République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire pour l'Obtention du Diplôme

D'Ingénieur d'Etat en Informatique

Option : Système d'Information

Thème :

Etude et Administration des Systèmes de Supervision dans un Réseau Local

Réalisé par :

- **Mme BELKHOUCHE Souheyla**

Présenté le 14/12/2011 devant le jury:

- *BENAMAR Abdelkarim* *(Président)*
- *BENAISSA Mohamed* *(Encadreur)*
- *BENZIAN Yaghmorasan Mohamed* *(Examinateur)*

Année universitaire : 2010-2011

Remerciements

En préambule à ce mémoire, je souhaitais adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Je tiens à remercier sincèrement Mr BENAISSA Mohamed, qui, en tant que encadreur de mémoire, s'est toujours montré à l'écoute et très disponible tout au long de la réalisation de ce mémoire, ainsi pour l'inspiration, l'aide et le temps qu'elle a bien voulu me consacrer et sans qui ce mémoire n'aurait jamais vu le jour.

J'exprime également ma gratitude aux membres du jury Mr BENAMAR AbdelKarim et Mr BENZIAN Yaghmorasan Mohamed, qui nous ont honorés en acceptant de juger ce modeste travail.

Je tiens à la fin de ce travail à remercier ALLAH le tout puissant de m'avoir donné la foi et de m'avoir permis d'en arriver là.

Dédicaces

A l'aide de DIEU tout puissant, qui trace le chemin de ma vie, j'ai pu arriver à réaliser ce modeste travail que je dédie:

A ma plus belle étoile qui puisse exister dans l'univers, ma très chère mère celle a qui je souhaite une longue vie et bonne santé ;

A mon père qui n'a pas cessé de m'encourager

A ma Seurre et sa fille, a mes frères

A ma petite famille mon époux et mon fils et a tout mes amis

Tables des Matières

Table des matières

Chapitre III: SUPERVISION RESEAU

Chapitre IV: OUTIL DE SUPERVISION NAGIOS

ChapitreV: ADMINISTRATION ET CONFIGURATION DE NAGIOS

Table des illustrations

Liste des figures

Liste des Acronymes

Liste des acronymes

A

API: application program interface

ACSE: Association Control Service Eléments

AE: Application Elément

ASE :Application Service Elément

D

DNS: Domain Name system

F

FTP: File Transfert Protocol

H

HTTP: Hyper Texte Transfert Protocol

I

IP: Internet Protocol

ISO: International Organization for Standardization

IANA: Internet Assigned Numbers Authority.

L

LSM: Local System Manager

M

MIB: : Management Information Base

N

NSCA: : Nagios Service Check Acceptor

NRPE: Nagios Remote Plugin Execut

O

OSI: Open Source Inder

OID: Object Identifier

R

RPC: Remote Procedure Call

ROSE: Remote Operation Service Elements

S

SNMP: Simple Network Management Protocol

T

TCP: Transmission Control Protocol

U

UDP: User Datagram Protocol

Introduction Générale

Introduction générale

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 en temps réel une vue de l'ensemble du parc informatique. Il peut être. informé (par email, par SMS) en cas de problème. Grâce à un tel système.

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.

Notre mémoire est composé de deux parties :

Une partie théorique contenant cinq chapitres ; le premier chapitre comporte une introduction à l'architecture client/serveur, le deuxième chapitre comporte les notions de base d'administration réseau, le troisième chapitre détaille les notions de la supervision réseaux, le quatrième chapitre comporte l'étude de l'outil de supervision Nagios et le dernier chapitre comporte la configuration et l'administration d'un serveur de supervision dans un réseau locale.

Une partie pratique qui fait l'objectif de notre projet, elle est réservée pour la configuration sous Linux l'outil de supervision Nagios dans un réseau local.

Chapitre I

LE MODELE CLIENT/SERVEUR

Chapitre I

LE MODELE CLIENT/SERVEUR

I.1 Introduction

Dans l'informatique moderne, de nombreuses applications fonctionnent selon un environnement client-serveur; cette dénomination signifie que des machines clientes (faisant partie du réseau) contactent un serveur - une machine généralement très puissante en termes de capacités d'entrées-sorties - qui leur fournit des services. Nous allons voir comment cette technologie permet d'exploiter au mieux les réseaux, et permet un haut niveau de coopération entre différentes machines sans que l'utilisateur se préoccupe des détails de compatibilité

I.2. Définition du modèle client/serveur

Le modèle client-serveur s'articule autour d'un réseau auquel sont connectés deux types d'ordinateurs le serveur et le client. Le client et le serveur communiquent via des protocoles. Les applications et les données sont réparties entre le client et le serveur de manière à réduire les coûts. Le client-serveur représente un dialogue entre deux processus informatiques par l'intermédiaire d'un échange de messages. Le processus client sous-traite au processus serveur des services à réaliser. Les processus sont généralement exécutés sur des machines, des OS et des réseaux hétérogènes.

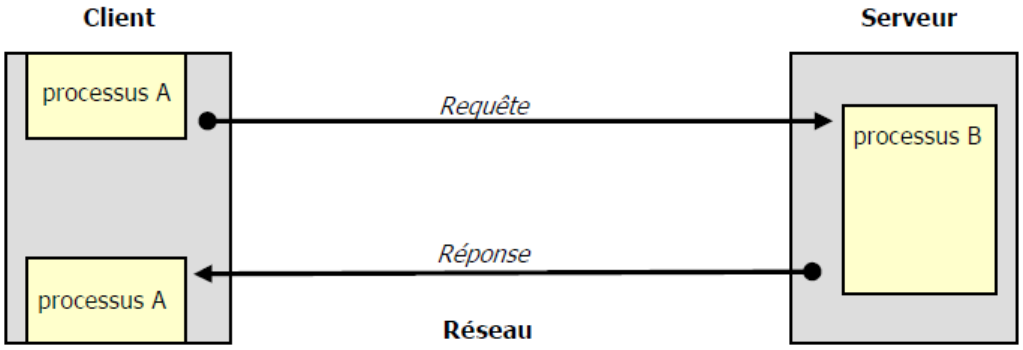


Figure I.1: Le modèle client/serveur

I.3. Caractéristiques des systèmes client serveur

Les éléments qui caractérisent une architecture client serveur sont :

- **Service**

Le modèle client serveur est une relation entre des processus qui tournent sur des machines séparées. Le serveur est un fournisseur de services. Le client est un consommateur de services.

- **Partage de ressources**

Un serveur traite plusieurs clients et contrôle leurs accès aux ressources

- **Protocole asymétrique**

Conséquence du partage de ressources, le protocole de communication est asymétrique le client déclenche le dialogue ; le serveur attend les requêtes des clients.

- **Transparence de la localisation**

L'architecture client serveur doit masquer au client la localisation du serveur (que le service soit sur la même machine ou accessible par le réseau). Transparence par rapport aux systèmes d'exploitation et aux plates-formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments

- **Message**

Les messages sont les moyens d'échanges entre client et serveur.

- **Encapsulation des services**

Un client demande un service. Le serveur décide de la façon de le rendre une mise à niveau du logiciel serveur doit être sans conséquence pour le client tant que l'interface message est identique.

- **Evolution**

Une architecture client serveur doit pouvoir évoluer horizontalement (évolution du nombre de clients) et verticalement (évolution du nombre et des caractéristiques des serveurs).

I.4. La répartition des tâches

Dans l'architecture client/serveur, une application est constituée de trois parties :

- L'interface utilisateur
- La logique des traitements
- La gestion des données

Le client n'exécute que l'interface utilisateur (souvent un interfaces graphique)

Ainsi que la logique des traitements (formuler la requête), laissant au serveur de bases de données la gestion complète des manipulations de données

La liaison entre le client et le serveur correspond a tout un ensemble complexe de logiciels appelé middleware qui se charge de toues les communication entre les processus.

I.5. Les différents modèles de client/serveur

En fait, les différences sont essentiellement liées aux services qui sont assurés par le serveur.

On distingue couramment:

I.5.1.Le client -serveur de donnée

Dans ce cas, le serveur assure des taches de gestion, stockage et de traitement de donnée .c'est le cas le plus connu de client- serveur est utilisé par tous les grands SGBD:

La base de données avec tous ses outils (maintenance, sauvegarde….) est installée sur un poste serveur.

Sur les clients, un logiciel d'accès est installé permettant d'accéder à la base de données du serveur

Tous les traitements sur les données sont effectués sur le serveur qui renvoie les informations demandées par le client.

I.5.2.Le client -serveur de présentation

Dans ce cas la présentation des pages affichées par le client est intégralement prise en charge par le serveur. Cette organisation présente l'inconvénient de générer un fort trafic réseaux.

I.5.3.Le client –serveur de traitement

Dans ce cas, le serveur effectue des traitements a la demande du client .Il peut S'agir de traitement particulier sur des données, de vérification de formulaire de saisie, de traitements d'alarmes

Ces traitements peuvent être réalisés par des programmes installés sur des serveurs mais également intégrés dans des bases de données, dans ce cas, la partie donnée et traitement sont intégrés.

I.6 La notion de protocole et port

I.6.1.Notion de port

Lors d'une communication en réseau, les différents ordinateurs s'échangent des informations qui sont généralement destinées à plusieurs applications (le client mail et le navigateur internet par exemple).

Seulement ces informations transitent par la même passerelle. Il faut donc savoir pour quelle application telle information est destinée. On attribue donc des ports pour chaque application. Un port est comme une porte en schématisant. Les informations sont multiplexées (comme dans les voitures récentes) et passent par la passerelle. A leur arrivée (vers le serveur) ou à leur réception (vers votre machine) elles sont dé multiplexées et chaque information distincte passe par le port qui lui est associé. Les informations sont ensuite traitées par l'application correspondante.

Un port est codé sur 16 bits, il y a donc 65536 ports.

L'adresse IP plus le port (exemple : **127.0.0.1:80**) est appelée socket.

Les ports ce sont vus attribuer une assignation par défaut pour aider à la configuration des réseaux.

Voici les principaux ports et le protocole les utilisant :

Port Service ou Application

21	FTP
23	Telnet
25	MTP
53	DNS
80	HTTP
110	POP3
119	NNTP

Les ports 0 à 1023 sont les ports reconnus ou réservés et sont assignés par l'IANA (Internet Assigned Numbers Authority).

Les ports 1024 à 49151 sont appelés ports enregistrés et les ports 49152 à 65535 sont les ports dynamiques (ou privés).

I.6.2Notion de protocoles

Un protocole est une série d'étapes à suivre pour permettre une communication harmonieuse Entre plusieurs ordinateurs.

Internet est un ensemble de protocoles regroupés sous le terme "TCP-IP" (**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol). Voici une liste non exhaustive des différents protocoles qui peuvent êtres utilisés :

HTTP : (Hyper Texte Transfert Protocol) : c'est celui que l'on utilise pour

Consulter les pages web.

- FTP : (File Transfert Protocol) : C'est un protocole utilisé pour transférer des fichiers.
- SMTP : (Simple Mail Transfert Protocol) : c'est le protocole utilisé pour envoyer des mails.
- POP : C'est le protocole utilisé pour recevoir des mails
- Telnet : utilisé surtout pour commander des applications côté serveur en lignes
- IP (internet Protocol) : L'adresse IP vous attribue une adresse lors de votre connexion à un serveur.

