre.

Sommaire

I – Introduction 5
II – Cahier des charges 6
A – Réseau à superviser 6
B – Règles sur le réseau 6
C – Que superviser? 7
III – Pré-requis 8
A – Choix d'une machine virtuelle 8
B – Choix de Linux 8
C – Installation et configuration des équipements 9
IV – Fonctionnement et installation de Nagios 11
A – Présentation de Nagios 11
B – Fonctionnement de Nagios 11
C – Installation de Nagios 13
D – Interface graphique de Nagios 13
V – Les plugins 14
A – Plugins principaux 14
B – Plugins retenus 15
1. Check_nt 15
2. Check_nrpe 16
3. Check_snmp 17
4. Check_ping 18
VI – Configuration de Nagios 19
VII – Oreon 21
A – Pourquoi Oreon ? 21
B – Installation d'Oreon 21
C – Configuration d'Oreon 22
VIII – Notifications 26
A – E-mail 26
B – SMS 27
IX – Utilisation de Nagios 28
A – Etats du réseau 28
B – Récupération des informations 28
X – Conclusion 30
Bibliographie 31
Annexes 32
MAP 44
Rapports intermédiaires 55

I - Introduction

Actuellement aucune entreprise ne peut se passer d'outils informatiques, et très souvent un réseau informatique de taille plus ou moins importante est mis en œuvre. Le nombre des machines dans ces réseaux peut parfois devenir extrêmement élevé; La maintenance ainsi que la gestion de ces parcs informatiques deviennent alors des enjeux cruciaux, d'autant plus qu'une panne du réseau peut parfois avoir des conséquences catastrophiques.

C'est pourquoi les administrateurs réseau font appel à des logiciels de surveillance et de supervision de réseaux. Ces logiciels vérifient l'état du réseau ainsi que des machines connectées et permettent à l'administrateur d'avoir une vue d'ensemble en temps réel de l'ensemble du parc informatique sous sa responsabilité. Il peut être aussi informé (par e-mail, par SMS) en cas de problème. Grâce à un tel système, les délais d'interventions sont fortement réduits.

Plusieurs logiciels réalisent ces taches, comme par exemple Websense, Tivoli, Observer, Hp Openview, Ciscoworks, Patrol et d'autres, mais certains sont payants.

Dans ce domaine, un logiciel fait office de référence: Nagios. En effet Nagios est très performant et possède une prise en main assez intuitive. Il s'installe sur une machine possédant un système d'exploitation Linux, mais peut superviser aussi bien des machines Linux que Windows. Cet outil permet également une supervision des équipements réseaux (routeur, switch), ce qui est primordial pour l'utilisation que l'on va en faire.

De plus, Nagios est un outil Open source: Chaque société peut l'adapter comme elle lui semble. Puis, la société ne payera pas de licence: Elle ne payera que les frais de formation, d'installation et de maintenance.

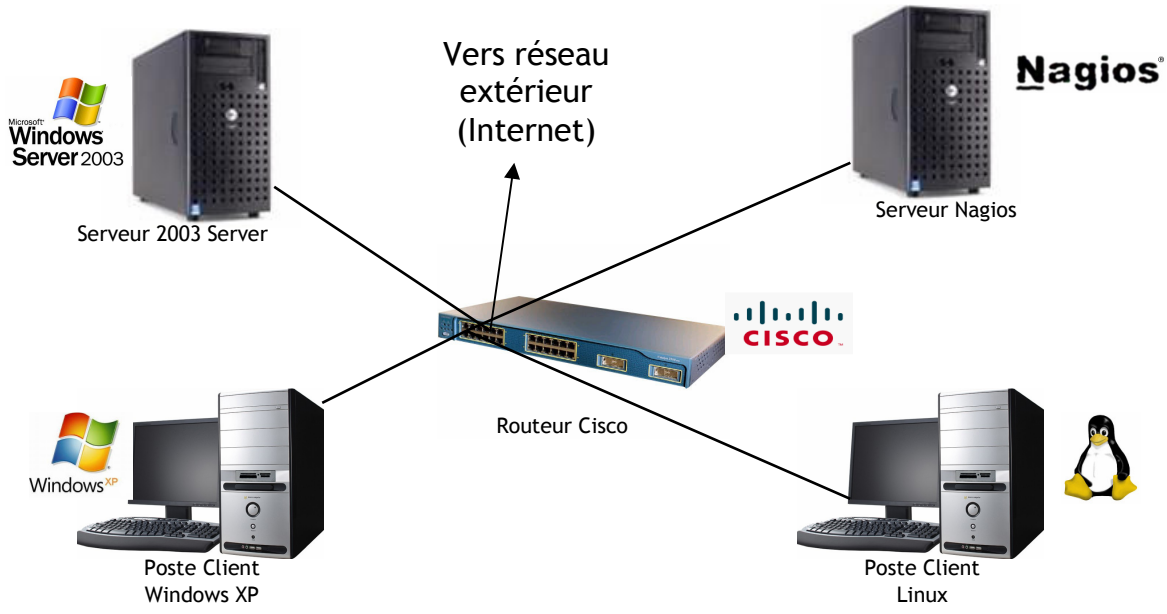
Enfin un autre avantage: Une grosse communauté est réunie autour de ce logiciel, ce qui facilite les recherches de documentations et de réponses à nos questions.

Notre projet consiste donc à superviser un réseau grâce à l'outil Nagios. Ce rapport résumera les trois étapes de notre projet : Compréhension, installation, et utilisation de Nagios.

II - Cahier des charges

A – Réseau à superviser

Le réseau que nous devons superviser est celui-ci :



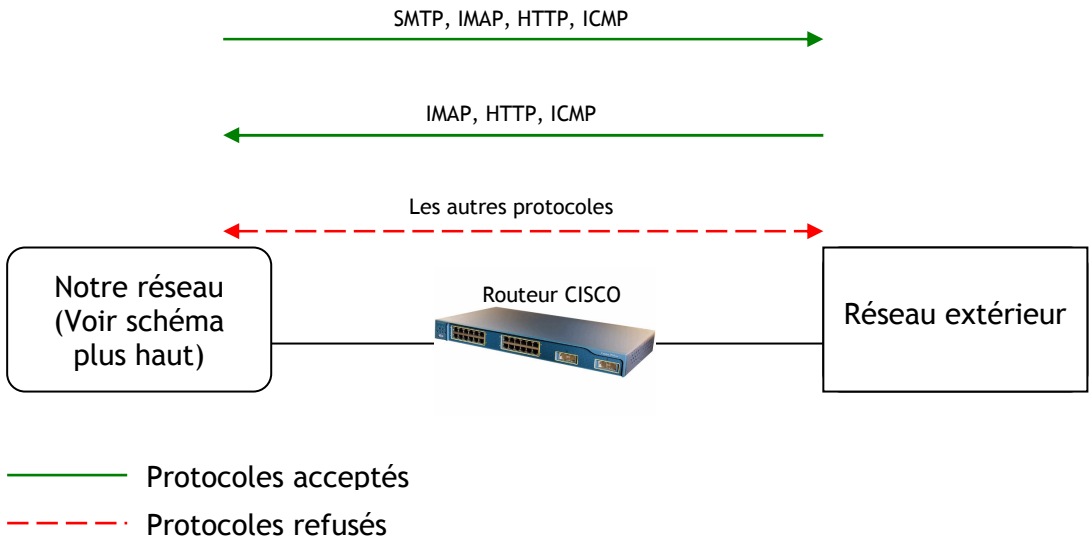
Il sera composé :
- D'un serveur "Windows Server 2003" qui permettra la gestion des utilisateurs du réseau : Stockage des données et identifications des utilisateurs
- D'un serveur "Nagios" qui s'occupera de la supervision du réseau, de la centralisation et de l'analyse des informations du réseau
- D'un poste client "Windows XP"
- D'un poste client "Linux"
- D'un routeur "Cisco" qui permettra de relier les différents équipements du réseau et d'être relié au réseau extérieur (à Internet).

B – Règles sur le réseau

Sur le routeur, un firewall sera configuré grâce à des ACL (Access Control List) permettant l'autorisation ou le refus de certaines connections.

Le firewall devra:
- Autoriser le protocole SMTP (pour l'envoi de mail) sortant mais pas entrant
- Autoriser le protocole IMAP (pour la réception de mail) entrant et sortant
- Autoriser le protocole HTTP entrant et sortant (pour le web)
- Autoriser le protocole ICMP entrant et sortant (pour l'envoi et la réception de PING)
- Refuser tous les autres protocoles dans les deux sens

Pour résumer :



C – Que superviser?

Avant tout, il faut définir les informations qui seront récupérées par Nagios, sur chaque équipement du réseau.

➔ Sur le serveur "2003 serveur", Nagios récupéra :
- La version du pluging qui envoie les informations au serveur Nagios : Si cette version n'est pas la dernière, un e-mail sera envoyé à l'administrateur réseau.
- La charge CPU du serveur. Dans notre cas, si la charge dépasse les 90 %, un e-mail sera envoyé à l'administrateur réseau.
- La durée depuis le dernier démarrage du serveur.
- La taille et l'occupation des disques durs. Dans notre cas, lorsque 90 % de l'un des disques durs est occupé, un e-mail sera envoyé à l'administrateur réseau.

➔ Sur les postes clients, Nagios récupéra :
- La version du pluging qui envoie les informations au serveur Nagios : Si cette version n'est pas la dernière, un e-mail sera envoyé à l'administrateur réseau.
- La taille et l'occupation des disques durs. Dans notre cas, lorsque 90 % de l'un des disques durs est occupé, un e-mail sera envoyé à l'administrateur réseau.
- Pour Windows XP: La taille du dossier "C:\Documents and Settings" qui stocke les données des utilisateurs en local. Si ce dossier a une taille supérieure à 2Go, un e-mail sera envoyé à l'administrateur réseau pour qu'il puisse vider ce répertoire.

➔ Si Internet n'est plus disponible, un SMS sera envoyé à l'administrateur réseau.

➔ Si le routeur ne répond plus (le réseau ne peut donc plus marcher), envoie d'un SMS à l'administrateur réseau.

➔ Nagios doit avoir un historique des paquets rejetés par le firewall (voir les règles d'autorisations / refus dans le paragraphe précédent)

III – Pré-requis

A – Choix d'une machine virtuelle

Après réflexion, nous avons décidé d'utiliser une machine virtuelle sur laquelle nous avons installé Nagios. Une machine virtuelle permet d'utiliser plusieurs systèmes d'exploitation sur une même machine simultanément.

Les avantages d'utiliser une machines virtuelle sont assez nombreux, et correspondent bien aux besoins de notre projet.
Tout d'abord cela nous permet d'avoir un PC de moins dans notre réseau, ce qui est assez agréable d'utilisation étant donné que nous en avons déjà trois (plus les câbles, le hub, le routeur...).

Etant donné que le pc sur lequel est installé la machine virtuelle est le pc personnel d'un membre du groupe, cela nous a permis de travailler très facilement en dehors des séances de projet, et sans avoir à utiliser la salle C274.

Ensuite, il est beaucoup plus aisé de faire des sauvegardes d'une machine virtuelle que d'une machine physique.
Au niveau sécurité, la mobilité de la machine est très intéressante : Si la machine physique tombe en panne, on peut mettre la machine virtuelle très rapidement sur une autre machine physique ; les délais de coupure en cas de panne sont réduits. Ce cas de figure c'est d'ailleurs présenté lors de notre projet: Heureusement il nous a suffit de reprendre notre backup qui était assez récent et ainsi nous avons pu limiter les dégâts.

Dans une société où les équipements réseaux sont très nombreux, les machines virtuelles peuvent faire gagner de la place dans les locaux.

Cette solution permettra également, en fin de projet, de pouvoir récupérer la machine sur laquelle est installé Nagios, qui pourra resservir ultérieurement.

Il existe plusieurs logiciels permettant de créer des machines virtuelles. Un des plus connus est Vmware. Nous n'avons pas retenu ce logiciel pour éviter que l'utilisation de notre serveur Nagios nécessite une licence Vmware, qui est payante.
Notre choix c'est porté sur VirtualBox, développé par InnoTeck. C'est un logiciel à licence gratuite fonctionnant sur les machines hôtes Windows, Linux et Mac OS X, et qui peut supporter Windows (dont Vista) et Linux comme systèmes invités.
De plus, VirtualBox est très simple d'utilisation avec une interface intuitive.

B – Choix de Linux

Le réseau sur lequel nous avons travaillé se compose de cinq équipements:
Un routeur sur lequel tous les équipements sont reliés, une machine possédant Windows 2003 Server qui servira à administrer le réseau, deux machines clientes, une sur Windows XP et une sur Mandriva, et enfin une machine possédant Debian sur laquelle est installé Nagios.

Le choix d'avoir utilisé Debian pour faire fonctionner Nagios n'est pas un hasard. En effet Debian est une version de Linux connu pour sa stabilité. De plus le logiciel Nagios a été développé sur ce même système d'exploitation, donc nous ne craignions pas les problèmes d'incompatibilité.

Lors de l'installation de Debian, nous avons choisi d'utiliser l'interface graphique XFCE. Cette interface a pour particularité d'être très légère et assez pauvre en outils intégrés. Cela nous convient parfaitement pour l'utilisation que nous allons en faire, car cette interface ne nécessite que très peu de ressources, ce qui est une caractéristique cruciale étant donner que nous utilisons une machine virtuelle.

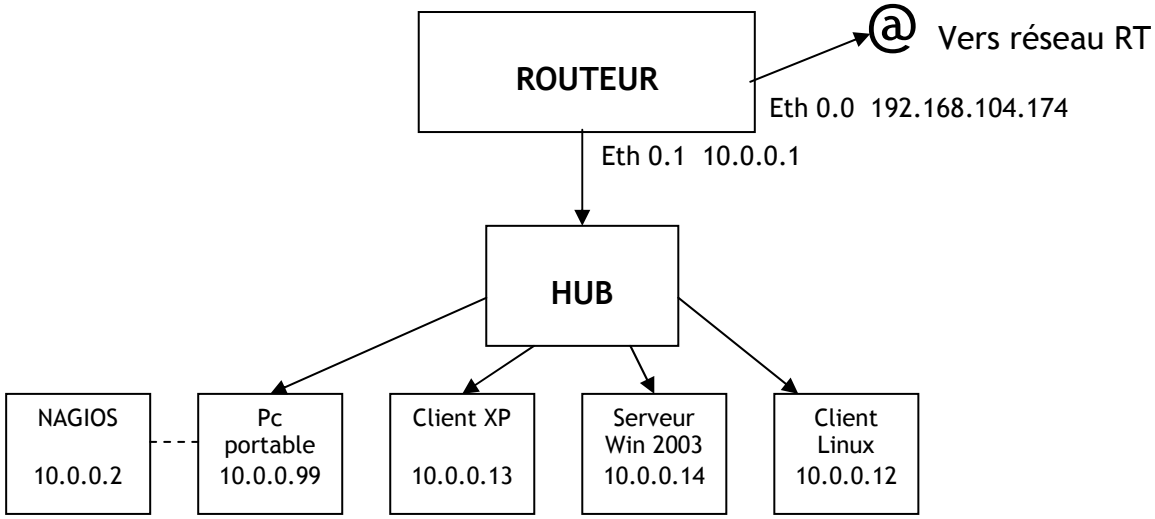
C – Installation et configuration des équipements

La machine fonctionnant sous Windows 2003 Server a pour but d'administrer le réseau. Lors de son installation nous avons mis en place un serveur Active Directory. L'organisation d'Active Directory est la plus simple possible. Nous avons créé un unique utilisateur dans un domaine afin de pouvoir utiliser ce compte avec le client Windows XP.

Ce client XP, nous l'avons introduit dans le domaine créé lors de l'installation d'Active Directory, afin de simuler une organisation d'entreprise (très) simplifiée. Aucun logiciel supplémentaire n'a été installé sur Windows XP car nous n'en avions pas l'utilité.

Concernant le client Linux, nous avons opté pour Mandriva. C'est une version de Linux très répandue dans les entreprises car elle comporte de nombreux logiciels de bureautique intégrés d'origines. C'est une version adaptée aux utilisateurs et donc susceptible d'être rencontré en entreprise.

Enfin le routeur est l'élément sur lequel tous les autres sont reliés et grâce auxquels ils communiquent entre eux et aussi avec les réseaux extérieurs.



Afin de permettre toutes les communications, la configuration suivante a été mise en place sur le routeur:

Tout d'abord nous définissons les adresses IP des deux interfaces du routeur:

- Eth 0/0 : 192.168.104.174
- Eth 0/1 : 10.0.0.1

Puis on active ces deux interfaces.
Il faut activer l'IP Forwarding (transfert de paquets) sur le routeur, puis créer une route par défaut vers le routeur RT dont l'adresse est 192.168.104.5 afin de pouvoir accéder à internet.
Nous devons mettre en place un système de NAT (translation d'adresses) pour convertir les

adresses privées de notre réseau en adresses publiques utilisées sur internet.

Puis, pour respecter le cahier des charges, nous avons mis en place des ACL (Access Control List) pour filtrer le trafic circulant par le routeur.
Nous avons établi une ACL nommée sur chaque interface avec les règles correspondantes au cahier des charges, et également les règles qui correspondent aux réponses des protocoles de l'autre interface.

Nous avons limité l'accès via telnet à nos interfaces en définissant un mot de passe pour les cinq lignes virtuelles, puis en autorisant uniquement les adresses du réseau 10.0.0.0 à se connecter.

Afin de pouvoir superviser le routeur avec Nagios, nous avons du définir des communautés SNMP sur le routeur. Une communauté « public », qui possède uniquement le droit de lecture dans la configuration et l'état du routeur, et une communauté « private » qui possède quand à elle le droit de lecture et d'écriture. Dans le cadre de notre projet, nous n'utiliserons que la communauté public car nous nous servons de Nagios que pour récupérer et surveiller l'état du routeur, et non pas pour l'administrer.

