



# RAPPORT DE STAGE

CONNECTONS NOS TALENTS



Tuteur entreprise : Nicolas KLINK

Tuteur pédagogique : Bruno BEAUFILS

MODIS

# SOMMAIRE

|  |    |
|--|----|
| SOMMAIRE .....                                       | 2  |
| REMERCIEMENTS.....                                   | 4  |
| 1 PRESENTATION ENTREPRISE.....                       | 5  |
| 1.1 HISTORIQUE .....                                 | 5  |
| 1.2 PRESENCE MONDIALE .....                          | 6  |
| 1.3 PRESENCE NATIONALE .....                         | 6  |
| 1.4 SECTEUR D ACTIVITE.....                          | 7  |
| 1.5 METIERS .....                                    | 7  |
| 1.6 ORGANIGRAMME .....                               | 8  |
| 2 PROJET : SUPERVISION SOUS SHINKEN.....             | 9  |
| 2.1 LE CONTEXTE .....                                | 9  |
| 2.2 PHASES DU PROJET.....                            | 9  |
| 2.3 ETUDE DU BESOIN.....                             | 10 |
| 2.4 LES SOLUTIONS.....                               | 10 |
| 2.5 LES RISQUES .....                                | 10 |
| 2.6 LA REALISATION .....                             | 11 |
| 2.6.1 Installation de shinken sur debian wheezy..... | 11 |
| 2.6.2 configuration générale .....                   | 13 |
| 2.6.3 Utilisation des checks .....                   | 16 |
| 2.6.4 Créer ses propres plugins.....                 | 20 |
| 2.6.5 Interface web de shinken.....                  | 22 |
| 2.6.6 configuration graphique .....                  | 24 |
| 3 PROJET : ADMINISTRATION DES VLANS.....             | 26 |

|   |    |
|---|----|
| 3.1 CONTEXTE.....                       | 26 |
| 3.2 TOPOLOGIE .....                     | 26 |
| 3.3 REPERAGE .....                      | 27 |
| 3.4 SECURITE .....                      | 28 |
| 3.5 MISE A JOUR.....                    | 29 |
| 3.6 ADMINISTRATION .....                | 31 |
| 3.6.1 Serveur d' administration .....   | 32 |
| 3.6.2 Protocole snmp.....               | 32 |
| 3.6.3 Les MIB .....                     | 32 |
| 3.6.4 Automatisation affichage .....    | 35 |
| 3.6.5 Améliorations.....                | 37 |
| 3.6.6 Evolution.....                    | 37 |
| 3.6.7 Avantages et Inconvénients.....   | 38 |
| 3.7 HP VLAN SIMPLE ADMINISTRATION ..... | 39 |
| 3.7.1 Mise en place .....               | 39 |
| 3.7.2 Configuration .....               | 40 |
| 3.7.3 Interface graphique.....          | 41 |
| 3.7.4 Avantages et inconvénients .....  | 42 |
| 3.8 EVOLUTION .....                     | 42 |
| CONCLUSION.....                         | 44 |
| GLOSSAIRE .....                         | 55 |

# REMERCIEMENTS

J'ai eu la chance d'effectuer mon contrat de Professionnalisation dans une SSII, la société Modis située à Villeneuve d'Ascq et je tiens à remercier dans un premier temps Eric BRIANT, responsable du centre des services de m'avoir accueilli au sein de la structure.

Je remercie également Nicolas KLINK, mon tuteur entreprise et Bruno BEAUFILS, mon tuteur Pédagogique pour l'aide et les conseils qu'ils m'ont apportés lors des différents suivis de mon projet.

Je tiens à remercier tout particulièrement et à témoigner toute ma reconnaissance à Florent HOUBART et Sébastien GENOS, pour l'expérience enrichissante et pleine d'intérêt qu'ils m'ont fait vivre durant cette période au sein de l'entreprise.

Je remercie Yvan PETER, Responsable pédagogique, de m'avoir accepté en Licence Pro CGIR ainsi que tous les enseignants participants internes et externes à cette formation.

Je tiens enfin à exprimer toute ma gratitude envers toutes les personnes qui m'ont accompagné, de près ou de loin dans cette aventure professionnelle et humaine.

---

**Pour des raisons de confidentialité et à la demande de la Direction de la société Modis, certaines informations ont été volontairement cachées ou modifiées.**

---

# 1 PRESENTATION ENTREPRISE

## 1.1 HISTORIQUE

- **1969** Création d'Ajilon aux Etats-Unis.
- **1990** Acquisition en France de Cenergys chez Adecco Consulting.
- 1996** Ajilon devient une filiale du Groupe Adecco.
- 2002** Regroupement de l'activité IT sous la marque Ajilon IT Consulting créée au sein de la société Ajilon.
- 2008** Datavance rejoint le groupe Adecco.
- 2012** Datavance et Ajilon IT Consulting se regroupent pour devenir Modis.

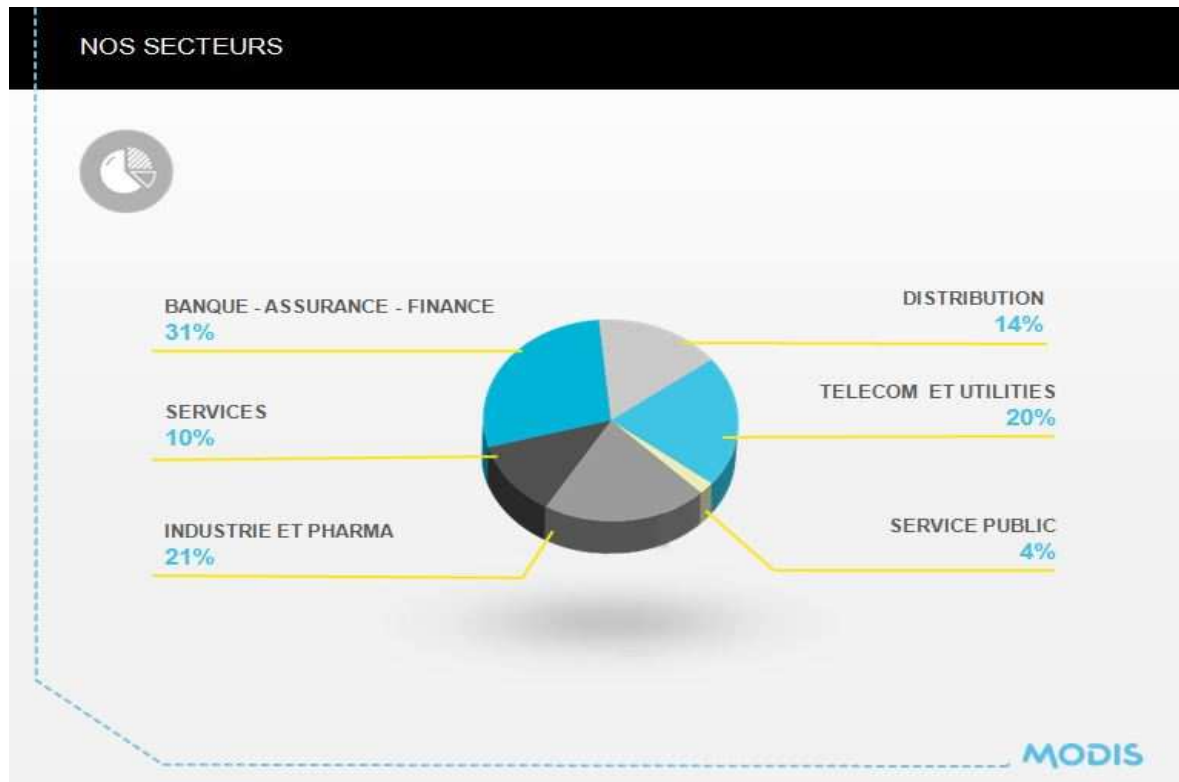
## 1.2 PRESENCE MONDIALE



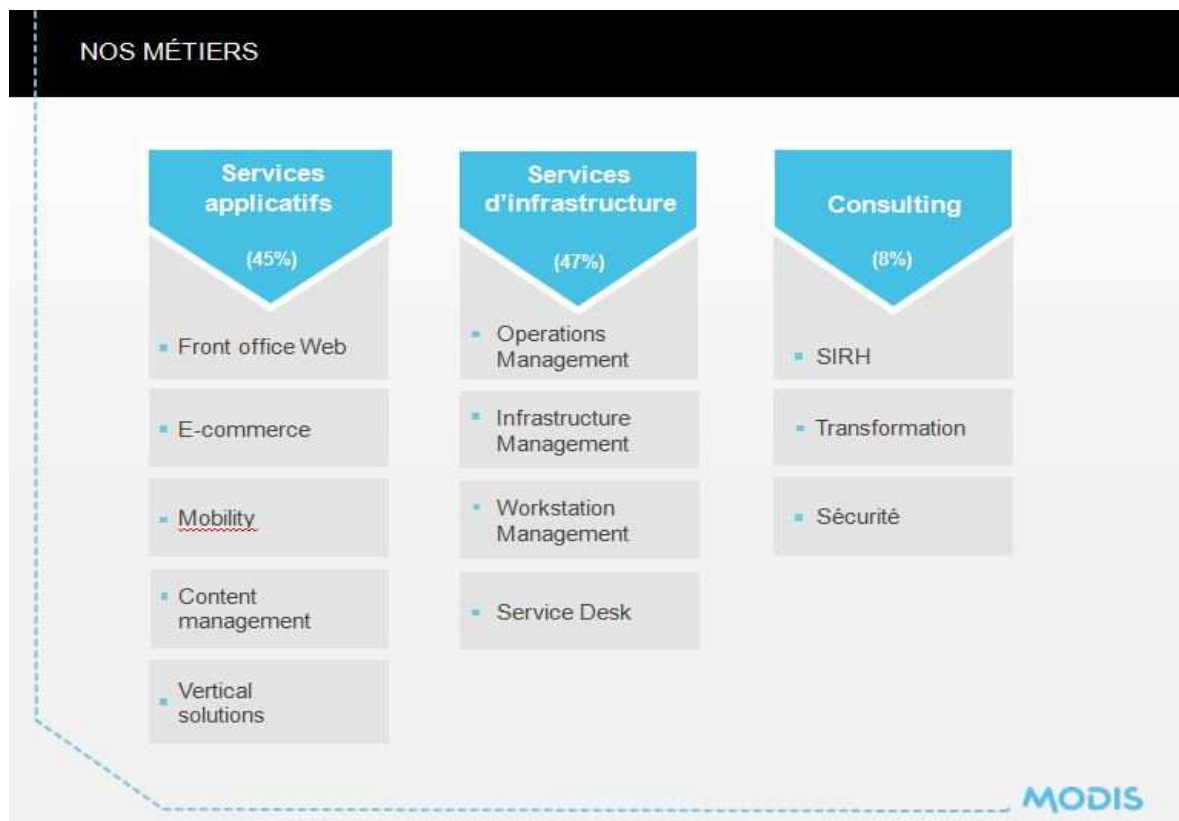
## 1.3 PRESENCE NATIONALE



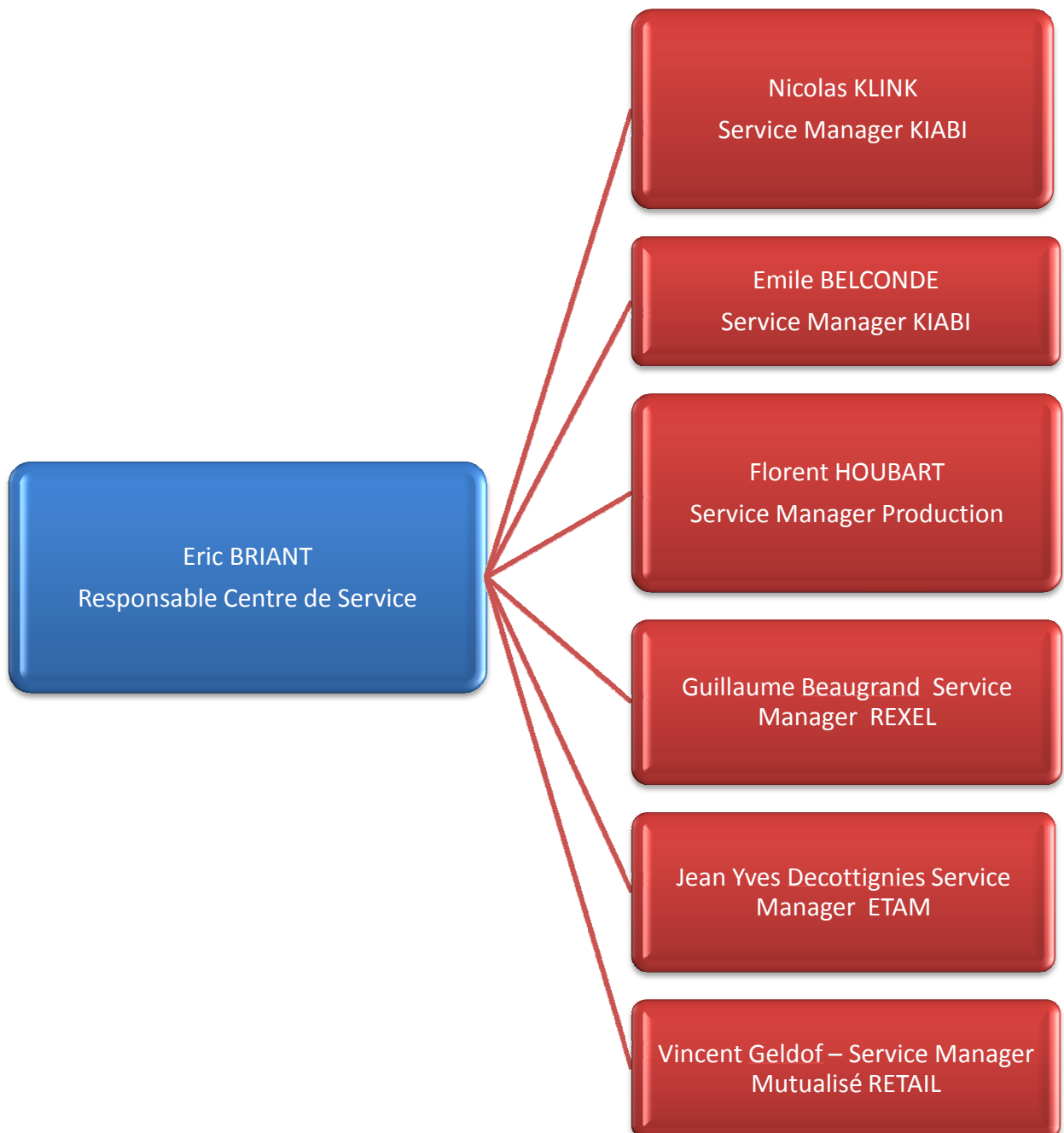
## 1.4 SECTEUR D ACTIVITE



## 1.5 METIERS



## 1.6 ORGANIGRAMME





## 2 PROJET : SUPERVISION SOUS SHINKEN

**Shinken\*** est un logiciel de supervision à code ouvert et sous licence libre qui offre d'immenses avantages. Cette refonte complète du moteur Nagios\* développé en python\* est multiplateforme\*. Avec sa multitude de greffons\*, il permet de maîtriser les coûts tout en facilitant l'intégration de matériels non encore supportés. L'équipe de Supervision de savoir-faire Linux\* est très active dans la communauté Shinken, qui ne cesse de développer de nouveaux modules\* et de les partager.

