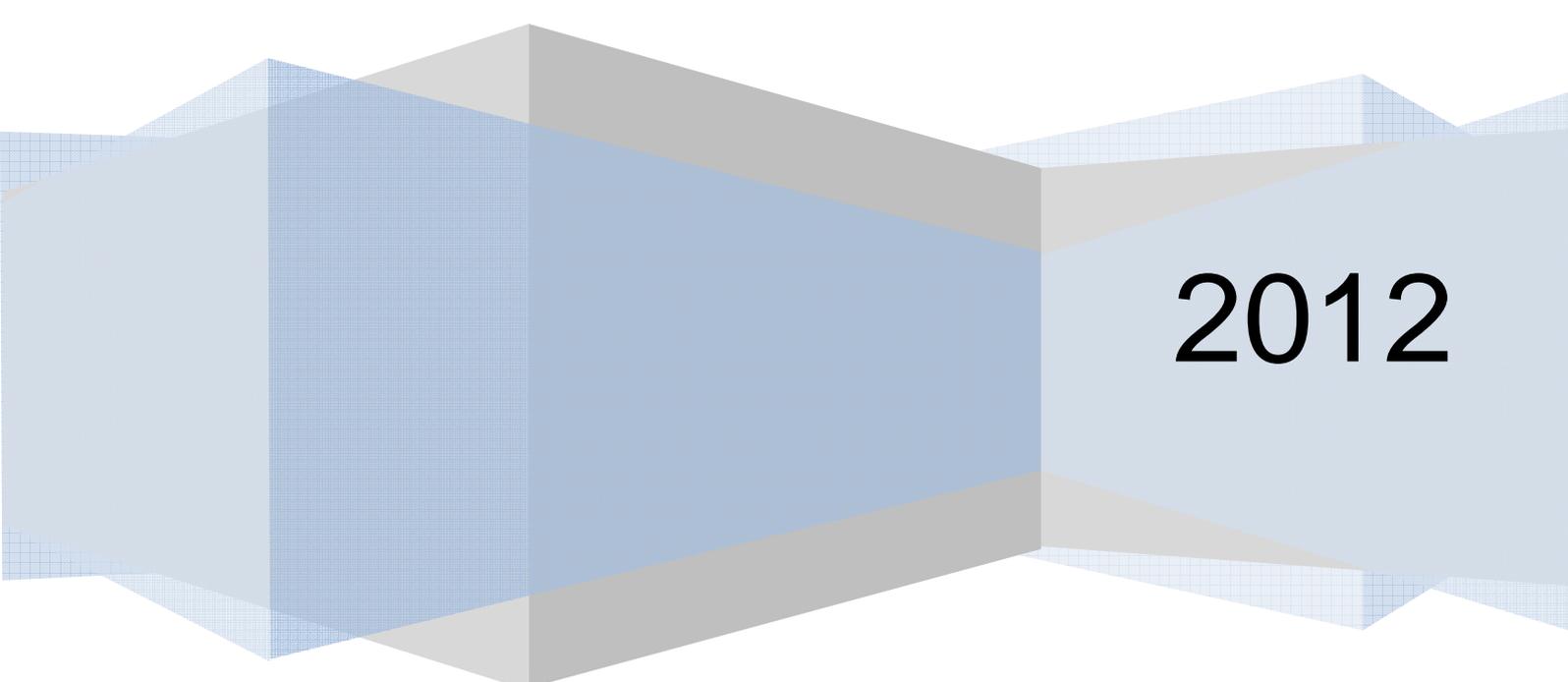


© 2012 – Les tutos à toto

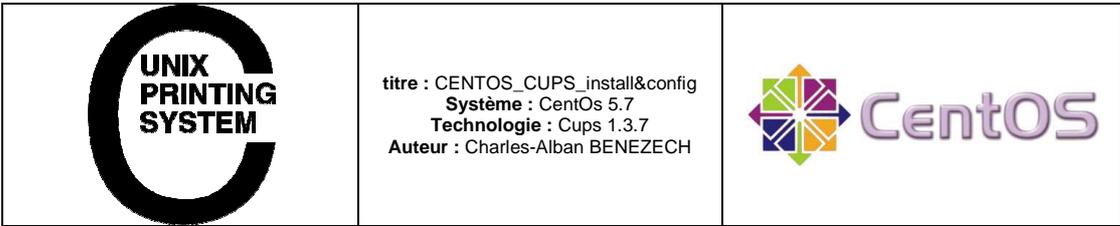
CUPS server - install and configure

Réalisée sur CentOS 5.7

Ecrit par Charles-Alban BENEZECH



2012



Sommaire

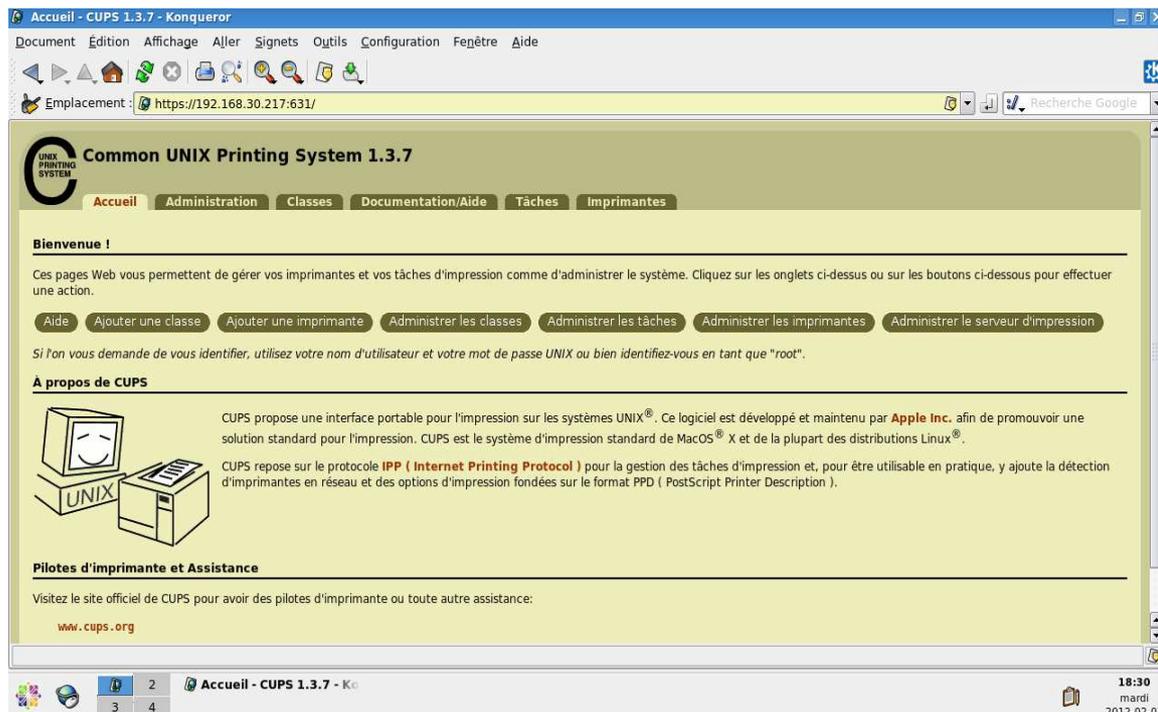
- 1. **CUPS** **3**
 - 1.1 Introduction 3
 - 1.2 Fonctionnement 4
 - 1.3 Sécurisation 4
 - 1.3.1 Algorithme 4
 - 1.3.2 cryptosystème asymétrique 5
- 2. **Les dépendances** **6**
 - 2.1 Téléchargement 6
 - 2.2 Installation 6
- 3. **installation** **7**
 - 3.1 Certification 7
 - 3.2 Mise en place 8
- 4. **configuration** **9**
 - 4.1 CUPS **Erreur ! Signet non défini.**

1. CUPS

1.1 Introduction

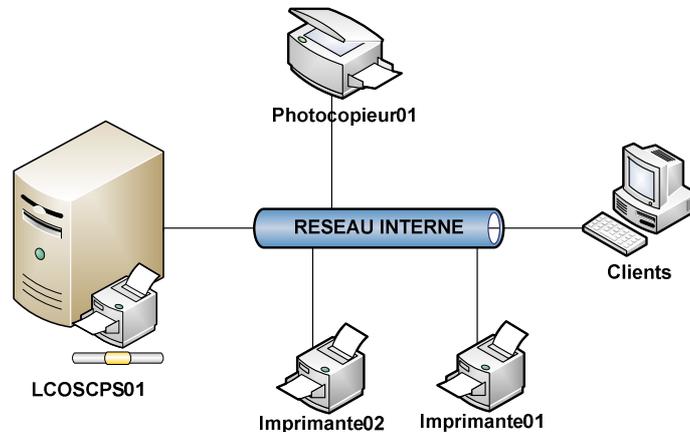
Common Unix Printing System (CUPS) est un logiciel serveur permettant d'administrer les imprimantes développées par Apple. Un serveur d'impression a pour mission d'assurer la gestion des imprimantes, fax et scanners d'un ou plusieurs réseaux interconnectés.