Enfin un système de redirection de ports a été mis en place afin de pouvoir accéder à notre serveur 2003 et à Nagios depuis les réseaux extérieurs au routeur.
Ainsi nous avons redirigé tous les paquets entrants sur le port 443 avec l'adresse de destination 192.168.104.174 vers l'adresse 10.0.0.2 afin d'accéder à l'interface de Nagios depuis internet.
Nous avons préféré utiliser le port 443 qui correspond au protocole HTTPS, au lieu du port 80 afin de sécuriser l'accès depuis l'extérieur à notre interface Nagios.

De même pour les paquets arrivant sur le port 25 avec l'adresse de destination 192.168.104.174 que nous avons redirigé vers l'adresse 10.0.0.14, qui est l'adresse de la machine Windows 2003 Server.

Nous nous sommes également servis du serveur DNS de l'université afin de faciliter la navigation sur internet à partir des postes de notre réseau, bien que cela ne soit pas indiqué dans le cahier des charges. Nous avons donc autorisé les connexions DNS.

Afin d'éviter de refaire cette configuration à chaque séance de projet, nous avons utilisé un serveur TFTP pour sauvegarder cette configuration. Ce serveur est lancé sur le PC hôte de Nagios. Le logiciel utilisé pour crée le serveur tftp est SolarWindws (sur Windows).
Les commandes permettant cette sauvegarde sont les suivantes:

`copy running-config tftp`: copie la configuration courante du routeur vers le serveur tftp. Ensuite le routeur nous demande l'adresse du serveur TFTP (10.0.0.99), puis le nom du fichier dans lequel enregistrer la configuration (r4-rc).
La commande pour restaurer cette configuration est

`copy tftp running-config` pour copier le contenu du fichier sur le serveur tftp dans la configuration courante du routeur.

Le fichier de configuration du routeur est donné en annexe.

IV - Fonctionnement et installation de Nagios

A – Présentation de Nagios

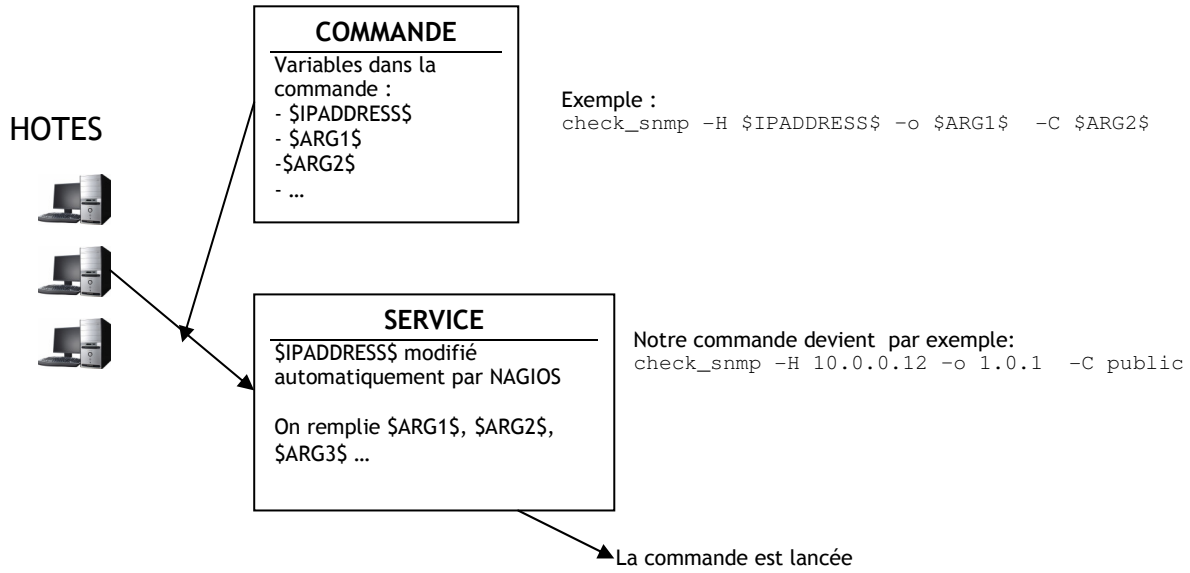
Nagios est un logiciel de supervision de réseau libre sous licence GPL qui fonctionne sous Linux.
Il a pour fonction de surveiller les hôtes et services spécifiés, alertant l'administrateur des états des machines et équipements présents sur le réseau.
Bien qu'il fonctionne dans un environnement Linux, ce logiciel est capable de superviser toutes sortes de systèmes d'exploitation (Windows XP, Windows 2000, Windows 2003 Server, Linux, Mac OS entre autres) et également des équipements réseaux grâce au protocole SNMP.
Cette polyvalence permet d'utiliser Nagios dans toutes sortes d'entreprises, quelque soit la topologie du réseau et les systèmes d'exploitation utilisés au sein de l'entreprise.
Ce logiciel est composé de trois parties:
-Le moteur de l'application, qui gère et ordonnance les supervisions des différents équipements
-Les Plugins qui servent d'intermédiaire entre les ressources que l'on souhaite superviser et le moteur de Nagios. Il faut bien noter que pour accéder à une certaine ressource sur un hôte, il faut un plugin coté Nagios et un autre coté hôte administré.
-L'interface web qui permet d'avoir une vue d'ensemble des états de chaque machine du parc informatique supervisé et ainsi pouvoir intervenir le plus rapidement possible en ciblant la bonne panne.

B – Fonctionnement de Nagios

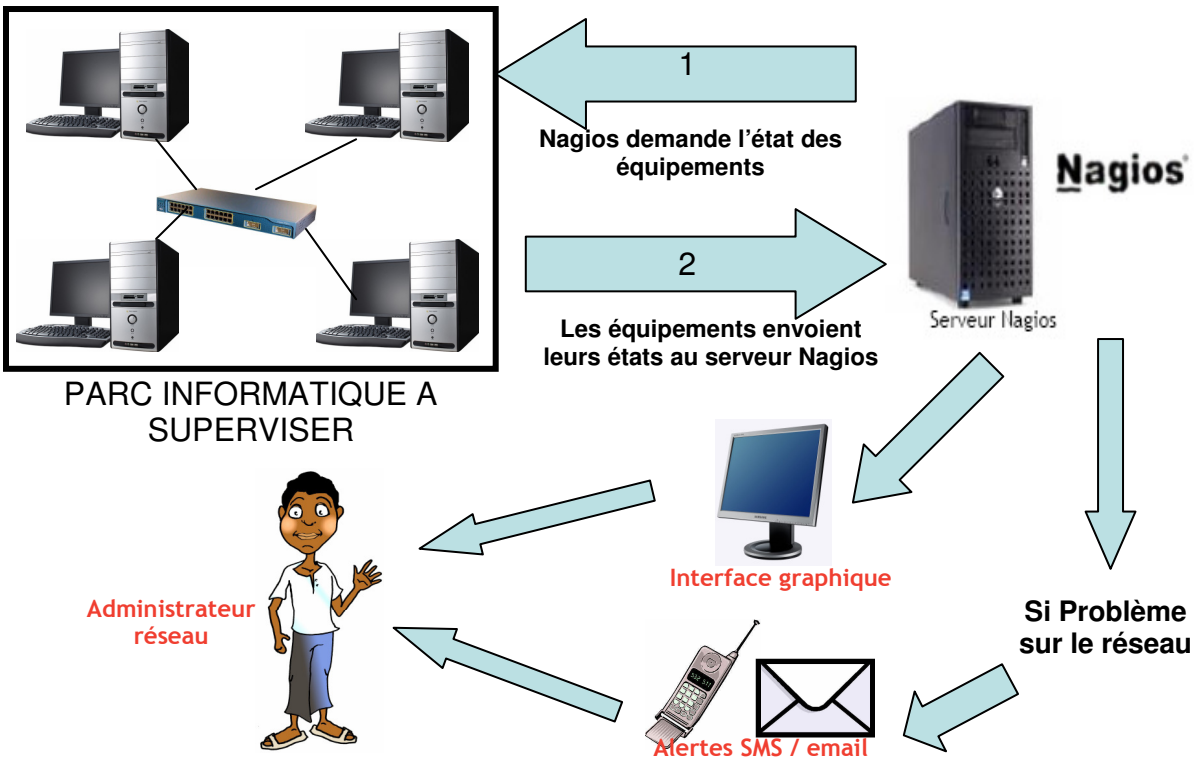
Le principe de supervision de Nagios repose sur l'utilisation de plugins, l'un installé sur la machine qui supporte Nagios, et l'autre sur la machine que l'on souhaite superviser. Un plugin est un programme modifiable, qui peut être écrit dans plusieurs langages possibles, selon les besoins, et qui servent à récupérer les informations souhaitées.
Nagios, par l'intermédiaire de son plugin, contact l'hôte souhaité et l'informe des informations qu'il souhaite recevoir.
Le plugin correspondant installé sur la machine concernée reçoit la requête envoyée par Nagios et ensuite va chercher dans le système de sa machine les informations demandées.
Il renvoi sa réponse au plugin Nagios, qui ensuite le transmet au moteur de Nagios afin d'analyser le résultat obtenu et ainsi mettre à jour l'interface web.
Il existe deux types de récupération d'informations: La récupération active et la récupération passive.
La différence entre les deux types est l'initiative de la récupération. Dans le premier type, à savoir le type actif, c'est Nagios qui a toujours cette initiative. C'est lui qui décide quand il envoie une requête lorsqu'il veut récupérer une information.
Alors que lors d'une récupération passive, l'envoi d'information est planifié en local, soi à partir d'une date, soit en réaction à un événement qui se déroule sur la machine administrée.
Pour notre projet, nous avons décidé d'utiliser le type de récupération active, c'est à dire que Nagios prend l'initiative d'envoyer une requête pour obtenir des informations. Ceci évite donc de configurer les postes à superviser.

La demande d'informations se fait grâce à l'exécution d'une commande de la part de Nagios. Une commande doit obligatoirement comporter des arguments afin de pouvoir

chercher les bonnes informations sur les bonnes machines.
Ces arguments sont l'adresse IP de l'hôte sur lequel aller chercher l'information, la limite de la valeur de l'information recherchée pour laquelle l'état 'attention' sera décidé, idem pour la valeur 'critique', et enfin d'autres options qui varient selon le plugin utilisé.
Pour ne pas devoir à créer une commande par machine supervisée et par information recherchée, nous pouvons remplacer les arguments par des variables, et ainsi réutiliser la commande plusieurs fois, en remplaçant la bonne variable. Nous avons alors la possibilité de travailler avec des services. Lors de la création d'un service, il faut l'associer à un ou plusieurs hôtes puis à une commande.
Ensuite Nagios remplace automatiquement la variable de l'adresse IP dans la commande, grâce à la liste d'hôtes associée au service.
Puis on doit définir manuellement dans le service les autres variables nécessaires à la commande.



Un fois que Nagios à reçu les informations dont il avait besoin sur l'état des hôtes, celui-ci peut construire des notifications sur l'état du réseau, afin d'en informer l'administrateur.
Lorsque Nagios effectue une notification, il attribut des états aux hôtes, ainsi qu'aux services.
Un hôte peut avoir les états suivants:
-Up : en fonctionnement
-Down : éteint
-Inaccessible
-En attente
Les différents états d'un service sont:
- OK
- Attention
- Critique
- En attente
- Inconnu



C – Installation de Nagios

Nous avons installé Nagios en suivant la documentation fournie par Nagios.
Les étapes de l'installation sont fournies en annexe.

Afin de sécuriser l'interface web de Nagios, nous avons mis en place le protocole "HTTPS" (web sécurisé). Ceci permet de crypter les échanges entre le serveur et l'utilisateur.
Pour cela nous avons ajouté un certificat SSL à Apache.

D – Interface graphique de Nagios

Pour accéder à l'interface de Nagios depuis l'extérieur de notre réseau, il suffit de taper dans un navigateur web https://192.168.104.174/nagios/ puis de s'identifier.
Pour y accéder depuis l'intérieur de notre réseau, l'adresse à utilisée est https://10.0.0.2/nagios/
L'identification s'effectue de la même manière.

L'interface graphique de Nagios est utilisée uniquement pour visualiser l'état du réseau supervisé. Cette interface ne peut en aucun cas servir pour la configuration de Nagios.

L'interface se compose d'une partie "menu" à gauche, et une partie centrale, beaucoup plus grande sur le reste de l'écran, qui servira à afficher les informations souhaitées Des captures d'écran sont disponibles en annexe.

Dans le menu, nous retrouvons en premier des liens vers le site de Nagios, et vers la documentation de ce logiciel. Ces liens sont dans la partie 'General'.

Puis une partie 'Monitoring' dans laquelle il est possible de sélectionner les informations que l'on souhaite visualiser. Il y a de nombreux sous-menus dans cette partie ce qui permet

d'afficher vraiment les informations précises qui nous interessent. Il y a également la possibilité de visualiser des statistiques que Nagios a construit, ce qui est très intéressant pour l'administrateur.

Dans la partie "Reporting" il y a la possibilité de créer des rapports et des historiques des évènements qui se sont produits sur le réseau.

Et enfin dans la dernière partie "Configuration", il est possible de visualiser toute les configuration grâce à laquelle Nagios sait qui et quoi superviser.

V - Les plugins

A – Plugins principaux

Nagios possède une importante communauté sur Internet. Grâce à celle-ci, de nombreux utilisateurs ont crées des plugins permettant à Nagios d'aller récupérer des informations sur des équipements du réseau (PC, routeurs, serveurs, …)

Les plugins n'utilisent pas tous le même protocole pour échanger les informations. Le protocole utilisé est dans la plupart des cas un facteur décisif sur le choix des plugins à utiliser.

Un seul plugin Nagios ne peut pas aller chercher toutes les informations sur les équipements du réseau: En effet, chaque plugin n'a accès qu'à certaines informations (exemple: un plugin peut aller chercher l'occupation du disque dur, et un autre l'occupation du processeur d'un PC). Pour superviser un parc informatique, il est donc nécessaire de mettre en place plusieurs plugins.

De plus, certains plugins peuvent aller chercher des informations sur des clients uniquement sur certains systèmes d'exploitation (c'est le cas du plugin check_nt qui peut chercher des informations uniquement sur des équipements Windows).

Les principaux plugins utilisés par nagios sont :
- check_disk : Vérifie l'espace occupé d'un disque dur
- check_http : Vérifie le service "http" d'un hôte
- check_ftp : Vérifie le service "ftp" d'un hôte
- check_mysql : Vérifie l'état d'une base de données MYSQL
- check_nt : Vérifie différentes informations (disque dur, processeur …) sur un système d'exploitation Windows
- check_nrpe: Permet de récupérer différentes informations sur les hôtes
- check_ping: Vérifie la présence d'un équipement, ainsi que sa durée de réponse
- check_pop: Vérifie l'état d'un service POP (serveur mail)
- check_snmp : Récupère divers informations sur un équipement grâce au protocole SNMP (Simple Network Management Protocol)

Il est possible de créer son propre plugin. Dans ce cas, il faudra les créer de la sorte que celui renvoie à nagios :
- L'état du résultat (OK, CRITICAL, DOWN, UP, …)
- Une chaine de caractères (pour donner le détail du résultat)

B – Plugins retenus

Après avoir consulté les différents plugins existants, nous avons choisi ceux qui correspondaient à notre cahier des charges.

Nous avons retenus les plugins suivants :
- check_nt
- check_nrpe
- check_snmp
- check_ping

1. Check_nt

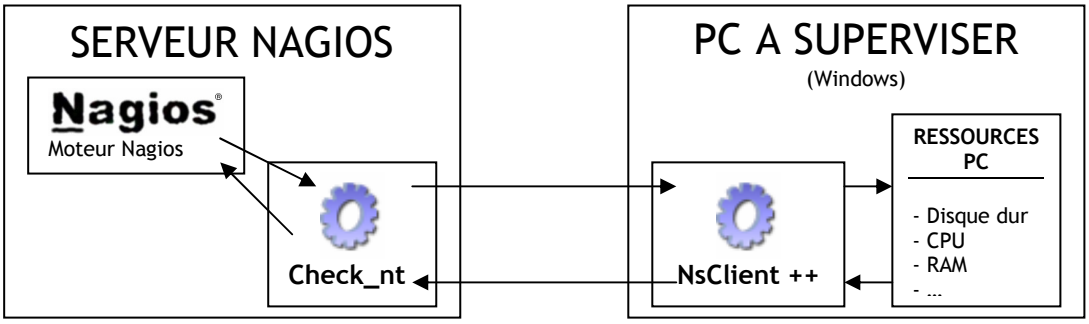
Le plugin Check_nt est un plugin récent qui permet de superviser très facilement des PC dont le système d'exploitation est Windows.

Check_nt permet de récupérer sur un système Windows les informations suivantes : L'espace occupé sur le disque dur, le temps depuis le démrrage de l'ordinateur, la version du plugin NsClient ++ (voir ci-dessous), occupation du processeur, occupation de la mémoire, état d'un service.

Mise en place de check_nt :
1/ Le plugin check_nt est à installer sur la machine NAGIOS. Dans notre cas, check_nt a été installé automatiquement (dans le dossier /etc/usr/local/nagios/libexec) lors de l'installation de Nagios.
2/ Sur les machines à superviser, on doit installer le logiciel NsClient++, téléchargeable sur le site http://sourceforge.net/projects/nscplus
3/ Sur les machines à superviser, on doit configurer le fichier "NSC.ini". C'est dans ce fichier que l'on doit définir :
- Le port sur lequel NsClient++ doit écouter les requêtes
- Les adresses des machines qui ont le droit de dialoguer avec NsClient++ (les machines qui ont le droit de récupérer les informations sur ce poste)
- Un mot de passe (les machines qui souhaiteront dialoguer avec celle-ci par NsClient++ devront fournir ce mot de passe)

=> Le fichier de configuration est fourni en annexe

Fonctionnement de check_nt :



Lorsque Nagios veut connaître une information sur un PC, il exécute le plugin check_nt. Celui envoie une requête au PC. Sur le PC, le programme NsClient++ reçoit la requête, va

chercher les informations dans les ressources du PC et renvoie le résultat au serveur Nagios.

