Labo SSH (90 min)

0	Introduction	`sudo ./c 4`

Le protocole SSH est très utilisé dans le monde Unix-Linux permettant à un administrateur d'accéder à distance à ses serveurs. SSH présente de nombreuses analogies avec SSL

Configuration pour ce travail pratique :

- **PC-Windows7** avec le **client putty**
  albert (compte administrateur), *password* : admin

- **PC-Fedora16 CLI** en tant que serveur et proxy SSH (sans interface graphique)
  labotd (compte utilisateur), password : labolabo
  root (compte administrateur), password : rootroot

- VM Ubuntu Server remplace le client putty au §4-5 :
  labotd (compte utilisateur), password : labolabo

Action Sur le **PC-Windows,** ouvrir une session
Copier le dossier partagé **`\\10.2.1.1\doclabo\RSX\1_SSH`** sur le bureau

1	Utilisation du client SSH	10 min

But 1.1 **Configuration du serveur**

Action Sur le **PC-Fedora (compte labotd)**, typer **`ifconfig`** pour trouver son adr. IP
Choisir la bonne interface **em1**
Noter cette valeur ……

Action Typer **`ssh-keygen –lf /etc/ssh/ssh_host_rsa_key.pub`**
Noter cette valeur (3 premiers et 3 derniers bytes) ……

Lien http://linux.die.net/man/1/ssh-keygen

But 1.2 **Connexion avec putty**

Action Sur le **PC-Windows**, ouvrir putty (Bureau)
Se connecter au serveur

Q_1a Pourquoi une fenêtre Security Alert apparaît ?

Q_1b Pouvez-vous faire confiance à ce serveur ?

Q_1c Que se passe-t-il en répondant Yes ?

But 1.3 **Contrôler que la clé publique du serveur est présente sur le PC-Windows**

Action Ouvrir l'éditeur de registre
Menu Start, typer **`regedit`** dans le champ de recherche
Aller dans **`HKEY_CURRENT_USER\SoftWare\SimonTatham\PuTTY\SshHostKeys`**

Q_1d Pourquoi la valeur présente dans la base de registre est différente de celle affichée dans la fenêtre Security Alert ?

But 1.4 **Connexion à un serveur en mode ligne de commande**

Remarque Putty peut également s'utiliser en ligne de commande (CLI) comme nous pourrions le faire sous Linux avec la commande **ssh**

Action Ouvrir le **`Command Prompt`** de Windows (raccourci bureau)
Typer **`cd Desktop`** pour sélectionner le bureau
Typer **`putty -ssh labotd@IP_Fedora`**

2	**Clés stockées sur le serveur SSH**	**20 min**

But 2.1 Sur le **PC-Fedora, ouvrir le dossier `/etc/ssh` contenant ces fichiers**

Action
```
cd /etc/ssh
ls –al
```

Répondre aux questions à l'aide du lien suivant
http://docs.fedoraproject.org/en-US/Fedora/15/html/Deployment_Guide/s1-ssh-configuration.html

Q_2a A quoi sert le fichier **`moduli`** ?

Q_2b A quoi sert le fichier **`sshd_config`** ?

Q_2c Que contient le fichier **`ssh_host_rsa_key`** ?

Q_2d Que contient le fichier **`ssh_host_rsa_key.pub`** ?

Remarque
`ssh_config` contient la configuration du client SSH
`ssh_host_key` et **`ssh_host_key.pub`** contiennent les clés pour la version 1 du protocole SSH
`ssh_host_dsa_key` → http://fr.wikipedia.org/wiki/Digital_Signature_Algorithm

But 2.2 **Calcul de l'empreinte (fingerprint) à partir de la clé pub du serveur**

Extrait de rfc4716 §4 → https://tools.ietf.org/html/rfc4716#section-4

4. