Sécurité des réseaux

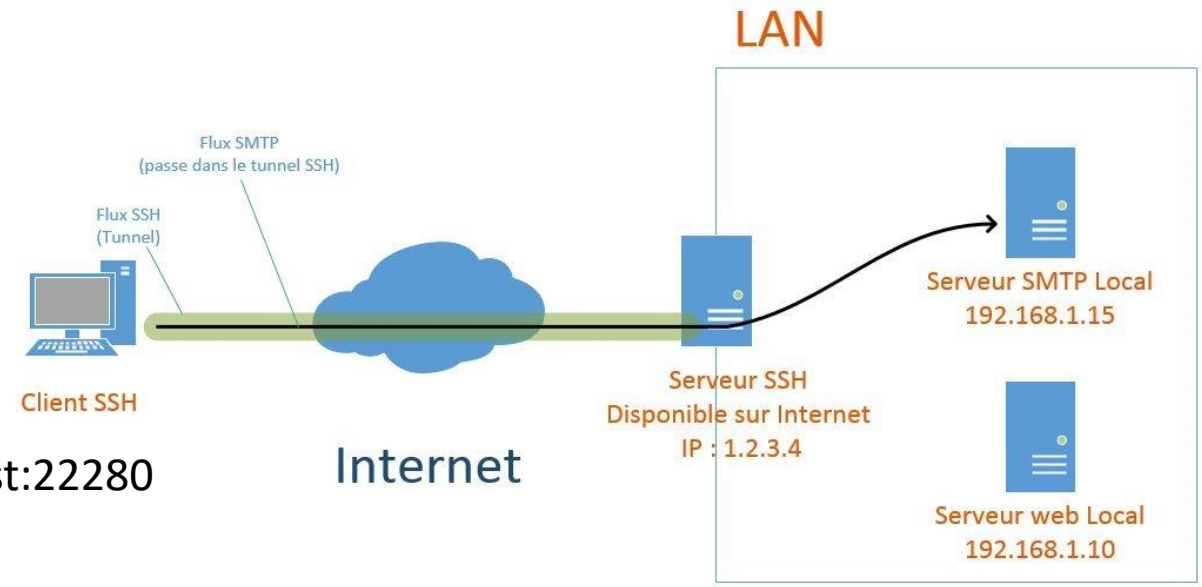
M. Jean DIOKH

Certifié Linux Professional Institute (**LPIC-3 Core** & **LPIC-3 Security**)
http://www.lpi.org/verify/LPI000201968/9ntq6grjtk