Usage :
Pour aller chercher les informations sur un PC grâce à check_nt, Nagios exécute une commande ayant la syntaxe suivante :

```
check_nt –H host –v variable [–p port] [–w warning] [–c critical][–l params]
```

Avec :
-H : Adresse IP de l'hôte à superviser
-v : ce qu'il faut superviser (ex : CPULOAD
-p : Port sur lequel il faut envoyer la requête
-w : Seuil pour lequel le résultat est considéré comme une alerte
-c : Seuil pour lequel le résultat est considéré comme critique
-l : Paramètres supplémentaires (nécessaire ou non en fonction du paramètre "v")

Pour notre projet, nous utiliserons ce plugin pour superviser tous les postes Windows (client XP + Serveur 2003 Server) sauf pour contrôler l'espace des dossiers des profils des utilisateurs. En effet, ce plugin ne permet pas d'effectuer cette vérification. Nous utiliserons un autre plugin pour cela.

2. Check_nrpe

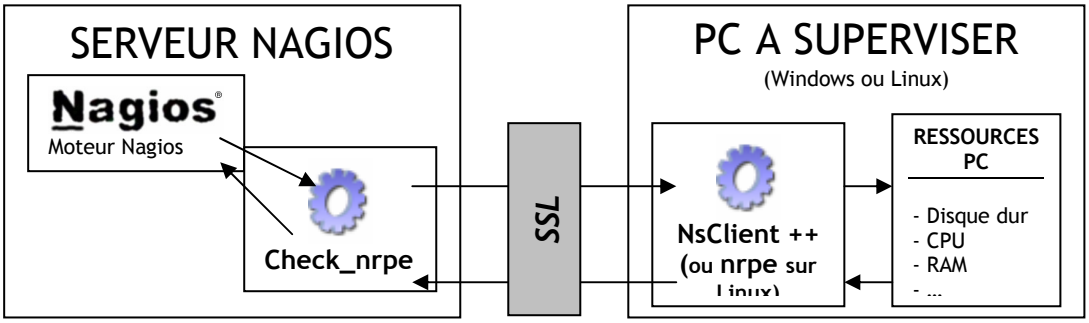
Le plugin Check_nrpe est un plugin qui permet de superviser des PC dont le système d'exploitation est Windows ou Linux.

Check_nrpe utilise une connexion SSL (Secure Socket Layout) pour aller chercher les informations sur les postes. Ceci permet de crypter les trames d'échanges.

Mise en place de check_nrpe (sur Windows) :
1/ Le plugin check_nrpe est à installer sur la machine NAGIOS. Dans notre cas, check_nrpe a été installé automatiquement (dans le dossier /etc/usr/local/nagios/libexec) lors de l'installation de Nagios.
2/ Sur les machines à superviser, on doit installer un logiciel permettant de dialoguer avec check_nrpe. Le programme le plus couramment utilisé est "nrpe pluging". Seulement, le logiciel NsClient++ permet aussi de faire des échanges avec le plugin check_nrpe. Comme nous utilisons déjà ce programme pour check_nt, nous le conservons aussi pour check_nrpe.
3/ Sur les machines à superviser, on doit configurer le fichier "NSC.ini". C'est dans ce fichier que l'on doit définir :
- Le port sur lequel NsClient++ doit écouter les requêtes de check_nrpe (différent de celui check_nt)
- Les adresses des machines qui ont le droit de dialoguer avec NsClient++ (les machines qui ont le droit de récupérer les informations sur ce poste)
=> Le fichier de configuration est fourni en annexe

Mise en place de check_nrpe (sur Linux) :
Même procédure que sur Windows sauf qu'on n'utilise pas NsClient ++ sur l'hôte à superviser mais le programme "nrpe", téléchargeable ici http://www.nagiosexchange.org/
Puis on configure le fichier /etc/nagios/nrpe.conf (le fichier est donné en annexe).

Fonctionnement de check_nrpe :



Lorsque Nagios veut connaître une information sur un PC, il exécute le plugin check_nrpe. Celui envoie une requête au PC. Sur le PC, le programme NsClient++ (ou nrpe si linux) reçoit la requête, va chercher les informations dans les ressources du PC et renvoie le résultat au serveur Nagios.

Usage :
Pour aller chercher les informations sur un PC grâce à check_nrpe, Nagios exécute une commande ayant la syntaxe suivante :

```
check_nrpe –H <adresse de l'hote à superviser> –c <nom de la
commande à excuter sur le serveur>
```

Puis sur les postes à superviser, dans le fichier de configuration (NSC.ini pour Windows, nrpe.conf pour Linux), on doit définir la commande à exécuter pour chaque nom de commande.

Exemple pour Windows :
```
command[check_cpu]=inject checkCPU warn=80 crit=90 5 10 15
```

Exemple pour Linux:
```
command[check_cpu]=/usr/local/nagios/libexec/check_load –w 15,10,5 –
c 30,25,20
```

Ces deux commandes vérifient la charge du processeur.
On remarque alors que la mise en place de nrpe dans une grande entreprise est très complexe car il faut configurer toutes les commandes sur chaque hôte à superviser (contrairement à check_nt qui ne nécessite pas de configuration). En revanche, nrpe offre une meilleure sécurité puisque les échanges client – serveur sont sécurisées (grâce à SSL).

Pour notre projet, nous utilisons check_nrpe pour :
- superviser les clients Linux
- récupérer la taille des dossiers de profils sous Windows

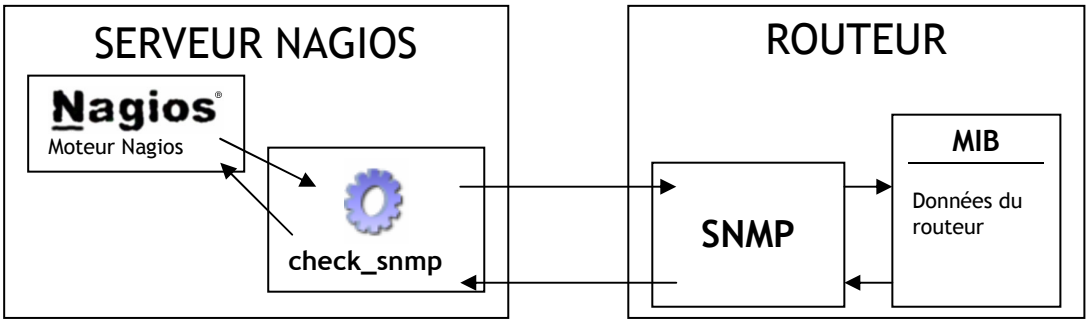
3. Check_snmp

Le plugin Check_snmp est un plugin qui permet de superviser tous les équipements. En revanche, il est très instable pour superviser les PC.

Dans notre projet, nous utiliserons check_snmp pour superviser le routeur.

Mise en place de check_snmp:
1/ Le plugin check_snmp est à installer sur la machine NAGIOS. Dans notre cas, check_snmp a été installé automatiquement (dans le dossier /etc/usr/local/nagios/libexec) lors de l'installation de Nagios.
2/ Il faut configurer le routeur pour qu'il accepte les échanges snmp (voir configuration du routeur en annexe)

Fonctionnement de check_nt :



La MIB (Management Information Base) est une base de données sur le routeur qui stocke toutes les informations de celui-ci (statistiques, débit, état des interfaces…).

Lorsque Nagios veut connaître une information sur le routeur, il exécute le plugin check_snmp. Celui envoie une requête au routeur. Le routeur reçoit la requête, va chercher les informations dans sa MIB et renvoie le résultat au serveur Nagios.

Usage :
Pour aller chercher les informations sur le routeur grâce à check_snmp, Nagios exécute une commande ayant la syntaxe suivante :

```
check_snmp –H <adresse de l'hote à superviser> –o <adresse de
l'information a récupérer dans la MIB>  –C <communauté SNMP>
```

Pour notre projet, on pensait récupérer directement dans la MIB le nombre de paquets rejetés par les ACLs, dans le routeur. Malheureusement, la documentation de la MIB de notre routeur est très incomplète et nous n'avons pas trouvé où était stockée cette information.
Nous avons donc décidé de récupérer dans la MIB le nombre de paquets ICMP envoyés par le routeur lors d'un " host unreachable" (message d'erreur envoyé à l'expéditeur quand le destinataire n'est pas joignable)

4. Check_ping

Le plugin Check_ping est un plugin qui permet de vérifier qu'un hôte est bien joignable.

Usage :
Pour vérifier qu'un hôte est joignable, Nagios exécute une commande ayant la syntaxe suivante :

```
check_ping -H <adresse de l'hote> -w <temps maxi de
reponse>,<Pourcentage de réussite des pings> -c <temps maxi de
reponse>,<Pourcentage de réussite des pings>
```

Avec:
-w : Seuil pour lequel le résultat est considéré comme une alerte
-c : Seuil pour lequel le résultat est considéré comme critique

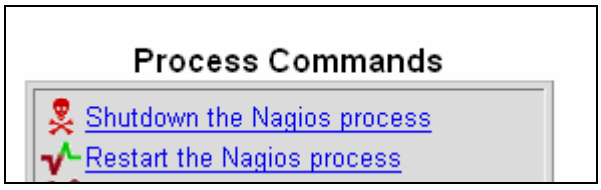
Pour notre projet, on testera la présence du routeur RT (192.168.104.5). En effet, ci celui-ci ne répond plus, on peut considérer que l'on est plus connecté à Internet.

VI - Configuration de Nagios

Les commandes permettant de démarrer, d'arrêter, de recharger Nagios sont les suivantes:
- Démarrer Nagios : `/etc/rc.d/init.d/nagios start`
- Arrêter Nagios : `/etc/rc.d/init.d/nagios stop`
- Recharger Nagios: `/etc/rc.d/init.d/nagios reload`

Après avoir modifié les fichiers de configuration de Nagios, il est très important de recharger Nagios pour que les modifications soient prises en compte.

Il est possible de réaliser ces mêmes commandes, en mode graphique, sur l'interface de Nagios :



Pour respecter notre cahier des charges, nous devons configurer dans Nagios :
- les hôtes à superviser
- les groupes d'hôtes
- les commandes de supervision
- les services de supervision
- les contacts (les personnes qui reçoivent les alertes)

L'interface graphique de Nagios ne permet pas de configurer celui-ci. La seule manière de le configurer, (sans utiliser d'autres outils) est de remplir les fichiers de configurations manuellement (dans le dossier /etc/usr/local/nagios/etc/) :

Exemple de configuration du fichier hosts.cfg (hosts à superviser):

```
define host{
    host_name
Client_XP
    alias                Client XP
    address               10.0.0.13
    hostgroups            Windows
    check_command
nsclient!UPTIME
```

Exemple de configuration du fichier services.cfg (services):

```
define service{
    hostgroup_name
Windows
    service_description        CPU
    check_command
nsclient!CPULOAD!-l 5,80,90
    max_check_attempts            10
    normal_check_interval         1
```

```
    max_check_attempts              10
    check_period             24x7
    contact_groups             email
    notification_interval          1
    notification_period          24x7
    notification_options
d,r,f
    notifications_enabled          0
}
```

Exemple de configuration du fichier hostgroup.cfg (groupes d'hosts):

```
define hostgroup{
    hostgroup_name
Windows
    alias                  Windows
    members
Client_XP, PC_hote_Nagios,
Serveur_Win2003
}
```

Exemple de configuration du fichier checkcommand.cfg (commandes):

```
define command{
    command_name              nsclient
    command_line
$USER1$/check_nt -H $HOSTADDRESS$ -p
12489 -s toto -v $ARG1$ $ARG2$
}
```

```
    retry_check_interval          2
    check_period             24x7
    notification_interval          1
    notification_period          24x7
    notification_options          w,c
    notifications_enabled          0
    contact_groups             email
}
```

Exemple de configuration du fichier contacts.cfg (contacts):

```
define contact{
    contact_name              email
    alias                   email
    contactgroups              email
    host_notification_period
24x7
    service_notification_period
24x7
    host_notification_options
d,u
    service_notification_options
w,u,c
    host_notification_commands
Email
    service_notification_commands
Email
    email
hubert.lecorche@etu.univ-savoie.fr
}
```

On remarque alors que la configuration de Nagios est très complexe pour une grande entreprise. En effet, si le parc informatique à superviser est grand, il faudra du temps pour remplir l'intégralité des fichiers de configuration.
De plus, plus ces fichiers sont grands, plus il sera difficile pour l'administrateur réseau de s'y retrouver.
Comme dans la plupart des cas, on supervise un réseau lorsque celui a une taille assez importante, la configuration de Nagios telle qu'elle sera rarement facile.

C'est pourquoi plusieurs outils ont été crées pour faciliter la configuration de Nagios. C'est le cas d'Oreon, un programme qui se fixe sur Nagios, et qui offre une nouvelle interface graphique. Cette nouvelle interface graphique permet de configurer Nagios "en graphique", sans toucher directement les fichiers de configuration.
Nous avons donc décidé de mettre en place cet outil.

VII - Oreon

A – Pourquoi Oreon ?

Oreon est un logiciel qui s'installe par dessus Nagios et qui permet d'améliorer l'interface graphique, mais surtout le très gros avantage d'Oreon est de pouvoir configurer Nagios par l'interface graphique.En effet la configuration de Nagios, qui s'effectue par modification de fichiers de configuration, devient très vite trop complexe lorsque le parc informatique à superviser prend de l'importance.

Le principe de fonctionnement d'Oreon est simple.L'administrateur configure les options de supervisions, hotes, services, plugins, etc grâce à l'interface d'Oreon.
Ensuite toutes les configurations effectuées par l'administrateur sont stockées dans une base de données, mais elle ne sont pas immédiatement appliquées au moteur Nagios. Lorsqu'il veut appliquer ces modifications, il doit relancer Oreon, qui va alors modifier automatiquement les fichiers de configuration de Nagios, grâce aux informations stockées dans la base de données.

De plus, si l'administrateur réseau configure Nagios depuis les fichiers et que celui-ci fait une faute de frappe, Nagios ne pourra pas fonctionner; dans certains cas, l'administrateur peut mettre du temps avant de retrouver son erreur. Oreon évite ce problème car il contrôle les données entrées par l'administrateur avant de les valider.

Cela permet de configurer Nagios avec une interface intuitive, plaisante, et moins complexe que les fichiers de configuration que l'administrateur devait modifier lui même, et en même temps pouvoir visualiser l'état complet du parc informatique. C'est donc un outil complet et indispensable lorsque le parc informatique à gérer devient complexe, comme cela est très souvent le cas dans les entreprises.

B – Installation d'Oreon

Oreon nécessite une base de données, il faut donc la créer. Nous avons choisi d'utiliser une base de données Mysql sur Apache.

Ensuite nous pouvons commencer l'installation d'Oreon :
Tout d'abord il faut décompresser dans le répertoire /tmp tous les fichiers d'installation d'Oreon, que l'on a pu télécharger sur: http://www.oreon-project.org/

install.ssh

Des lors, l'installeur va poser un certain nombre de questions concernant les emplacements des différents fichiers, quelques avertissement sur certains fichiers qui risquent d'être effacés. Pour la plupart des questions, il faut conserver la réponse par défaut.

Nous avons configuré l'interface web d'Oreon de la même manière que celle de nagios: Nous avons activé le SSL pour des raisons de sécurité.

Ensuite Oreon va installer ses plugins, puis pour finaliser l'installation, il faut se rendre sur l'interface graphique, à l'adresse https://192.168.104.174/oreon depuis internet ou https://10.0.0.2/oreon depuis le réseau local.

