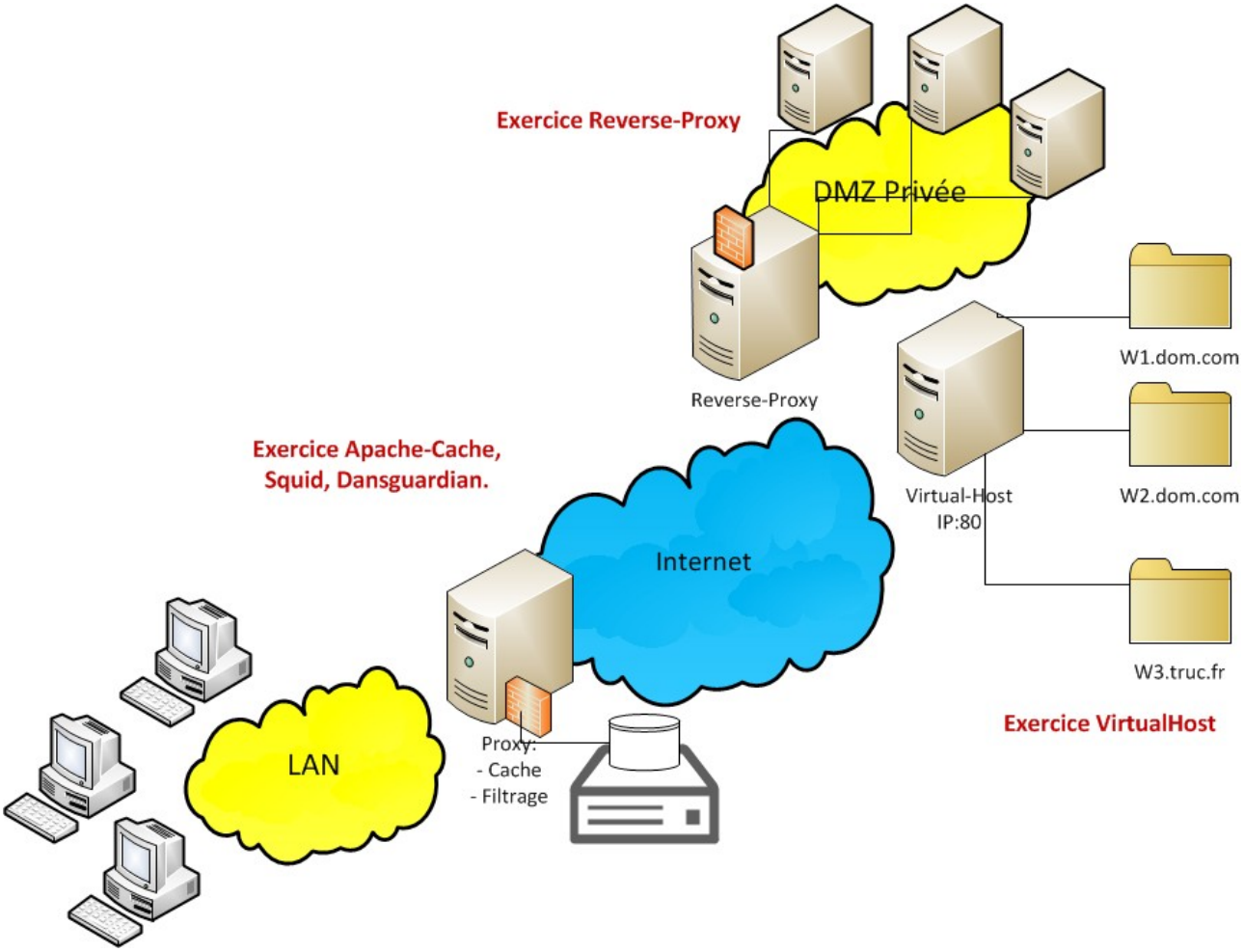
Installation du serveur apache



Téléchargement:
Vérifier que le paquet build-essential est installé (compilateur, make)
Vérifier la version installée sur votre distribution pour installer les sources correspondantes et ainsi bénéficier des bonnes versions des paquets dépendants.
Exemple:
cd /usr/local/src
apt-get source --download-only apache2
apt-get build-dep apache2
bzip2 -d apache2_2.4.18.orig.tar.bz2

tar -xvf apache2_2.4.18.orig.tar

La compilation:

Pour prendre en compte les options nécessaires pour faire fonctionner apache en mode proxy il faut le recompiler et choisir les bonnes options (./configure --help)

Exemple :
./configure --prefix=/usr/local/apache --enable-module=so --enable-proxy --enable-proxy-connect --enable-proxy-http --enable-module=rewrite --enable-cache --enable-cache-disk

Compiler:
make

Installer:
make install

Exécuter le démon Apache:
/usr/local/apache/bin/apachectl start
erreur il faut charger le module suivant :
LoadModule slotmem_shm_module modules/mod_slotmem_shm.so

Configuration basique de httpd.conf:

```
ServerRoot "/usr/local/apache"

Listen 80
User daemon
Group daemon
ServerAdmin you@u-picardie.fr
DocumentRoot "/usr/local/apache/htdocs"

ServerName Nomduserveur

...
```

Configuration d'un virtualHost :

La configuration de serveurs virtuels nécessite la configuration du DNS. Il *faut* que leurs noms soient définis dans le DNS, et qu'ils y soient résolus sur l'adresse IP du serveur en écoute. Il est possible d'ajouter des entrées dans le fichier hosts dans le cadre du TP si vous n'avez pas accès au DNS renseigné dans la configuration des machines (mais bien entendu cela ne fonctionnera que sur la machine possédant ces entrées (machine utilisant le navigateur).

Fonctionnement « VirtualHost »:

Si la requête contient un en-tête « Host» inconnu, ou si celui-ci est absent, elle sera toujours servie par le serveur virtuel primaire par nom (**celui correspondant à ce couple adresse/port trouvé en premier dans le fichier de configuration**).

Exemple :

```
# On écoute sur les ports 80 et 8080


Listen 80
Listen 8080

# Par defaut on route vers le répertoire htdocs (autre que les VirtualHost)
DocumentRoot /usr/local/apache/htdocs

# Pour les accès sur 192.168.34.136 sur le port 80 on passe à travers les déclarations VirtualHost (monsite1, monsite2). Pour un ServerName différent des déclarations www.monsite1.fr  est par défaut.
```

```
NameVirtualHost 192.168.34.136:80

<VirtualHost 192.168.34.136:80>
ServerName www.monsite1.fr
DocumentRoot /usr/local/apache/monsite1
</VirtualHost>

<VirtualHost 192.168.34.136:80>
 ServerName www.monsite2.fr
 DocumentRoot /usr/local/apache/monsite2
</VirtualHost>
```

C’est la variable d’entête « Host » du navigateur qui déterminera le « VirtualHost » conformément à l’Url renseignée par l’utilisateur et comme nous l 'avons vu lors du TD. Pour l'exercice ne pas oublier de mettre des droits pour rendre accessible les répertoires des différents virtualhost

ex :<Directory "/usr/local/apache/monsite2">

Options Indexes FollowSymLinks

Require all granted

</Directory>

Installation Squid via la distribution:

Apt-get install squid

Squid.conf :
Ex : Configurer en écoute sur votre IP port 3128

Une acl est une étiquette qui va en fonction des directives « squid » représenter une adresse IP, un réseau, un domaine, une plage horaire…
La directive http_access va permettre d'autoriser ou d'interdire cette acl (utilisation allow ou deny). Il est nécessaire de construire le cache avant d'exécuter l'application.

Pour autoriser l'utilisation du proxy par un réseau de machines :

```
acl monreseau src 10.1.24.0/255.255.255.0
http_access allow monreseau
```

Exemple pour une liste noire :

```
acl sites_ok url_regex "/usr/local/squid/var/sites_non_ok"
http_access deny sites_non_ok
```

Pour les tests positionner son navigateur avec les options de proxy.

Pour lancer squid avec des traces :

/usr/sbin/squid3 -N -d 1

les log dans /var/log/squid3/access.log

Installation de DansGuardian via le paquet de la distribution

```
# apt-get install dansguardian
```

L'installation de « DansGuardian pour ubuntu entraîne l'installation de l'antivirus « ***Clamav*** *».*

Configuration de DansGuardian dans ce mode d'installation :

DansGuardian utilise plusieurs fichiers de configuration situés dans « **/etc/dansguardian** ».