L'avantage de CUPS est qu'il met à disposition des administrateurs une interface web permettant de l'administrer à distance.



1.2 Fonctionnement

Un serveur d'impression aura pour rôle de fournir un panel de fonction permettant d'administrer les périphériques d'impression et de gérer l'impression. Lorsqu'un client voudra accéder à une imprimante en réseau il enverra une requête au serveur d'impression qui la traitera.



Une requête envoyée à un serveur d'impression contient plusieurs informations, l'émetteur, qui désigne l'utilisateur qui effectua la requête, la priorité, qui désigne un niveau défini dans le serveur, plus la requête aura un niveau de priorité bas plus elle sera traitée rapidement, le contenu, qui désigne le fichier qui doit être traité, et la date d'émission de la requête.

Un serveur d'impression doit être le plus transparent possible pour l'utilisateur; ainsi CUPS offrant des fonctions de partage d'imprimante réseau via le protocole IPP est très intéressant car permet une interopérabilité à toute épreuve, les protocoles web étant compris par tous les systèmes actuels.

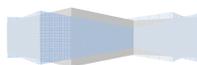
1.3 Sécurisation

Afin de nous permettre d'assurer la sécurité des transactions entre l'agent web et le serveur d'impression nous allons créer un tunnel SSL entre l'agent web et le serveur d'impression. Cette technique consiste à échanger des clés entre le client et le serveur. Ces clés permettront au client de chiffrer et de déchiffrer les données échangées.

Celui que nous allons utiliser est l'algorithme de chiffrement RSA, qui est un chiffrement à clé publique et de type asymétrique.

1.3.1 Algorithme

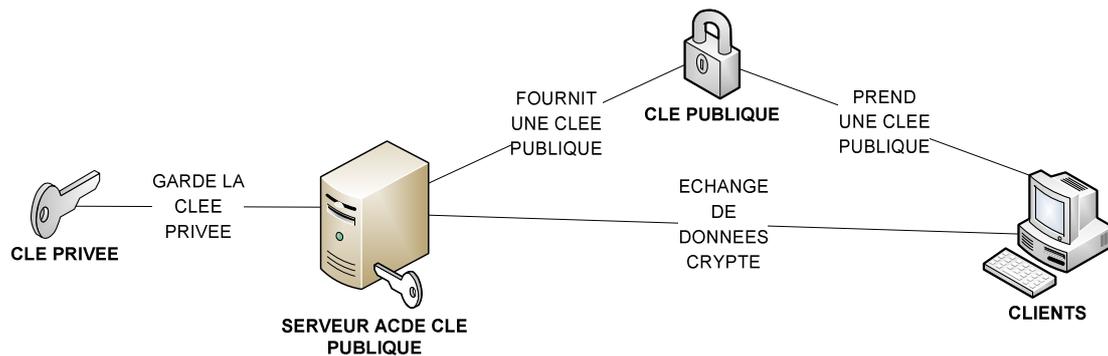
Un algorithme de chiffrement est une méthode permettant de transformer un message en clair (non chiffré) en chiffré. Par exemple l'algorithme de chiffrement RSA qui est basé sur la difficulté à factoriser de grand entier (longueur minimum 80): cet algorithme donnera ainsi la clé privée grâce à une suite d'opération arithmétique.



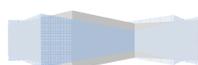
1.3.2 cryptosystème asymétrique

Dans système de cryptage asymétrique (à clé publique). Dans ce type de système, deux clés seront générées: une clé privée pour le chiffrement et une clé privée pour le déchiffrement.

Ce type de système fonctionne de la manière suivante: un utilisateur (souvent le serveur qui remplit le rôle d'autorité de contrôle) générera une clé privée qu'il gardera et met à disposition des autres utilisateurs un certain nombre de clés publiques. Lorsque qu'un utilisateur veut échanger des données avec cet utilisateur (le générateur de clé) il prend une clé publique et se sert de cette clé afin de chiffrer ces données avant de les envoyer à cet utilisateur qui utilisera la clé privée dont il dispose afin de les décrypter.