Les protocoles sont classés en deux catégories

-Les protocoles où les machines s'envoient des accusés de réception (pour permettre une gestion des erreurs). Ce sont les protocoles "orientés connexion"

-Les autres protocoles qui n'avertissent pas la machine qui va recevoir les données sont les protocoles "non orientés connexion"

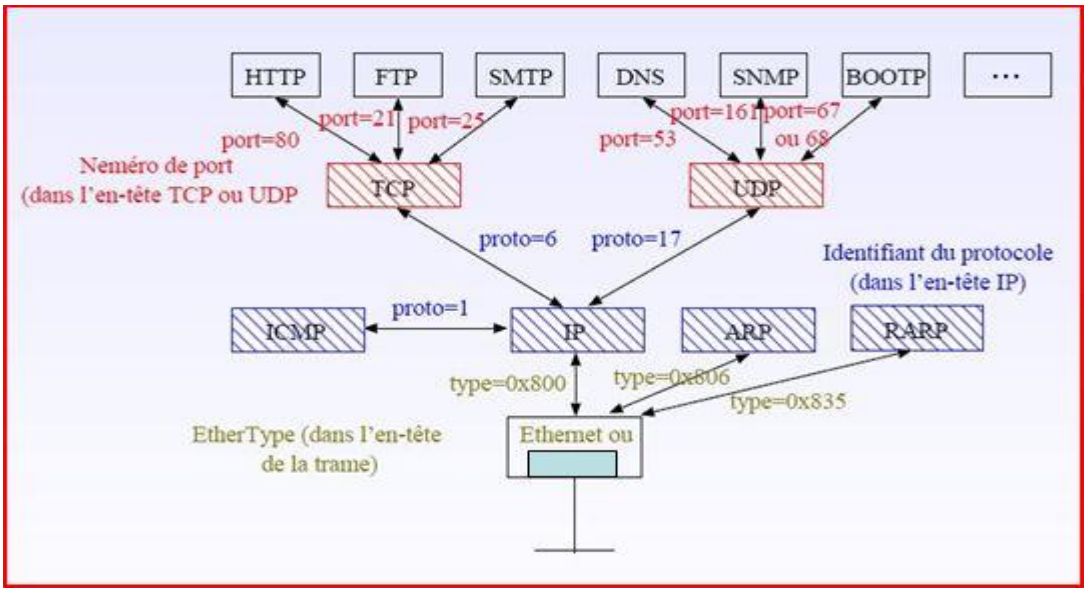


Figure I.2: protocole et port

I.7. Les Sockets

Port : Entrée réseau de la machine sur laquelle un serveur « écoute » en attendant des connexions / requêtes

Socket c'est un Tuyau entre deux programmes

Exemple

Client sur machine 1 appelle serveur sur machine 2 / port 53.

La connexion s'établit, le canal de communication est ouvert

Il devient possible de communiquer suivant un protocole application (par exemple DNS). [3]

I.7.1. API (application program interface) socket

Mécanisme d'interface de programmation Permet aux programmes d'échanger des données.

Les applications client/serveur ne voient les couches de communication qu'à travers API socket

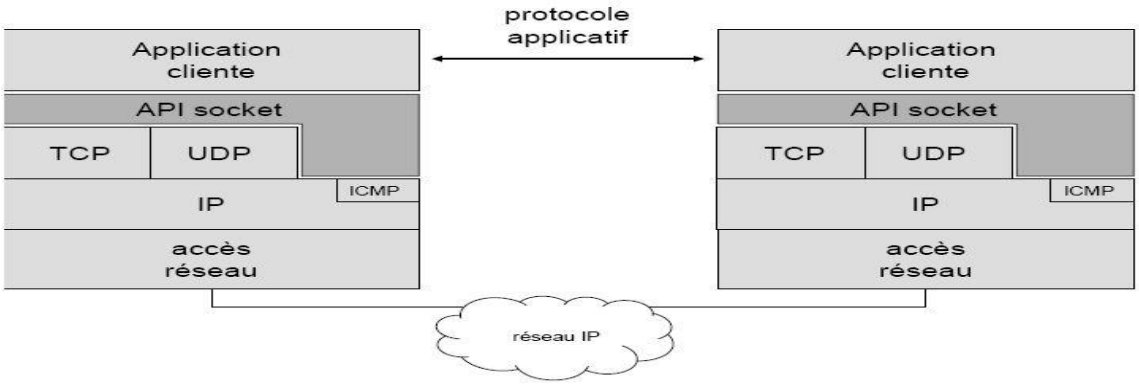


Figure I.3: Sockets

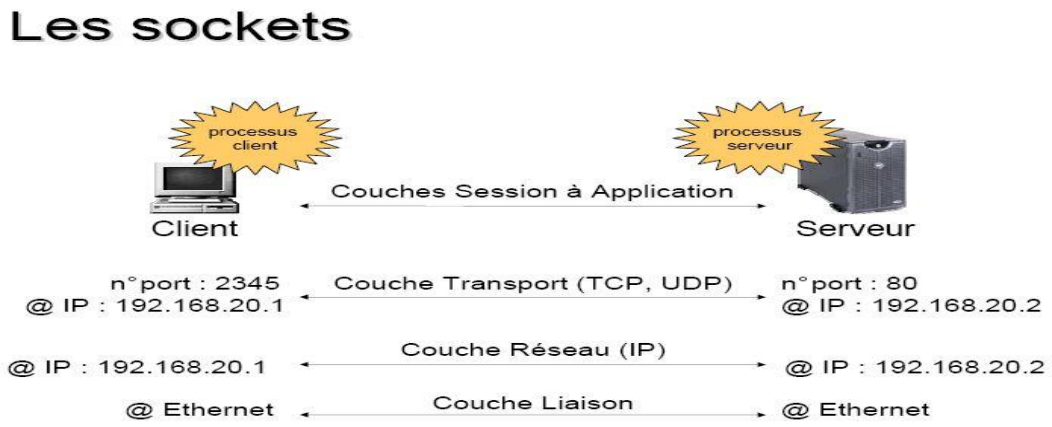


Figure I.4: Modèle OSI et sockets

I.8. Les middlewares

On appelle middleware (ou logiciel médiateur en français), littéralement " élément du milieu", l'ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différent composent d'une application répartie. Ce dialogue se base sur un protocole applicatif commun, défini par l'API du middleware.

Middleware c'est une interface de communication universelle entre processus. Il représente véritablement la clef de voûte de toute application client/serveur.

C'est logiciel qui assure les dialogues entre clients et serveurs hétérogènes, ou entre 2 applicatifs n'ayant pas les même API. Fait de l'« adaptation de protocole » des couches 5, 6, 7 du modèle OSI

L'objectif principal du middleware est d'unifier, pour les applications, l'accès et la manipulation de l'ensemble des services disponibles sur le réseau, afin de rendre l'utilisation de ces derniers presque transparente.

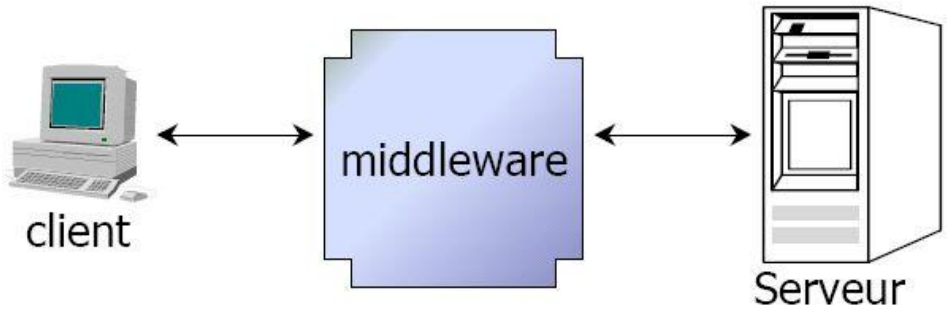


Figure I.5: Middlewares

I.8.1. Les services des middlewares

Un middleware susceptible de rendre les services suivants :

- **Conversion** :
  Services utilisé pour la communication entre machine mettant en œuvre des formats de données différentes

- **Adressage** :
  Permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès. Dans la mesure du possible.

- **Sécurité** :
  Permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations.

- **Communication** :
  Permet la transmission des messages entre les deux systèmes sans altération. Ce service doit gérer la connexion au serveur, la préparation de l'exécution des requêtes, la récupération des résultats et la déconnexion de l'utilisation.

Le middleware masque la complexité des échanges inter-applications et permet ainsi d'élever le niveau des API utilisées par les programmes. Sans ce mécanisme, la programmation d'une application client/serveur serait complexe et difficilement évolutive.

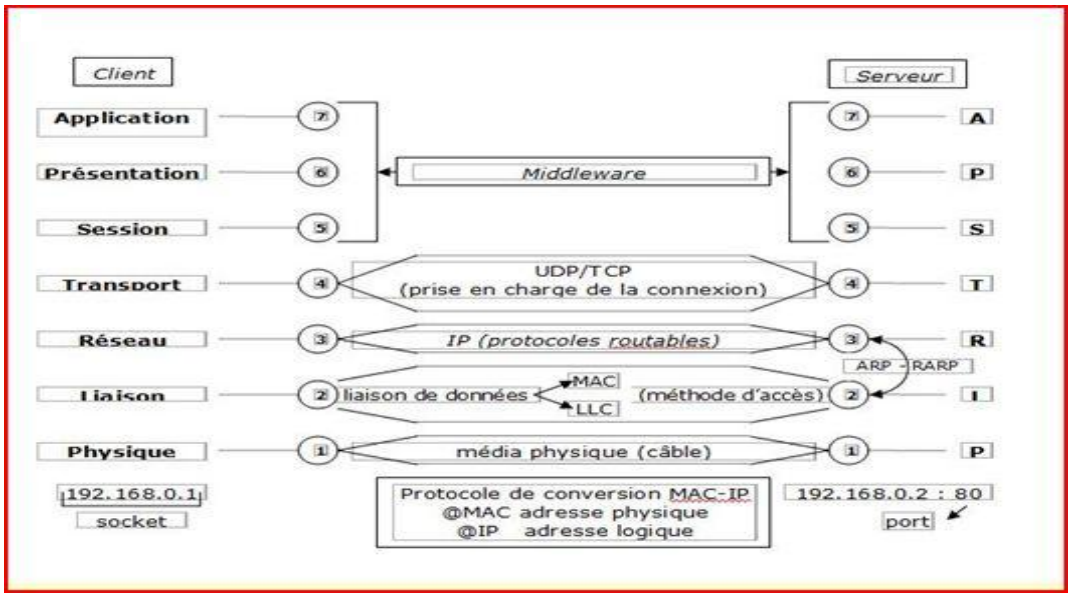


Figure I.6: Modèle OSI et middleware

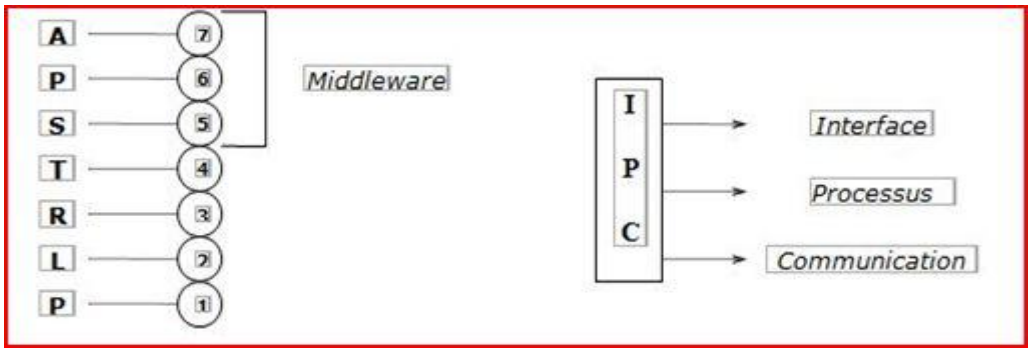


Figure I.7: IPC

I.9. RCP

Appel de procédure distance : RPC = Remote Procédure Call

Technique permettant d'appeler une procédure distante comme une procédure locale en rendant transparente les messages échangés