Configuration de « /etc/dansguardian/dansguardian.conf »

Pour activer la configuration, il faut commenter ou supprimer cette ligne :

```
#UNCONFIGURED
```

La ligne suivante permet d'indiquer le port utilisé par le proxy squid :

```
proxyport = 3128
```

Cette ligne permet d'avoir les messages en français en cas de blocage des sites :

```
language = 'french'
```

Cette ligne permet de désactiver l'antivirus :

```
virusscan = off
```

* Positionner votre navigateur avec les options de proxy sur dansguardian IP port 8888

Installation Dansguardian via les sources:

Télécharger le paquet en concordance avec votre distribution (utilisation par exemple des commandes apt-cache pkgnames et apt-cache showpkg)
wget http://dansguardian.org/downloads/2/Stable/dansguardian-X.tar.gz
tar -zxvf dansguardian-Version.tar.gz
cd dansguardian-Version
./configure --prefix=/usr/local/dansguardian

Si Erreur libpcre !!!

Recherche du paquet par un apt-file search (installer apt-file si nécessaire)

apt-file search libpcre
apt-get install libpcre3-dev

./configure --prefix=/usr/local/dansguardian
make
 make install
cd /usr/local/dansguardian/

Ajouter au fichier dansguardian.conf les directives:

daemonuser = 'root'
daemongroup = 'root'

« root » juste pour nos TPs bien entendu ! sinon utiliser nobody et donner les droits pour la création et l'écriture des logs.

Chaîner le squid et dansguardian :

Configurer l'Ip, le port d'écoute local et également le « forward » vers le serveur proxy squid

Exemple :
#Dansguardian
filterip = 10.1.24.220
filterport = 8080
#Squid (a modifier en conséquence)
proxyip = 127.0.0.1
proxyport = 3128

Lancer par /usr/local/dansguardian/bin/dansguardian

Attention : la configuration par défaut est très restrictive (voir naughtynesslimit)!

Configuration de filtrage de dansguardian

Dans « /etc/dansguardian/dansguardian.conf »

La ligne suivante permet de paramétrer la force du filtrage :

naughtynesslimit = 150

La valeur par défaut de « 50 » donne un filtrage très dur et peu de sites sont accessibles. Plus cette valeur est élevée et plus le filtrage est faible.

Cette ligne permet de désactiver l'antivirus :

virusscan = off

De nombreuses directives et fichiers permettent de régler le filtrage de dansguardian (interdiction, autorisation entêtes http, types mime, urls, mots clés... Sur une distribution « debian » ces fichiers se trouvent dans le répertoire /etc/dansguardian/lists. Il faut également activer la directive dans le fichier de configuration « dansguardian.conf ».

Quelques exemples :

Configuration du fichier « bannedextensionlist »

Bannedextensionlist = chemin/bannedextensionlist dans dansguardian.conf

Ce fichier permet de définir des types de fichiers qui seront interdits de télécharger. Par défaut, la liste est très restrictive. il faut donc commenter les types de fichiers que vous souhaitez autoriser.

Configuration du fichier « bannedmimetypelist »

Là encore il s'agit de restreindre des types de fichiers en fonction du type mime.

application/gzip
application/x-gzip
application/zip

Configuration du fichier « bannedregexpurllist »

L'ajout de cette ligne à la fin du fichier permet de bloquer la plupart des sites français de rencontres en ligne :

(webcam|tchat|t'chat|rencontre|meetic|amour)

Configuration du fichier weighted_*

Il est possible d'ajouter des mots ou des listes de mots dans ces fichiers, avec des coefficients positifs (mauvais mot) ou négatifs (bons mots) :

< tchat ><20>
< t'chat ><20>

Utilisation des blacklists de Toulouse :

Vous trouverez sur ce site, des listes noires régulièrement actualisées:
Les blacklists sont disponibles individuellement

http://dsi.ut-capitole.fr/blacklists/

Téléchargement également par rsync :

rsync -arpogvt rsync://ftp.ut-capitole.fr/blacklist

Exemple pour la blacklist adult.tar.gz

mkdir /etc/dansguardian/blacklists
pour la blacklist « Adult » par exemple :

wget ftp://ftp.univ-tlse1.fr/pub/reseau/cache/squidguard_contrib/adult.tar.gz

Ajouter les chemins d'accès aux blacklists que vous souhaitez utiliser dans **/etc/dansguardian/bannedsitelist**

.Include /etc/dansguardian/blacklists/blacklists/adult/domains ...