### 2.1 LE CONTEXTE

Aujourd'hui aucune supervision\* proactive n'est réalisée sur le centre de service Modis. Cela implique :

- L'incapacité de connaître le bon fonctionnement du centre de service sans déranger les techniciens.
- Aucune réactivité en cas d'incident. Ils ne sont pris en compte qu'une fois remontés par les techniciens et donc trop tard.

Modis cherche à être plus réactif et fournir un outil visuel permettant d'avoir un état de ses services en temps réel.

L'objectif du projet de supervision est :

- Améliorer la réactivité / Prendre les bonnes décisions en cas d'incident.
- Anticiper rapidement les impacts.
- Tracer les événements (récurrence / fréquence).

### 2.2 PHASES DU PROJET

La réalisation de ma mission s'est déroulée en phase Projet.

- Étude du besoin sur l'ensemble des prestations.
- Regroupement des données et ciblage des éléments critiques.
- Choix d'une solution technique adaptée.
- Anticipations des risques potentiels.
- Réalisation.

## 2.3 ETUDE DU BESOIN

Rencontre avec les responsables des prestations pour établir une liste des outils à superviser\* :

- Téléphonie.
- Reporting\*.
- Poste de travail.
- Réseau\*.
- Ticketing\*.
- Base de connaissances\*.
- Serveur de fichiers.

## 2.4 LES SOLUTIONS

Solutions « payantes » : (Patrol Openview, BMC, Tivoli).

Avantages : Outils reconnus sur le marché / Support (payant).

Inconvénients : Coût / Nécessite des ressources avec des connaissances spécifiques.

Solutions « Open source » (Nagios, Centreon, Shinken)

Avantages : Faible Coût / Communauté importante / Flexibilité

Inconvénients : Pas de support\* ou support payant / Nécessite des ressources avec des connaissances spécifiques.

Le choix Modis s'est orienté vers la solution « Open Source ».

## 2.5 LES RISQUES

Risques détectés :

Compétence Open-source : Capacité à se former rapidement.

Délai de réalisation du projet (fin de la mission prévue pour fin Décembre 2014).

## 2.6 LA REALISATION

La distribution utilisée sur les serveurs Linux chez Modis est Débian\* et dans le but d'homogénéiser le parc, j'ai choisi cette distribution pour le support d'installation de Shinken. La version à la date où j'ai réalisé ce projet étant Debian 7 wheezy\*.

### 2.6.1 Installation de shinken sur debian wheezy

#### Plusieurs versions existent :

Pour la version via python\* shinken 2.0 :

Src : <https://mespotesgeek.fr/installation-de-shinken-2-0-sur-debian-wheezy/>

J'ai utilisé la version 1.2 car c'est celle utilisée sur une installation existante. Le but dans un premier temps est de valider ma configuration sur une machine test en local et dans un second temps, après validation fonctionnelle et technique, d'importer mes fichiers de configuration sur une machine en place sur un Esx Vmware\*.

#### Prérequis

Avant l'installation de Shinken il faut que notre support soit à jour. Shinken , développé en python, nécessite l'installation de certains modules. La mise en place d'un serveur Lamp\* est également nécessaire pour le stockage des informations et l'affichage via navigateur.

– Mises à jour :

```
#apt-get update && apt-get upgrade
```

– Installation de Lamp server :

```
#apt-get install apache2 php5 mysql-server
```

– Installation des modules python :

```
#apt-get install pyro lsb-release
```

#### Installation

wget <http://shinken-monitoring.org/pub/shinken-1.2.tar.gz>

```
tar -xvzf shinken-1.2.tar.gz
cd shinken-1.2
/install -i
```



(src : img shinken.jpg)

Il ne faut pas oublier de changer le mot de passe

A ce sujet, il faut savoir que le système d'authentification s'appuie sur la configuration contenue dans le fichier **shinken-specific.cfg** (dans le répertoire **/etc/shinken**).

Dans le cas de l'interface web\* de shinken (avec sa configuration par défaut), ce dernier va s'appuyer sur ces fichiers de configuration. Dans le cas de l'utilisateur admin, il s'agit d'un contact (au sens nagios) défini dans les fichiers de configuration (par défaut, fichier *contacts.cfg*). Le mot de passe est par ailleurs défini de la manière suivante :

```
define contact{
    use                generic-contact
    contact_name       admin
    email              shinken@localhost
    pager              0600000000 ; contact phone
    number
    password           admin
    is_admin           1
}
```

De là, on peut changer le mot de passe et le faire prendre en compte par shinken

Src : <http://lesaventuresdeyannigdanslemondeit.blogspot.fr/2012/02/utilisation-du-mot-de-passe-admin-de.html>

## Installation des plugins Nagios :

Afin d'ajouter des fonctionnalités à l'outil de supervision Shinken, j'ai décidé d'installer les plugins\* Nagios.

```
apt-get install nagios-plugins
```

Il faut mettre les plugins au même endroit que ceux de shinken.

```
cp -R /usr/lib/nagios/plugins/*  
/var/lib/shinken/libexec/
```

## 2.6.2 configuration générale

Les fichiers de configuration de shinken se situent dans :

```
/etc/shinken/
```

Voici les éléments importants qui constituent sa configuration.

### Les « hostgroups »

On dit que tel test de service est fait sur tel groupe et pour chaque hôte on indique à quel groupe il appartient.

### Les « contacts »

Les contacts servent à indiquer qui notifier en cas de problème et comment le prévenir.

### Les « hosts »

Syntaxe de définition d'un hôte :

```
define host {  
    host_name      gingembre  
    alias          gingembre  
    address       gingembre.rhizome-fai.net  
    use           linux-host  
    contacts      rhizome-alert  
    hostgroups    serveur-linux,routeur  
}
```

### Les « services »

Les services correspondent aux différents tests / sondes\*. Pour chaque service on définit l'hôte ou le groupe d'hôtes concerné (grâce à "host\_name" ou "hostgroup\_name"). On indique ici le nom du test à réaliser (la check\_command) avec éventuellement des paramètres séparés par un point d'exclamation.

Exemple :

# Service à distance sur un groupe d'hôtes

```
define service {
    hostgroup_name      serveur-linux
    service_description Disk
    check_command       check_nrpe!check_disks
    use                 generic-service
}
```

# Service à distance sur un hôte

```
define service {
    host_name           igrname
    service_description MySQL
    check_command       check_nrpe!check_mysql_rhizome
    use                 generic-service
}
```

# Test lancé directement sur la machine locale (test de connexion SMTP\*), sur un hôte donné (igrname)

```
define service {
    host_name           igrname
    service_description SMTP
    check_command       check_smtp
    use                 generic-service
}
```

### Les « commandes »

Les commandes ne décrivent pas de quelle façon réaliser un test, mais plutôt quel test est à effectuer. La partie la plus technique du test se trouve dans ce fichier. On a le nom de la commande de test (qui est donnée dans la configuration des services) et la commande à réellement effectuer pour réaliser le test.

Dans la commande à exécuter, on peut utiliser des variables :

***\$PLUGINDIR\$*** : Le dossier où sont stockés les programmes de tests. Certains sont fournis directement avec shinken, d'autres sont inclus dans des plugins à installer. On peut bien sûr ajouter ses propres fichiers. Ce répertoire est, pour notre installation : */usr/local/shinken/libexec*.

***\$HOSTADDRESS\$*** : L'adresse de l'hôte sur lequel le test est effectué.

***\$ARGx\$*** : Les arguments donnés dans la configuration des services. ***\$ARG1\$*** correspond à la chaîne entre le premier et le deuxième point d'exclamation, ***\$ARG2\$*** après le deuxième.

**# Compare time between target and shinken**

```
define command {
    command_name      check_snmp_time
    command_line      $PLUGINDIR$/check_snmp_time.pl      -H
                    $HOSTADDRESS$ -C $SNMPCOMMUNITYREAD$ -f -w  $ARG1$      -c
                    $ARG2$
}
```

Utiliser des variables permet d'avoir des définitions de commandes génériques. En cas de changement d' »host », on ne modifie qu'une seule variable d'environnement. Ces paramètres sont spécifiés dans la définition de l'hôte ou du service en les séparant par des points d'exclamation comme ceci :

```
define service{
    host_name          linuxbox
    service_description PING
    check_command      check_ping!200.0,80%!400.0,40%
}
```

### Les checks

Un check\* est un contrôle utilisé pour prélever des informations sur une machine. On dit que ces checks sont actifs quand ils sont directement lancés par shinken. Dans le cas de checks passifs, ils sont lancés et exécutés par une application externe et sont soumis à Shinken pour le traitement.

Ces checks sont regroupés dans

[\*/usr/local/shinken/libexec\*](#)

Sinon , rechercher l'emplacement :

[\*Locate libexec\*](#)

Ma configuration est :

[\*/var/lib/shinken/libexec\*](#)

### Prise en compte des modifications par shinken

Lorsqu'on modifie un fichier de configuration, il faut redémarrer shinken pour la prise en compte de la modification et éventuellement vérifier qu'il n'y a pas d'erreurs.

A faire après chaque modification du paramétrage.

Tester la configuration.

```
/etc/init.d/shinken check
```

Si le check est Failed, vérifier les logs :

```
cat /tmp/shinken_checkconfig_result
```

Relancer les services Shinken:

```
/etc/init.d/shinken restart
```

Remarque(s) :

Afin d'avoir le plus de détails possible dans les logs\*, il est possible de démarrer en mode verbeux\*.

Démarrage en Mode debug\* :

```
/etc/init.d/shinken -d start
```

### 2.6.3 Utilisation des checks

#### ***Exemple : avec check\_NRPE***

**NRPE pour Nagios Remote Plugin Executor** est un plugin de Nagios qui permet d'exécuter des commandes de supervision sur des machines distantes.

Par exemple, dans le dossier `/var/lib/shinken/libexec`, nous avons plusieurs scripts\* de check comme **check\_load**, **check\_disk**, **check\_users** etc... qui ne fonctionnent que sur la machine locale.

NRPE permet donc d'utiliser ces différents scripts pour superviser des machines distantes. Par convention je nommerai les machines à superviser : « client » et le superviseur shinken : « serveur ».

#### **Pour un client Linux :**

Sur la machine client, il faut taper la commande suivante:

```
apt-get install nagios-nrpe-server
```

Editer le fichier nrpe.cfg présent dans le dossier `/etc/nagios/`

Et mettre à la place de 127.0.0.1 l'adresse IP\* de notre serveur shinken.



Maintenant, il faut descendre plus bas dans ce fichier de configuration jusqu'aux lignes suivantes.

```
# 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/che ck_procs -
w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_
procs -w 150 -c 200
```

Pour les check\_disk, je contrôle le nom des disques dans /dev

```
Ls /dev | grep *da
```

Il faut modifier la commande suivante en conséquence :

```
command[check_hda1]=/usr/lib/nagios/plugins/check_disk - w
20% -c 10% -p /dev/hda1 devient :
command[check_sda1]=/usr/lib/nagios/plugins/check_disk - u
GB -w 20% -c 10% -p /dev/sda1
```

NRPE utilise les scripts de check présents dans le dossier **/usr/lib/nagios/plugins** avec des options mises en place.

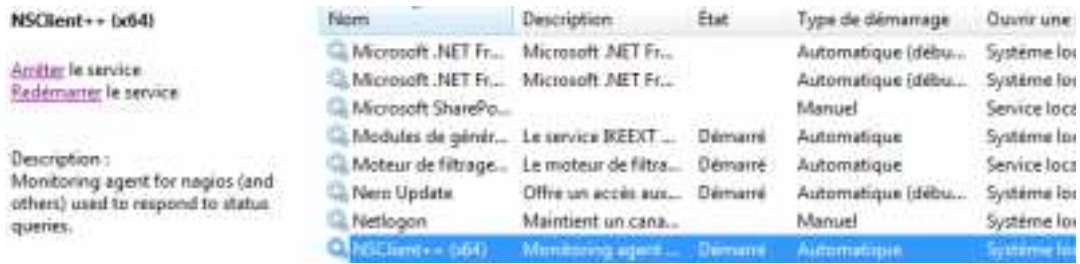
### **Pour un client Windows ex : windows serveur 2003 :**

Installer winrpe sur le client :

Ex : <https://www.itefix.net/winrpe>

Par défaut, en ayant précisé, durant l'installation de NSClient++, l'adresse IP du serveur Shinken, un mot-de-passe et les modules de base à utiliser (tel que par exemple **NSClientListener**), le mode NSClient est pour ainsi dire fonctionnel. Toutefois je peux m'en assurer dans le fichier de configuration. Normalement ce dernier se trouve dans **C:\Program Files\NSClient++\** (ou **C:\Program\NSClient++\** pour une version 64 bits), le fichier de configuration est **NSC.ini**.