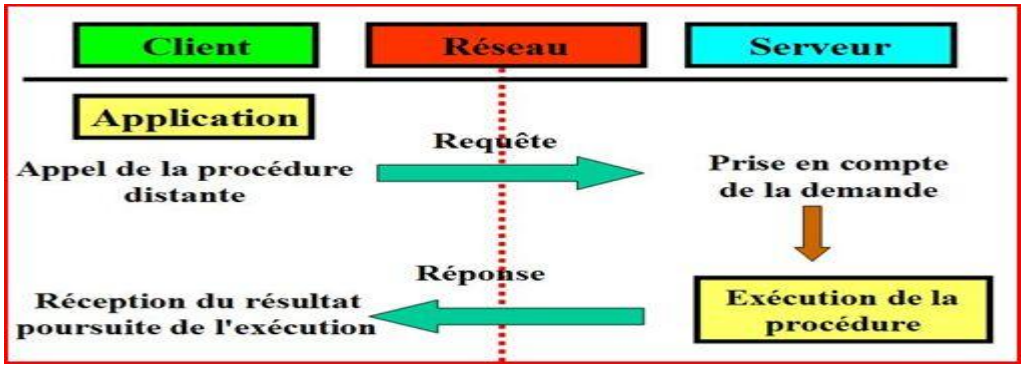


Figure I.8: Appel de procédure à distance

Dans les RPC, la communication client/serveur : peut se faire par datagramme (par paquets), ou par connexion (flux de données dans un canal).

Elle peut être :

Synchrone : le serveur attend la requête du client ; et pendant que le serveur fait le traitement, le client attend,

Asynchrone : pendant qu'un des acteurs traite les informations, l'autre acteur, au lieu d'attendre, continue de « vivre sa vie ». Il est interrompu (par une interruption système) quand l'autre acteur lui envoie de nouveau de l'information, afin qu'il aille traiter ce flot entrant. [3]

I.10.Conclusion

Le modèle client /serveur est la base de tous les services réseaux informatique, c'est pour cela nous sommes intéressé par l'étude de se modèle. Le but de ce chapitre c'est décrire les différentes notions de base de ce modèle comme le middleware, les protocoles, les sockets et l'appel de procédure à distance. Afin de mieux configurer et administrer les différents services qui font l'objectif de notre projet. La configuration de la carte réseaux est une étape fondamentale et importante pour faire cette configuration sur la distribution l'linux Ubuntu, qui fait l'objet du deuxième chapitre de notre mémoire.

Chapitre II
ADMINISTRATION
ET CONFIGURATION RESEAU

Chapitre II

ADMINISTRATION ET CONFIGURATION RESEAU

II.1 Introduction

Le terme administration de réseaux recouvre l'ensemble des fonctions qui sont nécessaires pour l'exploitation, la sécurité, le suivi et l'entretien du réseau. II est nécessaire de pouvoir initialiser de nouveaux services, installer de nouvelles stations raccordées au réseau, superviser l'état du réseau global et de chacun de ses sous ensembles, suivre de manière fine l'évolution des performances, évaluer et comparer diverses solutions, mettre fin à des situations anormales. L'administrateur a besoin de trois grands types d'actions pour agir et suivre son réseau :

Des actions en temps réel pour connaître l'état de fonctionnement de son réseau (surveillance et diagnostic des incidents, mesure de la charge réelle, maintenance, contrôle, information aux utilisateurs,...) et agir sur celui-ci (réparation, ajout de nouveaux utilisateurs, retraits,...), assurer la sécurité (contrôler les accès, créer/retirer des droits d'accès,...).

Des actions différées pour planifier, optimiser, quantifier et gérer les évolutions du réseau (statistiques, comptabilité, facturation, prévention, évaluation de charges,...).

Des actions prévisionnelles qui lui permettent d'avoir une vision à moyen et long terme, d'évaluer des solutions alternatives, de choisir les nouvelles générations de produits, d'envisager les configurations, de décider du plan d'extension, de vérifier la pertinence de la solution réseau pour un problème donné…

L'ensemble de ces objectifs ne peut être satisfait par un outil unique. I1 est nécessaire de faire appel à plusieurs techniques de l'informatique et des mathématiques pour répondre à ces divers besoins. Nous distinguerons les fonctions liées à la gestion au jour le jour du réseau, communément appelées outils d'administration les outils de configuration et les outils d'analyse et de mesure.

II.2 Objectifs de l'administration

Le rôle de l'administration du réseau est indissociable de la structure d'organisation de l'entreprise. Les fonctions assurées par un groupe d'utilisateurs (micro-ordinateurs, robots,...) sont de première importance dans la définition du service qui doit leur être fourni. L'administration du réseau doit posséder une bonne connaissance des entités réseau qu'il contrôle et une compréhension claire de la manière dont le réseau local est utilisé. Cette connaissance est nécessaire pour permettre des actions efficaces: réponses rapides aux questions posées par les utilisateurs, suivi précis de l'utilisation effective du réseau, évolution des logiciels, matériels, protocoles, applications.

La qualité de l'administration du réseau peut généralement être jugée en fonction de la disponibilité (i.e. durée de fonctionnement sans interruption) et du temps de réponse.

Pour effectuer une bonne administration, l'administrateur a besoin de procédures d'interventions et d'outils adaptés aux conditions d'exploitation du réseau.

Dans un environnement réseau les procédures les plus fréquemment citées sont :

- Sauvegardes
- Gestion de l'espace disque
- Implantation de logiciel
- Implantation de nouvelles versions
- Modification de configuration
- Rechargement de fichier
- Gestion des droits d'accès

II.3.La normalisation ISO

L'administration de réseaux, compte tenu de sa complexité et de la confusion qui régnait a été normalisée par l'ISO au niveau de la "couche application" du modèle OSI Cependant, le succès de TCP-IP développé pour INTERNET fait office de standard et a généré un autre protocole de gestion de réseaux relativement simple appelé SNMP (que nous verrons ensuite). CMISE: Common Management Information System Elément est le nom du système d'administration de réseau ISO. Au niveau application un processus appelé MAP (Management Application Process) réalise les fonctions d'administration. II est constitué au minimum:

D'un gestionnaire local, LSM (Local System Manager) qui permet d'accéder à la base de données MIB (Management Information base) locale. Celle-ci est constituée des mesures effectuées localement et d'informations d'état ou de routage.

D'un agent d'administration global appelé SMAE (System Management Application Entity) qui gère entre autre les interactions entre les SMAE, les interfaces humaines. SMAE utilise les services ISO, ROSE (Remote Operation Service Eléments) et ACSE (Association Control Service Eléments). SMAE est donc, au sens de la couche application un AE (Application Elément) et est constitué d'un ensemble d'ASE (Application Service Elément). Chaque ASE offrant lui même un ensemble de primitives pour l'administration du réseau.

L'exécution d'une primitive est faite par invocation d'appels de procédures (ASE) distants appelés RO (Remote Operation) par l'ISO. Pour ces appels l'ISO préconise l'utilisation d'associations d'application mises en œuvre à travers l'ACSE (Application Control Service Elément). La communication sur ces associations, donc entre deux processus d'administration, est régie par le protocole CMIP (Common Management Information Process), CMIP est largement inspiré des travaux et implémentations de DECnet.[11]

La normalisation ISO définit 4 niveaux dans une administration de réseaux :

- le niveau fonctionnel
- le niveau organisationnel,
- le niveau informationnel,
- le niveau communication

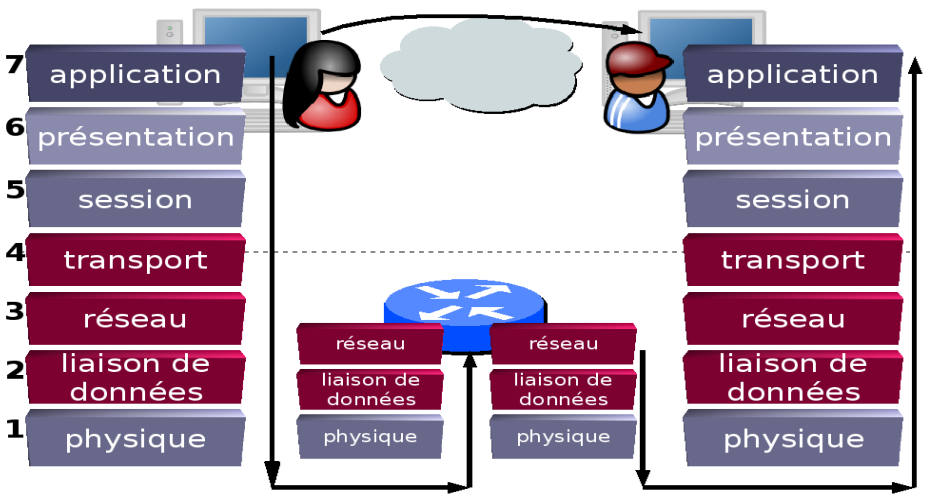


Figure II.1: Modèle ISO

II.4. Configuration réseau

II.4.1. Configurer les interfaces à la main avec ifconfig

- **Installation de la carte réseau :**

Les cartes réseaux sont souvent détectées au démarrage. Si ce n'est pas le cas il faudra charger les modules correspondants.

Pour obtenir la liste des interfaces réseaux qui ont été détectées, on peut utiliser la commande : **ifconfig -a**

Les sections qui commencent par ethX correspondent aux cartes Ethernet, ou X est le numéro de la carte.

Une fois votre carte reconnue par le noyau, vous devez au moins préciser l'adresse IP et masque de sous-réseau de la carte. Dans le cas d'un réseau local connecte à Internet, vous devez aussi ajouter l'adresse IP de la passerelle et l'adresse IP d'un ou plusieurs serveurs DNS.

- **Adresse IP :**

Pour attribuer une adresse IP a une interface réseau, on peut utiliser la commande :

ifconfig <interface> <adresse ip>

Par exemple : ifconfig eth0 192.168.0.4 up

Le masque de sous-réseau est déterminé automatiquement en fonction de la classe de l'adresse IP. S'il est différent on peut le spécifier avec l'option **netmask** :

ifconfig eth0 192.168.0.4 netmask 255.255.255.0

Pour voir si la carte réseau est bien configurée, on peut utiliser la commande : **ifconfig eth0**

- **Passerelle et routage :**

Pour ajouter une passerelle, on peut utiliser la commande **route** :

route add default gw <adresse ip >

Pour afficher les routes vers les différents réseaux : **route -n**

❖ **Tester le réseau**

Pour tester si la carte réseau fonctionne, on peut essayer de communiquer avec une autre machine avec la commande :

Ping <adresse ip>

La commande Ping envoi un paquet a l'adresse IP puis attend que la machine réponde. Elle affiche ensuite le temps qu'a pris toute l'opération, en millisecondes.