Sécurité des applications

Sécurité des applications

Tunnel SSH



http://localhost:22280

Sous Linux

ssh -LportLocal: adresse_serveur_non_securisé:port adresse_serveur

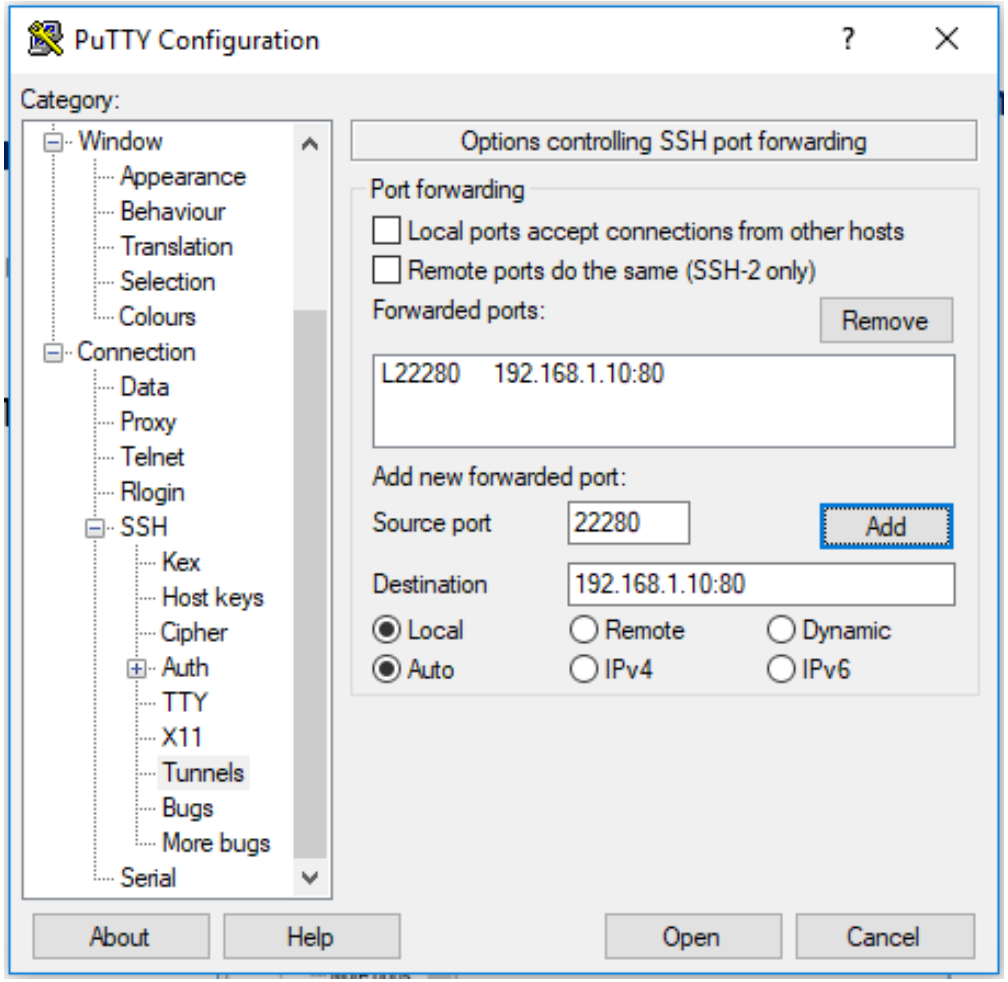
Exemple:
ssh -L22280:192.168.1.10:80 1.2.3.4

Sécurité des applications

Tunnel SSH

Putty

Après une connexion vers le serveur SSH, on active le tunnel



Sécurité des applications

DNS

Version de BIND

named.conf.options : version 'dns';

nslookup -type=txt -class=chaos version.bind adresseServeurDNS

Autorisation des requêtes:

- allow-query { RESEAUX-AUTORISES; };
- allow-recursion { RESEAUX-AUTORISES; };

Sécurité des applications

DNS

Transfert des éléments de zone :

- allow-transfer { adresseIP_slave; }

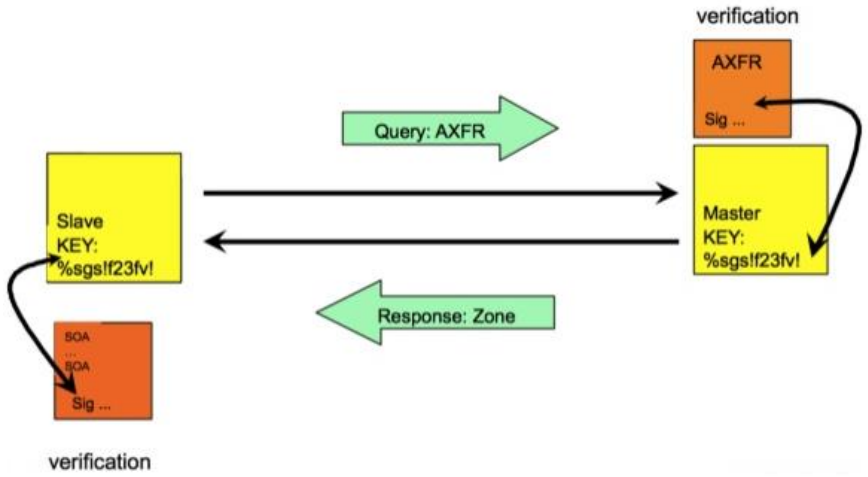
Sécurité des applications

DNS

Echange sécurisé entre le serveur primaire et le serveur secondaire - TSIG

La sécurisation reposera surtout sur l'authentification et l'intégrité des données. C'est-à-dire qu'on veut être certain que c'est bien le bon serveur qui nous répond, et que les données ne subissent pas de modification pendant le trajet.

Nous allons utiliser ici le mécanisme **TSIG** (Transaction SIGnature, signature des transactions). Ce mécanisme repose sur la présence d'un secret partagé par les serveurs qui échangent des données.



Sécurité des applications

DNS

Configuration côté Primaire

Génération de la clé

```
root@srv-diokh:/etc/bind# rndc-confgen -a
wrote key file "/etc/bind/rndc.key"
root@srv-diokh:/etc/bind#
```

Déclaration de clé

```
root@srv-diokh:/etc/bind# cat rndc.key >> named.conf.default-zones
root@srv-diokh:/etc/bind# _
```

Sécurité des applications

DNS

Configuration côté Secondaire

```
zone "test.sn" {
        type slave;
        file "/var/cache/bind/db.test.sn";
        masters { 192.168.1.18 key maCle; };
};

key "maCle" {
        algorithm hmac-md5;
        secret "fgoO8/ESmobf9YyGJEdTqg==";
};
```

Sécurité des applications

WEB - Apache

ServerTokens Prod

ServerSignature Off

L’utilisation de liaisons chiffrées et authentifiés: HTTPS

L’authentification des utilisateurs: mod_auth_basic, mod_authnz_ldap

Restreindre l’accès des pages: mod_authz_host, mod_security

Sécurité des applications

WEB - Apache

HTTPS - certificat auto-signé

Génération du certificat

```
root@srv-desktop:~# openssl req -x509 -nodes -newkey rsa:1024 -keyout /etc/ssl/test.sn.key -out /etc
/ssl/test.sn.crt -days 365
Generating a RSA private key
..........+++++
.......................+++++
writing new private key to '/etc/ssl/test.sn.key'
-----
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) [AU]:SN
State or Province Name (full name) [Some-State]:Senegal
Locality Name (eg, city) []:Dakar
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Test
Organizational Unit Name (eg, section) []:Securite
Common Name (e.g. server FQDN or YOUR name) []:www.test.sn
Email Address []:
root@srv-desktop:~#
```

Sécurité des applications

WEB - Apache

HTTPS - certificat auto-signé

Configuration

```
<VirtualHost _default_:443>
ServerName www.test.sn
DocumentRoot /var/www/html/test.sn
SSLEngine On
SSLCertificateFile /etc/ssl/test.sn.crt
SSLCertificateKeyFile /etc/ssl/test.sn.key
</VirtualHost>
```

NB: le module SSL doit être chargé (a2enmod ssl)

Sécurité des applications

WEB - Apache

HTTPS - certificat auto-signé

Redirection HTTP vers HTTPS

```
root@srv-diokh:~# cat /etc/apache2/sites-available/test.sn.conf
<VirtualHost *:80>
ServerName www.test.sn
DocumentRoot /var/www/html/test.sn/
redirect permanent / https://www.test.sn/
</VirtualHost>
root@srv-diokh:~#
```

Sécurité des applications

WEB - Apache

HTTPS - Let’s Encrypt

Sécurité des applications

WEB - Apache

HTTPS - Let’s Encrypt

Install Certbot

Run this command on the command line on the machine to install Certbot.

```
$ sudo apt-get install certbot python3-certbot-apache
```

Sécurité des applications

WEB - Apache

HTTPS - Let’s Encrypt

Choose how you'd like to run Certbot

Either get and install your certificates...

Run this command to get a certificate and have Certbot edit your Apache configuration automatically to serve it, turning on HTTPS access in a single step.

```
$ sudo certbot --apache
```

Or, just get a certificate

If you're feeling more conservative and would like to make the changes to your Apache configuration by hand, run this command.

```
$ sudo certbot certonly --apache
```

Sécurité des applications

WEB - Apache

HTTPS - Let's Encrypt

Test automatic renewal

The Certbot packages on your system come with a cron job or systemd timer that will renew your certificates automatically before they expire. You will not need to run Certbot again, unless you change your configuration. You can test automatic renewal for your certificates by running this command:

```
$ sudo certbot renew --dry-run
```

The command to renew certbot is installed in one of the following locations:

```
/etc/crontab/

/etc/cron.*/*

systemctl list-timers
```