Ensuite, il faut s'assurer que le service NSClient++ est bien lancé ou le redémarrer en cas de modification de la configuration. Pour cela, on se rend dans le gestionnaire de services Windows (cliquez sur **Démarrer** puis **Exécuter**, je lance ensuite la commande "services.msc" puis je valide avec la touche entrée).



(src : img nsc.jpg)

Je démarre le service si cela n'est pas le cas. Après chaque modification du fichier de configuration, il faudra aussi ne pas oublier de le redémarrer.

Une fois la machine client configurée pour être supervisée via NRPE, nous revenons au serveur Shinken.

Je regarde si cela fonctionne en tapant la commande suivante:

```
/var/lib/shinken/libexec/check_nrpe -H ipClient
```

Je dois obtenir la version de NRPE installée sur le client:

Ex : NRPE v2.12

Maintenant je teste mon premier check via NRPE, avec la commande `check_sda1` que j'ai modifiée précédemment sur le client.

```
./usr/local/nagios/libexec/check_nrpe -H ipClient -c check_sda1
```

Je dois obtenir quelque chose de cette forme:

```
DISK OK - free space: / 17 GB (95% inode=97%);| /-0GB;14;16;0;18
```

Lorsque cela fonctionne en ligne de commande, je le mets en place dans shinken en utilisant la commande située dans le fichier de configuration **check\_nrpe.cfg** présent dans le dossier **/etc/shinken/commands/**.

Si `check_nrpe.cfg` n'existe pas, il faut le créer :

```
# vi check_nrpe.cfg
```

Je tape les lignes suivantes:

```
##Commande check_nrpe
define command{
    command_name        check_nrpe
    command_line        $USER1$/check_nrpe -H      $HOSTADDRESS$
    -c $ARG1$
}
```

Puis, j'ajoute les services à superviser dans le fichier contenant les services shinken :

```
#vi /etc/shinken/services/services.conf
```

```
##Check PING
define service{
    use                               local-service
    host_name                          nagios,    portable,
asterisk
    service_description                PING
    check_command
    check_ping!100.0,20%!500.0,60%
}
##Check NRPE sda1
define service{
    use                               local-service
    host_name                          nagios,    portable,
asterisk
    service_description                Sda1
    check_command                      check_nrpe!check_sda1
}
##Check NRPE users
define service{
    use                               local-service
    host_name                          nagios,    portable,
asterisk
    service_description                Users
    check_command                      check_nrpe!check_users
}
##Check NRPE load
define service{
    use                               local-service
    host_name                          nagios,    portable,
asterisk
    service_description                Load
    check_command                      check_nrpe!check_load
}
##Check NRPE total procs
define service{
    use                               local-service
    host_name                          nagios,    portable,
asterisk
    service_description                Total Procs
    check_command                      check_nrpe!check_total_procs
}
```

Je redémarre shinken via la commande :

```
service shinken restart
```

## 2.6.4 Créer ses propres plugins

### *Introduction*

Il arrive parfois que l'on ne trouve pas le plugin adéquate sur ce que l'on désire contrôler et dans ce cas il faut le créer soit même. Le plugin que je vais écrire pourra me servir de base pour réaliser des plugins plus complexes.

Voici mon cahier des charges pour sa création:

- Ouvrir une session SSH\* vers le serveur distant
- Lancement d'un script vérifiant si un processus\* (testd) est lancé
- Renvoi de l'état du processus (OK / ERROR / WARNING)
  - OK: Le processus (testd) est lancé et fonctionne correctement.
  - ERROR: Le processus n'est pas lancé
  - WARNING: Le processus est lancé mais ne fonctionne pas correctement.

### *Création du plugin*

Je vais donc dans un premier temps créer le plugin qui va lancer la commande sur le serveur distant:

```
# vi check_testd
```

```
#!/bin/bash
#####
#####
#      Creation:BERTRAND.D
#      LastModification:
# This script checks testd daemon is running on a server
#####
#####
ssh nagios@$1 /usr/local/bin/nagios_testd.pl
```

Une fois l'automatisation du SSH effectuée entre le serveur Nagios et le serveur distant (il ne faut pas que le script demande le password...). Il reste à créer le script nagios\_testd.pl dans le répertoire /usr/local/bin du serveur distant.

```
# vi /usr/local/bin/nagios_testd.pl
```

```

#!/bin/bash
#####
#
#           Creation:                BERTRAND.D
#           Last           Modification:
# This script is polling if testd daemon is running
#####
#
STATE_OK=0
STATE_WARNING=1
STATE_CRITICAL=2
STATE_UNKNOWN=3
STATE_DEPENDENT=4
ps auxw | grep [t]estd | grep -v nagios > /dev/null
STATE=$?
if ["$STATE"="$STATE_OK"]
then
echo "TESTDOK"
exit 0
else
echo "TESTDFailed"
exit $STATE_CRITICAL
fi

```

Pour tester ce plugin, il faut lancer cette ligne de commande sur le serveur Nagios:

```

# /usr/lib/nagios/plugins/check_testd.sh
monserveurdistant.mondomaine.com
TESTD OK

```

Si cela ne fonctionne pas, il faut d'abord vérifier que les fichiers ont les bons droits (lecture et exécution) et que le SSH fonctionne correctement entre les deux machines.

### ***Intégration***

Une fois le plugin validé, il faut l'intégrer dans nos fichiers de configuration.

```

#'check_testd'      command      definition
definecommand{
command_name       check_testd
command_line       $USER1$/check_testd $HOSTADDRESS$
}

```

```

# vi /etc/nagios/checkcommands.cfg

```

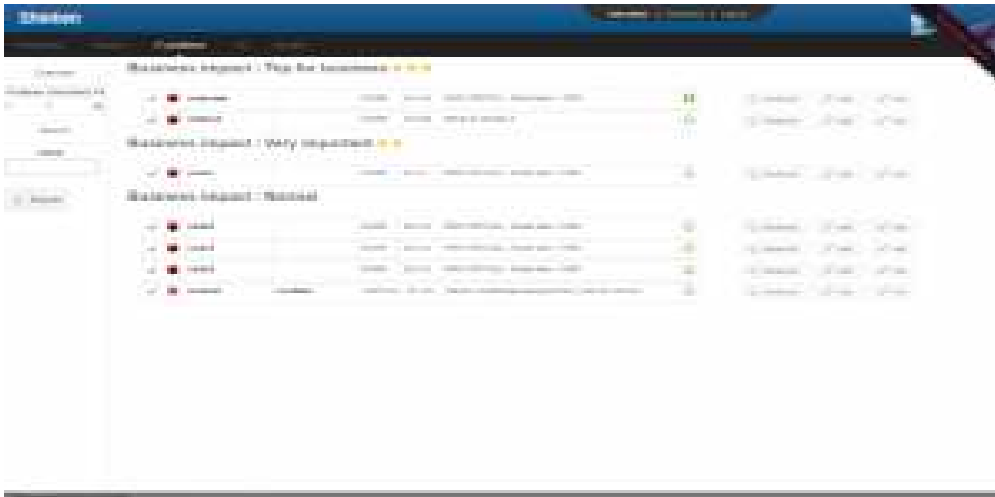
et enfin:

```
# vi /etc/nagios/services.cfg
```

```
define service{
use generic-service
host_name monserveurdistant.mondomaine.com
service_description TESTD
check_command check_testd
}
```

### 2.6.5 Interface web de shinken

Shinken WebUI\* est lancé au démarrage Shinken, et configuré à l'aide du fichier de configuration principal Shinken en y mettant quelques paramètres de base.



(src : img webui.jpg)

#### **Paramètres de base**

J'active le module webui dans shinken-specific.cfg sur le serveur.

```
#vi /etc/shinken/modules/webui.cfg
```

```

define module{
    module_name      WebUI
    module_type      webui

server
    host            0.0.0.0    ; mean all interfaces of your broker
    port            7767

    # CHANGE THIS VALUE or someone may forge your cookies
    auth_secret     TOCHANGE

    # Allow or not the html characters in plugins output
    # WARNING: so far, it can be a security issue
    allow_html_output 0

    # Option welcome message
    #login_text      Welcome to ACME Shinken WebUI.

    #http_backend    auto
    # ; can be also: wsgiref, cherrypy, paste, tornado,
twisted
    # ; or gevent. auto means best match in the system.
    modules
    Apache_passwd,ActiveDir_UI,Cfg_password,Mongodb

    # Modules available for the WebUI:
    #
    # Note: Choose one or more authentication methods.
    #
    # Apache_passwd: use an Apache htpasswd files for auth
    # ActiveDir_UI: use AD for auth and photo collect
    # Cfg_password: use passwords in contacts configuration
for authentication
    #
    # PNP_UI: Use PNP graphs in the UI
    # GRAPHITE_UI: Use graphs from Graphite
    #
    # Mongodb: Necessary for enabling user preferences in
WebUI
}

```

### ***Modules d'authentification***

Le module WebUI recherche le mot de passe utilisateur et permet l'authentification ou non. Le mot de passe par défaut se trouve dans le module **cfg\_password\_webui**.

```

define module{
    module_name Cfg_password
    module_type cfg_password_webui
}

```

Ce module utilise un fichier Apache passwd (htpasswd) pour l'authentification. Il faut indiquer le chemin complet du fichier,

```
define module{
    module_name      Apache_passwd
    module_type      passwd_webui

    # WARNING: put the full PATH for this value!
    passwd           /etc/shinken/htpasswd.users
}
```

Vérifier le propriétaire (l'utilisateur doit être Shinken) et les droits sur ce fichier (en lecture).

Si on ne dispose pas d'un tel fichier, on peut le générer avec la commande "htpasswd" (dans Debian "apache2-utils" package).

### **Important**

Pour être en mesure de se connecter à l'interface utilisateur Web, les utilisateurs doivent également être déclarés en tant que contacts Shinken, ainsi, l'ajout d'un utilisateur dans ce fichier sans l'ajouter dans les contacts n'aura aucun effet.

### **2.6.6 configuration graphique**

La partie graphique de base shinken est webui

Pour pouvoir créer des cartographies\* il existe des modules .J'ai utilisé ici : nagvis\*

### ***Installation de Nagvis***

```
#apt-get install nagvis
```

Nagvis communique avec Shinken à travers le module LiveStatus\*.

Pour activer le module Livestatus, Nagvis a besoin de savoir où l'API\* Shinken **Livestatus** est hébergé.

Dans Nagvis, le fichier de configuration est :

```
[Backend_live_1]
backendtype = " mklivestatus
htmlcgi = "/ nagios / cgi-bin"
socket = "tcp: localhost: 50000"
```

*/etc/nagvis/nagvis.ini.php :*

Pour ma configuration, le broker\* de shinken est installé en local donc je remplace localhost par l'adresse IP de mon serveur Shinken.



Il faut ensuite de créer des cartes, les charger via l'interface graphique et positionner nos voyants de supervisions.

### ***Cartes de supervision***

La partie création graphique de shinken est assez intuitive.

Il suffit de créer des images au format .jpg ou .png pour les cartes et placer nos voyants de supervision.

**(voir annexe 1)**

# 3 PROJET : ADMINISTRATION DES VLANS

## 3.1 CONTEXTE

Au sein de l'équipe technique Modis composé de 2 administrateurs et suite à l'arrivée d'un nouveau client au sein du centre de service, il est nécessaire de modifier nos installations techniques/organisationnelles. Nous voulons profiter de cette opportunité pour mettre au propre et à jour nos installations puis, trouver une solution afin d'être toujours plus réactif lorsque les problèmes se présentent.

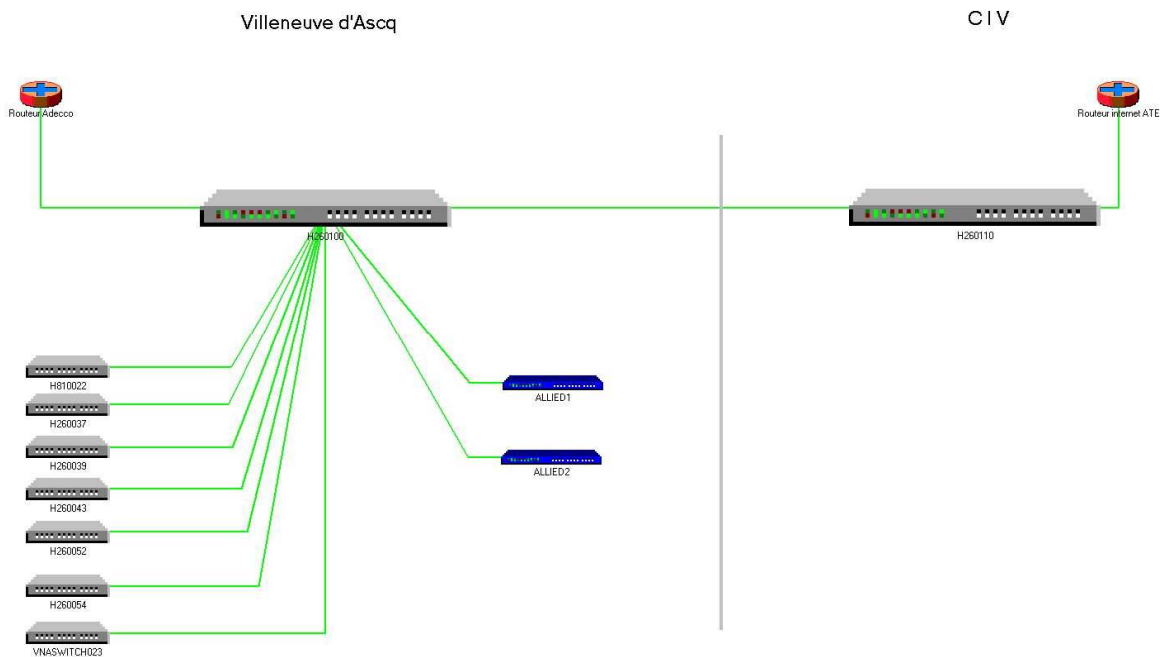
Les switchs\* en place dans la structure sont des HP\* Procurve 2626 j4900B 24 et 2650 j4899B 48 ports. Les moyens employés pour visualiser les informations sur nos switchs sont des fichiers Excel qui ne sont pas forcément mis à jour, faute de temps, et cela impose souvent une connexion sur le switch pour vérifier les informations. Les prises du bandeau de brassage\* sont également reportées sur ces fichiers.