❖ **Informations sur les interfaces**

Pour vérifier les statuts de toutes les interfaces on peut utiliser la commande :

netstat –i

❖ **Nom d'hôte (hostname)**

Le fichier **/etc/hostname** contient le nom de la machine et du domaine. Il est lu au démarrage du système ou lorsqu'on lance :

/etc/init.d/hostname

✓ **Configuration automatique au démarrage**

Le fichier /etc/network/interfaces permet de configurer les cartes réseau. Ce fichier est lu au démarrage du système et lorsqu'on utilise les commandes **ifup** et **ifdown**.

➢ Si l'interface eth0 doit être configurée automatique grâce à un serveur DHCP en remplis le fichier interfaces par : **iface eth0 inet dhcp**

❖ **Résolution de noms**

Le fichier **/etc/host.conf** indique comment les noms doivent être résolus (c'est-à-dire comment passer d'une adresse IP a un nom, et inversement). Par exemple :

❖ **Serveurs DNS**

Le fichier **/etc/resolv.conf** contient les adresses IP des serveurs DNS (domain name system). Par exemple : **nameserver 192.168.0.4**, **nameserver 127.0.0.1**, **search ubuntu-fr.lan**

ubuntu-fr.lan c’est le nom de domaine

❖ **Fichier hosts**

Le fichier **/etc/hosts** contient une liste de résolutions de noms (adresses IP et noms de machine). Par exemple: **192.168.0.4 Nabila**

Ce fichier indique que Nabila correspond a l'adresse IP 192.168.0.4 qui sera accessible par cet alias.

- **Fichier networks**

Le fichier **/etc/networks** permet d'affecter un nom logique à un réseau

Localhost 127.0.0.1

Nabila.ubuntu-fr.lan 192.168.0.4

Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse.

II.5. Conclusion

Administrer un réseau, c'est tirer le meilleur parti de la structure que l'on utilise. C'est un système dual car la conception d'une administration dépend étroitement de la structure gérée mais le comportement futur de cette structure dépendra fortement de son administration.

L'administrateur réalise la configuration des systèmes d'exploitation et veille au bon fonctionnement du matériel. Il met à la disposition des utilisateurs un environnement fiable, convivial et homogène. Il permet aussi à l'entreprise d'économiser les investissements matériels inutiles. Il tient un rôle essentiel dans l'entreprise.

Dans le prochain chapitre, nous présentons les notions de base de la supervision des réseaux.

Chapitre III

SUPERVISION RESEAU

Chapitre III

SUPERVISION RESEAU

III.1.Introduction

Les réseaux sont de partout à l'heure actuelle. Ils sont devenus indispensables au bon fonctionnement général de nombreuses entreprises et administrations. Tout problème ou panne peut avoir de lourdes conséquences aussi bien financières qu'organisationnelles. La supervision des réseaux est alors nécessaire et indispensable. Elle permet entre autre d'avoir une vue globale du fonctionnement et problèmes pouvant survenir sur un réseau mais aussi d'avoir des indicateurs sur la performance de son architecture. De nombreux logiciels qu'ils soient libres ou propriétaires existent sur le marché. La plupart s'appuie sur le protocole SNMP.

Dans une première partie nous allons faire une présentation de la supervision et tout ce qui touche au monitoring de réseau. Dans une seconde partie, nous verrons le fonctionnement du protocole le plus utilisé actuellement : le protocole SNMP.

III.2.Présentation

III.2.1 Définition de la supervision

En informatique, la supervision est une technique de suivi, qui permet de surveiller, analyser, rapporter et d'alerter les fonctionnements normaux et anormaux des systèmes informatiques.

Entre outre, La supervision informatique consiste à indiquer et/ou commander l'état d'un serveur, d'un équipement réseau ou d'un service software pour anticiper les plantages ou diagnostiquer rapidement une panne

III.2.2.Objectifs

Il est aujourd'hui de plus en plus difficile d'administrer un réseau. En effet le nombre d'équipements à gérer est souvent de plus en plus important : stations, serveurs, imprimantes… Le plus grand souci d'un administrateur est la panne. En effet, il doit pouvoir réagir le plus rapidement possible pour effectuer les réparations nécessaires.

Il faut pouvoir surveiller de manière continu l'état des systèmes d'information afin d'éviter un arrêt de production de trop longue durée. C'est là ou la supervision intervient. Elle doit permettre d'anticiper les problèmes et de faire remonter des informations sur l'état des équipements.

Plus le système est important et complexe, plus la supervision devient compliquée sans les outils indispensables.

III.2.3.Principe

Une grande majorité des logiciels de supervision sont basés sur le protocole SNMP qui existe depuis de nombreuses années. Nous en faisons la description dans la deuxième partie.

La plupart de ces outils permettent de nombreuses fonctions dont voici les principales :

- Surveiller le système d'information
- Visualiser l'architecture du système
- Analyser les problèmes
- Déclencher des alertes en cas de problèmes
- Effectuer des actions en fonction des alertes

La tache de l'administrateur est alors simplifiée. Il n'a plus qu'à faire une vérification ou réaliser une action en fonction d'une alerte déclenchée. Chaque outil doit aussi lui donner une vision globale du système d'information pour localiser les problèmes le plus rapidement possible.

III.3. Le protocole SNMP

III.3.1.Présentation

SNMP signifie Simple Network Management Protocol (protocole simple de gestion de réseau en Français). C'est un protocole qui permet comme son nom l'indique, de gérer les équipements réseaux ainsi que les machines informatiques. Ce protocole est donc utilisé par les administrateurs réseaux pour détecter à distance les problèmes qui surviennent sur leur réseau.

Chaque machine, que ce soit sous Windows ou sous Linux possède de nombreuses informations capitales pour l'administrateur réseaux. On retrouve des informations comme la quantité de RAM utilisé, l'utilisation du CPU, l'espace disque et encore bien d'autre indicateurs.

SNMP va permettre de remonter ces informations à l'administrateur de façon centralisé pour pouvoir réagir au plus vite aux pannes éventuelles.

III.3.2 .Fonctionnement

III.3.2.1 Les agents

Sur une machine à superviser, pour que SNMP envoie les informations que l'on souhaite il faut qu'un agent soit installé sur celle-ci. Cet agent écoute sur le port 161 et attend que le serveur lui envoie des requêtes pour lui répondre.

L'agent pourra aussi envoyer des alertes lui même si l'administrateur l'a configuré. Par exemple pour surveiller l'occupation CPU l'administrateur définira une valeur critique pour laquelle une alerte doit lui être émise.

Pour finir l'agent pourra aussi agir sur l'environnement local. C'est pourquoi ce protocole est critique car il peut servir a d'autres personnes mal intentionnées pour prendre le contrôle a distance de certains équipements sur le réseau.

III.3.2.2 Les systèmes de management de réseaux

Généralement, l'administrateur possède un outil permettant de centraliser ce que lui retournent ses agents. Et c'est donc cet outil qui va interroger les équipements du réseau. Il va donc pouvoir gérer un réseau entier grâce à cela.

III.3.2.3 La MIB

- • Présentation

Pour que SNMP fonctionne, il est nécessaire qu'un protocole d'échange soit définit. Il y a aussi une standardisation des informations que ce protocole peut transporter. C'est un protocole Internet, il doit être utilisable sur des plates-formes hétérogènes (matériel comme système d'exploitation).

C'est pour cette raison que l'on parlera de MIB (Management Information Base).En effet, la MIB est une base de données des informations de gestion maintenue par l'agent. C'est cette base à laquelle on va demander les informations.

- • Structure de la MIB

La structure de la MIB est hiérarchique : les informations sont regroupées en arbre. Chaque information a un OID (Object identifier), une suite de chiffres séparés par des points, qui l'identifie de façon unique et un nom, indiqué dans le document qui décrit la MIB.

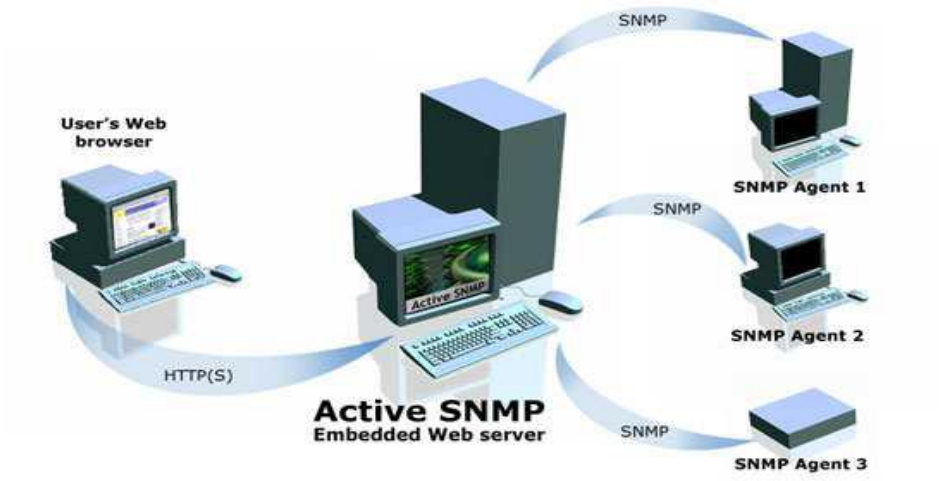


Figure III.1: Eléments de base du protocole SNMP

III.3.2.4.Les commandes SNMP

Il existe 4 types de requêtes SNMP :

- get-request : Le Manager SNMP demande une information à un agent SNMP
- get-next-request : Le Manager SNMP demande l'information suivante à l'agent SNMP

- set-request : Le Manager SNMP met à jour une information sur un agent SNMP
- trap : L'agent SNMP envoie une alerte au Manager

Les alertes sont transmises lorsqu'un événement non attendu se produit sur l'agent. Ce dernier informe le manager via une « trap ». Plusieurs types d'alertes sont alors possibles : ColdStart, WarmStart, LinkDown, LinkUp, AuthentificationFailure.

Pour chaque envoi de message, une réponse est retournée à l'exception de la commande « trap ». Les réponses sont du type suivant :

- get-response : L'information a bien été transmise.
- NoSuchObject : Aucune variable n'a été trouvée.
- NoAccess : Les droits d'accès ne sont pas bons.
- NoWritable : La variable ne peut être écrite.

III.3.2.5 Echange de message

Voici un schéma récapitulant les échanges pouvant être effectués entre un agent et le manager :

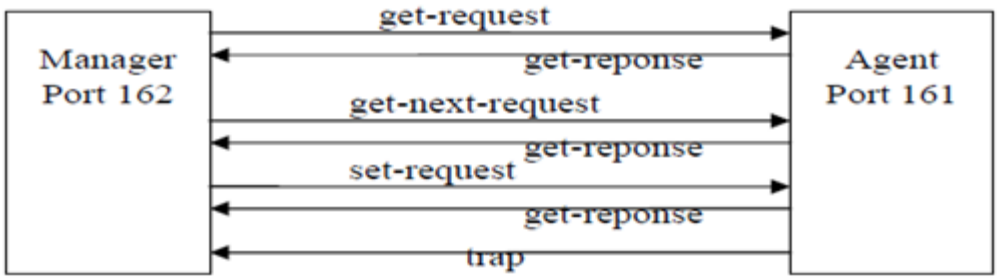


Figure III.2: Exemple d'échange SNMP

Le protocole SNMP est principalement utilisé avec UDP/IP. (Il peut aussi utiliser TCP). L'utilisation d'UDP permet un échange de message plus rapide que l'utilisation de TCP. L'inconvénient est qu'il est possible de perdre des trames lors de l'échange de messages (mode non connecté). Les ports UDP sont donc le 162 pour le manager et le 161 pour les agents.

III.3.3. SNMP en pratique

Concrètement, dans le cadre d'un réseau, SNMP est utilisé: pour administrer les équipements et pour surveiller le comportement des équipements Une requête SNMP est un datagramme UDP habituellement à destination du port 161. Les schémas de sécurité dépendent des versions de SNMP (v1, v2 ou v3). Dans les versions 1 et 2, une requête SNMP contient un nom appelé communauté, utilisé comme un mot de passe. Il y a un nom de communauté différent pour obtenir les droits en lecture et pour obtenir les droits en écriture.

Dans bien des cas, les colossales lacunes de sécurité que comportent les versions 1 et 2 de SNMP limitent l'utilisation de SNMP à la lecture des informations car la communauté circule sans chiffrement avec ces deux protocoles. Un grand nombre de logiciels libres et propriétaires utilisent SNMP pour interroger régulièrement les équipements et produire des graphes rendant compte de l'évolution des réseaux ou des systèmes informatiques (MRTG, Cacti, Nagios, Zabbix...)

III.4. Conclusion

La supervision est devenue indispensable dans tout système d'information. Elle est à la base du bon fonctionnement d'une architecture réseau et permet de réagir rapidement en cas de problèmes ou pannes. Elle se base à l'heure actuelle principalement sur le protocole SNMP qui depuis de nombreuses années a quand même du mal à évoluer. En effet, de nombreux logiciels sont encore basés sur la version 1 du protocole qui commence un peu à vieillir et qui n'est pas du tout sécurisé. En effet la version 2, apportant notamment la sécurité n'a été qu'une phase de transition vers la v3 qui est encore très peu utilisée.

Chapitre IV

OUTIL DE SUPERVISION NAGIOS

Chapitre IV

OUTIL DE SUPERVISION NAGIOS

IV.1. Introduction

La complexité et la grande quantité d'informations que Voyer sur des réseaux d'ordinateurs motiver la création d'équipements et de logiciels pour la gestion et le suivi de ces environnements informatiques. Une de ces ressources est le Nagios, outil qui vous permet de gérer plusieurs périphériques et services disponibles sur un réseau informatique. Le logiciel est conçu pour les entreprises cherchant des solutions pour gérer les réseaux locaux d'infrastructure ouverte et efficace. Il comprend des fonctions de surveillance, correction de gestion et de la faute. En outre, il a un grand nombre de plugins qui peuvent être regroupées, ce qui en fait un logiciel robuste et fiable.

IV.2. La supervision par nagios

Nagios est un logiciel qui fournit un ensemble de moyens et services pour assurer une supervision particulièrement simple, fiable, évolutive et non-propriétaire d'un parc informatique.

IV.2.1. 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) 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.

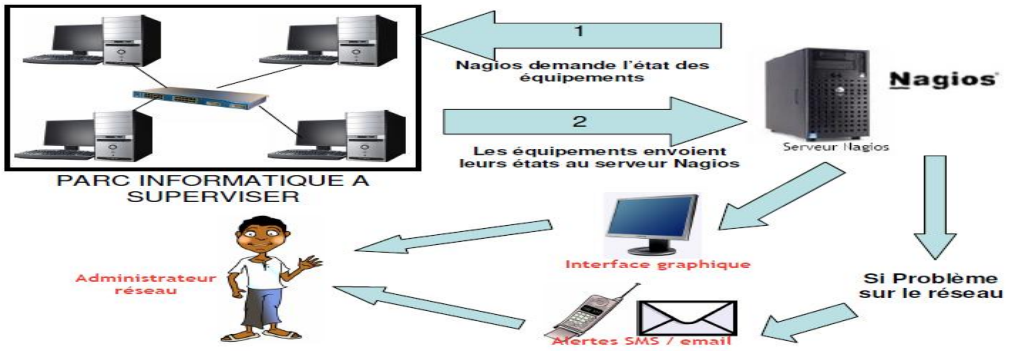


Figure IV.1 : l'interface graphique

IV.2.2. Architecture de Nagios

L'architecture de base de Nagios est simple :

- ***un ordonnanceur*** : Nagios est d'abord un moteur gérant l'ordonnancement des vérifications, ainsi que les actions à prendre sur incidents (alertes, escalades, prise d'action corrective) ;
- ***une IHM*** : la partie graphique visible à travers un simple serveur web, tel Apache est basée (pour les versions jusqu'à la 2.0) sur des CGI ;
- ***des sondes*** : les sondes de Nagios (les greffons ou *plugins*) sont de petits scripts ou programmes qui sont la base des vérifications.

Le projet Nagios fournit en standard bon nombre de greffons de base, mais la simplicité de leur mode de fonctionnement nous a permis d'en écrire un certain nombre pour nos besoins propres, que ce soit pour superviser dans notre environnement ou pour vérifier que nos clients peuvent bien se connecter chez nous. [7]

Figure IV.2 : Architecture de nagios

IV.2.3. Les plugins

Les plugins (greffons) sont des programmes exécutables ou des scripts (perl, Shell, etc..) qui peuvent être lancés depuis une ligne de commande pour tester un hôte ou un service. Le résultat de l'exécution d'un plugin est utilisé par Nagios pour déterminer le statut des hôtes ou des services sur le réseau.

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 diverses informations sur un équipement grâce au protocole SNMP.

VI.2.4 Fonctionnement de nagios

Nous pouvons distinguer deux modes de fonctionnement complémentaires de Nagios : le mode actif, ou de polling et le mode passif ou de traps.

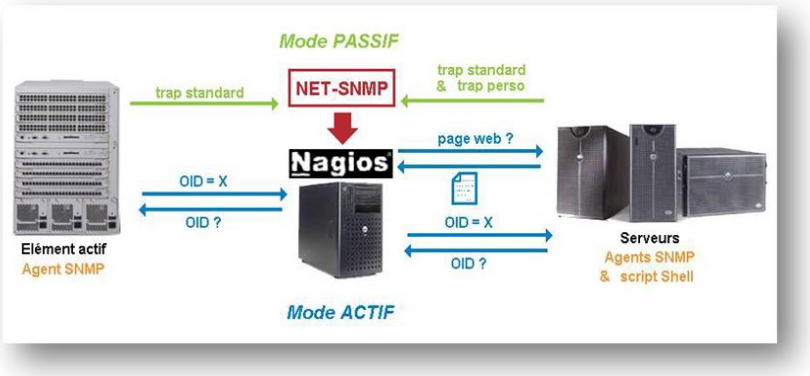


Figure IV.3 : Les deux modes de fonctionnement de Nagios

- En **mode polling**, Nagios exécute un plugin pour réaliser un test à des intervalles de temps réguliers. Il analyse ensuite la réponse et adopte un comportement en fonction de celle-ci. Ce mode de fonctionnement entraîne une génération du trafic sur le réseau.

- En **mode passif**, Nagios reste à l'écoute de tout ce qu'on peut lui dire. Pour communiquer avec lui, il suffit d'installer le programme client send_nsca sur les serveurs à superviser et de faire tourner le démon nsca sur le serveur Nagios. Dans notre configuration, c'est le démon snmptrapd de Net-SNMP qui utilise ce programme client via le script 'traitement-trap'.

Quelque soit le mode de fonctionnement, Nagios remonte des alertes aux administrateurs définis dans ses fichiers de configuration, que soit par mail, sms. Nagios met aussi en permanence à jour sont interface web qui reflète donc en temps réel l'état du réseau et des services.

Il est possible d'utiliser des agents de supervision permettant de récupérer des informations à distances. Ils offrent la possibilité de profiter de la puissance offerte par les plugins. Il existe 2 types d'agents :

- Les agents NRPE
- Les agents NCSA

Le principe de fonctionnement des *agents* **NRPE** *(pour Nagios Remote Plugin Executor)* est simple : les plugins sont installés sur l'équipement à superviser, compilés en fonction de son architecture car c'est elle qui va les exécuter, ainsi que le démon **NRPE** faisant office de serveur. Sur la plateforme de supervision Nagios, le plugin *check_nrpe* fera alors office de client nrpe, récupérant les informations en interrogeant le démon nrpe sur l'équipement concerné.

Le plugin *check_nrpe* sur le serveur Nagios initiera une connexion vers l'agent nrpe de la machine cible et lui demandera alors l'exécution d'une vérification. L'agent nrpe lancera alors le plugin configuré en local pour obtenir l'information et retournera le code retour de l'exécution ainsi que sa sortie standard.

Les *agents ncsa (pour Nagios Service Check Acceptor)* diffèrent des agents nrpe car la vérification est plannifiée en local sur l'équipement supervisé, exécutée, puis le résultat est

envoyé au serveur Nagios. De même que pour nrpe, l'architecture ncsa demande la présence du plugin *check_ncsa* sur la plateforme Nagios.

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. [5]

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.
Une 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***

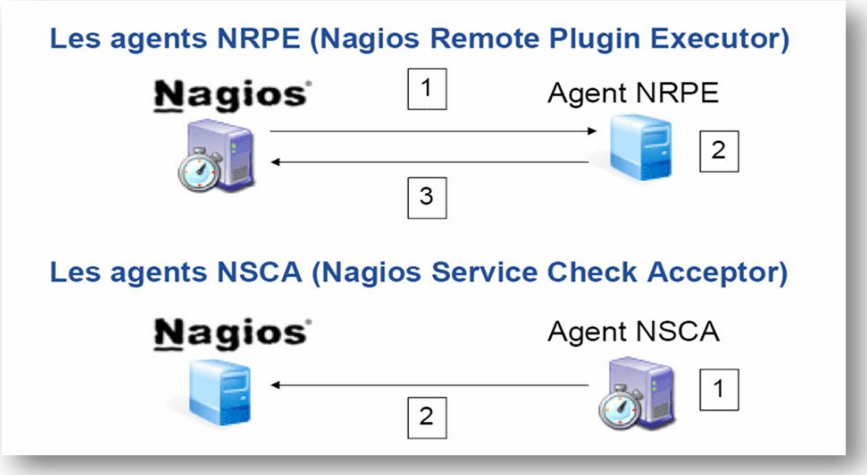


Figure IV.4 : Le fonctionnement de nagios

IV.2.4. Les fonctionnalités de nagios

Surveillance des services réseaux (SMTP, POP3, HTTP, NNTP, PING, etc.).

- Surveillance des ressources des hôtes (charge processeur, utilisation des disques, etc.).
- Système simple de plugins permettant aux utilisateurs de développer facilement leurs propres vérifications de services.
- Notifications des contacts quand un hôte ou un service a un problème et est résolu (via email, pager, ou par méthode définie par l'utilisateur).
- Possibilité de définir des gestionnaires d'évènements qui s'exécutent pour des évènements sur des hôtes ou des services, pour une résolution des problèmes
- Interface web, pour voir l'état actuel du réseau, notification et historique des Problèmes, fichiers etc.