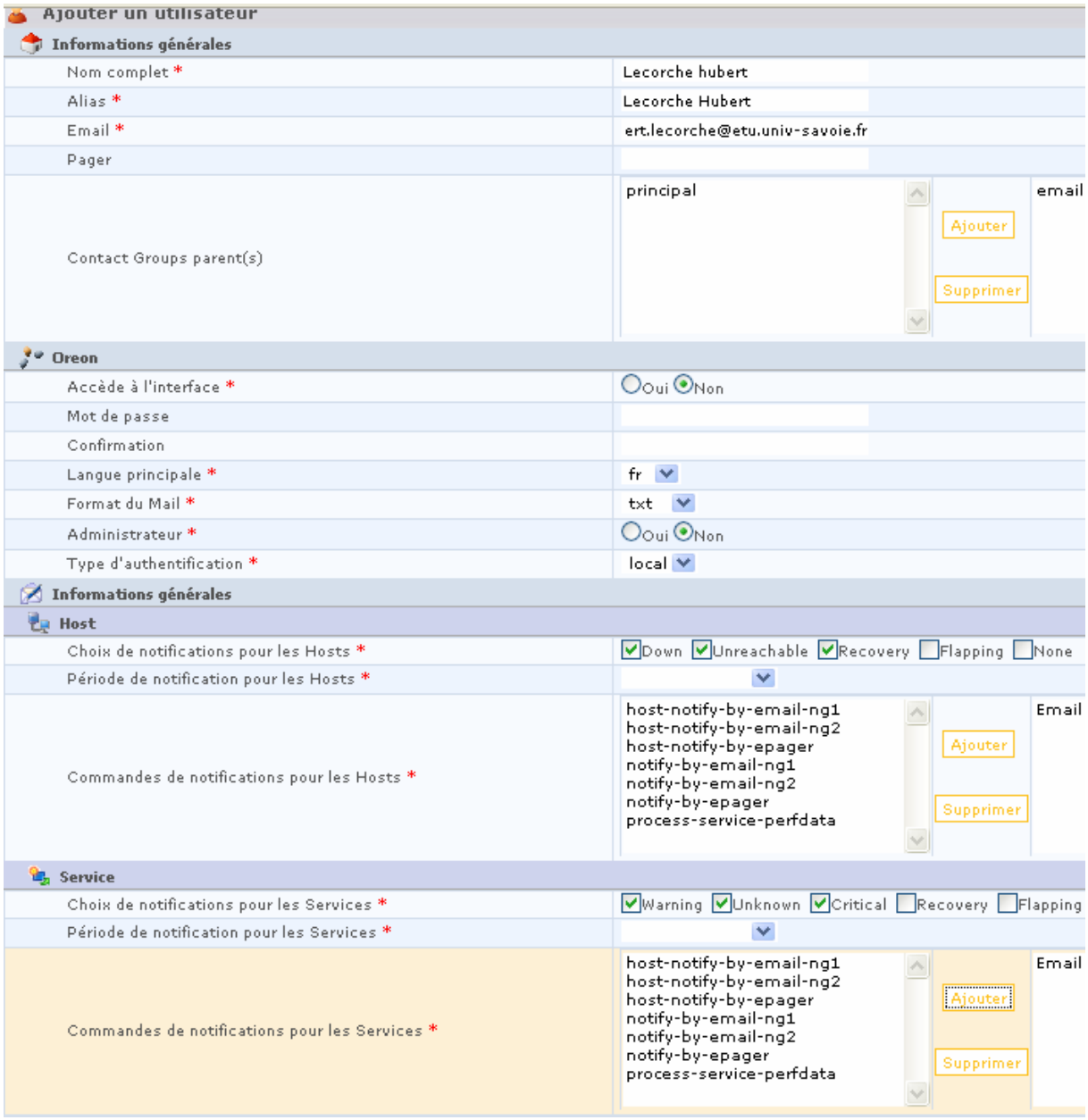
Une fois sur l'interface, il faut vérifier que tous les composants soient bien installés, puis

attribuer les mots de passes et les login pour accéder à l'interface et à la base de données.

C – Configuration d'Oreon

Pour pouvoir accéder à l'interface d'Oreon et pour pouvoir recevoir les alertes, nous avons crée 3 utilisateurs :
- L'utilisateur **nagiosadmin** : C'est l'administrateur : Il a accès à l'interface d'Oreon pour pouvoir visualiser l'état du réseau ainsi que pour configurer Nagios en mode graphique
- L'utilisateur **email** : Cet utilisateur ne peut pas se connecter à l'interface d'Oreon. En revanche c'est cette personne qui recevra les alertes par e-mail.
- L'utilisateur **sms** : Cet utilisateur ne peut pas se connecter à l'interface d'Oreon. En revanche c'est cette personne qui recevra les alertes par sms.

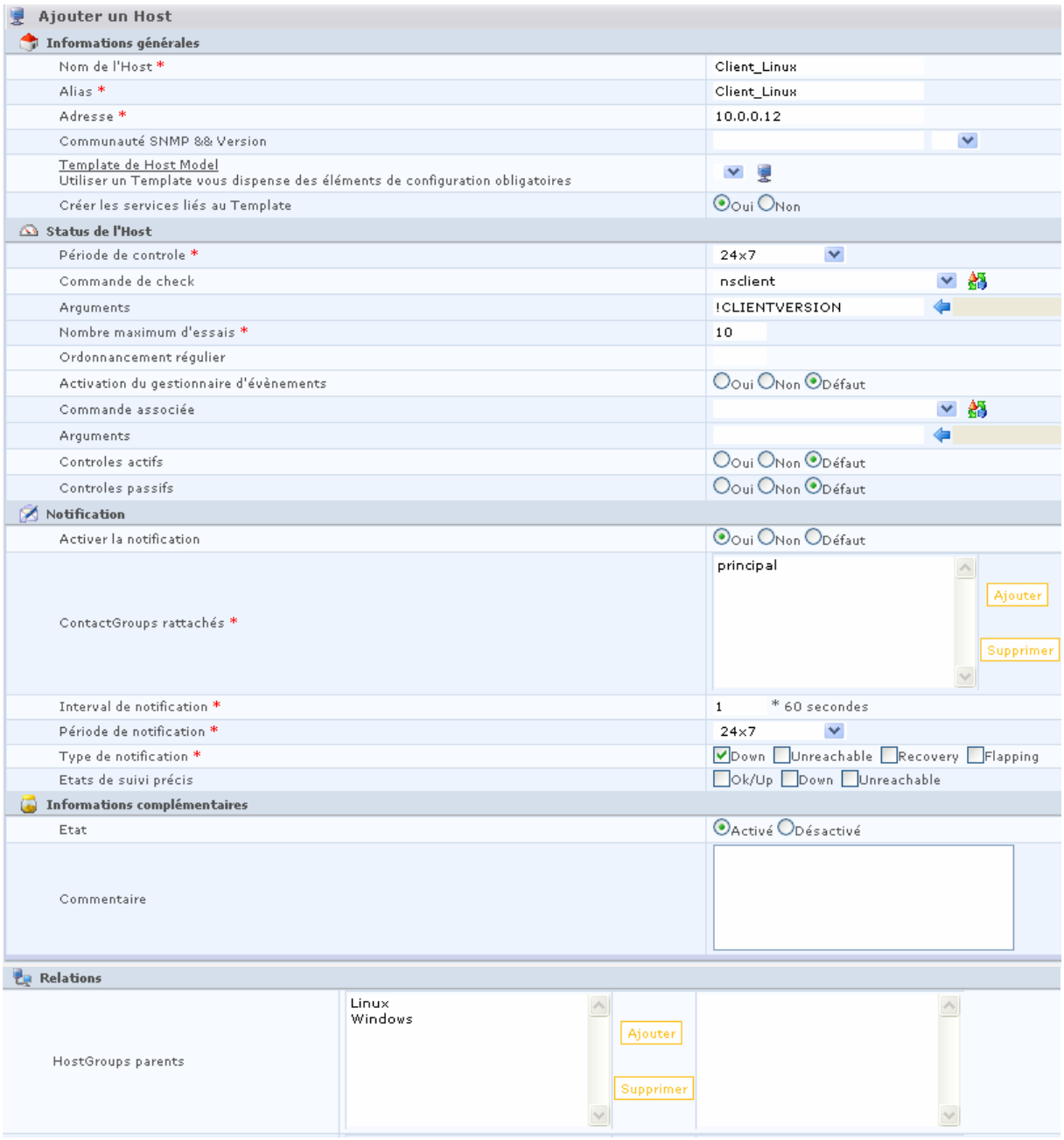
L'ajout d'un utilisateur sur Oreon est très simple :



Il faut crée ensuite déclarer les hôtes à superviser.

Chaque hôte possède un service principal. Ce servir principal permet de définir si l'hôte est allumé ou éteint: En effet, si l'hôte ne répond pas au service, il est soit éteint ou soit injoignable (problème de câblage par exemple).
Pour les Pc à superviser, nous avons choisi "la version du plugin" comme service.

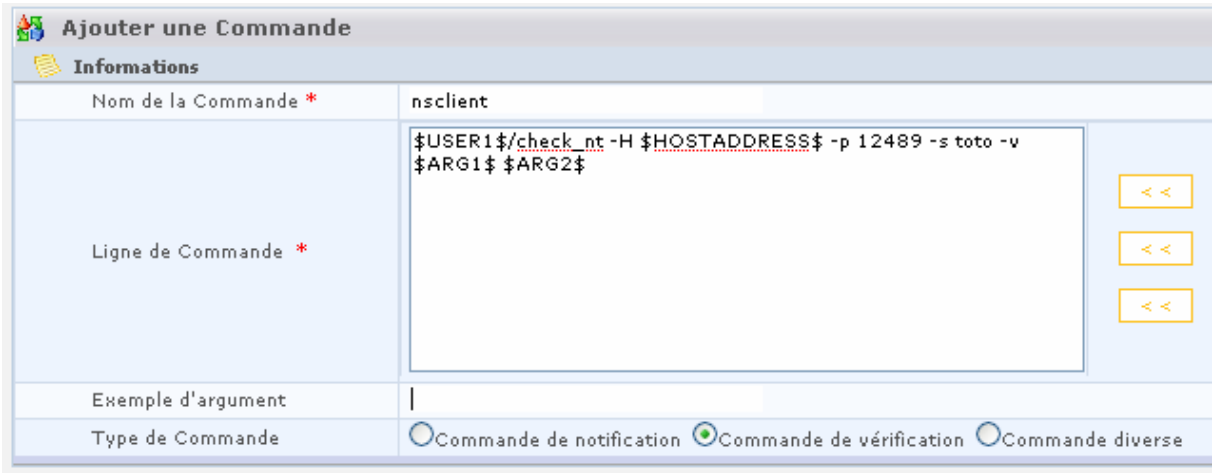
Dans l'exemple ci-dessous, la période de contrôle est fixée à "24x7" : L'hôte sera supervisé 24h/24h, 7j/7. Nous avons décidé que tous nos hôtes seront supervisés en permanence, car ceci permet d'avoir un historique sur celui-ci (on peut savoir exactement combien d'heures par semaine il est allumé, par exemple).



Il faut définir toutes les commandes de vérification / notification.

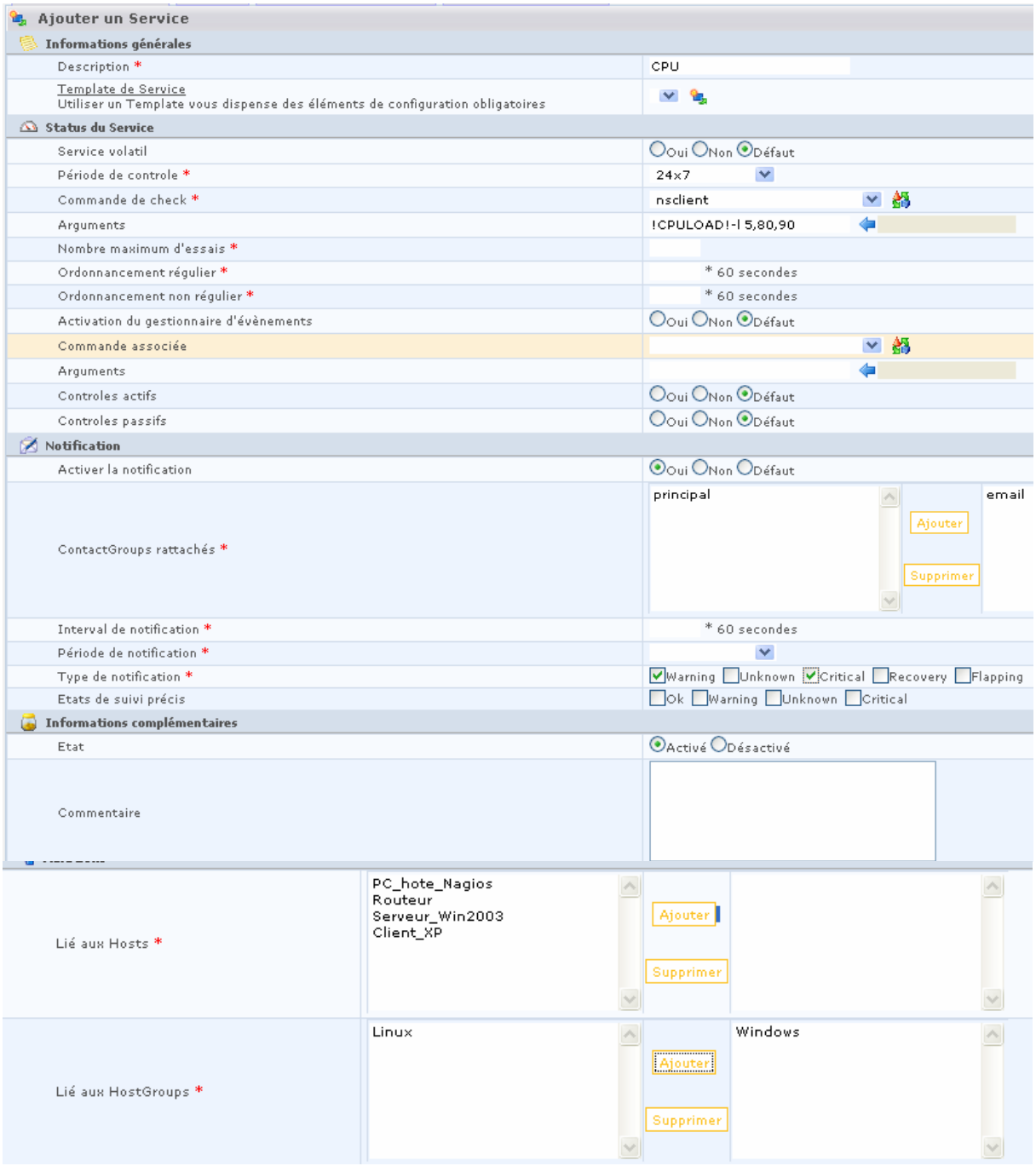
Dans l'exemple ci-dessous, la variable $USER1$ correspond à l'emplacement où sont stockés les plugins nagios.

Quand cette commande sera associée à un service, la variable $HOSTADDRESS$ sera automatiquement remplacée par l'adresse IP de l'hôte à superviser. Il faudra préciser dans le service les valeurs de $ARG1$ et $ARG2$.



Nous avons ensuite crée les services.
Les services doivent avoir une commande (dans notre exemple ci-dessous : nsclient) et doivent être associés à des hôtes (les hôtes dont lesquels on supervisera avec ce service).
Il faut ensuite donner les valeurs des variables de la commande. Dans le paragraphe précédent, nous avons crée la commande "nsclient" qui possedait 2 variables à définir.
Nous définissons ces deux variables dans "Arguments".

Dans l'exemple, $ARG1$ = **`CPULOAD`** et $ARG2 = **`-l 5,80,90`** (chaque variable commence par le symbole "!")



Enfin, une fois que la configuration est faite, il faut régénérer les fichiers de configuration de Nagios. En effet, toute la configuration crée jusqu'à présent a été stockée dans une base de données mais n'était pas effective dans Nagios. Il faut donc transférer cette configuration dans les fichiers de configuration Nagios.

Grace à cet outil, Odeon crée lui-même, à notre place, les fichiers de configuration cités dans le paragraphe "Installation et configuration de Nagios".

Après avoir comparé entre la configuration de Nagios faite en remplissant manuellement les fichiers de configuration puis entre la configuration de Nagios faite par l'interface d'Oreon nous confirmons que la deuxième méthode est beaucoup plus facile. Elle permet à l'administrateur de mieux se repérer, de gagner du temps et d'éviter des erreurs.

VIII - Notifications

A – E-mail

La première façon d'alerter l'administrateur réseau en cas de problème sur le réseau, est l'envoi d'un e-mail.
Pour pouvoir envoyer un e-mail, nous avons utilisé le programme "Swaks". Ce programme permet d'envoyer des mails en ligne de commande, sans configuration.

L'installation de cet outil a été réalisée grâce à la commande suivante :

```
apt-get install swaks
```

Nous avons configuré Oreon pour que celui-ci envoie un mail parmi deux modèles :
⇨ Un modèle en cas de problème sur un équipement (mais celui-ci est toujours joignable).
Dans ce cas, Oreon exécute la commande suivante (qui envoie un mail) :

```
swaks -t $CONTACTEMAIL$ -f hubert.lecorche@etu.univ-
savoie.fr -s mail.annecy.univ-savoie.fr --header
"Subject: Alerte Nagios" -body "***** Oreon
Notification *****\n\nNotification Type:
$NOTIFICATIONTYPE$\n\nService: $SERVICEDESC$\nHost:
$HOSTALIAS$\nAddress: $HOSTADDRESS$\nState:
$SERVICESTATE$\n\nDate/Time: $DATE$ Additional Info :
$SERVICEOUTPUT$"
```

Voici un exemple de mail reçu :

```
***** Oreon Notification *****
Notification Type: PROBLEM
Service: Disque C
Host: PC hote Nagios
Address: 10.0.0.99
State: CRITICAL
Date/Time: 13-03-2008 Additional Info : c:\ - total:
54,84 Gb - utilisee: 51,75 Gb (94%) - libre 3,09 Gb
(6%)
```

⇨ Un modèle si l'équipement n'est plus joignable
Dans ce cas, Oreon exécute la commande suivante (qui envoie un mail) :

```
swaks –t $CONTACTEMAIL$ –f hubert.lecorche@etu.univ-
savoie.fr –s mail.annecy.univ-savoie.fr --header
"Subject: Alerte Nagios" –body "***** Oreon
Notification *****\n\nType:$NOTIFICATIONTYPE$\nHost:
$HOSTNAME$\nState: $HOSTSTATE$\nAddress:
$HOSTADDRESS$\nInfo: $HOSTOUTPUT$\nDate/Time: $DATE$"
```

Voici un exemple de mail reçu :

```
***** Oreon Notification *****
Type:PROBLEM
Host: Serveur_Win2003
State: DOWN
Address: 10.0.0.14
Info: Aucun chemin d'acces pour atteindre l'hÃ´te
cible
Date/Time: 13-03-2008
```

Les trames d'envoi en mail passent en clair sur le réseau. Nous avons utilisé le serveur mail de l'IUT car celui ne demande pas d'authentification (donc pas de circulation de mot de passe en clair sur le réseau). Toutefois, l'outil swaks peut gérer l'authentification à un serveur.

B – SMS

D'après notre cahier des charges, nous devons envoyer un SMS lorsque :
- Le routeur ne répond plus
- Le réseau n'a plus accès à Internet

A notre échelle, ne plus avoir accès à Internet correspond à ce que le routeur du département RT (192.168.104.5) ne répond plus. Nous testons sa présence grâce à l'envoi de Ping.

