

# Sécurisation de services



Apache



PostFix

ProFTPd



MySQL



# Plan



## 1. Points communs

1. Généralités
2. Emprisonnement
3. SSL / TLS
4. SSH

## 2. Apache

## 3. MySQL

## 4. PostFix

## 5. ProFTPd

# Sécurité



**On ne juge la solidité d'une chaîne  
que par la résistance de son maillon le plus faible.**

*BL*

# Points communs : Généralités

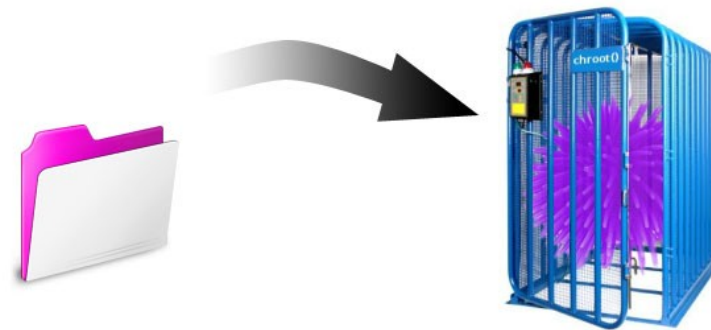
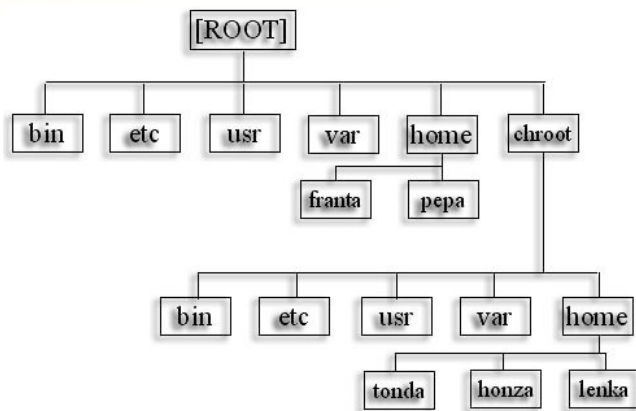


- Antivirus
- Firewall
- Éviter les configurations par défaut et courantes
- Limiter la divulgation d'information
- Conserver uniquement les fonctionnalités nécessaires
- Activer et analyser les fichiers log
- Maintenir à jour vos services
- Sécuriser le système d'exploitation

# Points communs : Emprisonnement



- Machines virtuelles
  - Isolation
  - Vulnérabilité
- ChRooting
  - La geôle



# Points communs : ChRooting



- Création de l'arborescence
- Droits utilisateurs
- Déplacement du service
- Déplacement des dépendances  
« `ldd` », « `strings` », « `truss` »

```
localhost# ldd /usr/local/apache/bin/httpd
```

```
/usr/local/apache/bin/httpd:
```

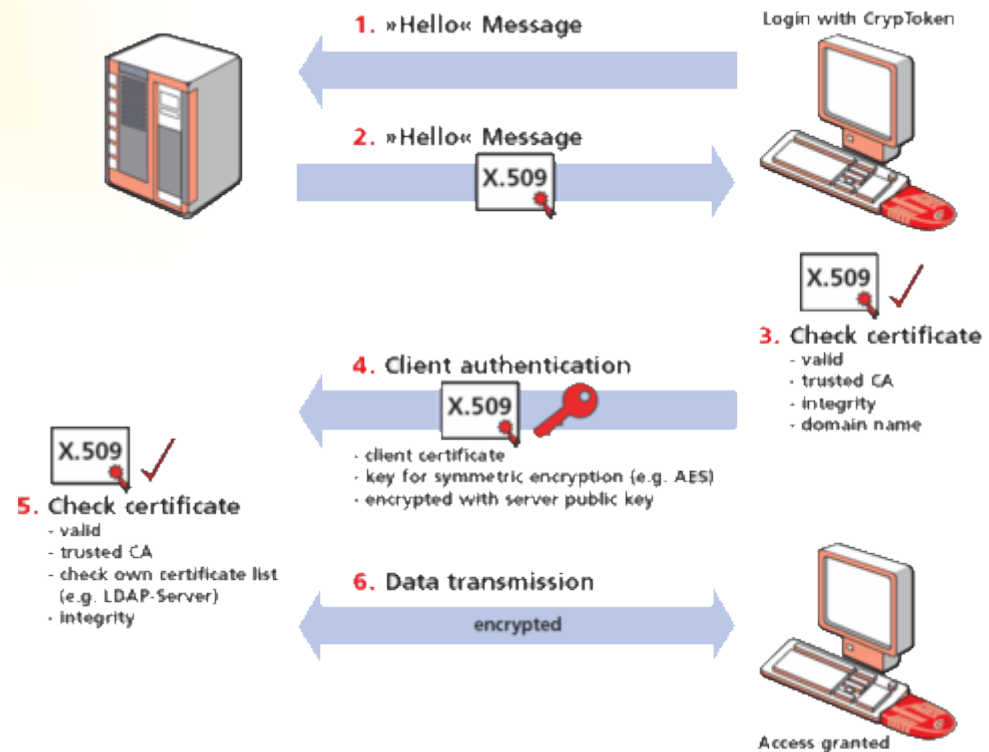
```
libcrypt.so.2 => /usr/lib/libcrypt.so.2 (0x280bd000)
```

```
libc.so.4 => /usr/lib/libc.so.4 (0x280d6000)
```