Inconvénients

- Configuration compliquée qui oblige une très bonne connaissance de Nagios.
- Graphes pas assez clairs.
- Administration compliquée. [2]

IV.3.Supervision de serveurs Windows : NSCLient++

Nous allons décrire l'installation de NSClient, un plugin permettant de récupérer un nombre important de d'informations à surveiller sur une machine Windows.

Comme les plugins NRPE et NSCA (disponible seulement sous Linux et Mac OS X), NSClient se base sur une architecture client/serveur. La partie cliente (nommée check_nt), doit être disponible sur le serveur Nagios. La partie serveur (NSClient++) est à installer sur chacune des machines Windows à surveiller.

IV.3.1. Principe fonctionnement

- **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 démarrage de l'ordinateur, la version du plugin NsClient ++, occupation du processeur, occupation de la mémoire, état d'un service.

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 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.

- **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.

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'hôte à superviser> -c <nom de la commande à exécuter 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 sécurisés (grâce à SSL).

- **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.

Nous utiliserons check_snmp pour superviser le routeur.)

Fonctionnement de check_snmp

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>

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'hôte> -w <temps maxi de réponse>,<Pourcentage de réussite des pings> -c<temps maxi de réponse>,<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 [6]

IV.4 Conclusion

Nous avons présentés dans ce chapitre les notions de base de la supervision par Nagios qui est indispensable dans tout système d'information. Elle est à la base du bon fonctionnement d'une architecture réseau et permet de réagir rapidement en cas de problèmes ou pannes. Dans le prochain chapitre nous présentons les étapes de configuration et administration de notre projet« Nagios ».

Chapitre V

CONFIGURATION ET ADMINISTRATION DE NAGIOS

Chapitre V

CONFIGURATION ET ADMINISTRATION

V.1 Introduction

Nagios est un outil libre et open-source qui est utilisé pour contrôler et monitorer les éléments et les services sur un réseau. Lorsqu'il détecte un problème il envoie des messages d'alerte, soit par mail, soit par d'autres techniques. Il peut aussi être configuré afin qu'un personnel désigné peut accéder à des informations, des services ou des équipements particulièrs. Ce chapitre vous explique comment mettre en place Nagios sur un Ubuntu 9.10 server.

V.2 Installation de Nagios

Avant d'installer Nagios, il est préférable d'installer le serveur web Apache (c'est plus commode pour tester le bon fonctionnement de Nagios). Sans entrer dans les détails d'installation d'Apache, vous pouvez déjà avoir un serveur web fonctionnel en installant le paquet **apache2**.

Ensuite, il ne vous reste plus qu'à installer Nagios proprement dit, installer le paquet **nagios-text**.

Installer le paquet nagios3 (apache2 s'installera automatiquement car c'est une dépendance).

A la fin de l'installation, Nagios va vous demander d'introduire un mot de passe pour «nagiosadmin ».

V.3. Configuration

Pour configurer le serveur Apache de telle manière que Nagios soit accessible, le paquet Nagios fait un lien symbolique /etc/apache2/conf.d/nagios.conf vers ***etc/nagios3/apache.conf.***

Ensuite, vous devez recharger la configuration d'Apache à l'aide de la commande suivante

/etc/init.d/apache2 restart

V.3.1. Création des informations de compte utilisateur

Créez un nouveau compte utilisateur *nagios* et donnez-lui un mot de passe.

/usr/sbin/useradd nagios

passwd nagios

Sur les versions server d'Ubuntu, vous allez devoir créer manuellement un groupe d'utilisateur *nagios* (il n'est pas créé par défaut).

/usr/sbin/groupadd nagios

Il vous faut maintenant placer l'utilisateur *nagios* dans ce nouveau groupe.

/usr/sbin/usermod -G nagios nagios

Créez un nouveau groupe *nagcmd*qui permettra d'exécuter certaines commandes externes par l'intermédiaire de l'interface WEB. Placez ensuite dans ce groupe les utilisateurs nagios et apache.

/usr/sbin/groupadd nagcmd

/usr/sbin/usermod -G nagcmd nagios

/usr/sbin/usermod -G nagcmd www-data

Créez un compte nagiosadmin pour se connecter à l'interface Web de Nagios. N'oubliez pas le motde passe, vous en aurez besoin plus tard.

htpasswd -c /etc/nagios3/htpasswd.users nagiosadmin

V.3.2. Personnalisation de la configuration

Des exemples de fichiers de configuration sont maintenant installés dans le répertoire /

/etc/nagios3/

Ces fichiers d'exemple peuvent fonctionner correctement pour démarrer avec Nagios. Vous allez avoir besoin d'effectuer une petite modification avant de continuer...

Editez le fichier de configuration */etc/nagios3/contacts.cfg* avec votre éditeur favori et remplacez l'adresse mail associée au contact *nagiosadmin* par votre adresse si vous désirez recevoir les alertes.

V.3.3. Configurez l'interface Web

Installez le fichier de configuration web de Nagios dans le répertoire conf.d d'Apache.

make install-webconf

Redémarrez Apache pour prendre en compte ces modifications.

/etc/init.d/apache2 restart

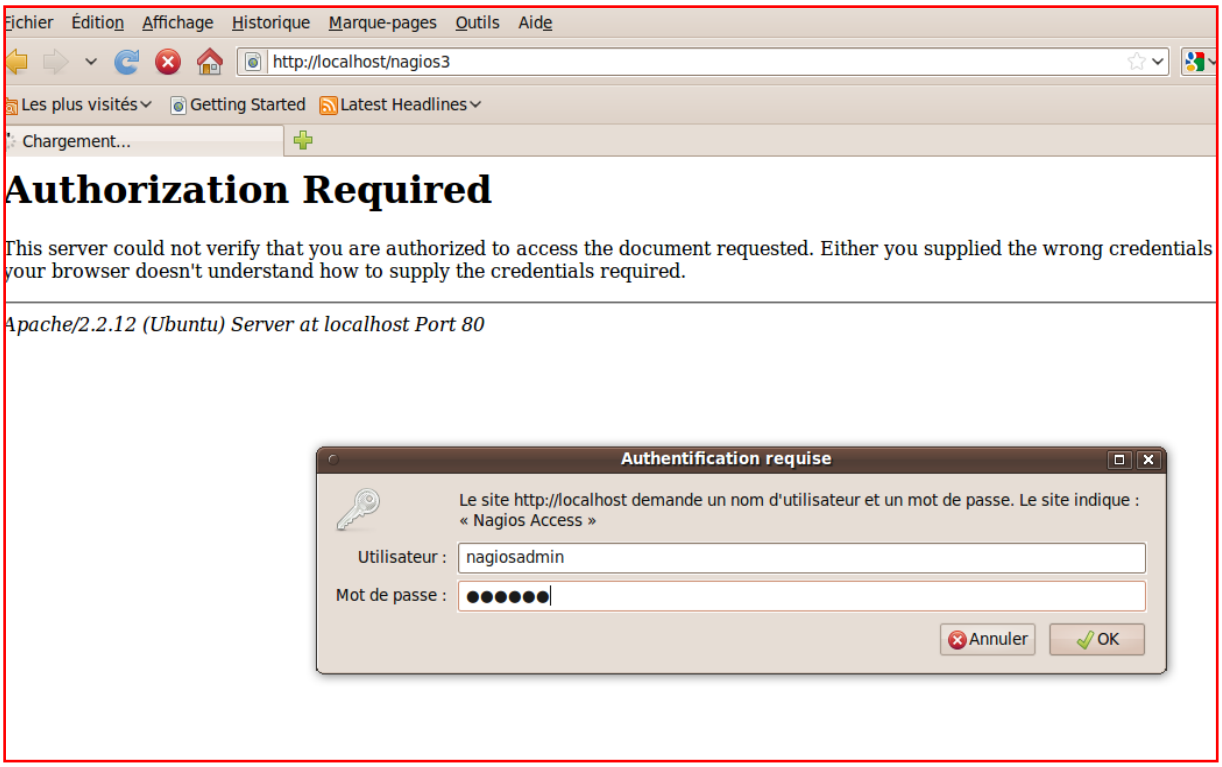
Démarrage de Nagios

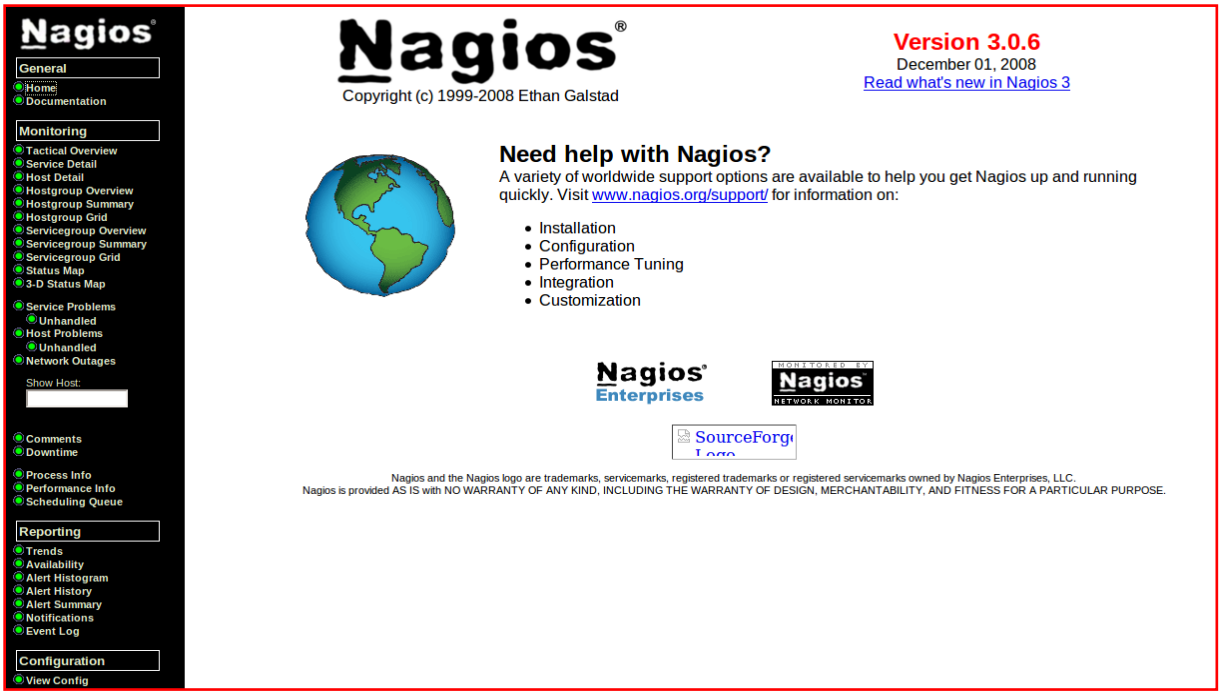
Configurez Nagios pour démarrer automatiquement au démarrage du système.

/etc/init.d/nagios3 restart

Connexion à l'interface Web

Vous devriez pouvoir maintenant accéder à l'interface Web de Nagios avec l'adresse ci-dessous. Le nom d'utilisateur (nagiosadmin) et le mot de passe définis précédemment vous sont demandés. http://localhost/nagios/





Cliquez sur le lien "Service Detail" de la barre de navigation pour voir ce qui est surveillé sur votre machine locale. Quelques minutes seront nécessaires à Nagios pour vérifier tous les services associés à votre machine.

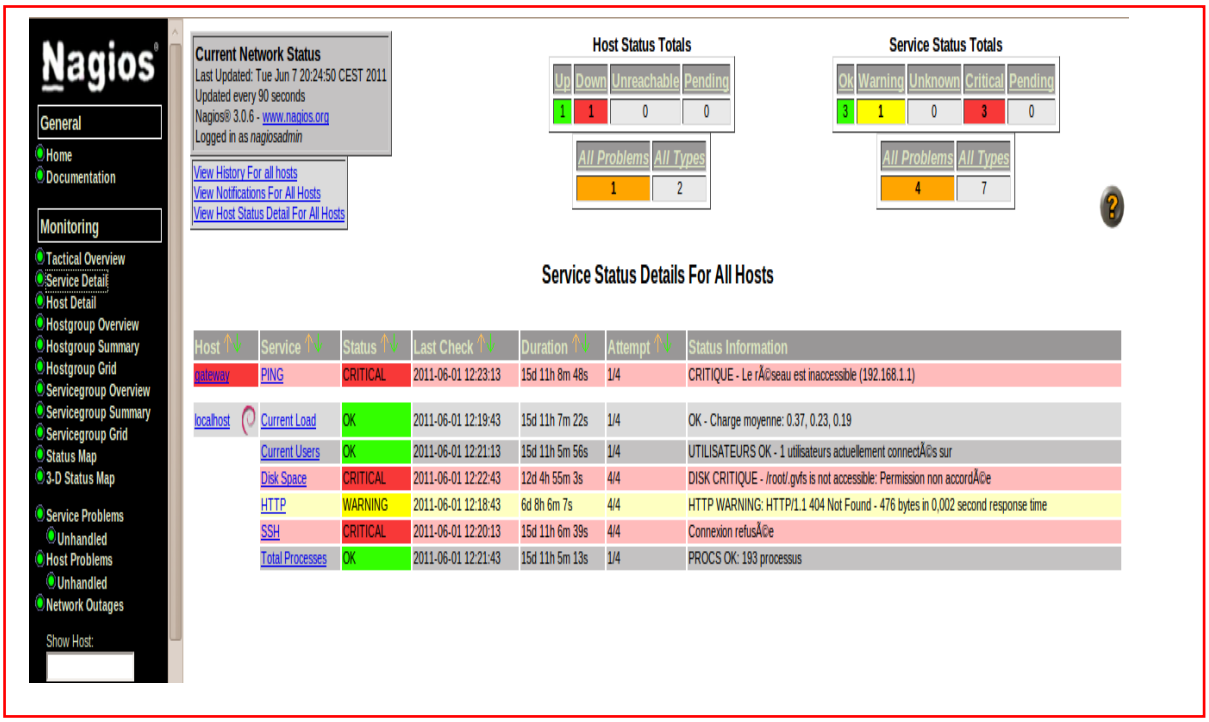


Figure V.1: service détail pour une machine localhost

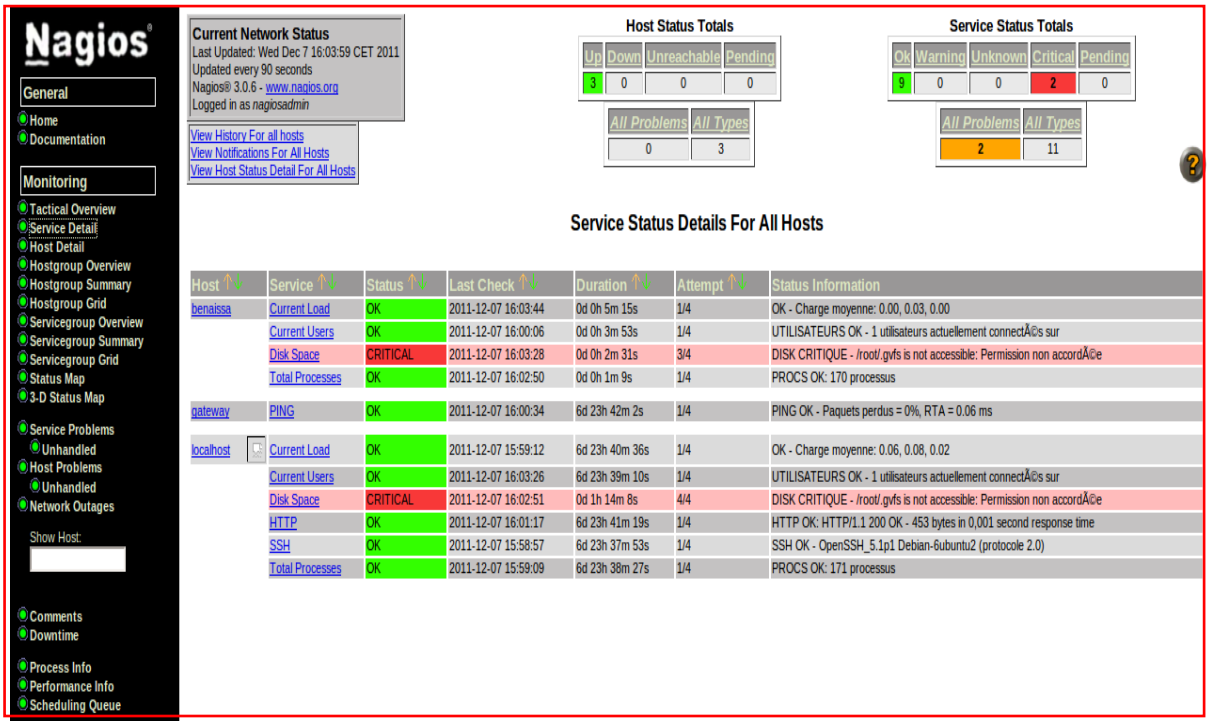


Figure V.2:service détail pour une machine l'inux

Maintenant, nous allons lister les principaux fichiers de configuration de Nagios. Ils ne sont pas tous mentionnés, seulement les plus importants. Ces fichiers se trouvent dans le répertoire /etc/nagios du répertoire d'installation de Nagios.

Fichiers	Description
cgi.cfg	Configuration du site web et des cgi (authorization).
checkcommands.cfg	Définition des tests.
contactgroups.cfg	Définition des groupes d'administrateurs.
contatcs.cfg	Définition des administrateurs (droits, adresse mail, nature des alertes…)
hostextinfo.cfg	Définissions complémentaires des machines pour la cartographie du réseau par les cgi de l'interface web (icône, emplacement…)
hostgroups.cfg	Définition des groupes de machines.
hosts.cfg	Définition des machines
miscommands.cfg	Définition des commandes. Notamment celle d'envoie par mail (host-notify-by-email)
nagios.cfg	Fichier de configuration principal (emplacement des fichiers, gestion des logs, user et group, comportement général…).
resource.cfg	Définition des variables. Notamment $USER1 = chemin d'accès aux plugins)
services.cfg	Définition des services à superviser. C'est le plus gros fichier à écrire. On y renseigne tous les services de toutes les machines que Nagios devra gérer.

Table V.1 : Les fichiers de configuration

Nous allons à présent voir à titre d'exemple quelques extraits choisis de ces fichiers de configuration. Le but est aussi pédagogique puisqu'il va nous permettre de concrétiser ce que nous avons vu depuis le début.

Exemple de fichier contacts.cfg : /etc/nagios3/conf.d/contacts.cfg

```
define contact{
        contact_name                    ostaquet
        alias                           Oscar Staquetowski
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,u,r
        service_notification_commands   notify-service-by-email
        host_notification_commands      notify-host-by-email
        email                           username@domaine.net
        pager                           +329999999999
        }
```

Exemple de fichier hostgroups : /etc/nagios3/conf.d/hostgroups.cfg

```
define hostgroup{
        hostgroup_name  connectique
        alias           Routeurs, firewalls et gateway
        contact_groups  admins-router
        members         router
        }

define hostgroup{
        hostgroup_name  mail-server
        alias           Serveurs de mails Ubuntu
        contact_groups  admins-ubuntu
        members         mail1, mail2
        }
```

Exemple dfichier services.cfg : /etc/nagios3/conf.d/services.cfg

```
define service{
        use                             generic-service
        host_name                       router
        service_description             PING
        contact_groups                  admins-routers,admins-ubuntu
        check_command                   check_ping!100.0,20%!500.0,60%
        }

define service{
        use                             generic-service
        hostgroup_name                  mail-server
        service_description             SMTP
        contact_groups                  admins-ubuntu
        check_command                   check_smtp
        flap_detection_enabled          0       ; Flap detection is disabled for this service
        }

define service{
        use                             generic-service
        host_name                       mail
        service_description             IMAP
        contact_groups                  admins-ubuntu
        check_command                   check_imap
```

V.4. Nsclient++

NSClient se base sur une architecture client/serveur. La partie cliente (nommée **check_nt**), doit être disponible sur le serveur Nagios. La partie serveur (**NSClient++**) est à installer sur chacune des machines Windows à surveiller.

V.4.1. Configuration de Nagios pour surveiller vos machines Windows

Une fois le client et le serveur installé, il faut configurer Nagios de la manière suivantes. Il faut dans un premier temps éditer votre fichier de configuration des hosts (hosts.cfg par défaut) et y ajouter votre machine Windows:

```
Define host {
use generic-hosthost_name nabila

alias Ma machine Win
address 192.168.0.4}
```

Puis ajouter les services offerts par NSClient (dans le fichier services.cfg):

```
# Affiche la version du NSClient
define service {
use generic-service
host_name benaissa
service_description VERSION
check_command check_nt!CLIENTVERSION
}

# Temps écoulé depuis le dernier reboot (uptime)
define service {
use generic-service
host_name  benaissa
service_description UPTIME
check_command check_nt!UPTIME
}

# Charge CPU
# WARNING si charge > 80% pendant plus de 5 minutes
# CRITICAL si charge > 90% pendant plus de 5 minutes
define service {
use generic-service
host_name  benaissa
```

```
   service_description CPU
   check_command check_nt!CPULOAD!-l 5,80,90}
   # Etat de la mémoire vive libre
   # WARNING si mémoire > 80%
   # CRITICAL si mémoire > 90%
   define service {
use generic-service
host_name  benaissa
   service_description MEM
   check_command check_nt!MEMUSE!-w 80 -c 90}
   # Etat de la mémoire disque libre (sur disque c:)
   # WARNING si mémoire > 80%
   # CRITICAL si mémoire > 90%
   define service {
   use generic-service
   host_name benaissa
   service_description DISK
   check_command check_nt!USEDDISKSPACE!-l c -w 80 -c 90}
```

Pour monitorer des clients Windows avec Nagios il faut passer par l'installation d'un agent nagios, ici le choix se portera sur **NSClient**

mais il en existe d'autres comme NCNET. NSClient communiquera directement avec Check NT (voir schéma fonctionnel).