Dans les deux cas d'envoi d'un SMS, nous ne pouvons pas passer par une passerelle internet puisque nous n'y avons plus l'accès. Nous avons donc du réfléchir à une autre solution.

Il existe un outil, **smstools**, permettant d'envoyer un SMS grâce à téléphone portable directement relié au serveur (avec un câble USB ou un sable série). Nous avons maintenu cette solution.

Une fois cet outil installé, son utilisation est très simple: Il suffit de créer un fichier dans le répertoire /var/spool/sms/outgoing . Le fichier doit avoir la forme suivante :

```
To: 33612345678


Hello World
```

Une fois le fichier crée, le SMS est envoyé directement.
Malheureusement, nos téléphones portables personnels ne sont pas compatibles avec ce programme.
Nous avons quand même installé ce programme ; Nagios crée le fichier dans le dossier d'envoi en cas d'alerte.

Il suffit de connecter un téléphone portable compatible et l'alerte par SMS est active.

IX - Utilisation de Nagios

A – Etats du réseau

Une fois Oreon configuré et les fichiers de configuration régénérés, on peut visualiser l'état du réseau en temps réel depuis l'interface d'Oreon.
Nous avons aussi la possibilité de consulter l'état du réseau depuis l'interface de Nagios.
Cependant ces deux interfaces offrent les mêmes outils pour consulter le réseau.
Or nous avons vu précédemment que l'interface d'Oreon permettait de consulter le réseau mais aussi de configurer Nagios (contrairement à l'interface de Nagios).
De plus Oreon offre une interface plus conviviale que celle de Nagios.
Donc l'interface graphique de Nagios est inutile et peut être supprimée ; l'administrateur peut parfaitement se contenter de l'interface d'Oreon.

Lors de notre projet, nous avons peu utilisé l'interface graphique de Nagios.
Des impressions écrans des interfaces sont disponibles en annexe.

B – Récupération des informations

Une fois la supervision démarrée, il est intéressant de comprendre le fonctionnement des plugins : Comment le serveur Nagios échange-t-il les informations avec les hôtes à superviser ? Ceci a été la dernière partie de notre projet.

⇨ **Echanges SNMP** :

Voici l'exemple d'une trame lorsque le serveur Nagios demande une information au routeur, par protocole SNMP :

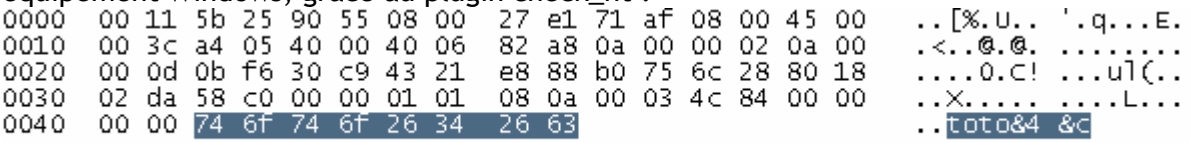
```
0000  00 1e 7a a6 00 23 08 00  27 e1 71 af 08 00 45 00   ..z..#.. '.q...E.
0010  00 47 00 00 40 00 40 11  26 a4 0a 00 00 02 0a 00   .G..@.@. &.......
0020  00 01 05 d1 00 a1 00 33  6c 5c 30 29 02 01 00 04   .......3 l\0)....
0030  06 70 75 62 6c 69 63 a0  1c 02 04 65 9f a9 09 02   .public. ...e....
0040  01 00 02 01 00 30 0e 30  0c 06 08 2b 06 01 02 01   .....0.0 ...+....
0050  04 01 00 05 00                                     .....
```

En analysant cette trame, on ne reconnaît qu'un seul paramètre: la communauté (ici égale à "public").
On remarque alors que le nom de la communauté est diffusé en clair. Ceci est problématique si cette communauté offre les droits d'écriture sur le routeur (un pirate peut intercepter le nom de la communauté et accéder au routeur).
Dans notre cas, la communauté "public" n'offre que les droits en lecture.

Dans la trame de réponse, on repère le résultat. Ci-dessous, le résultat est un entier égal à 1 (caractère surligné).

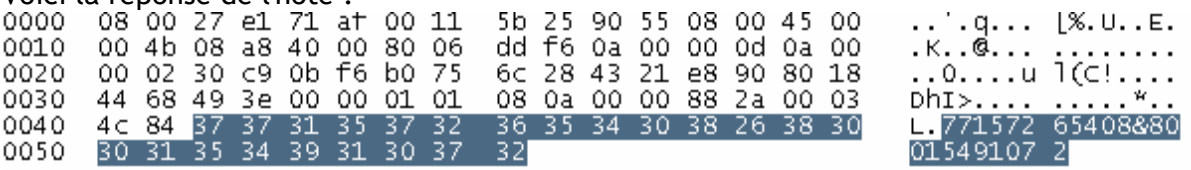
```
0000  08 00 27 e1 71 af 00 1e  7a a6 00 23 08 00 45 00   ..'.q... z..#..E.
0010  00 48 06 6d 00 00 ff 11  a1 35 0a 00 00 01 0a 00   .H.m.... .5......
0020  00 02 00 a1 05 d1 00 34  68 59 30 2a 02 01 00 04   .......4 hY0*....
0030  06 70 75 62 6c 69 63 a2  1d 02 04 65 9f a9 09 02   .public. ...e....
0040  01 00 02 01 00 30 0f 30  0d 06 08 2b 06 01 02 01   .....0.0 ...+....
0050  04 01 00 02 01 01                                  .....
```

⇨ **Echanges avec check_nt** :
Voici l'exemple d'une trame lorsque le serveur Nagios demande une information à un équipement Windows, grâce au plugin check_nt :



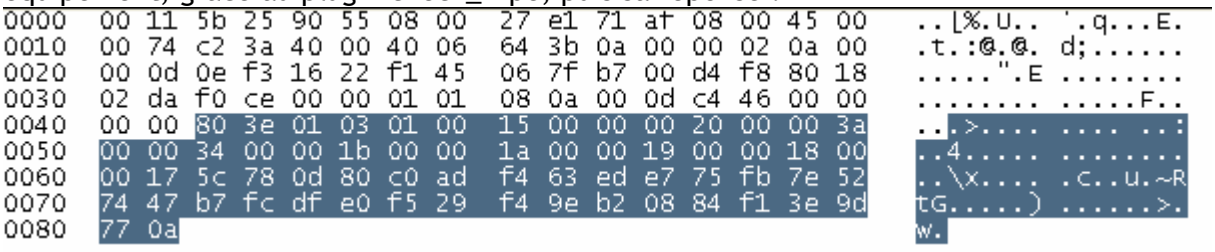
On repère dans la trame ce que demande le serveur Nagios au client (ici &4&c). On repère aussi le mot de passe de check_nt, en clair. On peut donc en conclure que l'utilisation de cet mot de passe est totalement inutile puisque celui circule en clair sur le réseau.

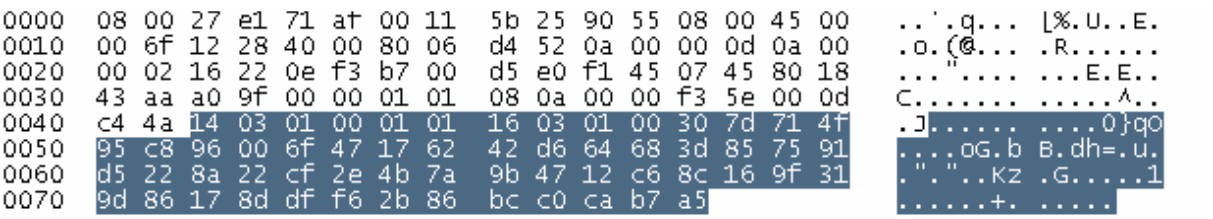
Voici la réponse de l'hôte :



Les résultats observés sont des valeurs numériques.

⇨ **Echanges avec check_nrpe** :
Voici l'exemple d'une trame lorsque le serveur Nagios demande une information à un équipement, grâce au plugin check_nrpe, puis sa réponse :





On remarque que les trames sont incompréhensibles. Ceci est tout à fait normal puisque les échanges sont cryptés avec SSL.

⇨ **Conclusion** :
Pour superviser les PC, nous avons utilisé deux plugins : check_nt et check_nrpe. Nous aurions pu utiliser un seul plugin : check_nrpe (compatible sur Linux et Windows).
Check_nt offre une très grande simplicité mais les trames circulent en clair.
Check_nrpe offre une mise en place assez complexe (car il faut configurer tous les postes à superviser) mais une bonne sécurité.

Pour la mise en place de plugins pour Nagios, l'administrateur réseau devra se poser la question s'il préfère privilégier le cryptage des échanges ou la simplicité.

X - Conclusion

Un logiciel de supervision de réseau comme Nagios est indispensable pour un administrateur lorsque le réseau devient complexe. Cela lui permet d'avoir une vue globale et en temps réel sur tout le parc informatique.

Mais cela nécessite une configuration qui devient elle aussi assez complexe en fonction du niveau de supervision que l'administrateur souhaite mettre en place. En effet, dans Nagios, il n'y a pas d'outils de simplification de la configuration, qui s'effectue entièrement par modification manuelle des fichiers de configuration.

Malheureusement nous avons estimé que la difficulté de configuration de Nagios n'était pas compensée par le service rendu à l'administrateur. Nous avons alors effectué des recherches pour savoir s'il existait des outils qui pourraient simplifier cette configuration.

Après réflexion, nous avons opté pour l'installation d'Oreon. C'est un logiciel qui s'installe par dessus Nagios et qui permet, grâce à son interface graphique, à la fois de visualiser l'état du réseau à la manière de Nagios, mais également de tout configurer en mode graphique.

Oreon agit comme un intermédiaire entre l'administrateur et les fichiers de configuration de Nagios.
Il enregistre dans une base de données les configurations effectuées par l'administrateur, puis il modifie les fichiers de configuration de Nagios en fonction du contenu de la base de données.

Cela permet de simplifier grandement le travail de l'administrateur, contrairement à l'utilisation de Nagios seul.

Pour pouvoir mettre en place la supervision, nous avons du choisir et installer les bons plugins sur les différents équipements. Il faut un plugin sur la machine sur laquelle est installé Nagios, et un autre plugin installé sur l'hôte à superviser.

Après avoir étudié le fonctionnement des différents plugins que nous avons mis en place, nous avons conclu que nous aurions pu utiliser le plugin NRPE à la fois sur Windows et Linux. Celui-ci est assez complexe à configurer mais performant en matière de sécurité.

Une fois tout cela mis en place nous avons pu constater l'efficacité de ce type de logiciel. Bien que le nombre de machines de notre réseau soit très limité, nous nous sommes rendu compte que l'interface de supervision est d'une grande aide. Cela est encore plus vrai dans une grande entreprise.

Mais il ne faut pas croire que l'installation et la configuration de cette solution de supervision soit très aisée à mettre en place au sein d'une entreprise de taille importante, même avec l'aide d'Oreon.
Et surtout il faut sans cesse adapter les configurations en fonction de l'évolution du parc informatique et du cahier des charges de cette solution.

Pour conclure, un projet comme celui-ci se révèle être une solution très intéressante au sein d'une entreprise, mais il ne doit pas être réalisée par n'importe qui, et ne constitue qu'un outil de travail pour un administrateur réseau. Il ne remplace en aucun cas celui-ci…

Bibliographie

- www.nagios.org
  Site officiel de Nagios

- www.centreon.com
  Site officiel d'Oreon

- www.nagiosexchange.org
  Site sur les plugins de Nagios

- www.cisco.com
  Documentation de notre routeur

Annexes

Annexe: Configuration du routeur

```
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
hostname R4
boot-start-marker
boot-end-marker
enable secret 5 $1$FMo9$6.lawEWsZjhnqChR.4FPc1
enable password bonjour
no aaa new-model
ip cef
!
interface FastEthernet0/0
 ip address 192.168.104.174 255.255.255.0
 ip access-group public in
 ip nat outside
 speed auto
 full-duplex
 no mop enabled
!
interface FastEthernet0/1
 ip address 10.0.0.1 255.0.0.0
 ip nat inside
 duplex auto
 speed 10
!
interface Serial0/1/0
 no ip address
 shutdown
 clock rate 125000
!
interface Serial0/1/1
 no ip address
 shutdown
 clock rate 125000
!
ip route 0.0.0.0 0.0.0.0 192.168.104.5
!
no ip http server
ip nat inside source list 22 interface FastEthernet0/0 overload
ip nat inside source static tcp 10.0.0.14 25 192.168.104.174 25 extendable
ip nat inside source static tcp 10.0.0.2 443 192.168.104.174 443 extendable
!
ip access-list standard R0
!
ip access-list extended privee
 permit icmp any any
 permit tcp 10.0.0.0 0.255.255.255 any eq www
 permit udp 10.0.0.0 0.255.255.255 any eq 80
 permit tcp any any eq 443
 permit udp 10.0.0.0 0.255.255.255 any eq domain
 permit tcp 10.0.0.0 0.255.255.255 any eq domain
 permit tcp 10.0.0.0 0.255.255.255 any eq smtp
```

```
 permit tcp any any eq 143
 permit tcp any eq 443 any
 permit tcp any eq 143 any
 evaluate sortant
 permit tcp 10.0.0.0 0.255.255.255 host 10.0.0.1 eq telnet
 permit udp 10.0.0.0 0.255.255.255 host 10.0.0.1 eq snmp
 permit udp 10.0.0.0 0.255.255.255 eq tftp host 10.0.0.1
 permit udp any eq tftp any
 permit udp any any eq tftp
ip access-list extended public
 permit tcp any any eq 443
 permit ip any any reflect sortant
 permit tcp any any eq 143
 deny   ip any any log
!
control-plane
!
line con 0
line aux 0
line vty 0 4
 password bonjour
 login
!
scheduler allocate 20000 1000
end
```

Annexe: Installation de Nagios

Avant de commencer l'installation de Nagios, il faut s'assurer de posséder les droits de « root » sur la machine qui va accueillir Nagios.

Voici les étapes à suivre pour installer correctement Nagios :

- Tout d'abord on télécharge la distribution Nagios sur son site officiel : www.nagios.org
- Ensuite, on extrait la distribution grâce à la commande suivante:

```
tar xzf nagios-version.tar.gz
```

Lorsque la commande aura été exécutée, un dossier nagios-version sera créé dans le répertoire courant. A l'intérieur de celui-ci, il y aura tous les fichiers qui constituent le noyau de la distribution Nagios.

- On crée un utilisateur Nagios, sous lequel sera exécuté le logiciel (c'est cet utilisateur qui aura les droits sur le logiciel)

```
adduser nagios
```

- On crée le répertoire d'installation dans lequel sera installé Nagios

```
mkdir /usr/local/nagios
```

- On change le propriétaire du répertoire d'installation en le remplaçant par l'utilisateur nagios

```
chown nagios.nagios /usr/local/nagios
```

- On ajoute un groupe 'nagcmd' pour les commandes dont les utilisateur Web et Nagios feront partis

```
/usr/sbin/groupadd nagcmd
```

- Ensuite, on ajoute au nouveau groupe créé les utilisateurs Web et Nagios avec les commandes suivantes :

```
/usr/sbin/usermod -G nagcmd apache
/usr/sbin/usermod -G nagcmd nagios
```

- On exécute le script configure pour initialiser les variables et créer un fichier Makefile...(les deux dernières options : --with-command-xxx sont optionnelles mais nécessaires si pour pouvoir utiliser les commandes externes)

```
./configure --prefix=/usr/local/nagios --with-cgiurl=/nagios/cgi-bin--with-htmurl=/nagios/ --with-nagios-user= nagios --with-nagios-group= nagios --with-command-group= nagios
```

- On compile Nagios et les CGIs avec la commande suivante:

```
make all
```

- On installe les binaires et les fichiers HTML (documentation et page web principale) avec la commande suivante :

```
make install
```

-on installe le script d'initialisation /etc/rc.d/init.d/nagios avec la commande suivante :

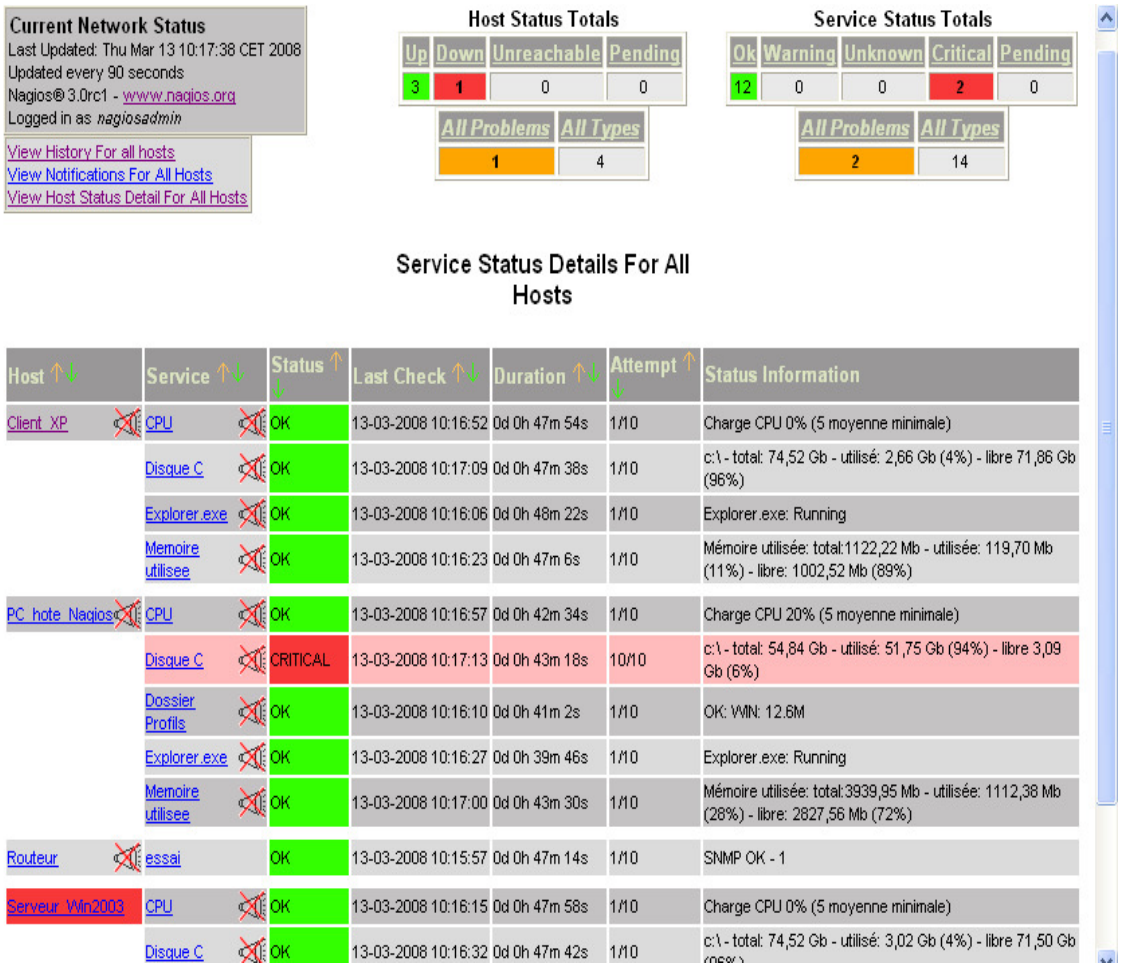
```
make install-init
```

Il faut ensuite spécifier qui peut avoir accès à l'interface Web de Nagios grâce à un fichier htpasswd.users. Voici la commande pour créer ce fichier:

```
htpasswd -c /usr/local/nagios/etc/htpasswd.users nagiosadmin
```

Cette commande crée également une entrée nom d'utilisateur/mot de passe pour nagiosadmin.

Annexe: Interface de Nagios



Current Network Status
Last Updated: Thu Mar 13 10:17:38 CET 2008
Updated every 90 seconds
Nagios® 3.0rc1 - www.nagios.org
Logged in as *nagiosadmin*

View History For all hosts
View Notifications For All Hosts
View Host Status Detail For All Hosts

Host Status Totals

Up	Down	Unreachable	Pending
3	1	0	0

All Problems	All Types
1	4

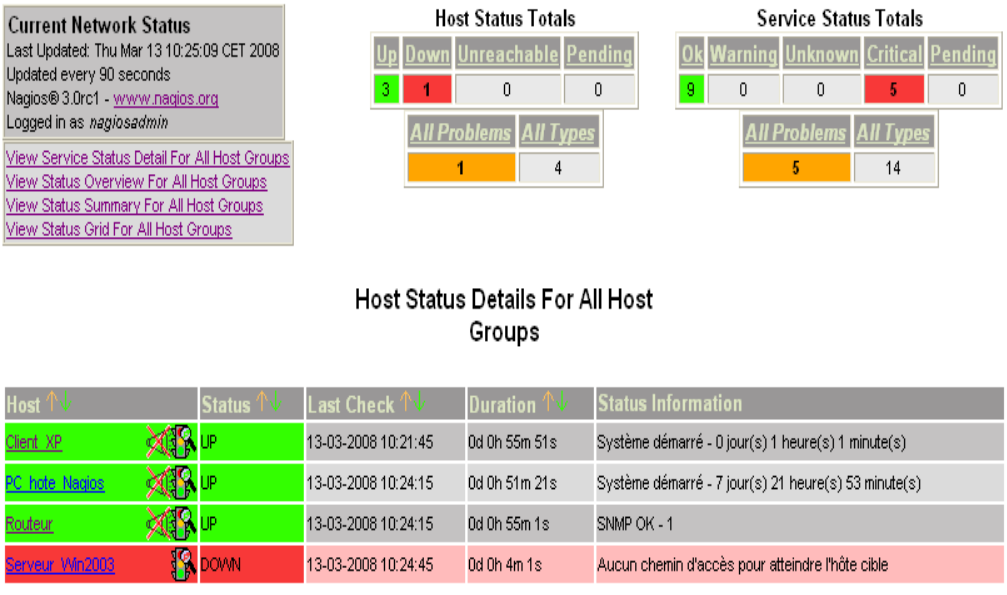
Service Status Totals

Ok	Warning	Unknown	Critical	Pending
12	0	0	2	0

All Problems	All Types
2	14

Service Status Details For All Hosts

Host	Service	Status	Last Check	Duration	Attempt	Status Information
Client_XP	CPU	OK	13-03-2008 10:16:52	0d 0h 47m 54s	1/10	Charge CPU 0% (5 moyenne minimale)
	Disque C	OK	13-03-2008 10:17:09	0d 0h 47m 38s	1/10	c:\ - total: 74,52 Gb - utilisé: 2,66 Gb (4%) - libre 71,86 Gb (96%)
	Explorer.exe	OK	13-03-2008 10:16:06	0d 0h 48m 22s	1/10	Explorer.exe: Running
	Memoire utilisee	OK	13-03-2008 10:16:23	0d 0h 47m 6s	1/10	Mémoire utilisée: total:1122,22 Mb - utilisée: 119,70 Mb (11%) - libre: 1002,52 Mb (89%)
PC_hote_Nagios	CPU	OK	13-03-2008 10:16:57	0d 0h 42m 34s	1/10	Charge CPU 20% (5 moyenne minimale)
	Disque C	CRITICAL	13-03-2008 10:17:13	0d 0h 43m 18s	10/10	c:\ - total: 54,84 Gb - utilisé: 51,75 Gb (94%) - libre 3,09 Gb (6%)
	Dossier Profils	OK	13-03-2008 10:16:10	0d 0h 41m 2s	1/10	OK: WIN: 12.6M
	Explorer.exe	OK	13-03-2008 10:16:27	0d 0h 39m 46s	1/10	Explorer.exe: Running
	Memoire utilisee	OK	13-03-2008 10:17:00	0d 0h 43m 30s	1/10	Mémoire utilisée: total:3939,95 Mb - utilisée: 1112,38 Mb (28%) - libre: 2827,56 Mb (72%)
Routeur	essai	OK	13-03-2008 10:15:57	0d 0h 47m 14s	1/10	SNMP OK - 1
Serveur_Win2003	CPU	OK	13-03-2008 10:16:15	0d 0h 47m 58s	1/10	Charge CPU 0% (5 moyenne minimale)
	Disque C	OK	13-03-2008 10:16:32	0d 0h 47m 42s	1/10	c:\ - total: 74,52 Gb - utilisé: 3,02 Gb (4%) - libre 71,50 Gb



Current Network Status
Last Updated: Thu Mar 13 10:25:09 CET 2008
Updated every 90 seconds
Nagios® 3.0rc1 - www.nagios.org
Logged in as *nagiosadmin*

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
3	1	0	0

All Problems	All Types
1	4

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
9	0	0	5	0

All Problems	All Types
5	14

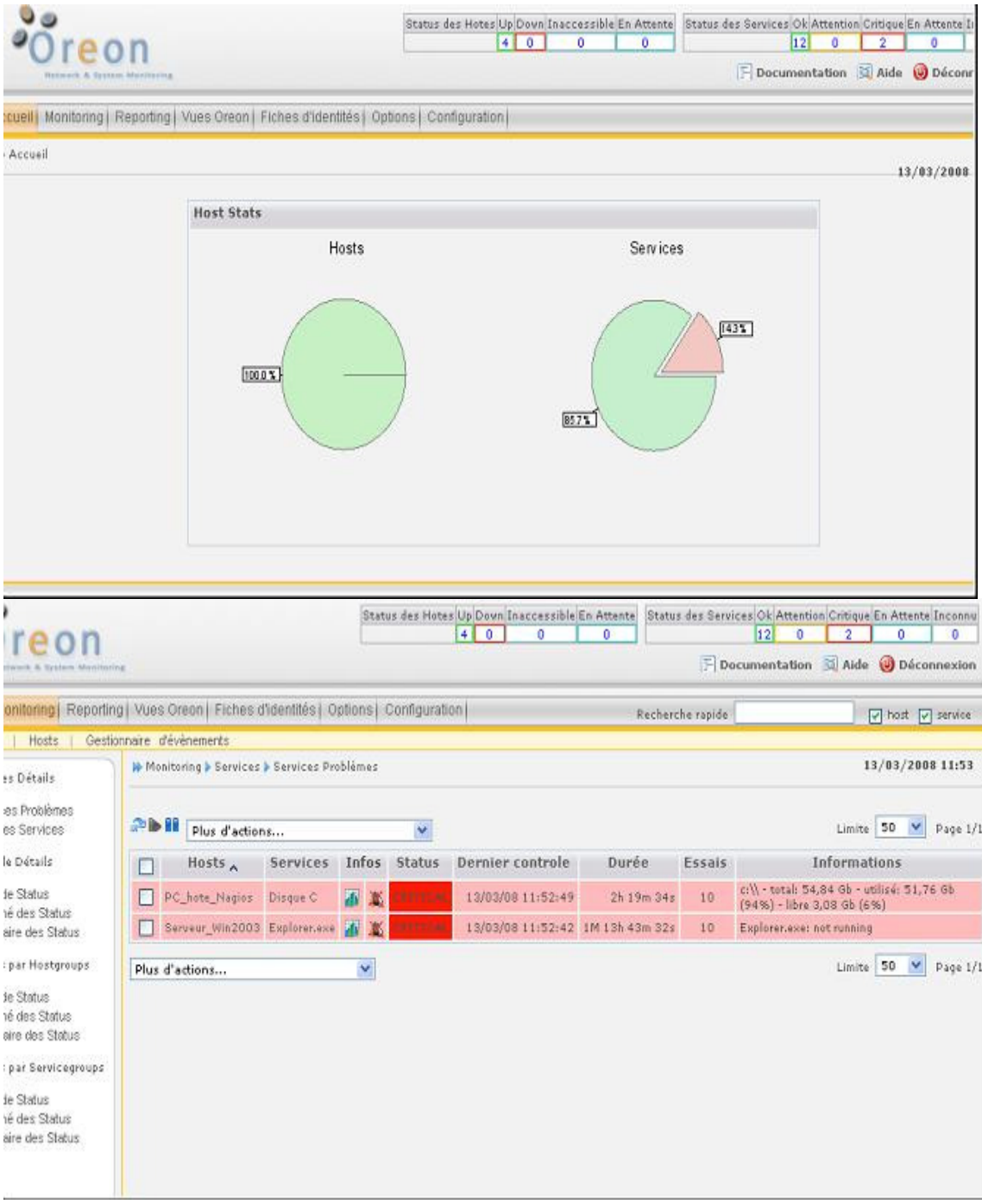
Host Status Details For All Host Groups

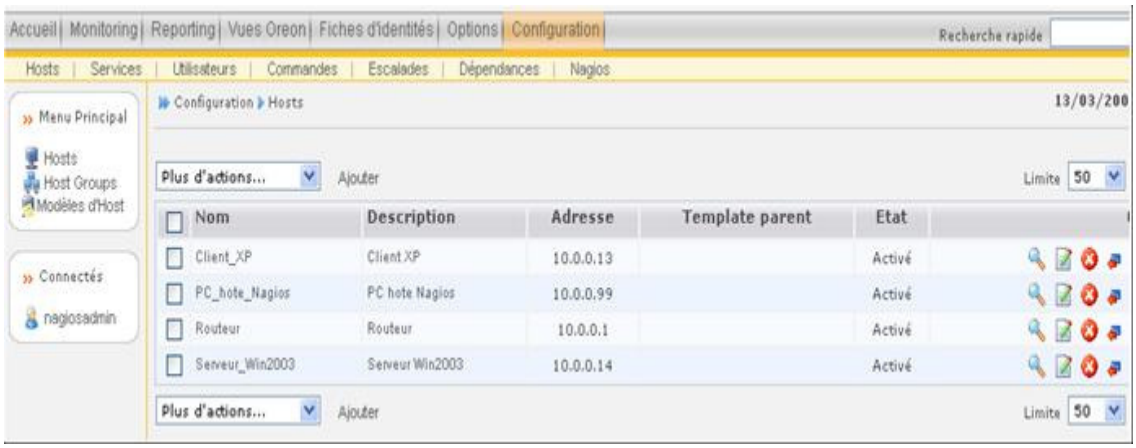
Host	Status	Last Check	Duration	Status Information
Client_XP	UP	13-03-2008 10:21:45	0d 0h 55m 51s	Système démarré - 0 jour(s) 1 heure(s) 1 minute(s)
PC_hote_Nagios	UP	13-03-2008 10:24:15	0d 0h 51m 21s	Système démarré - 7 jour(s) 21 heure(s) 53 minute(s)
Routeur	UP	13-03-2008 10:24:15	0d 0h 55m 1s	SNMP OK - 1
Serveur_Win2003	DOWN	13-03-2008 10:24:45	0d 0h 4m 1s	Aucun chemin d'accès pour atteindre l'hôte cible

4 Matching Host Entries Displayed


Current Event Log
Last Updated: Thu Mar 13 10:27:58 CET 2008
Nagios® 3.0rc1 - www.nagios.org
Logged in as nagiosadmin
Latest Archive
Log File Navigation
Thu Mar 13 00:00:00 CET 2008 to Present..
Older Entries First:
Update
File: /usr/local/nagios/var/nagios.log
March 13, 2008 10:00
[13-03-2008 10:27:24] SERVICE ALERT: Serveur_Win2003;CPU;CRITICAL;SOFT;5;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:27:23] HOST NOTIFICATION: email;Serveur_Win2003;DOWN;Email-hosts;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:27:00] SERVICE ALERT: Serveur_Win2003;Memoire utilisee;CRITICAL;HARD;1;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:26:53] SERVICE ALERT: Serveur_Win2003;Explorer.exe;CRITICAL;SOFT;2;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:26:10] INITIAL SERVICE STATE: Serveur_Win2003;Memoire utilisee;CRITICAL;SOFT;5;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:26:10] INITIAL SERVICE STATE: Serveur_Win2003;Explorer.exe;CRITICAL;SOFT;1;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:26:10] INITIAL SERVICE STATE: Serveur_Win2003;Disque C;CRITICAL;SOFT;1;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:26:10] INITIAL SERVICE STATE: Serveur_Win2003;CPU;CRITICAL;SOFT;4;Aucun chemin d'accès pour atteindre l'hôte cible
[13-03-2008 10:26:10] INITIAL SERVICE STATE: Routeur;essai;OK;HARD;1;SNMP OK - 1
[13-03-2008 10:26:10] INITIAL SERVICE STATE: PC_hote_Nagios;Memoire utilisee;OK;HARD;1;Mémoire utilisée: total:3939,95 Mb - utilisée: 1111,21 Mb (28%) - libre: 2828,74 Mb (72%)
[13-03-2008 10:26:10] INITIAL SERVICE STATE: PC_hote_Nagios;Explorer.exe;OK;HARD;1;Explorer.exe: Running
[13-03-2008 10:26:10] INITIAL SERVICE STATE: PC_hote_Nagios;Dossier Profils;OK;HARD;1;OK: WIN: 12.6M
[13-03-2008 10:26:10] INITIAL SERVICE STATE: PC_hote_Nagios;Disque C;CRITICAL;HARD;10;c:\\ - total: 54,84 Gb - utilisé: 51,75 Gb (94%) - libre 3,09 Gb (6%)
[13-03-2008 10:26:10] INITIAL SERVICE STATE: PC_hote_Nagios;CPU;OK;HARD;1;Charge CPU 22% (5 moyenne minimale)

Annexe: Interface d'Oreon




Accueil
Monitoring
Reporting
Vues Oreon
Fiches d'identités
Options
Configuration
Recherche rapide
Hosts
Services
Utilisateurs
Commandes
Escalades
Dépendances
Nagios
Menu Principal
Hosts
Host Groups
Modèles d'Host
Connectés
nagiosadmin
Configuration ▸ Hosts
13/03/200
Plus d'actions...
Ajouter
Limite 50
Nom
Description
Adresse
Template parent
Etat
Client_XP
Client XP
10.0.0.13
Activé
PC_hote_Nagios
PC hote Nagios
10.0.0.99
Activé
Routeur
Routeur
10.0.0.1
Activé
Serveur_Win2003
Serveur Win2003
10.0.0.14
Activé
Plus d'actions...
Ajouter
Limite 50