# Points communs : SSL / TLS



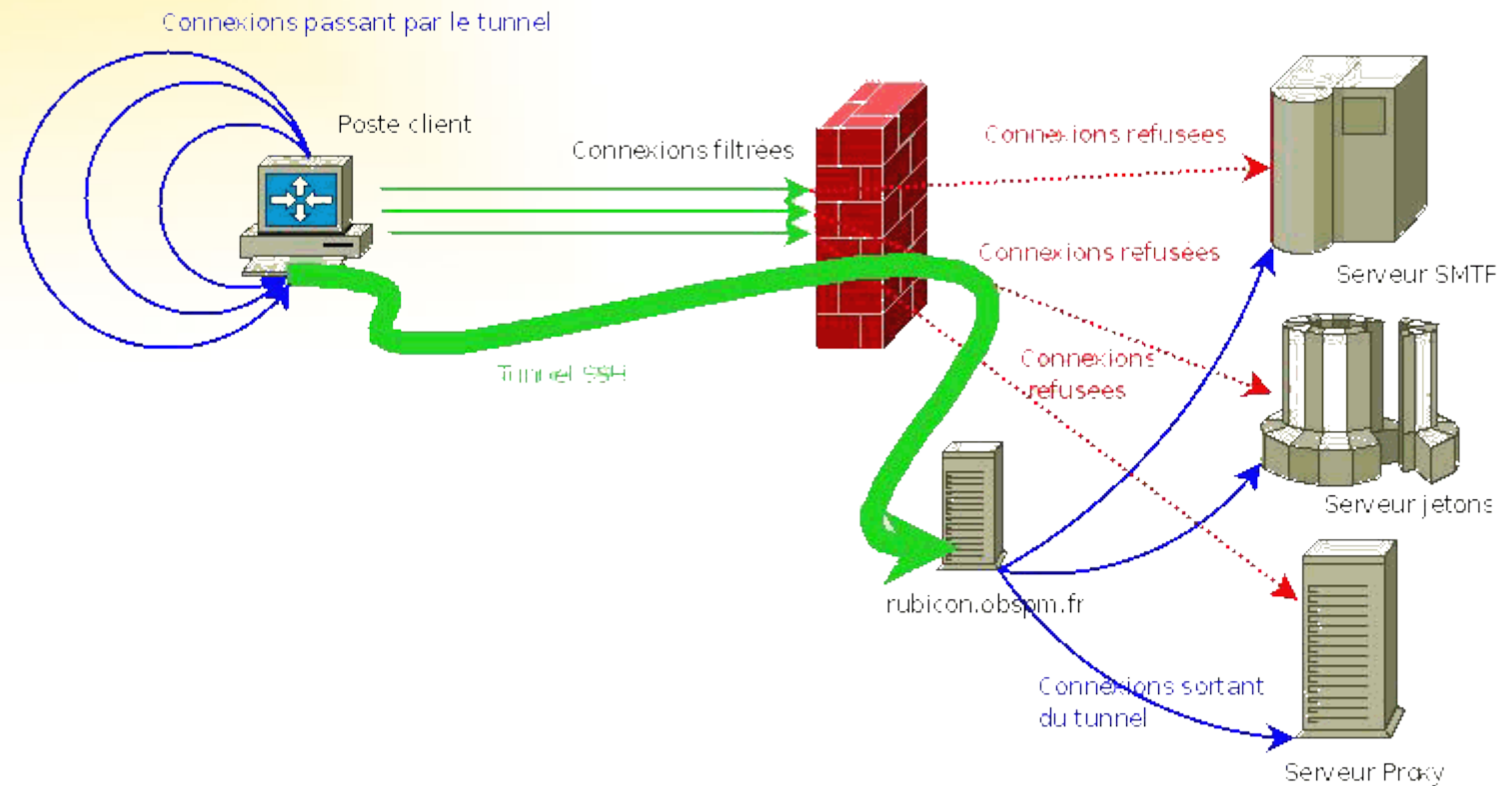
- TLS anciennement SSL
- Couche session
- Objectifs
  - Authentification
  - Confidentialité
  - Intégrité
- Certificats