Pour modifier la configuration des Vlans\*, on doit se connecter aux switchs à l'aide de Putty. Ces switchs, qui ont quelques disfonctionnements aléatoires, n'ont pas été mis à jour depuis longtemps. Les points de sécurité sont également à vérifier car certains switchs permettent par exemple une connexion en Telnet\*.

On m'a donc demandé de trouver une solution qui faciliterait l'administration, le repérage, la sécurité et leur mise à jour.

## 3.2 TOPOLOGIE

Des switchs 24 et 48 ports HP Procurve sont connectés, avec une topologie\* en étoile, à un switch, « cœur de réseau \* », modulable (ports 1 Gbps\*). Ce switch est équipé d'une alimentation redondante. Notre réseau est implémenté pour une partie dans une salle serveur\* à Villeneuve d'Ascq et pour l'autre dans un Datacenter\* (CIV) à Sainghin-en-Mélantois.



### 3.3 REPERAGE

#### *Brassage*

Ma première mission a été le repérage des connectiques, le nettoyage (retirer les câbles déconnectés) et de créer des tableaux via Excel\*.



(src : img brassage.jpg)

Ensuite, en me connectant aux switches avec Putty\*, ces tableaux ont été complétés avec leurs Vlan respectifs afin de déterminer les « tags »\* des ports\* et ceux qui ne sont plus utilisés.

| Port | Branchement  | AJILON | IDGROUP  | REXEL    | AJIT_PROD | CONTENTIA_WA | TRANSPOLE | RYWEAR_WA | ETAM_WAN | VIT_PROD | CONTENTIA | NOCIBE | FLAKT_INT | HC_INT | ID_RECETT | DMZ_ADMIN | PHILDAR2 | Port |
|------|--------------|--------|----------|----------|-----------|--------------|-----------|-----------|----------|----------|-----------|--------|-----------|--------|-----------|-----------|----------|------|
| 1    | 312          |        |          |          | Untagged  |              |           |           |          |          |           |        |           |        |           |           |          | 1    |
| 2    |              |        |          |          | Untagged  |              |           |           |          |          |           |        |           |        |           |           |          | 2    |
| 3    | 203          |        |          |          | Untagged  |              |           |           |          |          |           |        |           |        |           |           |          | 3    |
| 4    | 305          |        |          |          |           |              |           |           | Untagged |          |           |        |           |        |           |           |          | 4    |
| 5    | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 5    |
| 6    | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 6    |
| 7    | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 7    |
| 8    | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 8    |
| 9    | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 9    |
| 10   | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 10   |
| 11   | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 11   |
| 12   | IDGROUP      |        | Untagged |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 12   |
| 13   | IDGROUP/ETAM |        | Tagged   |          | Tagged    |              |           |           | Tagged   |          |           |        |           |        |           |           |          | 13   |
| 14   | IDGROUP/ETAM |        | Tagged   |          | Tagged    |              |           |           | Tagged   |          |           |        |           |        |           |           |          | 14   |
| 15   | IDGROUP/ETAM |        | Tagged   |          | Tagged    |              |           |           | Tagged   |          |           |        |           |        |           |           |          | 15   |
| 16   | IDGROUP/ETAM |        | Tagged   |          | Tagged    |              |           |           | Tagged   |          |           |        |           |        |           |           |          | 16   |
| 17   |              |        |          |          | Untagged  |              |           |           |          |          |           |        |           |        |           | Tagged    |          | 17   |
| 18   |              |        |          |          | Untagged  |              |           |           |          |          |           |        |           |        |           | Tagged    |          | 18   |
| 19   |              |        |          |          | Untagged  |              |           |           |          |          |           |        |           |        |           | Tagged    |          | 19   |
| 20   | 327          |        |          | Untagged |           |              |           |           |          |          |           |        |           |        |           |           |          | 20   |
| 21   | 345          |        |          |          |           |              |           |           |          |          |           |        | Untagged  |        |           |           |          | 21   |
| 22   | 328          |        |          | Untagged |           |              |           |           |          |          |           |        |           |        |           |           |          | 22   |
| 23   | 335          |        |          |          |           |              |           |           |          |          |           |        |           |        |           |           | Untagged | 23   |
| 24   |              |        |          |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 24   |
| 25   | H260100 D21  | Tagged | Tagged   |          |           | Tagged       | Tagged    | Tagged    | Tagged   |          | Tagged    | Tagged | Tagged    |        | Tagged    | Tagged    | Tagged   | 25   |
| 26   |              |        |          |          |           |              |           |           |          |          |           |        |           |        |           |           |          | 26   |

(src : img tabswitch.xls)

### Accès configuration des ports

Se connecter en ssh à l'aide de Putty par exemple.

Entrer le **log** et **passwd**.

Taper **Menu** entrer.

Choisir **2 switch configuration** entrer.

Choisir **8 VLAN menu** entrer.

Choisir **3 Vlan Port Assignment** entrer.

## 3.4 SECURITE

Les switches sont administrés actuellement sur un vlan historique : Ajilon situé en structure. Le seul moyen de les gérer est donc d'utiliser une machine en structure, ce qui n'est pas très pratique.

Le but est donc de ramener les switches sur un vlan d'administration : Dmz\_admin situé en production\*.

Une fois cette tache réalisée, certains points liés à la sécurité sont à vérifier.

### Adresse IP de management

Taper **Menu** entrer.

Choisir **2 switch configuration** entrer.

Choisir **5 IP configuration** entrer.

J'édite l'adresse IP de management et la passerelle.

### Adresses IP autorisées

Taper **Menu** entrer.

Choisir **2 switch configuration** entrer.

Choisir *7 IP Authorized Managers* entrer.

On édite les adresses IP autorisées à administrer le switch.

### **Telnet et ssh**

Désactivation du Telnet et activation du SSH.

Taper *Menu* entrer.

Choisir *5 Commande Line (CLI)* entrer.

```
# conf t
# crypto key generate ssh
# ip ssh
# no telnet-server
```

### **http et https**

Nos Switchs peuvent être administrés via navigateurs web grâce à l'environnement java.

Préférer HTTPS à HTTP:

```
#crypto key generate cert [512|768|1024]
#crypto host-cert generate self-signed
MM/DD[/[YY]YY] MM/DD[/[YY]YY] <ip du switch>
# web-management ssl
```

La première ligne permet de générer une clé de certificat\* sur le switch, il faut choisir la taille de cette clé en bits parmi 512, 768 et 1024 (ne pas utiliser les crochets !!).

La 2eme ligne permet de générer le certificat, les 2 MM/DD/YYYY correspondant respectivement à la date de début de certificat et à la date de fin du certificat.

La 3eme ligne permet enfin d'activer SSL\* et donc le service HTTPS. SSL ne peut être activé sans avoir préalablement créé et paramétré un certificat réservé à cet effet.

## **3.5 MISE A JOUR**

Pour effectuer ces mises à jour il faut que je libère les switchs qui sont en production, en regroupant les machines sur les ports des

autre switches et en configurant les vlans, ainsi que la continuité de propagation sur les autre switches.

### **Versions IOS**

Les images des IOS\* sont stockées sur 2 partitions de boot\* bien distinctes ( primary et secondary).

Pour avoir les versions d'ios et connaître la partition de boot :

*Show flash*

Changement de l'ordre de boot des partitions :

*boot system flash primary/secondary*

Il faut rechercher la mise à jour sur le site du constructeur :

<http://support.hp.com/fr-fr/drivers/>

exemple : **HP 2626 J4900B** , puis on compare avec les versions de notre switch.

### **Mise à jour sous windows**

Avant toute manipulation il vaut mieux sauvegarder la configuration.

Télécharger un serveur Tftp\*, exemple :

<http://sourceforge.net/projects/tftp-server/>

Il faut flasher\* la partition de l'ios non utilisée, la moins récente.

Après avoir lancé l'application Tftp, on entre l'adresse IP de notre machine et le chemin du fichier de mise à jour.

Sur le switch, on lance la procédure de mise à jour :

*Download IOS*

Il ne reste plus qu'à indiquer l'adresse IP de notre serveur Tftp puis le nom exact du fichier de mise à jour.

Si tout est correct, le switch redémarre avec son nouvelle ios.

### **Mise à jour sous Linux (Debian)**

On installe le serveur Tftp :

*apt-get install tftpd*

Création du répertoire de travail :

*mkdir /sauvegarde\_switchs*

*chmod -R 777 /sauvegarde\_switchs*

*chown -R nobody /sauvegarde\_switchs*

Configuration du serveur tftp :

*Vi /etc/inetd.conf*

Il faut spécifier le chemin du répertoire /sauvegarde\_switchs  
Si modifications, redémarrer tftpd :

```
/etc/init.d/openbsd-inetd restart
```

Sauvegarde de la configuration :

Créer un fichier vide

```
touch /sauvegarde_switchs/5604_Procurve  
chmod -R 777 /sauvegarde_switchs/5604_Procurve  
chown -R nobody /sauvegarde_switchs/5604_Procurve
```

Il ne reste plus qu'à effectuer la sauvegarde à partir du switch :

```
copy running-config tftp « IP_serveur_tftp »  
5604_Procurve
```

Mise à jour :

Il faut télécharger le fichier de mise à jour sur le site constructeur et le placer dans le dossier de sauvegarde du serveur tftp :  
/sauvegarde\_switchs.

Sur le switch :

On flashe la partie non utilisée (secondaire pour moi) par sécurité.  
Dans le cas où cette version est défectueuse ou instable, j'ai toujours  
La possibilité de booter sur l'ancienne version.

```
copy tftp flash « IP_serveur_tftp » H_08_8x.swi  
secondary
```

Pour booter le switch sur la partie secondary

```
5406zl-A(config)# configure terminale  
5406zl-A(config)# boot system flash secondary
```

## 3.6 ADMINISTRATION

Afin de pouvoir administrer plus facilement nos switches, on m'a demandé de trouver une solution de virtualisation regroupant tout les switches avec des infos sur les ports, les vlans et les prises du bandeau de brassage. Au début de mes recherches, je n'ai pas trouvé de solutions clés en main et j'ai donc décidé de remonter mes informations avec un script Shell contenant des requêtes snmp, puis de créer un affichage via une interface web en php et

stocker mes informations dans une base de donnée afin d' y ajouter l'ensemble des informations de brassage\*.

### 3.6.1 Serveur d' administration

J'ai installé une machine Debian 7 wheezy (installation minimale) sur un serveur ESX Vmware pour travailler sur le protocole\* snmp\*.

Il ne faut pas oublier d'ajouter dans les adresses IP autorisées de chaque Switch l'adresse IP de notre machine.

### 3.6.2 Protocole snmp

Dans un premier temps il faut configurer le switch pour activer et sécuriser le protocole snmp.

Je travaille avec la version 2c du protocole.

Taper *Menu* entrer

Choisir *2 Switch Configuration* entrer

Choisir *6 SNMP Community Names* entrer

Il faut éditer cette ligne :

Community : « ma communauté » MIB View : « manager » Write

Access : « Unrestricted ».

Pour la réalisation, il faut connaître les métriques\* de ce que l'on veut superviser, quels sont les MIB\* à ma disposition (celle de mon Switch) puis les parcourir pour savoir quels OID\* vont répondre à mon problème.

### 3.6.3 Les MIB

#### ***Caractéristiques constructeur***

En lisant la documentation d'HP sur les switchs procureur 2626 j4900B. Nous avons :

SNMPv1/v2c/v3

Gestion telnet et HTML

MIB de pont RFC\* 1493

MIB II RFC 1213

MIB de table de réacheminement IP RFC 2096

MIB d'entités RFC 2737

Evolution d'interface RFC 2863



MIB Ethernet RFC 2665

Quatre groupes de RMON\* RFC 2819 : 1 (statistiques), 2 (historiques), 3 (alarmes) et 9 (événements)

Syslog RFC 3164

Configuration de sonde RMON (RMON v2) RFC 2021

MIB MAU\* 802.3 RFC 2668

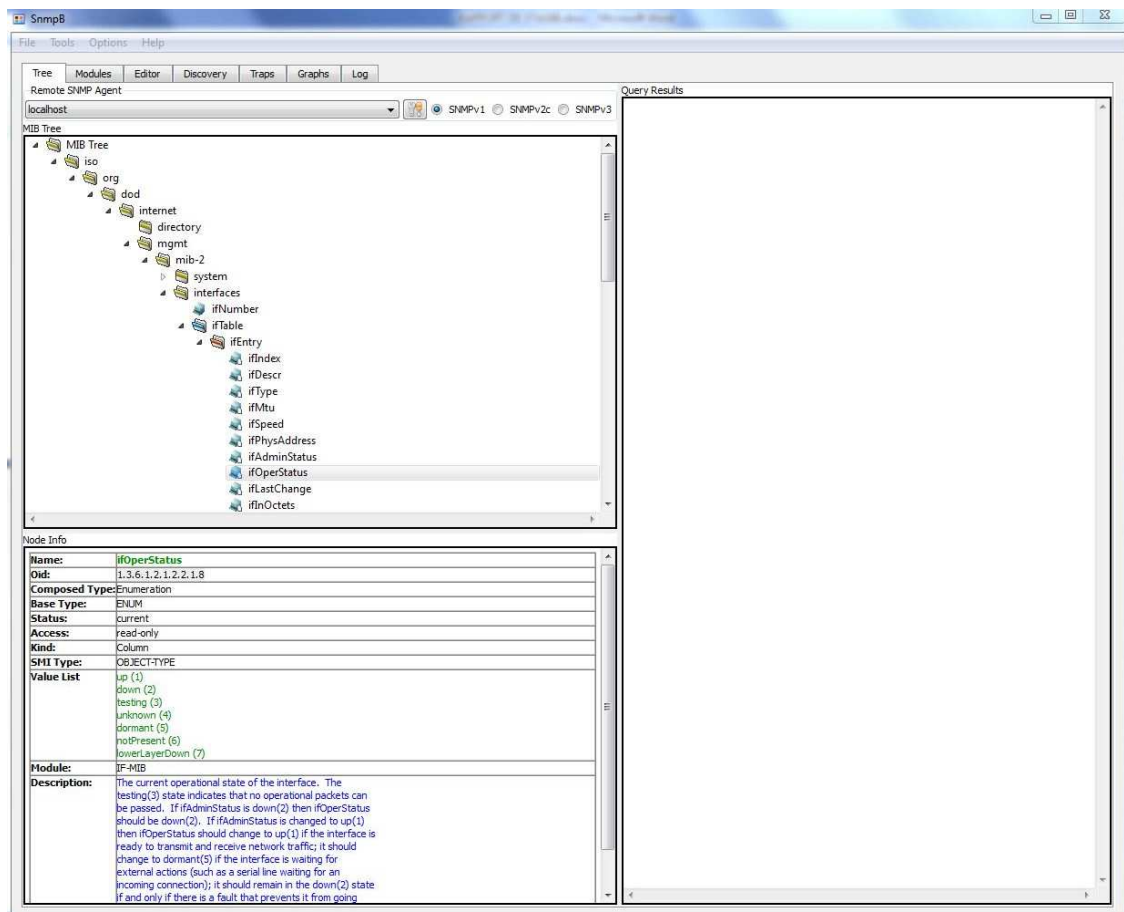
SMON\* RFC 2613

MIB de pont IEEE\* 802.1p et 802.1Q RFC 2674

MIB de client RADIUS\* RFC 2618

### ***Les outils de découverte des OID.***

Pour visualiser l'arborescence standard de la base MIB, j'ai utilisé un outil graphique. Mon poste de travail étant sous windows seven, j'ai installé l'outil snmpB.