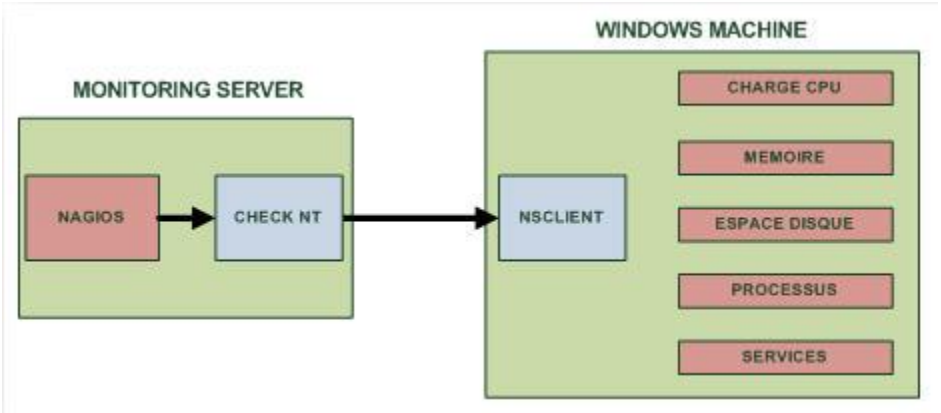


Figure V.3 Schéma fonctionnel de Nagios couplé à NSClient :

Configuration de NAGIOS pour accueillir des hôtes Windows

On va modifier la configuration de Nagios pour qu'ils connaissent l'hôte que l'on va superviser, pour cela on va modifier le fichier de config principal de Nagios pour accepter les clients Windows:

```
vim /usr/nagios/etc/nagios.cfg
```

Dans ce fichier on va dé commenter cette ligne :

```
#cfg_file=/usr/nagios/etc/objects/windows.cfg
```

Une fois décommenté on l'enregistre et on ferme. Maintenant on va ouvrir le fichier **windows.cfg** pour y rajouter le nom d'hôte à monitorer et les services à surveiller

```
vim /usr/nagios/etc/objects/windows.cfg
```

Une fois ce fichier ouvert il faut rajouter le nom du serveur :

```
define host{
        use             windows-server
        host_name       servfichier
        alias           servfichier
        address         192.168.0.225
        }
```

Ensuite suivant les services que vous voulez surveiller il faut rajouter le nom d'hôte toujours dans le même fichier :

```
define service{
        use                     generic-service
        host_name               servfichier
        service_description     NSClient++ Version
        check_command           check_nt!CLIENTVERSION
        }
```

Maintenant il faut ouvrir le fichier de configuration commands.cfg pour mettre un mot de passe pour la communication entre NSClient et le CHECK NT de Nagios

```
vim /usr/nagios/etc/objects/commands.cf
```

```
# 'check_nt' command definition
define command{
        command_name    check_nt
        command_line    $USER1$/check_nt -H $HOSTADDRESS$ -p 12489 -v $ARG1$
$ARG2$ -s Ton_password
        }
```

Il faudra se rappeler de ce mot de passe car on l'utilisera plus tard pour la config client.

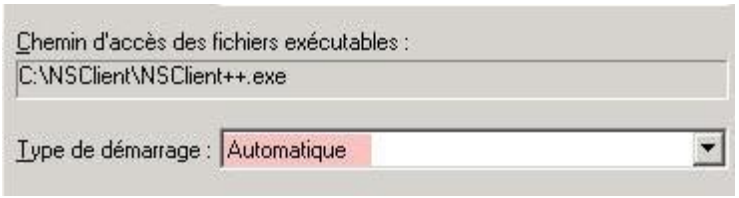
Installation de NSClient sur le serveur Windows:

Le logiciel NSClient est disponible à cette adresse : http://sourceforge.net/projects/nscplus

Une fois télécharger il faut dézipper l'archive par exemple dans C : maintenant il faut ouvrir une invite de commande dans C:\NSClient Et tapez ce qui suit :

```
nsclient++.exe /install
nstray.exe
```

Ensuite il faut ouvrir la mmc **services.msc** et configurer le démarrage automatique du service et l'autoriser à interagir avec le bureau



Ensuite on va éditer le fichier **NSC.ini** pour configurer la connexion entre le serveur à monitorer et nagios. Dans ce fichier il faut décommenter tous les modules de la section **[MODULES]** à l'exception de **checkWMI.dll** et **RemoteConfiguration.dll**

```
[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.dll
;
; RemoteConfiguration IS AN EXTREM EARLY IDEA SO DONT USE FOR PRODUCTION ENVIROMNEMTS!
;RemoteConfiguration.dll
; NSCA Agent is a new beta module use with care!
NSCAAgent.dll
; LUA script module used to write your own "check deamon" (sort of) early beta.
LUAScript.dll
; Script to check external scripts and/or internal aliases, early beta.
CheckExternalScripts.dll
; Check other hosts through NRPE extreme beta and probably a bit dangerous! :)
NRPEClient.dll
; Extreamly early beta of a task-schedule checker
CheckTaskSched.dll
```

Ensuite il faut changer le **password** dans la section **[Settings]** pour que le client communique avec Nagios. On a entré le password pour nagios un peu plus haut, bien entendu il faut que ce soit le même.

```
;# PASSWORD
;  This is the password (-s)
access the daemon remotly.
password=Ton_Password
```

Ensuite il faut décommenter **allowed_hosts** option toujours dans la section **[Settings]**. Et il faut rajouter l'**adresse IP du serveur Nagios** avec lequel il va communiquer.

```
;# ALLOWED HOST ADDRESSES
;  This is a comma-delimited list of IP
;  If leave this blank anyone can access
;  The syntax is host or ip/mask so 192.
allowed_hosts=Adresse IP de Nagios
```

Ensuite il faut vérifier la ligne ou se configure le **port** sur lequel NSClient va communiquer par défaut c'est le **12489** (décommenter la ligne si elle est commentée et penser bien à l'ouvrir dans le pare-feu en TCP)

```
;# NSCLIENT PORT NUMBER
;  This is the port the NSClientListener.dll will listen to.
port=12489
```

Voilà la configuration de NSClient et Nagios est terminé donc maintenant on va démarrer NSClient :

```
nsclient++.exe /start
```

Maintenant on vérifie la configuration de nagios

```
/usr/nagios/bin/nagios -v /usr/ nagios/etc/nagios.cfg
```

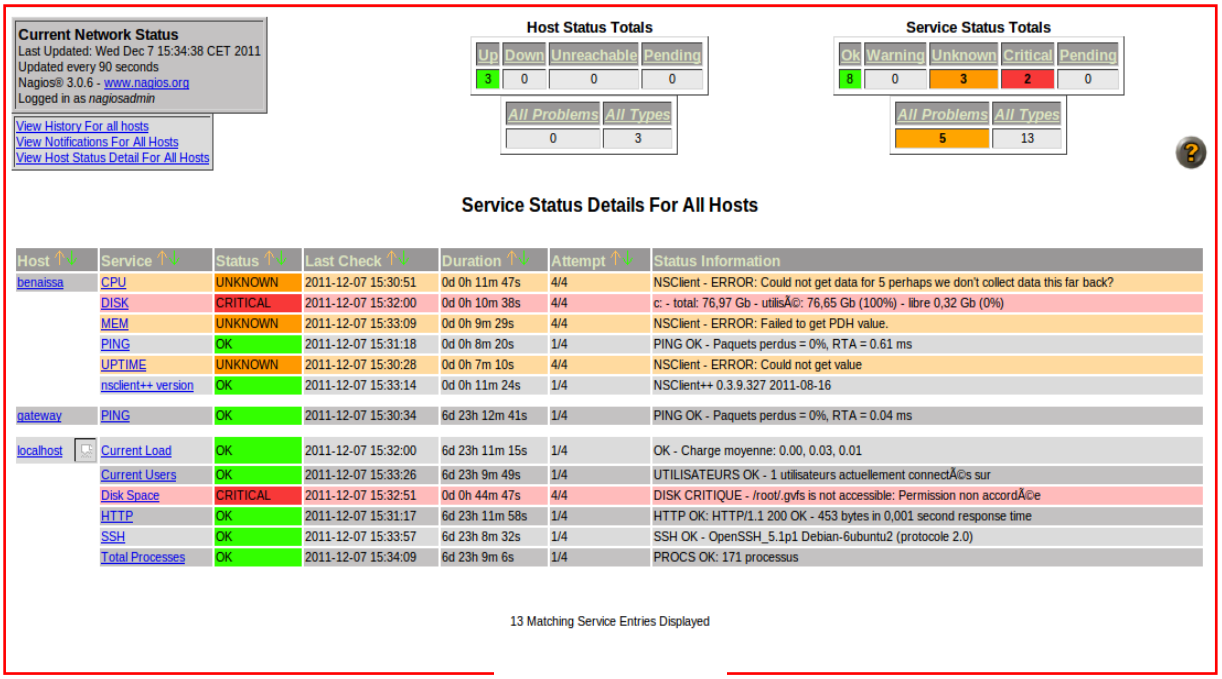


Figure V.4:Services détaillé pour une machines Windows

V.5.Conclusion

Avec les tests que nous pouvons conclure que Nagios est un outil qui fournit une analyse du trafic, le contrôle des liens, services de vérification et même de dispositifs qui prennent en charge SNMP avec Nagios. Malgré la complexité dans la mise en, pourrait déployer un système qui permet au gouvernement central pour contrôler l'ensemble du réseau et d'alerter la personne responsable pour les points de défaillance sont rapidement résolus.

Conclusion Générale

Conclusion générale

Nous sommes intéressées dans notre projet par un service importante pour un administrateur réseau notre travail a été consacré sur la notion de base de la supervision qui est devenue indispensable dans tout système d'information.

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.

Il faut pouvoir surveiller de manière continu l'état des systèmes d'information afin d'éviter un arrêt de production de trop longue durée. C'est là ou la supervision intervient. Elle doit permettre d'anticiper les problèmes et de faire remonter des informations sur l'état des équipements. Donc nous avons donnés tous les étapes nécessaires de l'installation et de configuration d'un service de supervision Nagios.

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.

De plus notre projet peut être développé par ceux qui veulent continuer ce travail pour rendre le système plus sécurisé et performant.

Bibliographie

Bibliographie

[1] LECORCHE Hubert - JEANDROZ Sylvain, SUPERVISION RESEAU AVEC NAGIOS, projet

[2] M. Grégory Bernard, Présentation de l'outil d'administration de réseau Nagios, projet, octobre 2003,

[3] Mr BENAISSA Mohamed,Chapitre N°2 : Architecture client-serveur, cour, 4 avril 2011,

[4] François Borderies, Olivier Chatel, Jean-Christophe Denis, Didier Reis, Administration Réseau,1 juillet 1993.

[5] www.nagios.org/download/

[6] www.guellec.fr/ressources/articles/nagios.php

[7] http://doc.ubuntu-fr.org/nagios

[8] igm .univ-mlv.fr

[9] WWW.RESEAUMAROC.COM

[10] www.nicosphere.net/lunix

[11] http://lunixetleschoses.tuxfamily.org

[12]. http://www.guill.net

ملخص

إن الشبكات المعلوماتية أصبحت كثيرة الاستعمال والانتشار فهي تعمل على تسهيل كثير من مهام الإدارة والمؤسسات.
إنَّ هدفنا الأساسي قائم على إدارة وتنصيب خادم المراقبة Nagios
كلمات رئيسية :, الشبكة المراقبة.Nagios

Résumé

Les réseaux d'information sont largement utilisés. Ils facilitent des de nombreuses tâches d'administration et des institutions
Notre objectif principal est d'administrer et configurer un service de supervision représenté par l'outil Nagios
Mots clés : Supervision, Réseaux NAGIOS

Abstract

Information networks are widely used. They facilitate the many tasks of administration and institutions
Our principal objective of this work is to manage and configure a service of supervision represented by the Nagios tool.
Keywords: Supervision; Network, NAGIOS