# Points communs : SSH (Secure SHell)



- Protocole de communication sécurisé
- Tunnel



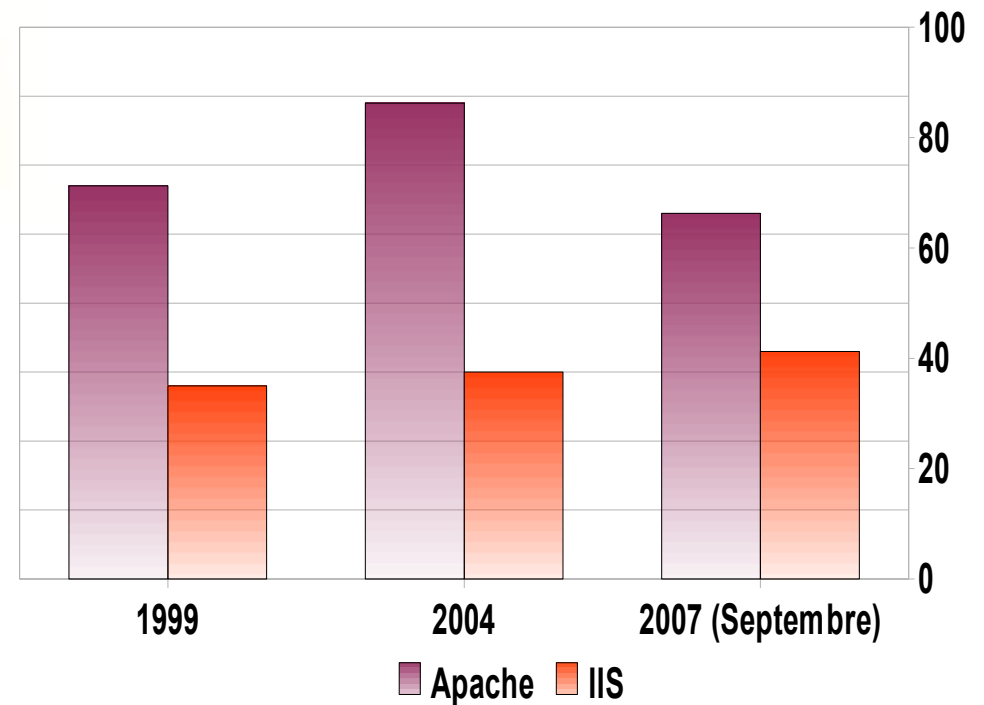


# Apache HTTP Server

# Apache HTTP Server



- C'est un logiciel de serveur HTTP **libre** qui apparut en 1995 et Open Source
- C'est le serveur le plus populaire
- Points forts :
  - Prend en charge de nombreux modules
  - Sa stabilité et ses performances
  - Son système de configuration
  - Gestion des hôtes virtuels



# Apache HTTP Server



- Hypothèses de sécurité
  - Résister aux attaques **locales** et **distantes**
  - Utiliser ce seul service et le seul service réseau HTTP si possible
  - **Désactiver** les diagnostics et listages automatiques
  - Exécuter le service sous un **unique** UID/GID
- Fichier de configuration
  - « `/apache/conf/httpd.conf` »

# Apache HTTP Server



- « **ChRouter** » le serveur
  - Recréer la structure de l'arborescence du service  
(avec les droits 0755)
  - Avoir un accès limité aux **fichiers systèmes**
  - **Ôter** tous les shell de l'environnement ChRooté
  - Lister les **dépendances** à l'aide de la commande  
`ldd /usr/local/apache/bin/httpd`
  - Changer le « **DocumentRoot** » en fonction