(src : img Vmibs)

## **Méthode**

Par défaut le protocole snmp utilise les versions 1 et 2 mais il est toujours possible de renforcer la sécurité avec la version 3.

J'ai employé plusieurs méthodes pour trouver les bons Oid :

- L'outil snmpb.
- La commande snmpwalk avec filtrage.
- Sur le site constructeur et les forums.

### **Première méthode :**

L'outil snmpB nous donne des infos intéressantes sur ce que l'on recherche. Par exemple : des infos sur les ports du switch :

Il faut connaître la communauté dans les paramètres du switch, exemple : communauté = michu .

On interroge la base de donnée de la façon suivante en sachant que X.X.X.X = adresse IP du switch:

Il ne reste plus qu'à effectuer une requête pour obtenir l'info désirée, par exemple, pour l'état du port 15 up ou Down.

```
snmpget -c MICHU -v 2c X.X.X.X ifOperStatus.15
```

### **Deuxième méthode :**

Je me suis servi des commandes de filtrage pour rechercher ce dont j'ai besoin, par exemple, ce qui concerne les Vlan :

```
snmpwalk -c MICHU -v 2c X.X.X.X | grep Vlan
```

```
Q-BRIDGE-MIB::dot1qPortAcceptableFrameTypes.13 = INTEGER:
admitOnlyVlanTagged(2)
Q-BRIDGE-MIB::dot1qPortAcceptableFrameTypes.14 = INTEGER:
admitOnlyVlanTagged(2)
Q-BRIDGE-MIB::dot1qPortAcceptableFrameTypes.15 = INTEGER:
admitOnlyVlanTagged(2)
Q-BRIDGE-MIB::dot1qPortAcceptableFrameTypes.16 = INTEGER:
admitOnlyVlanTagged(2)
Q-BRIDGE-MIB::dot1qPortAcceptableFrameTypes.25 = INTEGER:
admitOnlyVlanTagged(2)
```

En comparant avec les infos du switch, j'ai remarqué que ces lignes m'indiquent les ports du switch qui sont « tagués ».

### **Troisième méthode :**

On peut rechercher des informations dans la Mib privée entreprise du constructeur:

<https://h10145.www1.hp.com/Downloads/ProductsList.aspx?lang=en&cc=us&prodSeriesId=329892>

On peut également trouver les OID souhaités sur internet et les forums.

### ***Difficultés rencontrées***

La lecture des RFC m'a posé des problèmes tant au niveau de la longueur de cette documentation que du repérage des informations.

Lancer un snmpwalk avec un filtrage n'est pas non plus pratique quand on ne connaît pas le nom exact de ce que l'on cherche.

Les méthodes qui m'ont le plus aidé sont l'outil snmpb et les forums.

## **3.6.4 Automatisation affichage**

### ***Script shell***

Lorsque j'ai obtenu toutes les informations souhaitées, à savoir les références de mon switch et les renseignements sur les ports, j'ai décidé d'automatiser les requêtes avec un script shell.

Mon script reprenait la liste des switches à visualiser avec une sélection du switch au départ, puis affichait sur la sortie standard les requêtes snmp du switch sélectionné. Cela m'a permis de vérifier les informations.

Après vérification du fonctionnement de ce script, je l'ai modifié avec une boucle sur tous les switches référencés.

### ***Crontab***

Crontab est un outil qui permet de lancer automatiquement une tâche à une date déterminée.

Je l'ai utilisé pour lancer mon script tous les jours à 0h00.

Ce temps a été déterminé pour limiter la saturation du réseau pendant les heures de travail et pour pouvoir mieux gérer la latence des requêtes. Pour 5 switches à scanner, mon script met à peu près 10 minutes.

```
# Edit this file to introduce tasks to be run by cron.  
#  
# Each task to run has to be defined through a single line  
  
•  
•
```

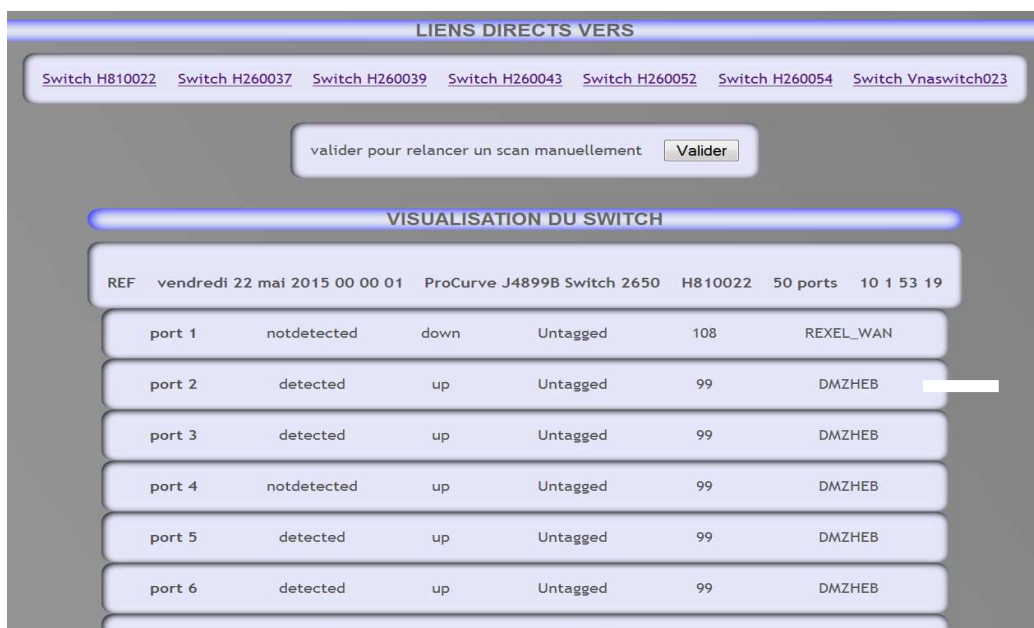
`#Crontab -e`

```
# For more information see the manual pages of crontab(5)  
and cron(8)  
#  
# m h dom mon dow   command  
0 0 * * * /home/scripts_david/snmhtml.sh >  
/var/www/switchs.txt 2>/var/log/snmhtml.log
```

**(voir annexe 2)**

## PHP

J'ai modifié le script pour insérer des « , » de séparation des champs, à la manière d'un .csv. et un repère de changement de switch « REF ». Pour arriver à une interface web plus lisible, j'ai créé en php un affichage des infos du fichier texte.



LIENS DIRECTS VERS

Switch H810022 Switch H260037 Switch H260039 Switch H260043 Switch H260052 Switch H260054 Switch Vnaswitch023

valider pour relancer un scan manuellement

VISUALISATION DU SWITCH

| REF    | vendredi 22 mai 2015 00 00 01 | ProCurve J4899B Switch 2650 | H810022  | 50 ports | 10 1 53 19 |
|--------|-------------------------------|-----------------------------|----------|----------|------------|
| port 1 | notdetected                   | down                        | Untagged | 108      | REXEL_WAN  |
| port 2 | detected                      | up                          | Untagged | 99       | DMZHEB     |
| port 3 | detected                      | up                          | Untagged | 99       | DMZHEB     |
| port 4 | notdetected                   | up                          | Untagged | 99       | DMZHEB     |
| port 5 | detected                      | up                          | Untagged | 99       | DMZHEB     |
| port 6 | detected                      | up                          | Untagged | 99       | DMZHEB     |

(src : img switches)

**(voir annexe 3)**

### 3.6.5 Améliorations

#### **Scan manuel et infos prises**

Suite à un point de suivi de projet, il m'a été demandé d'ajouter des fonctionnalités qui sont détaillées ci après :

- Relance manuelle du script de scan des Switchs.
- Insertion dans une base de données et ajout des numéros de prises de brassage.

Pour solutionner ces demandes, j'ai modifié le code de ce fichier avec l'ajout de 2 boutons qui me renvoient pour l'un vers une page qui traite le scan manuel: scanswitchs.php,

**(voir annexe 4)**

et pour l'autre, une page de création base et table: connexion.php,

**(voir annexe 5)**

d'insertion en base: tableau.php,

**(voir annexe 6)**

de modification: insertion.php,

**(voir annexe 7)**

de traitement: traitement.php,

**(voir annexe 8)**

et d'affichage: affichage.php.

**(voir annexe 9)**

#### **Utilisateurs et droits**

Pour sécuriser le serveur, il ne faut pas travailler avec le compte root. J'ai décidé de créer un utilisateur webadmin appartenant au groupe apache : www-data et lui attribuer les droits nécessaires au fonctionnement sur le dossier et les fichiers de travail.

```
adduser webadmin
chown -Rf webadmin:www-data /var/www
chmod 770 /var/www
chmod 760 /var/www/*
```

### 3.6.6 Evolution

Cette solution est évolutive. Pour ajouter un switch supplémentaire, il suffit de configurer le switch sur la version 2c du protocole snmp avec la même communauté et de suivre les 2 étapes suivantes.

### **Modification Script shell**

Ajout du switch au tableau du fichier snmphtml.sh. Il ne faut modifier que la troisième ligne. Il faut ajouter le nouveau switch à la fin du tableau pour que la colonne « prises » de ma base de données ne soit pas décalée.

On ajoute par exemple « monswitch » avec son « adresse IP » séparé par « : ».

```
#!/bin/bash
tableau=("H810022:192.168.0.1" "H260037:192.168.0.2"
"H260039:192.168.0.3" "monswitch:x.x.x.x" )
```

### **Modification Script php**

Pour pouvoir accéder au switch via le lien direct, il faut modifier le tableau qui contient les ancrages\*. Il faut modifier la 24 ème ligne du fichier avec une attention particulière : Garder le même ordre que le script shell.

```
$ancre = array(1 => "H810022", 2 => "H260037", 3 =>
"H260039", 4 => "monswitch") ;
```

### **3.6.7 Avantages et Inconvénients**

Cette solution est intéressante dans la mesure où je peux désormais visualiser l'ensemble des Vlans de mon parc avec les prises de brassage.

La méthode de conception de cette solution doit être améliorée car j'ai pu remarquer plusieurs problèmes :

- les requêtes du script shell sont lentes.  
Il faut revoir ce script en minimisant les requêtes « snmpwalk » et en travaillant avec les filtres.
- Si je supprime un switch, autre que le dernier, ou que je le remplace par un autre avec un nombre de ports différents, je dois modifier toute mes entrées « prises » à partir de celui-ci.  
Le problème peu être résolu en travaillant autrement sur les tables, par exemple une table par type d'infos avec association par clés étrangères.

- Générer un fichier par switch serait peut être une meilleure solution pour traiter l'affichage et rentrer les données dans une base.

## 3.7 HP VLAN SIMPLE ADMINISTRATION

Après la réalisation de ce projet et en poursuivant mes recherches sur des améliorations possibles, j'ai trouvé une solution gratuite et open-source\* qui m'offrait des fonctionnalités supplémentaires. J'ai choisit de la mettre en place en parallèle « Hp Vlan Simple Administration », sur le même serveur.

### 3.7.1 Mise en place

#### **Installation**

Lien de téléchargement de l'application :

[http://sourceforge.net/projects/procurve-admin/?source=typ\\_redirect](http://sourceforge.net/projects/procurve-admin/?source=typ_redirect)

Il faut décompresser le fichier dans le répertoire d'apache\*.

Sur mon serveur, j'utilise le répertoire par défaut **`/var/www`**

#### **Sécurisation**

Pour sécuriser les fichiers de configuration, il faut les déplacer dans un répertoire non accessible par apache. Je les ai placés dans un répertoire à la racine par exemple :

```
Mkdir /paramswitchs
```

Les 2 fichiers à déplacer se trouvent dans le répertoire

**`/var/www/hpvlanadmin/params`**

Ce sont les fichiers : `config.conf` et `switches.xml`

```
mv /var/www/hpvlanadmin/params/config.conf  
/paramswitchs/
```

```
mv /var/www/hpvlanadmin/params/switches.xml  
/paramswitchs
```

Il faut indiquer le nouveau chemin à ces fichiers dans le fichier

```
/params/conf_path.php.
```

```
path=/paramswitchs
```

Tout le paramétrage de notre installation se situe dans ces fichiers de configuration.

## 3.7.2 Configuration

### *Config.conf*

Ce fichier est très bien documenté.