.Include /etc/dansguardian/blacklists/blacklists/warez/domains

Pour prendre en compte les modifications, redémarrer DansGuardian :

/etc/init.d/dansguardian restart

Documentation :

http://doc.ubuntu-fr.org/tutoriel/installation_configuration

sarg pour éditer des stats squid

Télécharger le paquet source :

wget http://prdownloads.sourceforge.net/sarg/*sarg-2.2.5.tar.gz*?download

Déplier par un tar zxvf *sarg-2.2.5.tar.gz*

tar zxvf

Dans le répertoire des sources sarg :

```
./configure --prefix=/usr/local/sarg
 make
 make install
```

l'installation à placé le programme sarg dans /usr/bin/sarg
-rwxr-xr-x 1 root root 777098 2009-02-04 14:11 /usr/bin/sarg

Créer un alias pour apache (ex : connexion dans httpd.conf) :

mkdir /usr/local/httpd/htdocs/connexion

dans httpd.conf

```
  Alias /connexion/ "/usr/local/httpd/htdocs/connexion/"
    <Directory "/usr/local/httpd/htdocs/connexion">
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
```

Dans /usr/local/sarg

Créer le fichier de conf de sarg :

Sarg.conf :
max_elapsed 28800000
language French
access_log /var/log/squid/access.log En function de votre conf
output_dir /usr/local/httpd/htdocs/connexion En function de votre conf
title " Rapport des logs du proxy squid "

On peut maintenant lancer la commande « sarg » qui va générer un rapport dans /connexion/

Reste à programmer l'analyse via la crontab :

```
50 3 * * *   root /usr/local/bin/sarg >/dev/null 2>&1
```

Awstats pour éditer des stats apache

Télécharger le paquet source :

wget http://prdownloads.sourceforge.net/awstats/awstats-6.9.tar.gz

Vous pouvez aussi utiliser les paquets de la distribution.

Dans le cadre de ce TP vous utilisez la dernière version 7.X :

Pour les versions Ubuntu (18.04, 20.04…)

https://installati.one/ubuntu/20.04/awstats/

Déplier par un tar zxvf

tar zxvf awstats-6.9.tar.gz

Créer le script /usr/local/bin/analyse.sh :

cd /usr/local/httpd/conf

/usr/local/src/awstats-6.9/wwwroot/cgi-bin/awstats.pl -config=monsite.fr –update

/usr/local/src/awstats-6.9/tools/awstats_buildstaticpages.pl -config=monsite.fr -update=fr -dir=/usr/local/httpd/htdocs/analyse/ -awstatsprog=/usr/local/src/awstats-6.9/wwwroot/cgi-bin/awstats.pl –output

mv /usr/local/httpd/htdocs/analyse/awstats.monsite.fr.html /usr/local/httpd/htdocs/analyse/index.html

Créer le fichier ***awstats.monsite.fr.conf*** dans ***/usr/local/etc/awstats (chemin connu de l'outil)***

Contenu du fichier :

/usr/local/etc/awstats/awstats.monsite.fr.conf

Vérifier et modifier les variables suivantes si nécessaire:

LogFile="/usr/local/httpd/logs/access_log"
LogType=W
LogFormat=1 # Si apache
LogSeparator=" "
SiteDomain="www.monsite.fr"
DNSLookup=1
DirData="/usr/local/httpd/htdocs/analyse"
Lang="fr"

Créer un répertoire pour accueillir le résultat de l'analyse au format texte et rapport en html :

mkdir /usr/local/httpd/htdocs/analyse

Déclencher l'analyse tous les jours à travers la crontab :

cat /etc/crontab :