# Apache HTTP Server



- Denial of Service

- «MaxClients» et «MaxKeepAliveRequests»

Exemple pour un petit serveur :

```
MaxClients 150
```

```
KeepAlive On
```

```
MaxKeepAliveRequests 100
```

```
KeepAliveTimeout 5
```

- Virtual Hosting

```
<VirtualHost 194.57.201.103>
```

```
    ServerName www.esiea.fr
```

```
    ...
```

```
</VirtualHost>
```

# Apache HTTP Server



- Gérer son historique d'évènement
  - Par le module «**mod\_log\_config**»

- › Définir son propre format :

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\" combined
```

- › Choisir sa destination :

```
CustomLog /var/log/httpd/access_log common
```

- Gérer les droits
  - Par le module «**mod\_access**», on manipule principalement
    - › Directory : un répertoire
    - › Files : un fichier
    - › Location : une arborescence

# Apache HTTP Server



- Options et AllowOverride
  - Suivi des liens symboliques  
`SymLinksIfOwnerMatch`
  - Execution de scripts CGI  
`ExecCGI`
  - Génération de pages d'index  
`Indexes`
  - Serveur Side Includes  
`Includes`  
`IncludesNOEXEC`

# Apache HTTP Server



- Protection par mot de passe
  - «`mod_auth`» protège l'accès à un répertoire

```
<Directory /home/*/public_html>
  AllowOverride AuthConfig
  Options SymLinksIfOwnerMatch
</Directory>
```
  - Fichiers «`.htaccess`»

```
AuthName "Accès restreint"
AuthType Basic
AuthUserFile "/home/toto/.htpasswd"
Require valid-user
```
  - Fichiers «`.htpasswd`»

On peut créer ce fichier à l'aide de la commande «`passwd`»

```
nom_utilisateur:mot_de_passe_crypté
```

# Apache HTTP Server



- Autres Modules
  - Module «`mod_perl`» permet de meilleures performances qu'un script CGI mais présente des risques identiques
  - Fichiers «`mod_php`»
    - «`/etc/php.ini`» : Fichier de configuration
    - «`safe_mode`» : Contrôle la surveillance des fichiers accédés et l'interdiction des commandes à risques
    - «`max_execution_time`» et «`memory_limit`» : Pour limiter les risques de DoS
    - «`magic_quotes_gpc`» : Pour ajouter des guillemets aux données reçues des GET/POST et cookies
    - «`display_errors`» : Retirer l'affichage des lignes d'erreurs



# MySQL

# MySQL



- Serveur libre de bases de données relationnelles SQL
- Fonctionne sur les plate-formes  
AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetWare, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000, XP, Vista...
- Accessible en utilisant les langages  
C++, C#, Delphi / Kylix, Eiffel, Java, Perl, PHP, Python, Ruby, Tcl...

# MySQL : mysql\_secure\_installation



- Définition d'un **mot de passe** pour le compte « **root** »
- Destruction de la base **test**
- Suppression du **compte anonyme**
- Permet également de désactiver **l'accès « root » à distance**

# MySQL : Préparation



- Définir le mot de passe « root »  

```
user@machine$ mysqladmin -u root password 'mon_password' -p
```
- Supprimer les comptes sans mot de passe  

```
mysql> SELECT * FROM mysql.user WHERE Password='';  
mysql> FLUSH PRIVILEGES;
```
- Détruire la base de données test  

```
mysql> DROP DATABASE test;  
mysql> DELETE FROM mysql.db WHERE Db='test' OR Db='test\_%';
```
- Désactiver l'accès distant à l'utilisateur « root »  