Ce mode de fonctionnement permet d'éviter de faire transiter la clé privée qui permet de déchiffrer les données; cependant l'inconvénient consiste à être sûr que la clé publique utilisée par l'utilisateur pour crypter les données a bien été fournie par le bon serveur.



2. Les dépendances

Avant d'installer CUPS nous allons devoir installer les dépendances dont il a besoin pour fonctionner.

2.1 Téléchargement

Afin de pouvoir fonctionner correctement nous allons installer le dépôt rpmforge car il contient la plupart des derniers paquets disponibles, nous installerons ensuite openssl.

```
$ su -
  Password: *****
# mkdir /root/SETUP
# cd /root/SETUP
# wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el5.rf.i386.rpm
# rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
# wget http://ftp.gnu.org/pub/gnu/gettext/gettext-0.18.1.1.tar.gz
# rpm -K rpmforge-release-0.5.2-2.el5.rf.*.rpm
# rpm -i rpmforge-release-0.5.2-2.el5.rf.*.rpm
# yum upgrade
# yum update
# yum install openssl
```

2.2 Installation

Nous allons maintenant pouvoir installer gettext.

```
# cd /root/SETUP
# tar -xvzf gettext-0.18.1.1.tar.gz
# cd /root/SETUP/gettext-0.18.1.1
# ./configure && make && make install
```



3. installation

Nous allons maintenant pouvoir installer notre serveur d'impression CUPS. Voici la configuration du serveur CUPS.

```
nom: LCOSCP01
domain: tutoatoto.loc
IP: 192.168.13.22 /24
```

3.1 Certification

Nous allons utiliser des certificats SSL afin de sécuriser nos communications grâce à la commande openssl.

Cette tâche consistera dans un premier temps à créer une clé privée qui nous permettra ensuite de générer un certificat auto-signé. Une fois la clé privée et le certificat générés nous les validerons (signature) nous pourrions passer à la configuration d'apache.

```
# cd /etc/pki/tls/certs

# make server.key
umask 77 ; \
/usr/bin/openssl genrsa -des3 1024 > server.key
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 61251 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:

# openssl rsa -in server.key -out server.key
Enter pass phrase for server.key:
writing RSA key

# make server.csr
umask 77 ; \
/usr/bin/openssl req -utf8 -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]: FR
State or Province Name (full name) [Berkshire]: FRANCE
Locality Name (eg, city) [Newbury]: Perpignan
Organization Name (eg, company) [My Company Ltd]: tutoAtoto
Organizational Unit Name (eg, section) []: AdminSys
Common Name (eg, your server's hostname) []: LCOSCP01.tutoatoto.loc
Email Address []: OVampO@tutoatoto.fr
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

```
# openssl x509 -in server.csr -out server.crt -req -signkey server.key -days 3650
Signature ok
subject=/C=JP/ST=Hiroshima/L=Hiroshima/O=Server Linux/OU=IT Solution/CN=www.server-
linux.info/emailAddress=root@server-linux.info Getting Private key

# chmod 400 server.*
```

3.2 Mise en place

Maintenant que nous avons mis en place l'autorité de contrôle nous allons installer les composant de CUPS. Pour cela nous allons utiliser les packages disponibles dans les dépôts.

```
# yum install cups
```



4. configuration

Maintenant que nous avons installé tous les composants nécessaires au bon fonctionnement de notre serveur nous allons pouvoir passer à sa configuration.

Afin de faire fonctionner notre serveur CUPS nous allons modifier le fichier cupsd.conf afin de rendre l'interface web accessible à tout le monde suivant les permissions.

```
# vi /etc/cups/cupsd.conf
Listen 631
<Location />
    Order allow,deny
    Allow localhost
    Allow from all
</Location>
<Location /admin>
    Encryption Required
    Order allow,deny
    Allow localhost
    Allow from all
</Location>
<Location /admin/conf>
    AuthType Basic
    Require user @SYSTEM
    Order allow,deny
    Allow from all
</Location>
ServerCertificate /etc/pki/tls/certs/server.crt
ServerKey /etc/pki/tls/certs/server.key

# vi /etc/cups/mime.convs
application/octet-stream    application/vnd.cups-raw    0    -
```

ATTENTION: ne modifiez que les lignes suivantes de manière à ce qu'elles ressemblent à celles explicitement mentionnées!