50 2 * * * root /usr/local/bin/analyse.sh >/dev/null 2>&1

Vérifier conf apache pour la génération des logs (mode combined):

Pour les logs generals et pour les VirtualHost:

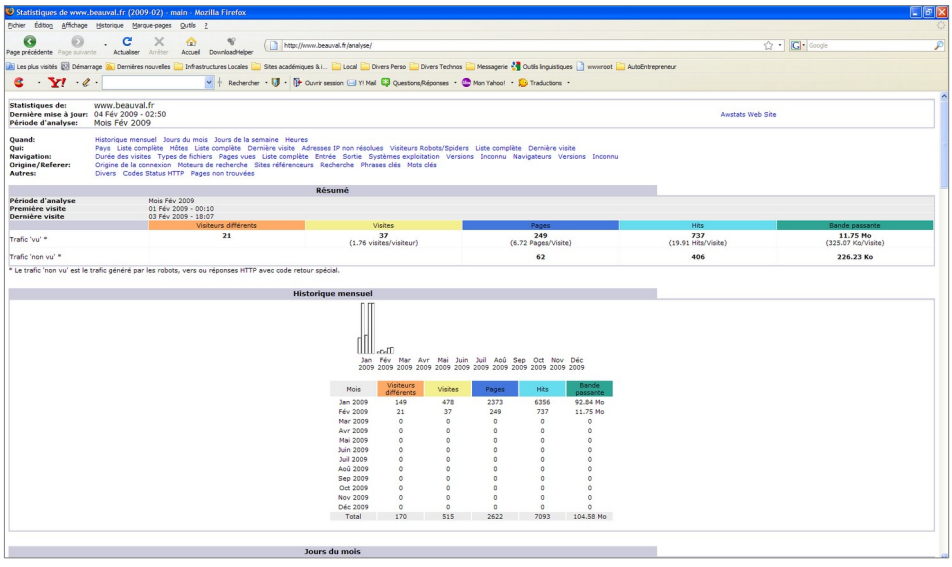
<VirtualHost 192.168.1.240>
ServerName www.monsite.fr
DocumentRoot "/usr/local/httpd/monsite/"
ServerAdmin reseauetab@ac-amiens.fr
ErrorLog /usr/local/httpd/logs/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
CustomLog logs/access_log combined
TransferLog /usr/local/httpd/logs/access_log
</VirtualHost>

Modifier le fichier de conf Apache httpd.conf pour créer un alias vers /analyse/ :

Alias /analyse/ "/usr/local/httpd/htdocs/analyse/"
    <Directory "/usr/local/httpd/htdocs/analyse">
        Options Indexes FollowSymlinks MultiViews
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>

Pour verifier le fonctionnement lancer directement la commande:

/usr/local/bin/analyse.sh et se connecter sur le site http://www.domaine/analyse/ pour avoir :



Travail à réaliser et à rendre via Moodle

1. Apache en Reverse Proxy

Vous pouvez utiliser nginx

Configuration en mode reverse proxy:

C'est le serveur Apache ou nginx qui va faire la requête vers un autre site (local ou distant) et transmettre la ressource comme si elle était locale.

Voir le schéma joint.

Avant de commencer:

- Définir un reverse-proxy

- Avantages du reverse-proxy

Faire tourner 3 virtual-hosts sur votre Apache ou nginx (ou sur 3 serveurs autonomes) . Le premier va servir de frontal reverse proxy (IP:80)et tourne sur l'Ip publique. Les 2 autres tournent sur 127.0.0.1:81 et 127.0.0.1:82 (ou une IP privée). Le frontal va rediriger les requêtes de façon transparente de IP:80/site1 vers 127.0.0.1:81 et IP:80/site2 vers 127.0.0.1:82.

2. Installation d'un module Apache à partir des sources

Un nouveau projet dans votre entreprise nécessite le développement d'un site apache en python. On vous demande d'installer le module WSGI (python) sous Apache à partir des sources. Vous installez un site virtuel /test pour faire vos tests. Vous développez une petite page Web en python.

Finalement le chef de projet opte pour le micro framework "flask". Vous l'installez et vous testez le fonctionnement.