```
mysql> UPDATE mysql.user SET Host='localhost' WHERE user='root';  
mysql> FLUSH PRIVILEGES;
```

# MySQL : Utilisateurs / Hôtes



- MySQL associe un hôte à un utilisateur
- L'hôte peut être une adresse IP ou un nom
  - `mysql> CREATE USER 'marc@192.168.110.1' IDENTIFIED BY 'motdepasse';`
  - `mysql> CREATE USER 'marc@192.168.110.%' IDENTIFIED BY 'motdepasse';`
  - `mysql> CREATE USER 'marc@192.168.110.0/255.255.255.0' IDENTIFIED BY 'motdepasse';`
  - `mysql> CREATE USER 'marc@mondomaine.com' IDENTIFIED BY 'motdepasse';`
  - `mysql> CREATE USER 'marc@machine' IDENTIFIED BY 'motdepasse';`
  - `mysql> CREATE USER 'marc@%' IDENTIFIED BY 'motdepasse';`

# MySQL : Interdiction de l'accès distant



- Accès distant inutile quand le serveur est interrogé uniquement en **local**.  
Ex: Apache/PHP et MySQL installés sur la même machine
- **Bloquer le port** entrant de MySQL (en général 3306)
  - Démarrer le service avec l'option : `-skip-networking`
  - Ajouter l'option `-skip-networking` dans la section `[mysqld]` du fichier de configuration, généralement: `/etc/mysql/my.conf`
- Supprimer les utilisateur qui peuvent se connecter à distance
  - `mysql> DELETE FROM mysql.user WHERE Host <> 'localhost';`
  - `Mysql> FLUSH PRIVILEGES`

# MySQL : Chiffrement des données



- Même avec un serveur sécurisé, les données peuvent être en danger
- MySQL propose un certain nombre de fonctions de chiffrement et de hachage. Il supporte les algorithmes :  
**MD5, SHA1, DES** (si compilé avec le support SSL), **AES**
- **Attention des failles sérieuses ont été trouvées dans les algorithmes MD5 et SHA1**
- MySQL possède une fonction de hachage « **PASSWORD ()** » pour crypter les mots de passe, elle n'est pas non plus recommandée
- Fonction « **ENCRYPT ()** » sur système UNIX: **combinaison de MD5 et DES** associé à une phrase (générée si non précisée)

# MySQL : Limites des utilisateurs



- Un minimum de droit = un maximum de sécurité
- Limitations globales sur une base, sur une table, une colonne
- **GRANT** : définir une limitation
  - `mysql> GRANT autorisation ON base.table TO 'utilisateur'@'hôte' IDENTIFIED BY 'password' ;`
  - `Mysql> GRANT SELECT (prix) ON magasin.article TO 'marc'@'%' IDENTIFIED BY 'pouet' ;`
- **REVOKE** : enlever une limitation
  - `mysql> REVOKE SELECT ON magasin.article FROM 'marc'@'%' ;`

# MySQL : Limitations



- Limiter les connexions
  - « **MAX\_CONNECTIONS** » : Connexion simultanées au serveur
  - « **MAX\_USER\_CONNECTIONS** » : Connexion simultanées par user
  - « **MAX\_CONNECTIONS\_PER\_HOUR** »
- Limiter le nombre de requêtes
  - « **MAX\_QUERIES\_PER\_HOUR** »
  - « **MAX\_UPDATE\_PER\_HOUR** »



# PostFix

# PostFix



- C'est un serveur de **messagerie** électronique libre (MTA)
- Alternative plus **rapide**, plus **simple** à administrer et plus sécurisé que l'historique « Sendmail »
- Présent sur plusieurs **OS** (Unix, Linux, Mac OS X)
- **Facilité** de configuration
- Il peut gérer un grand nombre de cas d'utilisations
- Et un bon nombre de pourriels
- Peut déléguer la gestion d'**analyse** de courriels

# PostFix



- Vérifier que le serveur **DNS** pointe vers le serveur de courrier **MX cloomcloom.com**
- Fichiers de configurations :
  - «**main.cf**» : dans « **/etc/postfix/** »
  - «**master.cf**» : dans « **/etc/postfix/** »

# PostFix : main.cf



- Filtrer le courriel  
`content_filter = smtp-amavis:[127.0.0.1]:10024`
- Authentification avec «`helo`»  
`smtp_helo_required = yes`
- Changer la bannière d'indication  
`smtp_banner = bienvenue sur mon serveur...`
- Eviter les messages trop volumineux  
`message_size_limit = 4096000`

# PostFix : main.cf



- Interdire le « **relaying** » et les connexions **sans DNS** inverse

```
smtpd_recipient_restrictions =  
    permit_mynetworks,check_relay_domains
```

```
relay_domain = postfix.com
```

```
smtpd_client_restrictions = reject_unknown_client,  
    permit_mynetworks, check_client_access  
    hash:/etc/postfix/access, reject_maps_rbl
```