Une règle d'or, pour toujours avoir la possibilité de revenir sur une configuration d'origine il faut systématiquement faire des backups\* des fichiers que l'ont va modifier.

```
cp /paramswitchs/config.conf
/paramswitchs/config.conf.backup
```

```
# passer la valeur de ce paramètre a un pour utiliser
l'application
DISABLE_FIRST_RUN_WARNINGS=1
```

Ce paramètre est indispensable.

J'ai modifié le nom de mon répertoire : `/var/www/hp-vlan-admin`  
à la décompression par : `/var/www/hpvlanadmin`

```
APPLI_FOLDER_NAME=hpvlanadmin
```

Mon serveur est une machine debian 7 wheezy :

```
# choose between "UNIX" or "WINDOWS"
OS_TYPE=UNIX
```

Le langage d'utilisation :

```
# choose between "en" or "fr"
LANGUAGE=fr
```

Version snmp :

```
SNMP_DEFAULT_VERSION=2c
```

D'autres paramètres nous donnent la possibilité de sécuriser l'application comme on le désire. Dans un premier temps, j'ai uniquement sécurisé l'accès à l'application :

```
# Passez cette valeur a 1 pour restreindre les accès a
l'application
LOCAL_AUTHENTICATION=1
LOCAL_USER=admin
# Please change the password with something more
complicated.
LOCAL_PASSWORD=admin
```



## Switches.xml

Ce fichier nous donne la possibilité de travailler avec les 3 versions du protocole snmp. Nos switches sont configurés sur la version 2c. Il faut supprimer les lignes qui ne nous intéressent pas puis garder celles correspondant à notre version.

Exemple de configuration pour notre version :

```
<?xml version="1.0" encoding="UTF-8"?>
<switchs>
  <switch>
    <id>1</id>
    <name>R2-D2</name>
    <ip>192.168.0.2</ip>
    <snmpVersion>2</snmpVersion>
    <community>MICHU</community>
  </switch>
  <switch>
    <id>2</id>
    <name>C-3PO</name>
    <ip>192.168.0.3</ip>
    <snmpVersion>2</snmpVersion>
    <community>MICHU</community>
  </switch>
</switchs>
```

### 3.7.3 Interface graphique

Il ne reste plus qu'à se connecter à l'interface et le tour est joué. Cette solution offre la possibilité de visualiser et paramétrer notre parc de switch via interface graphique.

## Hp Vlan Simple Administration

| Menu                      |
|---------------------------|
| Accueil                   |
| Tableau de bord           |
| Voir                      |
| Sauvegrade                |
| Faire une sauvegarde ▲    |
| Parcourir les sauvegardes |
| Switchs                   |
| H260037                   |
| H260052                   |
| H260043                   |
| H260054                   |
| Vriaswitch023             |
| Vue comparative           |
| Voir                      |

HP VLAN ADMIN v3.3.1.1 Copyright 2010 - 2015, Etienne CORBET

### H260037 web ssh

Cliquer sur un port pour le passer à tag/untag/no untag...

| H260037             |                                |
|---------------------|--------------------------------|
| 1<br>DEFAULT        |                                |
| 33<br>AJILON        | 3 8 10 13 15 16 17 18 19 26 25 |
| 99<br>HEB           | 20 23 24 25                    |
| 109<br>AJIT_PROD    | 1 2 4 5 6 7 11 12 14 21 25     |
| 159<br>AJIT_PROD_IN |                                |
| 168<br>FORMATION    | 9 22 25                        |
| 222<br>TOIP_ADECCO  | 25 26                          |
| 899<br>DMZ_ADMIN    | 25                             |

(src : img hpvsa)

### 3.7.4 Avantages et inconvénients

Cette application offre des avantages complémentaires à la première solution de visualisation par ports:

- Affichage par vlan.
- Possibilité d'affichage par comparaison (2 switchs en même temps).
- paramétrage des Vlans.
- Sauvegarde de la configuration.

Cependant, étant donné que cette application permet de modifier la configuration des switchs qui, je le rappelle, sont tous en production, je prends toute les précautions d'usage. Il faut avant tout s'assurer que l'application ne modifie réellement que ce que l'on veut et applique correctement les paramètres souhaités. On peut activer et désactiver des fonctionnalités dans le fichier de paramétrage de l'application. J'ai tout de même pu remarquer qu'en désactivant toute les fonctionnalités dans ce fichier, j'ai réussi à modifier la configuration d'un port (non utilisé !!) sur le vlan par défaut.

Toutes les fonctionnalités de l'application n'ont donc pas été testées aujourd'hui. Ces tests seront effectués dès que je peux me procurer un switch libre. Pour l'instant, elle n'est utilisée qu'en visualisation.

## 3.8 EVOLUTION

Ces 2 solutions sont complémentaires et j'ai donc décidé de les mettre en place avec une première page index.php laissant le choix de la solution aux administrateurs.

Cette solution est open-source et je prévois d'étudier le code pour mixer ces 2 solutions lorsque les tests auront été validés sur HP Vlans.

# CHOISIR LA SOLUTION

## Visualisation Vlan ou Administration Vlan

LIENS DIRECTS VERS

Switch H260022   Switch H260037   Switch H260039   Switch H260043   Switch H260052   Switch H260054   Switch Ynawtsch023

valider pour relancer un scan manuellement

VISUALISATION DU SWITCH

| REF    | mercredi 13 mai 2015 00:08:13 | ProCurve J49008 Switch 2626 | H260052  | 26 ports | 10.1.53.15 |
|--------|-------------------------------|-----------------------------|----------|----------|------------|
| port 1 | notdetected                   | down                        | Untagged | 109      | AJIT_PROD  |
| port 2 | notdetected                   | down                        | Untagged | 103      | IDGROUP    |
| port 3 | notdetected                   | down                        | Untagged | 109      | AJIT_PROD  |
| port 4 | notdetected                   | down                        | Untagged | 119      | ETAM_WAN   |

### Hp Vlan Simple Administration

Menu

- Accueil
- Tableau de bord
- Vue
- Sauvegarde
- Faire une sauvegarde
- Parcourir les sauvegardes
- Switchs
- H260037
- H260052**
- H260043
- H260054
- Ynawtsch023
- Vue comparative
- Vue

**H260052** web ssh

Cliquer sur un port pour le passer à tag/untag/no untag...

| H260052      |                            |
|--------------|----------------------------|
| 1            | 26                         |
| DEFAULT_VLAN |                            |
| 33           | 25                         |
| AJILON       |                            |
| 103          | 2 5 6 7 8 9 10 11 12 17 18 |
| IDGROUP      | 19 24 13 14 15 16 25       |
| 108          | 20 22 25                   |
| REXEL        |                            |
| 109          | 1 3 13 14 15 16 25         |
| AJIT_PROD    |                            |
| 111          | 25                         |
| CONTENTIA_WA |                            |
| 113          | 25                         |

# CONCLUSION

Je suis depuis 4 ans dans une démarche de reclassement professionnel et le désir de me reconvertir dans les métiers de l'informatique à été validé à la base par un bilan de compétence.

J'ai eu l'opportunité d'être accepté dans la société Modis pour pouvoir préparer une licence contrat de professionnalisation. Cela m'a appris de nombreuses choses sur le fonctionnement d'une ESN.

Passionné par linux (Debian, ubuntu), les projets qui m'ont été confié m'ont beaucoup intéressé. En tant qu'Administrateur Système, les solutions open sources sont souvent utilisées dans le domaine de la supervision et de l'administration technique.

En constante évolution, les solutions libres ont prouvé qu'elles avaient leur place dans la sphère professionnelle.

La supervision est un moyen indispensable pour être réactif sur les équipements informatiques d'une entreprise. La maîtrise de cet outil permet à long terme de contrôler tout son parc informatique.

Shinken grâce aux plugins Nagios a permis de mettre en place une solution de monitoring à la fois pratique et efficace dans la société.

La maîtrise des protocoles snmp et lldp implémentés dans tout équipements réseaux est très intéressante pour le management de son parc.

J'ai pris conscience également lors du deuxième projet qu'il est important de maîtriser les langages de programmation (php), le scripting (shell sous linux) et de savoir utiliser une base de donnée (mysql).

Grace à ces outils, on peut créer des solutions adaptées à ses besoins.

Lors de cette seconde partie, aucun plan n'a été défini et je suis parti dans toutes les directions. Cela m'a conforté dans l'idée que tout projet doit être géré par de la gestion projet. Quels sont les besoins, quels sont les différentes solutions, pourquoi je choisi cette solution, comment la mettre en place... sont la base essentielle a toute réalisation efficace.

Ces projets m'ont permis de parfaire mes connaissances dans le monde de la supervision, de l'administration des réseaux, du développement, ainsi que la maîtrise de l'environnement Linux.

# Annexe 1

The screenshot shows the Shinken monitoring dashboard. The main heading is "Business impact: Normal". Below this, there is a table of services with their status and performance metrics.

| Service     | Status | Age   | Details  | Response Time |
|-------------|--------|-------|--|---------------|
| service-now | UP     | 6d 2h | PING OK - Packet loss = 0%, RTA = 13.26 ms                           | 13.26ms       |
| vs260031    | UP     | 6d 2h | PING OK - Packet loss = 0%, RTA = 7.20 ms                            | 7.20ms        |
| vs260031    | OK     | 6d 2h | HTTP OK: HTTP/1.1 301 Moved Permanently - 808 bytes in 0.001420s     | 0.001420s     |
| vs260031    | OK     | 6m 6s | HTTPS OK: HTTP/1.1 200 OK - 2992 bytes in 0.053 second response time | 0.052998s     |
| vs260031    | OK     | 6d 2h | PING OK - 10.1.52.80: rta 0.273ms, lost 0%                           | 0.273ms       |

Active filters: Group: base-de-co

No bookmarks  
No common bookmarks

Shinken UI, 2011-2013 | Page generated in 0.00 seconds



The screenshot shows the NagVis monitoring interface. The main heading is "24/24". Below this, there is a grid of service tiles for various companies, each with a logo and a green checkmark indicating a successful status.

| Company               | Status |
|-----------------------|--------|
| Etam                  | OK     |
| Adecco                | OK     |
| Alphatec Spine        | OK     |
| nexify                | OK     |
| contentia             | OK     |
| NOCIBÉ                | OK     |
| Cofisun international | OK     |
| nextiraOne            | OK     |
| Transpole             | OK     |
| Kiabi                 | OK     |
| Rexel                 | OK     |
| RougeGorge            | OK     |
| veryweor              | OK     |
| NOCIBÉ                | OK     |



## Annexe 2

```
#!/bin/bash
clear

tableau=("10.0.0.1:17.0.11" "10.0.0.10:17.0.11" "10.0.0.10:17.0.11" "10.0.0.10:17.0.11" "10.0.0.10:17.0.11" "10.0.0.10:17.0.11")

for valeur in "${tableau[@]}"; do
    KEY=${valeur%%:*}
    host=${valeur#*:}

    ##creation des variables arg1 et arg2 pour le choix d' affichage des commandes snmp

    community=""
    args1="-v 2c -c $community -OQv -On $host"
    args2="-v 2c -c $community -OQv -Os $host"
    []

    echo -n `snmpget $args1 entPhysicalDescr.1 2>/dev/null` " " ##infos switch .1.3.6.1.2.1.47.1.1.1.2.1
    echo -n `snmpget $args1 sysName.0 2>/dev/null` " " ##nom du switch .1.3.6.1.2.1.1.5.0
    echo -n `snmpget $args1 dot1dBaseNumPorts.0 2>/dev/null` " " ## nombre de ports .1.3.6.1.2.1.17.1.2.0
    ##récupération dans une variable du nbre de ports pour la boucle
    nbrport=`snmpget $args2 .1.3.6.1.2.1.17.1.2.0 2>/dev/null| tr -s " " | cut -d " " -f 1`

    ##recupération adresse ip ou affichage

    echo "$host"
    ### echo "$nbrport"
    ###boucle de parcours des ports du switch
    for ((i=1;i<=$nbrport;i++));
    do
        echo -n "etat du port $i. "
        echo -n `snmpget $args1 ifOperStatus.$i 2>/dev/null` " " ##si l'interface est prete a émettre et recevoir du trafic
        echo -n `snmpget $args1 ifMauAutoNegRemoteSignaling.$i.1 2>/dev/null` " " ##si l'autonégociation est possible
        tag=`snmpget -OQv -c PRILIA -v 2c 10.1.53.15 dot1qPortAcceptableFrameTypes.$i 2>/dev/null` ##infos port tagged .1.3.6.1.2.1.17.7.1.4.5.1.2
        if [ "$tag" = "2" ]; then
            echo -n "Tagged. "
        else
            echo -n "Untagged. "
        fi
        vlan=`snmpget $args2 dot1qPvid.$i 2>/dev/null`
        if [ "$vlan" = "1" ]; then ##ne pas afficher le Vlan par défaut
            echo -n "#### "
        else
            echo -n `snmpget $args2 dot1qPvid.$i 2>/dev/null` " " ##numero de vlan
        echo -n `snmpget $args2 dot1qVlanStaticName.$vlan 2>/dev/null` " " ## nom du vlan associé au numéro
        fi
    done
done
done
exit 0
```