Annexe: Configuration de NsClient++

```
[modules]
;# NSCLIENT++ MODULES
;# A list with DLLs to load at startup.
;  You will need to enable some of these for NSClient++ to work.
; ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
; *                                                               *
; * N O T I C E ! ! ! - Y O U   H A V E   T O   E D I T   T H I S *
; *                                                               *
; ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! ! !
FileLogger.dll
CheckSystem.dll
CheckDisk.dll
NSClientListener.dll
NRPEListener.dll
SysTray.dll
CheckEventLog.dll
CheckHelpers.dll
;
; CheckWMI IS AN EXTREM EARLY IDEA SO DONT USE FOR PRODUCTION ENVIROMNEMTS!
;CheckWMI.dll
;
; RemoteConfiguration IS AN EXTREM EARLY IDEA SO DONT USE FOR PRODUCTION
ENVIROMNEMTS!
;RemoteConfiguration.dll

[Settings]
;# OBFUSCATED PASSWORD
;  This is the same as the password option but here you can store the
password in an obfuscated manner.
;  *NOTICE* obfuscation is *NOT* the same as encryption, someone with
access to this file can still figure out the
;  password. Its just a bit harder to do it at first glance.
;obfuscated_password=Jw0KAUUdXlAAUwASDAAB
;
;# PASSWORD
;  This is the password (-s) that is required to access NSClient remotely.
If you leave this blank everyone will be able to access the daemon remotly.
password=toto
;
;# ALLOWED HOST ADDRESSES
;  This is a comma-delimited list of IP address of hosts that are allowed
to talk to the all daemons.
;  If leave this blank anyone can access the deamon remotly (NSClient still
requires a valid password).
;  The syntax is host or ip/mask so 192.168.0.0/24 will allow anyone on
that subnet access
allowed_hosts=10.0.0.0/8
;
;# USE THIS FILE
;  Use the INI file as opposed to the registry if this is 0 and the use_reg
in the registry is set to 1
;  the registry will be used instead.
use_file=1

[log]
;# LOG DEBUG
```

```
;  Set to 1 if you want debug message printed in the log file (debug
messages are always printed to stdout when run with -test)
;debug=1
;
;# LOG FILE
;  The file to print log statements to
file=NSC.log
;
;# LOG DATE MASK
;  The format to for the date/time part of the log entry written to file.
;date_mask=%Y-%m-%d %H:%M:%S


[NSClient]
;# ALLOWED HOST ADDRESSES
;  This is a comma-delimited list of IP address of hosts that are allowed
to talk to NSClient deamon.
;  If you leave this blank the global version will be used instead.
;allowed_hosts=
;
;# NSCLIENT PORT NUMBER
;  This is the port the NSClientListener.dll will listen to.
port=12489
;
;# BIND TO ADDRESS
;  Allows you to bind server to a specific local address. This has to be a
dotted ip adress not a hostname.
;  Leaving this blank will bind to all avalible IP adresses.
;bind_to_address=


[Check System]
;# CPU BUFFER SIZE
;  Can be anything ranging from 1s (for 1 second) to 10w for 10 weeks.
Notice that a larger buffer will waste memory
;  so don't use a larger buffer then you need (ie. the longest check you do
+1).
;CPUBufferSize=1h
;
;# CHECK RESOLUTION
;  The resolution to check values (currently only CPU).
;  The value is entered in 1/10:th of a second and the default is 10 (which
means ones every second)
;CheckResolution=10

[NRPE]
;# NRPE PORT NUMBER
;  This is the port the NRPEListener.dll will listen to.
port=5666
;
;# COMMAND TIMEOUT
;  This specifies the maximum number of seconds that the NRPE daemon will
allow plug-ins to finish executing before killing them off.
command_timeout=60
;
;# COMMAND ARGUMENT PROCESSING
;  This option determines whether or not the NRPE daemon will allow clients
to specify arguments to commands that are executed.
allow_arguments=0
;
;# COMMAND ALLOW NASTY META CHARS
```

```
;  This option determines whether or not the NRPE daemon will allow clients
to specify nasty (as in |`&><'"\[]{}) characters in arguments.
allow_nasty_meta_chars=0
;
;# USE SSL SOCKET
;  This option controls if SSL should be used on the socket.
use_ssl=1
;
;# BIND TO ADDRESS
;  Allows you to bind server to a specific local address. This has to be a
dotted ip adress not a hostname.
;  Leaving this blank will bind to all avalible IP adresses.
; bind_to_address=
;
;# ALLOWED HOST ADDRESSES
;  This is a comma-delimited list of IP address of hosts that are allowed
to talk to NRPE deamon.
;  If you leave this blank the global version will be used instead.
allowed_hosts=10.0.0.0/8
;
;# SCRIPT DIRECTORY
;  All files in this directory will become check commands.
;  *WARNING* This is undoubtedly dangerous so use with care!
;script_dir=scripts\


[NRPE Handlers]
;# COMMAND DEFINITIONS
;# Command definitions that this daemon will run.
;# Can be either NRPE syntax:
;command[check_users]=/usr/local/nagios/libexec/check_users -w 5 -c 10
;# Or simplified syntax:
;test=c:\test.bat foo $ARG1$ bar
;check_disk1=/usr/local/nagios/libexec/check_disk -w 5 -c 10
;# Or even loopback (inject) syntax (to run internal commands)
;# This is a way to run "NSClient" commands and other internal module
commands such as check eventlog etc.
check_cpu=inject checkCPU warn=80 crit=90 5 10 15
;check_eventlog=inject CheckEventLog Application
warn.require.eventType=error warn.require.eventType=warning
critical.require.eventType=error critical.exclude.eventType=info
truncate=1024 descriptions
check_disk_c=inject CheckFileSize ShowAll MaxWarn=500M MaxCrit=4096M
File:WIN=c:\ATI\*.*
check_espace_docs=inject CheckFileSize ShowAll MaxWarn=1500M MaxCrit=2000M
"File:WIN=C:\Documents and Settings\*.*"
;# But be careful:
; dont_check=inject dont_check This will "loop forever" so be careful with
the inject command...
;# Check some escapings...
; check_escape=inject CheckFileSize ShowAll MaxWarn=1024M MaxCrit=4096M
"File: foo \" WIN=c:\\WINDOWS\\*.*"
;# Some real world samples
;nrpe_cpu=inject checkCPU warn=80 crit=90 5 10 15
command[nrpe_ok]=C:\NSClient++\scripts\ok.bat
```

Annexe: Configuration de Nrpe sous Linux

```
#############################################################################
##
# Sample NRPE Config File
# Written by: Ethan Galstad (nagios@nagios.org)
#
# Last Modified: 03-09-2007
#
# NOTES:
# This is a sample configuration file for the NRPE daemon.  It needs to be
# located on the remote host that is running the NRPE daemon, not the host
# from which the check_nrpe client is being executed.
#############################################################################
##
# PID FILE
# The name of the file in which the NRPE daemon should write it's process
ID
# number.  The file is only written if the NRPE daemon is started by the
root
# user and is running in standalone mode.
pid_file=/var/run/nrpe/nrpe.pid
# PORT NUMBER
# Port number we should wait for connections on.
# NOTE: This must be a non-priviledged port (i.e. > 1024).
# NOTE: This option is ignored if NRPE is running under either inetd or
xinetd
server_port=5666
# SERVER ADDRESS
# Address that nrpe should bind to in case there are more than one
interface
# and you do not want nrpe to bind on all interfaces.
# NOTE: This option is ignored if NRPE is running under either inetd or
xinetd
#server_address=127.0.0.1
# NRPE USER
# This determines the effective user that the NRPE daemon should run as.

allowed_hosts=127.0.0.1,10.0.0.2
dont_blame_nrpe=0
debug=0
command_timeout=60

connection_timeout=300

# The following examples use hardcoded command arguments...

command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -w 15,10,5 -c
30,25,20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20 -c 10 -p
/dev/hda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10
-s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c
200
```

LECORCHE Hubert
JEANDROZ Sylvain

Groupe Y2
Année 2007-2008

SUPERVISION RESEAU AVEC NAGIOS

MEMOIRE D'AVANT PROJET

Tuteur: R. Protière

Sommaire

I – Cahier des charges .....
A – Utilisation de Nagios .....
B – Structure du réseau .....
C – Informations récupérées par Nagios et alertes.....
II – Les tâches .....
III – Diagramme de Gantt .....
IV – MPM.....

I - Cahier des charges

A – Utilisation de Nagios

De nos jours, l'informatique étant devenu un outil indispensable dans toutes les professions, les réseaux informatiques se doivent d'être les plus performants et les plus stables possibles.
Lorsque l'architecture des réseaux devient complexe, comme cela est le cas dans la plupart des entreprises, l'administrateur réseau doit utiliser des logiciels de surveillance et de supervision de réseaux.
Ces logiciels lui permettent de détecter rapidement les dysfonctionnements des équipements réseau (switch, hub, routeurs, etc...), ainsi que des machines connectées sur le réseau.
Cela lui permet de remédier aux problèmes dans les délais les plus courts possible, car une panne prolongée d'un réseau informatique est extrêmement pénalisant au sein d'une entreprise.

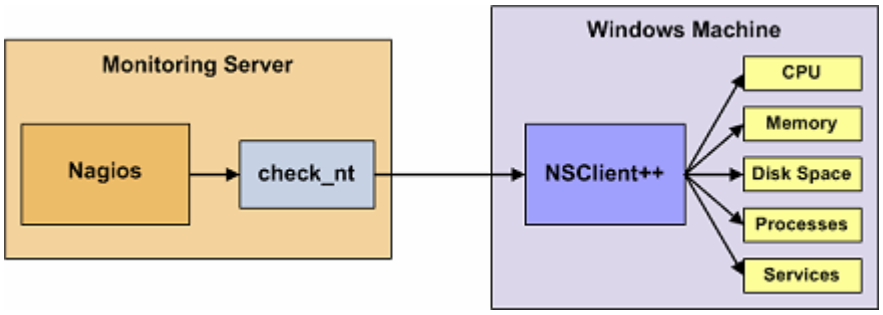
Plusieurs logiciels réalisent ces taches, comme par exemple *websense, tivoli, observer, hp openview, ciscoworks, patrol* et d'autres, mais certains sont payants.

Nous avons décidé d'utiliser Nagios, un logiciel libre de superviseur réseau, fonctionnant sous linux.

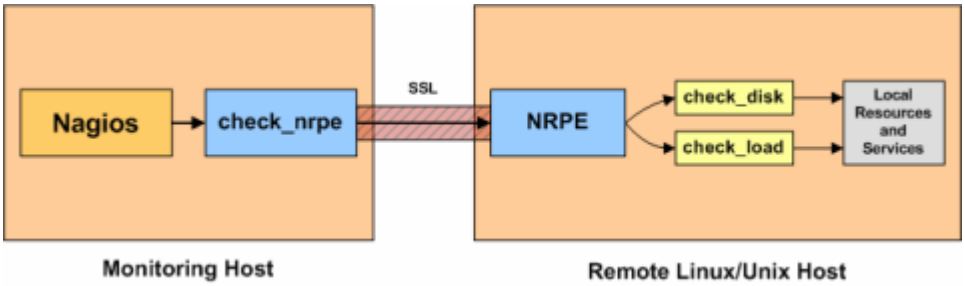
Nagios est un logiciel connu dans le domaine des réseaux, performant et assez simple d'utilisation. Il tourne sur une machine possédant un système d'exploitation UNIX mais il peut superviser des réseaux dont les machines possèdent UNIX et Windows.
La qualité majeur de Nagios est de pouvoir également superviser des équipements réseau grâce aux protocoles SNMP et RMON.

Nagios récupère les informations dont il a besoin (provenants des équipements qu'il supervise), les analyse afin de déterminer s'il y a dysfonctionnement et éventuellement le degré de celui-ci.
Ensuite les informations sont stockées dans une base de données afin de pouvoir faire des statistiques, des comparaisons, etc...

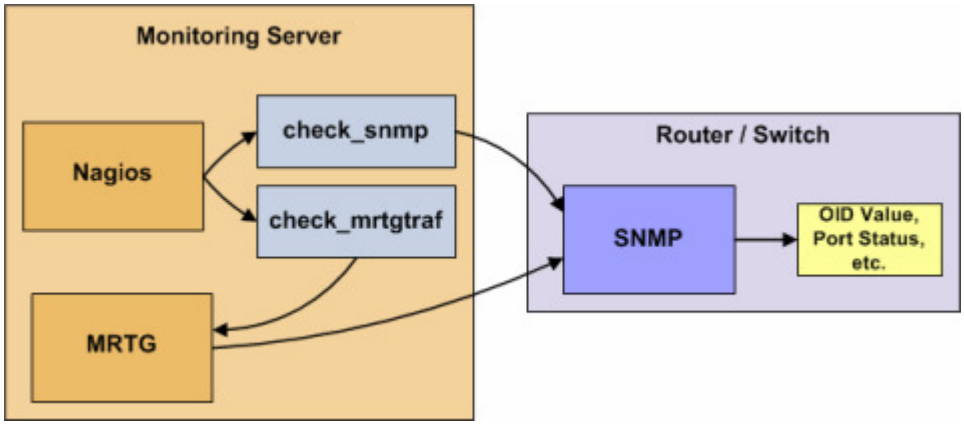
Pour pouvoir récupérer des informations sur les machines clientes Windows, un "client" doit être préalablement installé sur ces dernières. Cet agent agit comme un intermédiaire entre le "plugin" Nagios qui tourne sur le serveur Nagios et le service windows qui concerne les informations désirées.
Par exemple on peut installer *NSCLIENT++* sur windows et utiliser le plugin *check_nt* sur Nagios.



Concernant le monitoring sur les machines UNIX il existe de nombreux moyens de récupérer les informations mais le plus utilisé est d'installer *NRPE addon* sur la machine cliente et permet ainsi la communication avec le plugin de Nagios



Le monitoring des équipements réseau un peu évolués, utilisant SNMP repose en gros sur le même principe que précédemment.
Les plugins Nagios récupèrent les informations des équipements.
MRTG est utilisé pour le monitoring de la bande passante.



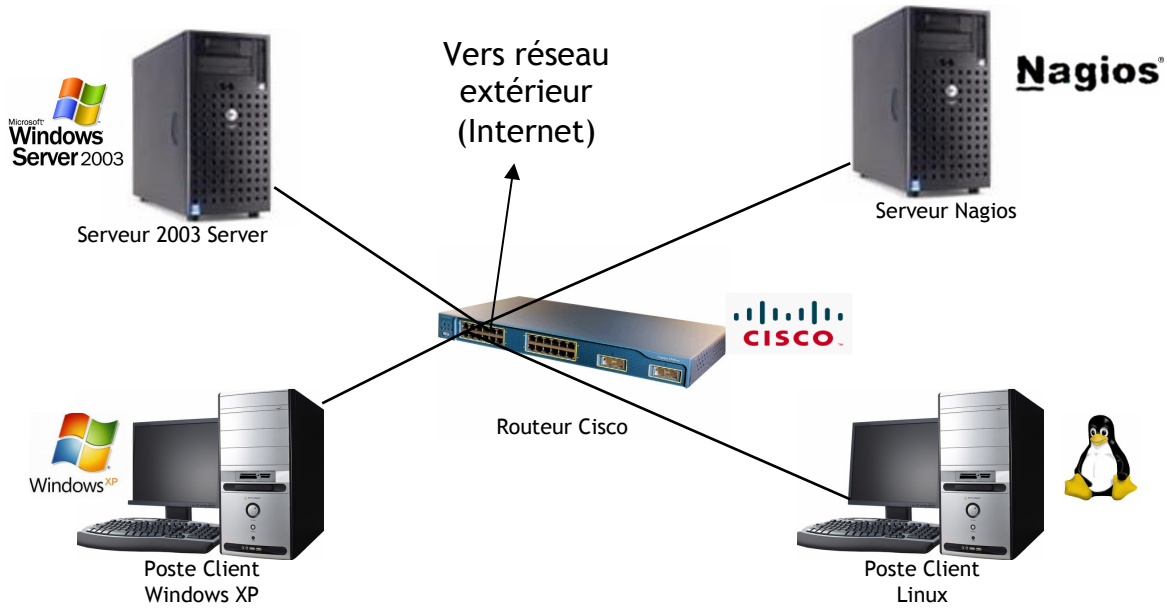
On peut distinguer deux types de fonctionnement de la récupération des informations: Actif et Passif

-Lors du fonctionnement actif, c'est Nagios qui envoie une demande d'information au client. Celui-ci renvoi alors les informations demandées.

-Alors que lors du fonctionnement passif, l'envoi d'information est planifié en local (soi planifié à l'avance selon une date donnée, soi selon un certain événement produit sur la machine cliente)

B – Structure du réseau

Le réseau sur lequel nous allons travailler sera celui-ci:



Il sera composé :
- D'un serveur "Windows Server 2003" qui permettra la gestion des utilisateurs du réseau : Stockage des données et identifications des utilisateurs
- D'un serveur "Nagios" qui s'occupera de la supervision du réseau, de la centralisation et de l'analyse des informations du réseau
- D'un poste client "Windows XP"
- D'un poste client "Linux"
- D'un routeur "Cisco" qui permettra de relier les différents équipements du réseau et d'être relié au réseau extérieur

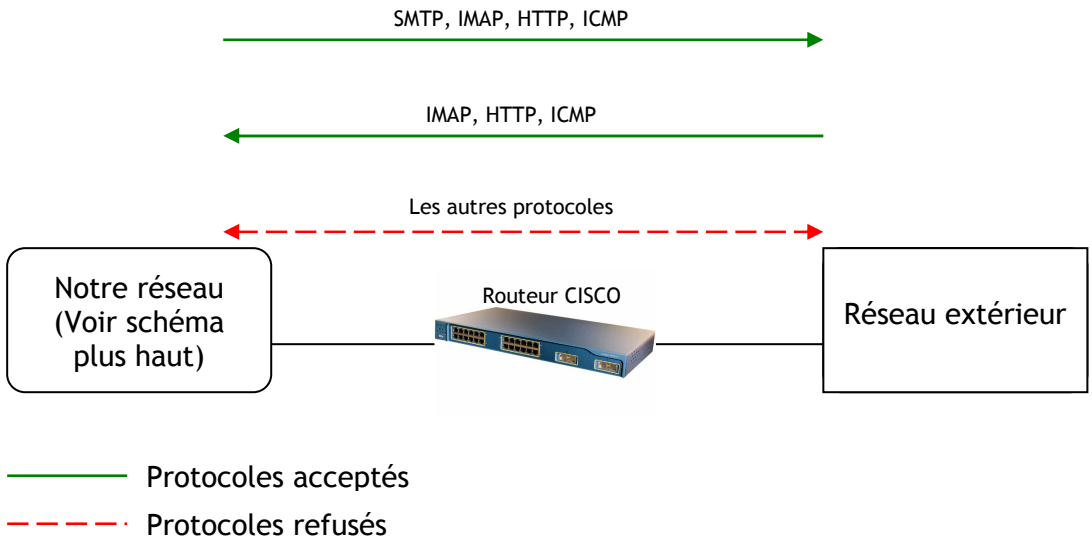
Le routeur CISCO devra faire une translation d'adresse (NAT) pour que notre réseau puisse dialoguer avec l'extérieur.