```
smtpd_sender_restrictions = reject_unknown_sender_domain,  
    check_sender_access hash:/etc/postfix/access,  
    reject_non_fqdn_sender, reject_maps_rbl
```

- Liste de rejet dans le fichier « **/postfix/access** »

```
aol.com REJECT
```

```
msn.com DISCARD
```

# PostFix : master.cf



- Filtrer les pourriels

```
smtp inet n - n - - smtpd -o
content_filter = spamassassin
spamassassin unix - n n - - pipe
```
- Scanner les courriels

```
smtp-amavis unix - - n - 2 smtp -o
```

# PostFix + AMaViS



- **A Mail Virus Scanner**
- Il s'**intercale** avant la délivrance locale par PostFix
- Scanné puis déposé soit dans la boîte, soit en quarantaine
- Configuration par « **/etc/amavis/conf.d/50.user** »
  - **Bloquer** les connections à l'interface locales

```
$inet_socket_bind = '127.0.0.1';  
@inet_acl = gw(127.0.0.1);
```
  - **Remplacement** du courriel infecté par un sain avertissement
  - **Avertir** l'expéditeur, le destinataire et l'administrateur

# PostFix + SpamAssassin

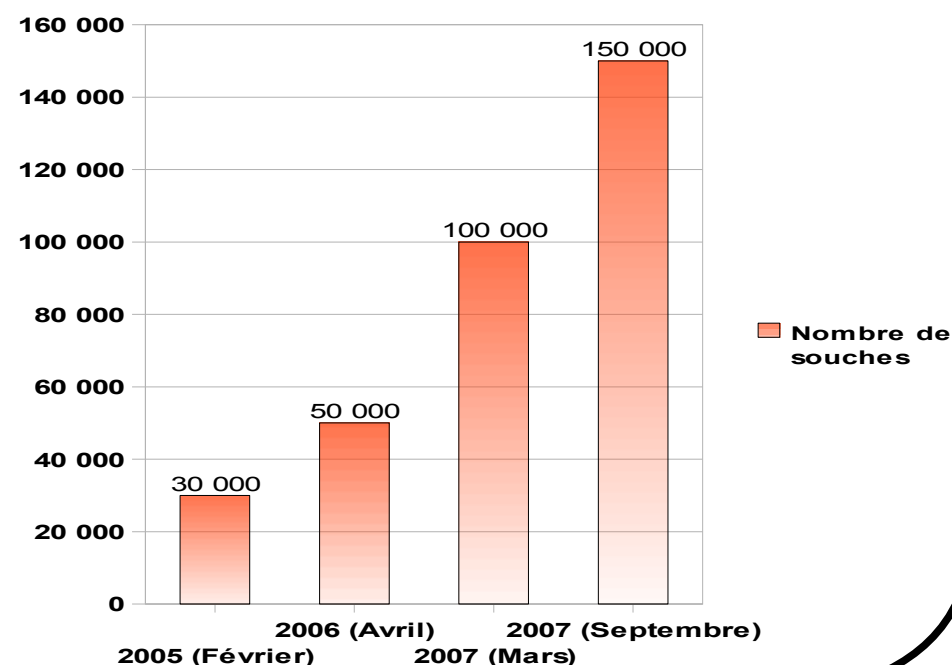


- Logiciel de **neutralisation** des pourriels
- Quelques chiffres
  - **130 000 000 000**
  - **50 %** des courriels
  - **1** internaute **sur 2**
  - **2,8** minutes
- Configuration par « `/spamassassin/local.cf` »
  - Un système de **score** : **required\_score 5.0**

# PostFix + ClamAV



- **Clam AntiVirus**
- Il est généralement utilisé avec les serveurs de courriels
- Son but? **Contrer** les virus s'attaquant aux OS Microsoft (principalement)
- Fichiers de configurations :
  - Générales  
« `/clamav/clamav.conf` »
  - Mise à jour  
« `/clamav/freshclam.conf` »





# ProFTPd

# ProFTPd



- Libre et Open Source sous licence GPL
- Configuration proche de celle d'Apache HTTP Server
- Serveurs et utilisateurs virtuels
- Comptes facilement ChRooté
- Pas besoin de binaires dans les geôles ou dans les comptes ChRootés

# ProFTPd



- Le fichier de configuration est situé généralement  
« `/etc/proftpd/proftpd.conf` »
- Vérifier l'utilisateur (et le groupe) pour lancer le service  
`User proftpd`  
`Group nogroup`
- Utilisateurs bannis du service
  - `UseFTPUUsers On`
  - « `/etc/ftpusers` »
- Refuser les utilisateurs avec un shell inexistant dans  
« `/etc/shell` »  
`Require ValidShell On`

# ProFTPD



- Ne pas afficher la version de ProFTPD
  - **ServerIdent off (DeferWelcome On)**
    - › 220 FTP server ready
    - › 220 ProFTPD 1.3.0 Server ('Server Name') [IP\_Serveur]
- Changer le port par défaut
  - **Port [Numéro\_port]**
- Changer le fichier des mots de passe
  - **AuthUserFile [chemin\_fichier]**
  - Création du fichier contenant les mots de passe