# Annexe 3

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <meta charset="UTF-8" />
5 <link rel="stylesheet" href="styles.css" />
6 <title>Visualisation des Vlans</title>
7 <style>...</style>
8 </head>
9 <body>
10 <?php
11 $ligne = 1; // compteur de ligne
12 $fic = fopen("switchs.txt", "a+");
13 while($stab=fgetcsv($fic,1024,',, '))
14 {
15 $champs = count($stab); //nombre de champ dans la ligne en question
16 $ligne ++;
17 //affichage de chaque champ de la ligne en question
18 $compt=0 ; //initialisation d'un compteur pour gerer les lignes
19 echo "<tx>" ;
20 $i = 0 ;
21 if ( $stab[$i] == "REF" ) {
22 echo '<center><table bgcolor="#E6E6FA" border="1">';
23 echo '<caption ><h1 >VISUALISATION DU SWITCH</h1></caption><tx/>';
24 for($i=0; $i<$champs; $i ++)
25 {
26 $compt++;
27 if ( $compt == 7 ) {
28 echo "<th>". $stab[$i] . "</th>". " </tx>";
29 $compt=1;
30 }
31 else {
32 echo "<th>". $stab[$i] . "</th>";
33 }
34 }
35 echo '</table></center>';
36 }
37 else {
38 echo '<center><table bgcolor="#E6E6FA" border="1">';
39 for($i=0; $i<$champs; $i ++)
40 {
41 $compt++;
42 if ( $compt == 7 ) {
43 echo " ". $compt;
44 echo "<td >". $stab[$i] . "</td>". " </tx>";
45 $compt=1;
46 }
47 else {
48 switch ( $compt ) {
49 case 1:
50 echo '<th align="center" width=60px >'. $stab[$i] . "</th>";
51 break;
52 case 2:
53 echo '<td align="center" width=80px>'. $stab[$i] . "</td>";
54 break;
55 case 3:
56 echo '<td align="center" width=40px>'. $stab[$i] . "</td>";
57 break;
58 case 4:
59 echo '<td align="center" width=80px>'. $stab[$i] . "</td>";
60 break;
61 case 5:
62 echo '<td align="center" width=40px>'. $stab[$i] . "</td>";
63 break;
64 case 6:
65 echo '<td align="center" width=100px>'. $stab[$i] . "</td>";
66 break;
67 }
68 }
69 }
70 echo '</table></center>';
71 }
72 }
73 <?>
74 </body>
75 </html>
```



# Annexe 4

```
1 <html lang="fr">
2 <head>
3 <meta charset="UTF-8" />
4 <link rel="stylesheet" href="styles.css" />
5 <title>Scan des Vians</title>
6 <style>.</style>
7 </head>
8 <body>
9 <?php
10     if ( file_exists('switchsbis.txt') ) {
11         echo '<center><p><h1> SCAN ACTIF </h1></p></center><br/>' ;
12         echo '<center><form name="form_associatif" id="form_associatif" method="post" action="scanswitchs.php">' ;
13
14         echo '<br/><center><p>Le Scan est actif, vous pouvez retourner sur la page de visualisation</p></center>' ;
15         echo '<center><p>soyez patient</p></center>' ;
16         echo '<td><a href="switchs.php" > <input type="button" value="Retour"> </a></td>' ;
17         echo '</form></center>' ;
18
19         //affichage du temps restant
20         $contenu_fichier1 = file_get_contents('switchsbis.txt');
21         $contenu_fichier2 = file_get_contents('switchs.txt');
22         $vert = substr_count($contenu_fichier1, "\n") ;
23         $rouge = substr_count($contenu_fichier2, "\n") ;
24         $rouge2 = (substr_count($contenu_fichier2, "\n") - substr_count($contenu_fichier1, "\n")) ;
25         echo "<br/><center><p>.$vert." sur ".$rouge."</p></center>";
26         echo '<center></center>' ;
27     }
28     else
29     {
30         if(isset($_POST['valider']) || ! file_exists('switchsbis.txt')){
31             //monfichier = fopen('switchsbis.txt', 'w+');
32             exec('/var/www/html.sh 1> /var/www/switchsbis.txt 2>/dev/null &');
33             //fclose($monfichier);
34             echo '<center><p><h1> LANCEMENT DU SCAN </h1></p></center><br/>' ;
35             echo '<center><form name="form_associatif" id="form_associatif" method="post" action="scanswitchs.php">' ;
36             echo '<br/><center><p>Le Scan a bien été lancé, vous pouvez retourner sur la page de visualisation</p></center>' ;
37             echo '<td><a href="switchs.php" > <input type="button" value="Retour"> </a></td>' ;
38             echo '</form></center>' ;
39         }
40         else {
41             echo '<center><p><h1> ATTENTION CE SCAN DURERA PLUS DE 10 MINUTES ET EFFACERA LES ANCIENNES DONNEES </h1></p></center><br/>' ;
42             echo '<center><form name="form_associatif" id="form_associatif" method="post" action="scanswitchs.php">' ;
43             echo '<br/>' ;
44             echo '<td>êtes vous sûr ? </td>' ;
45             echo '<td><input type="submit" value="Oui" name="valider" >' . ' ' .</td>' ;
46             echo '<td><a href="switchs.php" > <input type="button" value="Retour"> </a></td>' ;
47             echo '</td>' ;
48             echo '</form></center>' ;
49         }
50     }
51 </body>
52 </html>
```

## Annexe 5

```
1
2 <?php
3 // creating the database
4 $user = "root";
5 $password = "root";
6 $database_name = "vlans";
7 $mysql = new PDO("mysql:host=localhost", $user, $password);
8 $statement = $mysql->prepare("CREATE DATABASE IF NOT EXISTS $database_name");
9 $statement->execute();
10 $dbconn = new PDO("mysql:host=localhost;dbname=$database_name", $user, $password);
11
12 //setting connection parameters
13
14 $hostname = "localhost";
15 $dsn = 'mysql:dbname=' . $database_name . ';host=' . $hostname;
16
17
18
19 //initialize the connection
20 $conn = new PDO($dsn, $user, $password);
21
22 //creating the table if it not exists
23 $sqlCreate = "CREATE TABLE IF NOT EXISTS `switchs` (
24     `id` INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
25     `col1` VARCHAR(255) NOT NULL,
26     `col2` VARCHAR(255) NOT NULL,
27     `col3` VARCHAR(255) NOT NULL,
28     `col4` VARCHAR(255) NOT NULL,
29     `col5` VARCHAR(255) NOT NULL,
30     `col6` VARCHAR(255) NOT NULL,
31     `col7` VARCHAR(255) NOT NULL,
32     PRIMARY KEY (`id`));";
33
34 $conn->query($sqlCreate);
35 ?>
```

## Annexe 6

```
1 <?php
2 require "connexion.php";
3 $sth = $dbconn->prepare("SELECT col7 FROM switches");//recuperation de la colonne7
4 $sth->execute();
5 $result = $sth->fetchAll(PDO::FETCH_COLUMN, 0);
6 //destruction de toute la table
7 $sql=$dbconn->prepare("DROP TABLE switches ");
8 if($sql->execute()){
9 //echo " Table deleted ";
10 }else{
11 print_r($sql->errorInfo());
12 }
13 // mon fichier texte dans un tableau $tab
14 $row = 1;
15 $tab= array();
16 if (($handle = fopen("switches.txt", "r")) !== FALSE) {
17 while (($data = fgetcsv($handle, 1000, ",")) !== FALSE) {
18     $num = count($data);
19     $row++;
20     for ($c=0; $c < $num; $c++) {
21         $tab[$row - 2][$c] = $data[$c] ;
22     }
23 }
24 fclose($handle);
25 }
26 /*echo "controle visuel du tableau </br>";
27 $long = count ($tab) ;
28 $num = 6;
29 for ( $i=0 ; $i < $long ; $i++){
30     for ( $j=0 ; $j < $num ; $j++){
31         echo $tab [$i] [$j] ;
32     }
33     echo "\n\n";
34     echo "</br>";
35 } */
36 ?>
37
38 <?php
39 include "connexion.php";
40 $long = count ($tab) ;
41 for ( $i=0 ; $i < $long ; $i++){
42     $sql = "INSERT INTO switches (col1, col2, col3, col4, col5, col6)
43     VALUES ('.$tab[$i][0].', '.$tab[$i][1].', '.$tab[$i][2].', '.$tab[$i][3].', '.$tab[$i][4].', '.$tab[$i][5].')";
44     $req = $dbconn->exec ($sql);
45 }
46 $long = count($result);
47 for ($i=0; $i<$long; $i++){
48     $sql = 'INSERT INTO col7 VALUES $result[$i]'; //restitution de la colonne 7
49     $req = $dbconn->exec($sql);
50 }
51 flush();
52 ?>
```

# Annexe 7

```
1 <?php
2 $switchSelected = isset($_GET["s"]) ? $_GET["s"] : "";
3 $portSelected = isset($_GET["p"]) ? $_GET["p"] : "";
4 ?>
5 <!DOCTYPE html>
6 <html lang="fr">
7 <head>
8 <meta charset="UTF-8" />
9 <link rel="stylesheet" href="styles.css" />
10 <title>Visualisation des Vlan</title>
11 </head>
12 <body>
13 <?php
14 include "connexion.php";
15 // Si tout va bien, on peut continuer
16 // construction du formulaire de selection switch port et affectation prises
17 ?>
18 <form method="post" action="traitement.php">
19 <center><table bordre="0">
20 <tr>
21 <td>
22 switch
23 <select name="switch" >
24 <?php
25 foreach(array("H810022", "H260037", "H260039", "H260043", "H260052", "H260054", "vnaSwitch023") as $switch) {
26 echo "<option value='\$switch' ". ($switchSelected == $switch ? "selected":"" ). "> \$switch</option>\n";
27 }
28 ?>
29 </select>
30 </td>
31 <td>
32 port
33 <select name="port">
34 <?php for ($i=1; $i<50; $i++){
35 echo '<option value="'. $i. "' '. ($portSelected == $i ? "selected":"" ). ">'. $i. '</option>' ;
36 }?>
37 </select>
38 </td>
39 <td>
40 prise: <input id="inputPrise" type="text" name="prise" />
41 </tr>
42 <tr>
43 <td></td><td></td><td></td>
44 <td>
45 <INPUT TYPE="submit" NAME="envoyer" VALUE=" Envoyer ">
46 </td>
47 </tr>
48 </table></center>
49 </form>
50 <script type="text/javascript">
51 document.getElementById('inputPrise').focus();
52 </script>
53 </body>
54 </html>
```

## Annexe 8

```
1 <?php
2 header('Location: affichage.php');
3 include "connexion.php";
4 $switchSelected=$_POST['switch'] ;
5 //var_dump ($switch);
6 $portSelected=$_POST['port'];
7 $prise=$_POST['prise'];
8 //pour controle
9 //echo $switch ;
10 //echo $port;
11 //echo $prise;
12 //recuperation des valeurs de toute les lignes avec le marquage REF dans la colonne 2 (en clair le nom du switch et son numero id)
13 $sql = "SELECT id, col4 FROM switchs WHERE col1='REF'";
14 $req = $dbconn->query($sql);
15 $stab = $req->fetchAll();
16 //controle d'enregistrement
17 //var_dump ($stab);
18 /*comparaison et affectation d'une variable pour positionner l'entrée d'affectation
19 $long = count($stab);
20 for ($i=0; $i<$long; $i++){
21     if ($switchSelected == $stab[$i][1]){
22         //var_dump($stab[$i][1]);
23         $indics = $stab[$i][0];
24     }
25     else{
26     }
27 } */
28 //echo $indics;
29 //echo $port."</br>";
30 // $portSelected=$portSelected+$indics;
31 echo $prise."</br>";
32 echo $portSelected."</br>";
33 // Si tout va bien, on peut continuer
34 $sql = "UPDATE switchs
35     SET col7=?
36     WHERE id=?";
37 $q = $dbconn->prepare($sql);
38 $q->execute(array($prise,$portSelected)) ;
39 //insertion du nom 'PRISE' dans colonne 7 quand je vois REF colonne 1
40 $sql = "UPDATE switchs
41     SET col7=?
42     WHERE col1=?";
43 $q = $dbconn->prepare($sql);
44 $q->execute(array('PRISE', 'REF'));
45 ?>
46
```

# Annexe 9

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4 <meta charset="UTF-8" />
5 <link rel="stylesheet" href="styles.css" />
6 <link rel="stylesheet" href="//maxcdn.bootstrapcdn.com/font-awesome/4.3.0/css/font-awesome.min.css">
7 <title>Visualisation des prises</title>
8 <style type="text/css">
9     a.editBouton, a.editBouton:visited {
10         color: #333;
11     }
12     a.editBouton:hover {
13         color: #aa0;
14     }
15 </style>
16 </head>
17 <body>
18 <?php
19 include "connexion.php";
20 echo <center><table bgcolor="#E6E6FA" border="0"><tr>
21 echo <td><a href="switchs.php">Retour</a></td>
22 echo </tr></table></center>;
23 // Si tout va bien, on peut continuer
24 // On récupère tout le contenu de la table switchs
25 $reponse = $dbconn->query('SELECT * FROM switchs');
26 $nomDuSwitch = "";
27 $numPort = "";
28 echo <center><div><table>; // pour ouvrir la première balise qui est fermée à la première itération
29 while ($donnees = $reponse->fetch())
30 {
31     if ($donnees['col1'] == 'REF'){
32         echo </table></div><div class="tableaux"><table bgcolor="#E6E6FA" border="0">;
33         $nomDuSwitch = $donnees['col4'];
34     } else {
35         $portArray = explode(" ", $donnees['col1']);
36         $numPort = $portArray[1];
37     }
38     echo <tr><td>.$donnees['col1']. </td>;
39 //echo $donnees['col2'];
40 //echo $donnees['col3'];
41     if ($donnees['col4'] == 'Untagged' || $donnees['col4'] == 'Tagged' || $donnees['col4'] == 'PORT LIBRE'){ //on affiche que le nom du switch
42         echo <td>-----</td>; //ne rien afficher
43     }
44     else{
45         echo <td>.$donnees['col4'].</td>;
46     }
47 //echo $donnees['col5'];
48 //echo $donnees['col6'];
49     echo <td>.$donnees['col7'].</td><td><a class='editBouton' href='insertion.php?s=$nomDuSwitch&p=$numPort'><i class='fa fa-pencil'></i></a></td></td>;
50 }
51 echo </table></div></center>'
52 <?>
53 </body>
54 </html>
55
```

# GLOSSAIRE

## A

**Ancrage :** Les ancres sont des **positions** dans une page HTML, qui pourront être utilisées comme **références**. Les ancres sont **invisibles** sur la celui qui a créé cette page sait où elles sont placées. Elles permettent de faire défiler un long texte et de se rendre à la partie sans passer par la glissière de défilement.

**API :** une interface de programmation (souvent désignée par le terme API pour *Application Programming Interface*) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels.