Afin d'éviter de configurer le routeur chaque semaine, nous utiliserons le protocole TFTP (Trivial File Transfert Protocol) pour sauvegarder sa configuration.

De plus, sur le routeur, un firewall sera configuré grâce à des ACL (Access Control List) permettant l'autorisation ou le refus de certaines connections.
Le firewall devra:

- Autoriser le protocole SMTP (pour l'envoi de mail) sortant mais pas entrant
- Autoriser le protocole IMAP (pour la réception de mail) entrant et sortant
- Autoriser le protocole HTTP entrant et sortant (pour le web)
- Autoriser le protocole ICMP entrant et sortant (pour l'envoi et la réception de PING)
- Refuser tous les autres protocoles dans les deux sens

Pour résumer :



C – Informations récupérées par Nagios et alertes

Avant tout, il faut définir les informations qui seront récupérées par Nagios, sur chaque équipement du réseau.

➔ Sur le serveur "2003 serveur", Nagios récupéra :
- La version du pluging qui envoie les informations au serveur Nagios : Si cette version n'est pas la dernière, un e-mail sera envoyé à l'administrateur réseau.
- La charge CPU du serveur. Dans notre cas, si la charge dépasse les 90 %, un e-mail sera envoyé à l'administrateur réseau.
- La durée depuis le dernier démarrage du serveur.
- La taille et l'occupation des disques durs. Dans notre cas, lorsque 90 % de l'un des disques durs est occupé, un e-mail sera envoyé à l'administrateur réseau.

➔ Sur les postes clients, Nagios récupéra :
- La version du pluging qui envoie les informations au serveur Nagios : Si cette version n'est pas la dernière, un e-mail sera envoyé à l'administrateur réseau.
- La taille et l'occupation des disques durs. Dans notre cas, lorsque 90 % de l'un des disques durs est occupé, un e-mail sera envoyé à l'administrateur réseau.
- Pour Windows XP: La taille du dossier "C:\Documents and Settings" qui stocke les données des utilisateurs en local. Si ce dossier a une taille supérieure à 2Go, un e-mail sera envoyé à l'administrateur réseau pour qu'il puisse vider ce répertoire.

➔ Si Internet n'est plus disponible, un SMS sera envoyé à l'administrateur réseau.

➔ Si le routeur ne répond plus (le réseau ne peut donc plus marcher), envoie d'un SMS à l'administrateur réseau.

➔ Nagios doit avoir un historique des paquets rejetés par le firewall (voir les règles d'autorisations / refus dans la paragraphe précédent)

II - Les tâches

Tache 1 : Documentation Nagios
- Lecture de la documentation Nagios - Compréhension des différents fonctionnements de Nagios
Personnes: LECORCHE Hubert & JEANDROZ Sylvain **Durée:** 2 semaines **Coût:** 0 €

Tache 2 : Choix et installation d'une interface Linux
- Choix d'une interface Linux à installer - Installation de l'interface Linux sur le serveur Nagios
Personnes: LECORCHE Hubert **Durée:** 1 semaine **Coût:** 0 €

Tache 3 : Installation des équipements actifs
- Installation du serveur Windows 2003 serveur - Configuration du serveur Windows 2003 serveur - Installation d'un routeur CISCO - Configuration du routeur CISCO
Personnes: JEANDROZ Sylvain **Durée:** 1 semaine **Coût:** 0 €

Tache 4 : Installation de Nagios
- Téléchargements des fichiers Nagios - Compilation des fichiers Nagios - Installation du logiciel Nagios
Personnes: LECORCHE Hubert **Durée:** 1 semaine **Coût:** 0 €

Tache 5 : Installation des équipements passifs
- Installation du client Windows XP Professionnel - Installation du client Linux
Personnes: JEANDROZ Sylvain

Durée: 1 semaine **Coût:** 0 €

Tache 6 : Premiers tests
Après avoir installé le logiciel NAGIOS ainsi que les différents équipements du réseau, nous testons les différentes fonctionnalités de Nagios
Personnes: LECORCHE Hubert & JEANDROZ Sylvain **Durée:** 2 semaines **Coût:** 0 €

Tache 7 : Choix et installation des plugins
- Choix des plugins à installer sur les postes client pour qu'ils puissent communiquer avec le serveur Nagios afin que celui-ci récupère toutes les informations du réseau - Installation des plugins - Configuration des plugins
Personnes: LECORCHE Hubert **Durée:** 2 semaines **Coût:** 0 €

Tache 8 : Configuration de l'interface Web Nagios
- Configuration des paramètres de l'interface Nagios pour qu'il soit le plus optimal possible - Configuration de l'accès à l'interface Nagios (donner des droits aux utilisateurs)
Personnes: JEANDROZ Sylvain **Durée:** 2 semaines **Coût:** 0 €

Tache 9 : Services réseaux
Mise en place des protocoles pour que Nagios puisse dialoguer avec le routeur Mise en place des Access Control List (ACL) sur le routeur
Personnes: LECORCHE Hubert **Durée:** 2 semaines **Coût:** 0 €

Tache 10 : Détection des pannes
Configuration de Nagios pour qu'il puisse repérer les éventuelles pannes sur le réseau (traitement des informations qu'il reçoit des

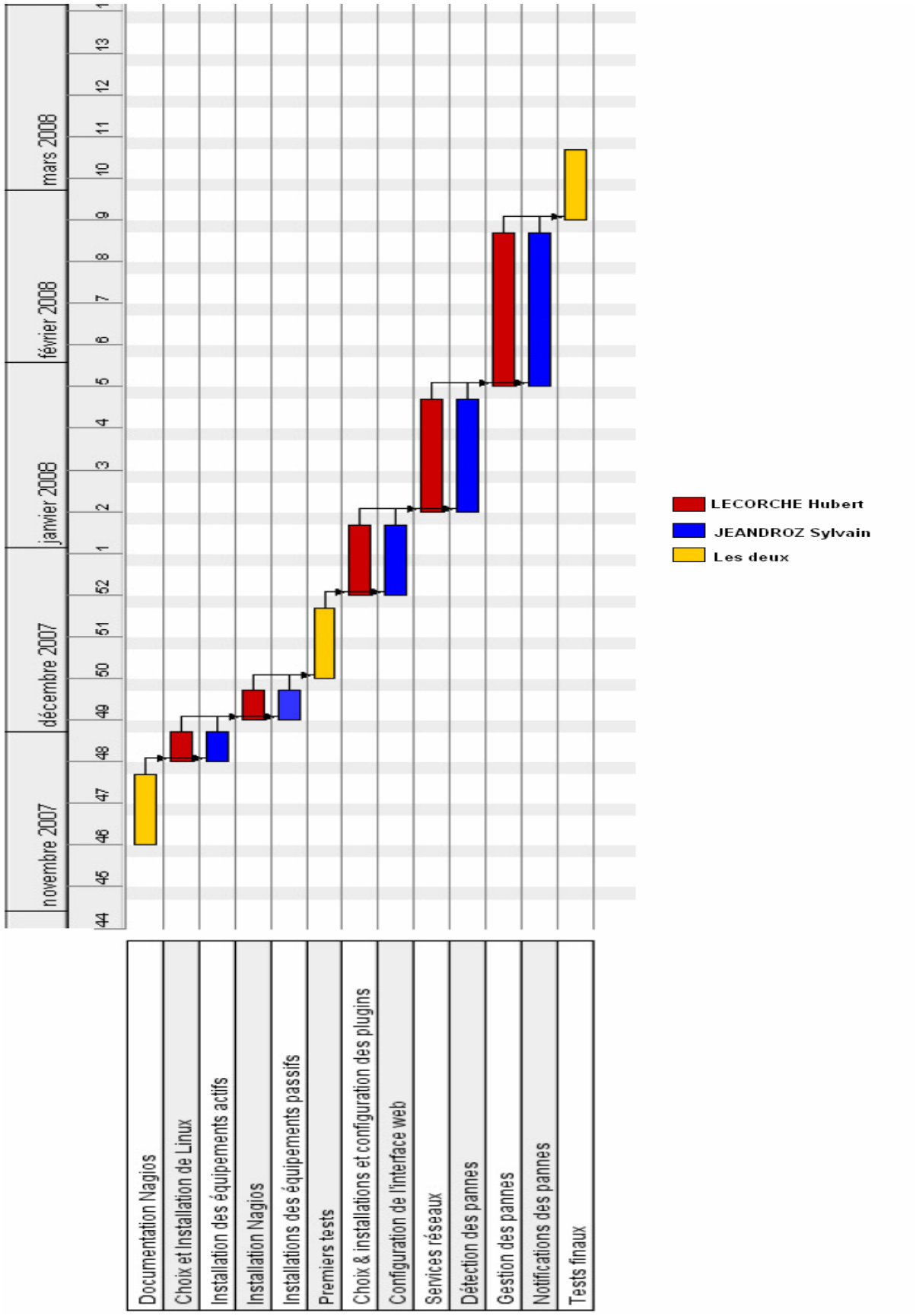
clients)
Personnes: JEANDROZ Sylvain **Durée:** 2 semaines **Coût:** 0 €

Tache 11 : Gestion des pannes
Configuration de Nagios pour qu'il puisse agir en cas de panne
Personnes: LECORCHE Hubert **Durée:** 4 semaines **Coût:** 0 €

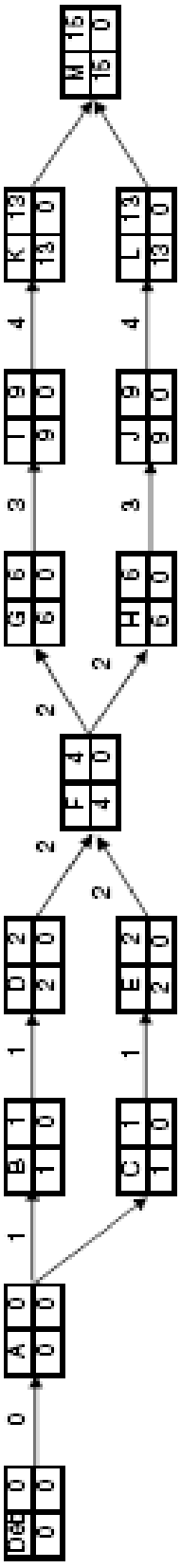
Tache 12 : Notification des pannes
Configuration de Nagios pour qu'il puisse avertir l'administrateur de différentes manières en cas de pannes sur le réseau afin qu'il puisse agir rapidement
Personnes: JEANDROZ Sylvain **Durée:** 4 semaines **Coût:** 0 €

Tache 13 : Tests finaux
On vérifie si toutes les configurations marchent correctement
Personnes: LECORCHE Hubert JEANDROZ Sylvain **Durée:** 2 semaines **Coût:** 0 €

III - Diagramme de Gantt



IV - MPM


M 15 15 0
K 13 13 0
L 13 13 0
I 9 9 0
J 9 9 0
G 6 6 0
H 6 6 0
F 4 4 0
D 2 2 0
E 2 2 0
B 1 1 0
C 1 1 0
A 0 0 0
Début 0 0
4
4
3
3
2
2
2
2
1
1
1
1
0

JEANDROZ Sylvain
LECORCHE Hubert

R. Protière

16/01/2008

PROJET n° 8b

Supervision de réseau avec Nagios

Compte rendu n° : 1

Travail réalisé depuis le dernier compte rendu (description succincte) :

Découverte de Nagios grâce à la documentation ainsi que des sites internet
Nous avons choisi un système d'exploitation Linux pour l'installation de Nagios : Debian. Nous l'avons installé sur une machine virtuelle (Virtual Box). Pour relier cette machine virtuelle à notre réseau, nous avons mis en place un pont réseau.
Nous avons installé les postes sur le réseau: Client XP + Client Linux (Mandriva) + Serveur 2003 Server. Nous avons configuré le routeur pour le fonctionnement de notre réseau local ainsi que l'accès au réseau extérieur (internet)
Nous avons installé / configuré les programmes sur les postes clients afin qu'ils puissent envoyer leurs performances au serveur Nagios.
Nous avons réalisé les tests pour savoir si Nagios recevait bien les messages envoyés par les PC. Tous nos tests ont été concluants.

Toutes les tâches sont conformes au planning
☑ oui
non. **Prenez contact avec votre tuteur** le plus rapidement possible.
Etablir un nouveau planning (GANT etc…) et le soumettre à votre tuteur.
Fournir les raisons:

-
-
-

Donner la liste des tâches concernées par la non-conformité
- tache " " : titre/contenu
- *etc…*

Indiquez les tâches terminées depuis le dernier rapport (le cas échéant) :

Tache 1 : Documentation de Nagios
Tache 2: Choix et installation d'une interface Linux
Tache 3: Installation des équipements passifs
Tache 4: Installation de Nagios
Tache 5: Installation des équipements passifs
Tache 6: Premiers tests

Tache 7: Choix et installation des pluging
Tache 8: Configuration de l'interface web

JEANDROZ Sylvain
LECORCHE Hubert

R. Protière

06/02/2008

PROJET n° 8b

Supervision de réseau avec Nagios

Compte rendu n° : 2

Travail réalisé depuis le dernier compte rendu (description succincte) :

- Installation des plugins sur les clients pour la récuperation des informations par Nagios
- Installation de l'interface ODEON qui permet de simplifier la configuration de Nagios
- Configuration du routeur pour n'autoriser que certains protocoles (ACL)
- Debut de création d'un script permettant l'envoie de sms à l'administrateur en cas de perte d'acces à internet

Toutes les tâches sont conformes au planning
☑ oui
non. **Prenez contact avec votre tuteur** le plus rapidement possible.
Etablir un nouveau planning (GANT etc…) et le soumettre à votre tuteur.
Fournir les raisons:

-
-
-

Donner la liste des tâches concernées par la non-conformité
- tache " " : titre/contenu
- *etc…*

Indiquez les tâches terminées depuis le dernier rapport (le cas échéant) :
Tache 6 : Premiers tests
Tache 7 : Choix et installation des plugins
Tache 8 : Configuration de l'interface Web Nagios
Tache 9 : Services réseaux

JEANDROZ Sylvain
LECORCHE Hubert

R. Protière

15/02/2008

PROJET n° 8b

Supervision de réseau avec Nagios

Compte rendu n° : 3

Travail réalisé depuis le dernier compte rendu (description succincte) :

--Configuration de l'interface OREON qui permet de simplifier la configuration de Nagios

-script permettant l'envoie de sms à l'administrateur en cas
de perte d'acces à internet

-Acces à l'interface Nagios depuis n'importe quel poste du réseau
Toutes les tâches sont conformes au planning
☑ oui
non. **Prenez contact avec votre tuteur** le plus rapidement possible.
Etablir un nouveau planning (GANT etc…) et le soumettre à votre tuteur.
Fournir les raisons:

-
-
-

Donner la liste des tâches concernées par la non-conformité
- tache " " : titre/contenu
- *etc…*

Indiquez les tâches terminées depuis le dernier rapport (le cas échéant) :
Tache 9 : Services réseaux
Tache 10 : Détection des pannes

JEANDROZ Sylvain
LECORCHE Hubert

R. Protière

29/02/2008

PROJET n° 8b

Supervision de réseau avec Nagios

Compte rendu n° : 4

Travail réalisé depuis le dernier compte rendu (description succincte) :

- Gestion des pannes par Nagios
- Tests finaux en vue de la démonstration du projet
- Ecriture du rapport de projet

Toutes les tâches sont conformes au planning

☑ oui

non. **Prenez contact avec votre tuteur** le plus rapidement possible.
Etablir un nouveau planning (GANT etc…) et le soumettre à votre tuteur.
Fournir les raisons:

-
-
-

Donner la liste des tâches concernées par la non-conformité
- tache " " : titre/contenu
- *etc…*

Indiquez les tâches terminées depuis le dernier rapport (le cas échéant) :
Tache 11 : Gestion des pannes
Tache 12 : Notification des pannes
Tache 13 : Tests finaux

Lorsqu'il y a un nombre important d'ordinateurs dans une entreprise, cela devient très difficile à gérer. C'est pourquoi il est utile d'utiliser un logiciel qui aide l'administrateur à superviser tout son parc informatique.
Nagios est un logiciel qui fonctionne sous Linux et qui permet d'effectuer cette supervision. Il utilise des plugins pour communiquer avec les machines hôtes et ainsi avoir une vue globale du réseau, avec les états des différentes machines.
Nous avons aussi utilisé Oreon, qui est un logiciel qui s'installe par dessus Nagios et qui permet de simplifier la configuration de celui-ci.
Pour notre projet, nous avons utilisé un réseau composé d'un routeur, d'un client XP, d'une machine Linux sur laquelle est installé Nagios, d'un client Linux et d'un serveur 2003 Serveur.
Une fois Nagios et Oreon configurés, nous pouvons surveiller les postes clients, ainsi que le routeur.
Des alertes, soit par mail, soit par SMS, sont envoyées lorsqu'il y a un problème sur le réseau.
Notre projet consiste donc à superviser un réseau grâce à l'outil Nagios. Notre projet comprend trois étapes: Compréhension, installation, et utilisation de Nagios.

When there is an important number of computers in a firm, it becomes very difficult to manage it. That's why it is useful to use a software which helps the Administrater to supervise all his computer park.
Nagios is a software which works under Linux and it allows this supervision. It uses plugins to communicate with hosts and so have a gobale view of network, with the state of the differents hosts.
We also used Oreon, which is a software which is installed on top of Nagios and simplifies it's configuration.
For our project, we used a network wich is composed by a router, an XP client, a computer on wich Nagios is installed, a Linux client, and a 2003 Server.
Once Nagios and Oreon configurated, we can surveil the hosts, as well as the router. Alerts, either by mail, or by sms, are sent when there is a serious probleme on network.
The goal of our project is to supervise a network with the tool Nagios. This project involves tree steps: undestanding, installation and use of Nagios.