```
ftpasswd --passwd --name=<login> --uid=<user_id> --gid=<group_id> --home=<ftp_root_path> --shell=/bin/false
```

# ProFTPD : Les directives



- Peuvent être appliqué à un **dossier** ou à un **serveur virtuel**

```
<Limit [action]>  
  [Ordres]  
</Limit>
```

- Les actions sont divisés en 5 groupes :

- **ALL** : toutes les commandes FTP
- **DIRS** : commandes sur les répertoires: CDUP, CWD, LIST, MDTM, NLST, PWD, RNFR, STAT, XCUP, XCWD, XPWD
- **LOGIN** : commandes sur les logins
- **READ** : commandes sur la lecture des fichier: RETR, SIZE
- **WRITE** : commande sur l'écriture des fichiers/dossiers: APPE, DELE, MKD, RMD, RNTD, STOR, STOU, XMKD, XRMD

- Les ordres peuvent être :

- Allow, Deny, AllowAll, DenyAll, AllowUser, DenyUser, AllowGroup, DenyGroup...

# ProFTPD : Authentification renforcée



- 2 modules sont présent pour renforcer l'authentification
  - « **mod\_wrap** » : emballage TCP supposant 2 sous-modules :
    - « **mod\_wrap\_file** » : table d'accès dans un **fichier**
    - « **mod\_wrap\_sql** » : table d'accès dans une **base SQL**
  - « **mod\_radius** »

# ProFTPD



- Désactiver les modules inutiles
  - Commenter les modules non souhaités dans « `/etc/proftpd/modules.conf` »
  - Passer à « off » dans le fichier de configuration

```
<IfModule mod_tls.c>  
    TLSEngine off  
</IfModule>
```
- Ne pas activer les utilisateurs anonymes
  - Ajouter anonyme, ftp dans le fichier « `/etc/ftpusers` »
  - Commenter dans le fichier de configuration les lignes comprises entre

```
<Anonymous ~ftp> et </Anonymous>
```

# ProFTPD : Chiffrement SSL



- Chiffrement SSL implicite
  - Semblable à l'HTTPS
  - Connexion au serveur sur le port 990 (port des commandes)
  - Port des données 989 chiffré
- Chiffrement SSL explicite
  - Connexion sur le port 21
  - Demande de chiffrage à travers la commande « **AUTH TLS** »
  - Commandes et données sont chiffrées
- Chiffrement TLS implicite
  - Identique au SSL implicite
  - Les données sont chiffrées avec la commande « **PROT P** »

# Bibliographie



- **Général**

- <http://fr.wikipedia.org/wiki/Accueil>
- [http://www.institut.math.jussieu.fr/informatique/tunnel/tunnel\\_ssh.html](http://www.institut.math.jussieu.fr/informatique/tunnel/tunnel_ssh.html)

- **Apache**

- <http://www.linux-pour-lesnuls.com/chroot.php>
- <http://www.cgsecurity.org/Articles/apache.html>

- **MySQL**

- <http://krierjon.developpez.com/mysql/securiser/index.php#sommaire>
- <http://dev.mysql.com/doc/refman/5.0/fr/security.html>
- <http://www.cgsecurity.org/Articles/mysql.html>

- **PostFix**

- <http://www.linux-pour-lesnuls.com/serveurcourrier.php>
- <http://www.cgsecurity.org/wiki/Articles>

- **ProFTPd**

- <http://www.proftpd.org/>
- <http://www.castaglia.org/proftpd/doc/contrib/ProFTPD-mini-HOWTO-TLS.html>
- <http://www.linux-pour-lesnuls.com/ftp.php>



# Questions ?