## B

**Bandeau de brassage :** Dans un réseau informatique, un **panneau (bandeau) de brassage** (anglais : *patch panel*) est un support d'interconnexions qui se place en général dans une baie de brassage. On y connecte des cordons de brassage afin de relier les différents périphériques entre eux.

**Base de connaissances :** Une base de connaissance regroupe des connaissances spécifiques à un domaine spécialisé donné, sous une forme exploitable par un ordinateur.

**Boot :** Le **boot** (ou bootstrap), l'**amorce**, ou encore l'**amorçage**, est la procédure de démarrage d'un ordinateur, qui comporte notamment le chargement du programme initial.

**Brassage :** un panneau de brassage (anglais : patch panel) est un support d'interconnexions qui se place en général dans une baie de brassage.

**Broker :** Il est en charge de récupérer les informations d'états des ordonnanceurs et de les traiter. Ces traitements sont fait par des modules. De nombreux modules existent : un pour l'export en base merlin (MySQL), un autre pour l'export des données de métrologie

dans le fichier service-perfdata, une présentation des données de style Livestatus, un export en base Oracle ou encore un export en base CouchDB.

## C

**Cartographie** : Ensemble des opérations ayant pour objet l'élaboration, la rédaction et l'édition de cartes.

**Certificat** : Le certificat est en quelque sorte la carte d'identité de la clé publique, délivré par un organisme appelé *autorité de certification* (souvent notée **CA** pour *Certification Authority*).

**Check** : un Check ou **test** désigne une procédure de vérification partielle d'un système.

**Cœur de réseau** : Il constitue le centre névralgique d'un réseau à haut débit. Littéralement épine dorsale. Dans le contexte des réseaux de télécommunications désigne la partie qui supporte le gros du trafic.

**.CSV** : ***Comma-separated values***, connu sous le sigle **CSV**, est un format informatique ouvert représentant des données tabulaires sous forme de valeurs séparées par des virgules.

## D

**Datacenter** : Un **centre de données** (*data center* en anglais) est un site physique sur lequel se trouvent regroupés des équipements constituant du système d'information de l'entreprise.

**Debian** : Debian est une distribution GNU/Linux non commerciale, lancée en 1993 par Ian Murdock avec le soutien de la Free Software Foundation ; elle a pour principal but de fournir un système d'exploitation composé uniquement de logiciels libres.

**Debug** : C'est un argument de ligne de commande déclenchant un mode de fonctionnement réclamant au programme beaucoup de messages relatifs à son propre fonctionnement : il dit à mesure aussi exactement que possible tout ce qu'il fait, de façon que son débogage soit plus facile. Ce paramètre est aussi utilisé



couramment par les administrateurs pour la configuration et la mise au point de nouveaux systèmes.

## E

**Excel :** **Microsoft Excel** est un logiciel tableur de la suite bureautique Microsoft Office, développée et distribuée par l'éditeur Microsoft. La version la plus récente est Excel 2013.

**ESX Vmware :** Vmware est une société informatique américaine fondée en 1998, filiale d'EMC Corporation depuis 2004, qui propose plusieurs produits propriétaires liés à la virtualisation d'architectures x86. C'est aussi par extension le nom d'une gamme de logiciels de virtualisation.

## F

**Flasher :** Pour effacer le contenu d'une mémoire morte de type EEPROM (Electrically-Erasable Programmable Read-Only Memory - Mémoire flash), il faut lui appliquer un courant électrique, "un flash". C'est pour cette raison que l'on emploie le terme "flasher" pour dire "effacer" une mémoire de type EEPROM.

## G

**GBPS :** "**GigaBits Par Seconde.**" 1Gbps est égale à 1,000 mégabits par seconde ( Mbps ), ou milliard de bits par seconde. Gbps est couramment utilisé pour mesurer la vitesse de transfert de données entre matériels périphériques.

**Greffons :** Greffon est un terme français préconisé pour remplacer *plugin*, qui vient de l'anglais.

## H

**HP :** **Hewlett-Packard Company**, officiellement abrégée en **HP**, est une entreprise multinationale américaine initialement d'électronique et d'instrumentation qui a évolué vers l'informatique, les imprimantes, les Serveurs & Réseaux et le multimédia.

**HTTPS :** L'**HyperText Transfer Protocol Secure**, plus connu sous l'abréviation **HTTPS** — littéralement « protocole de transfert

hypertexte sécurisé » — est la combinaison du HTTP avec une couche de chiffrement comme SSL ou TLS.

**I**  
**IEEE :** L'*Institute of Electrical and Electronics Engineers* ou **IEEE**<sup>note 1</sup>, en français l'« **Institut des ingénieurs électriciens et électroniciens** », est une association professionnelle. L'IEEE compte plus de 400 000 membres et possède différentes branches dans plusieurs parties du monde<sup>1</sup>. L'IEEE est constituée d'ingénieurs électriciens, d'informaticiens, de professionnels du domaine des télécommunications, etc. L'organisation a pour but de promouvoir la connaissance dans le domaine de l'ingénierie électrique (électricité et électronique). Juridiquement, l'IEEE est une organisation à but non lucratif de droit américain.

**IOS :** (*Internetworking Operating System*), est un système d'exploitation implémenté dans les équipements réseaux.

**IP :** *Internet Protocol*, le protocole de communication fondamental de la suite des protocoles internet. Il nécessite l'attribution d'**adresses IP**.

**J**  
**K**  
**L**

**LAMP :** Lamp est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites web.

**Linux :** **Linux** est le nom couramment donné à tout système d'exploitation libre fonctionnant avec le noyau Linux.

**Livestatus :** Depuis la version 1.1.0rc1 check\_mk embarque un nouvel élément qui mérite à lui seul le détour. Cet élément se nomme **livestatus** et vient combler à lui seul les derniers manques de Nagios, pouvoir enfin **dialoguer avec nagios sans middleware type ndo ou merlin**.

**Log :** Le concept d'**historique des événements** ou de **logging** désigne l'enregistrement séquentiel dans un fichier ou une base de

données de tous les événements affectant un processus particulier (application, activité d'un réseau informatique...).

## M

**MAU :** (**Medium Access Unit**). Cette MIB contient des liens et infos sur les ports et les appareils connectés aux ports.

**Métrique :** Le protocole snmp permet d'exposer **diverses métriques** telles que les compteurs liés aux interfaces réseau.

**MIB :** Une **MIB** (*Management Information Base*, base d'information pour la gestion du réseau) est un ensemble d'informations structuré sur une entité réseau, par exemple un routeur, un commutateur ou un serveur.

**Module :** Le terme module en programmation identifie une structure de programmation.

**Multiplateforme :** Un logiciel **multiplate-forme** ou **multiplateforme** est un logiciel conçu pour fonctionner sur plusieurs plates-formes, c'est-à-dire le couple liant ordinateur et système d'exploitation.

## N

**Nagios :** Nagios est une application permettant la surveillance système et réseau

**Nagvis :** NagVis est un addon de visualisation pour Nagios qui permet de générer des vues métier de la supervision. Facile à installer, à utiliser, très intuitif avec Nagios.

## O

**OID :** Les OID sont utilisés dans le monde LDAP mais aussi dans d'autres domaines, comme le protocole SNMP pour identifier des ressources. Il est possible d'obtenir un OID, et par conséquent toute une branche, auprès de l'IANA.

**Open-source :** La désignation **open source**, ou « **code source ouvert** », s'applique aux logiciels dont la licence respecte des critères précisément établis par l'*Open Source Initiative*, c'est-à-dire les

possibilités de libre redistribution, d'accès au code source et de création de travaux dérivés.

## P

**Plugins :** En informatique, un *plugin* ou *plug-in*, aussi nommé **module d'extension**, **module externe**, **greffon**, **plugiciel**, ainsi que *add-in* ou *add-on* en France, est un paquet qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités.

**Port :** un **port matériel** est une prise permettant de brancher des périphériques sur une machine.

**Processus :** gestion de la production, ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie.

**Protocole :** Un protocole est une méthode standard qui permet la communication entre des processus (s'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau.

**Putty :** **PuTTY** est un émulateur de terminal doublé d'un client pour les protocoles SSH, Telnet, rlogin, et TCP brut.

**Python :** Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet.

## Q

## R

**Radius :** **RADIUS** (*Remote Authentication Dial-In User Service*) est un protocole client-serveur permettant de centraliser des données d'authentification. Le protocole RADIUS a été inventé et développé en 1991 par la société Livingston, qui fabriquait des serveurs d'accès au réseau pour des matériels uniquement équipés d'interfaces série ; il a fait ultérieurement l'objet d'une normalisation par l'IETF.

- Reporting** : La **communication de données** (en anglais *reporting*) est l'opération consistant, pour une entreprise, à faire rapport de son activité.
- Reseau** : Ensemble des moyens matériels et logiciels mis en œuvre pour assurer les communications entre ordinateurs, stations de travail et terminaux informatiques.
- RFC** : Les **requests for comments (RFC)**, littéralement « demande de commentaires », sont une série numérotée de documents officiels décrivant les aspects techniques d'Internet, ou de différents matériels informatiques (routeurs, serveur DHCP).
- RMON** : Le standard **RMON (Remote network Monitoring)** est fondé sur l'utilisation du protocole de gestion SNMP et met en jeu deux composants, un gestionnaire SNMP et des agents SNMP prenant RMON en charge. L'association des deux composants est équivalente à un système d'analyseurs réseaux distribués.
- S**
- Script** : Un **langage de script** est un langage de programmation qui permet de manipuler les fonctionnalités d'un système informatique configuré pour fournir à l'interpréteur de ce langage un environnement et une interface qui déterminent les possibilités de celui-ci.
- Serveur** : Un **serveur informatique** est un dispositif informatique matériel ou logiciel qui offre des services, à différents clients.
- Shinken** : **Shinken** est une application permettant la surveillance système et réseau.
- SMTP** : **Simple Mail Transfer Protocol (SMTP)**, littéralement « protocole simple de transfert de courrier ») est un protocole de communication utilisé pour transférer le courrier électronique (courriel) vers les serveurs de messagerie électronique.

**SMON :** Le Réseau **R**emote **MON**veillance (**RMON**) MIB a été développé par l' IETF pour appuyer l'analyse de protocole de contrôle et des réseaux locaux . La version originale (parfois dénommé RMON1) axé sur la couche OSI 1 et de niveau 2 information dans les réseaux Ethernet et Token Ring. Il a été prolongé par RMON2 qui ajoute le support pour Network- et la couche application et la surveillance par SMON qui ajoute le support pour les réseaux . Il est une spécification standard de l'industrie qui fournit la plupart des fonctionnalités offertes par les analyseurs de réseaux s. Agents RMON sont intégrés dans de nombreux commutateurs haut de gamme et les routeurs.

**SNMP :** **Simple Network Management Protocol** (abrégé **SNMP**), en français « protocole simple de gestion de réseau », est un protocole de communication qui permet aux administrateurs réseau de gérer les équipements du réseau, de superviser et de diagnostiquer des problèmes réseaux et matériels à distance.

**SNMPB :** SnmpB est un navigateur SNMP MIB écrit en QT. Il prend en charge SNMPv1, SNMPv2c et SNMPv3.

**Sonde :** Une sonde est un exécutable, invoqué régulièrement par le daemon Nagios. Le protocole de communication entre les deux parties est simple et robuste. En entrée, cette sonde récupère ses paramètres de deux façons différentes :

- les paramètres définis par l'utilisateur sont passés explicitement
- un certain nombre de paramètres contextuels sont passés sous forme de variables d'environnement.

**SSH :** **Secure Shell (SSH)** est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion.

**SSL :** **Secure Sockets Layer**, un protocole de sécurisation des échanges sur Internet, devenu **Transport Layer Security (TLS)** en 2001.

**Superviser** : voir supervision.

**Supervision** : La **supervision** est une technique industrielle de suivi et de pilotage informatique de procédés de fabrication automatisés.

**Support** : les **services d'assistance**, ou support aux utilisateurs, consistent à garantir que les utilisateurs d'un système puissent continuer à profiter de la disponibilité de l'ensemble de ses composants pour l'accomplissement de leurs tâches.

**Switch** : Un **switch** désigne un commutateur réseau, équipement ou appareil qui permet l'interconnexion d'appareils communicants, terminaux, ordinateurs, serveurs, périphériques reliés à un même réseau physique

## T

**Tag** : balise accompagnant un flux de données sur un réseau informatique afin d'en préciser l'adressage (VLAN), le degré d'urgence ou de priorité (QoS) ou la nature.

**Telnet** : **Telnet** (**TE**rminal **NE**Twork ou **TELE**communication **NE**Twork, ou encore **TELE**type **NE**Twork) est un protocole réseau utilisé sur tout réseau prenant en charge le protocole TCP/IP.

**TFTP** : **TFTP** (pour **Trivial File Transfer Protocol** ou **Protocole simplifié de transfert de fichiers**) est un protocole simplifié de transfert de fichiers.

**Ticketing** : Centre d'assistance, système de gestion de requêtes créant un ticket pour chacune, assigné à une seule personne à la fois jusqu'à sa résolution et son archivage.

**Topologie** : Une **topologie de réseau** est en informatique une définition de l'architecture (physique ou logique) d'un réseau, définissant les connexions entre ces postes et une hiérarchie éventuelle entre

eux ; elle peut définir la façon dont les équipements sont interconnectés ou la représentation, spatial du réseau (topologie physique).

U

V

**Verbeux :** Indique le fait qu'un programme ou une commande affiche explicitement toutes les opérations qu'il effectue.

**Vlans :** Un **réseau local virtuel**, communément appelé **VLAN** (pour *Virtual LAN*), est un réseau informatique logique indépendant. De nombreux VLAN peuvent coexister sur un même commutateur réseau.

W

**WEB :** Le **World Wide Web (WWW)**, littéralement la « toile (d'araignée) mondiale », communément appelé le **Web**, et parfois la **Toile**, est un système hypertexte public fonctionnant sur Internet.

**Webui :** Interface web pour Shinken.

**Wheezy :** Wheezy est le nom de code de développement de Debian 7.0.

X

Y

Z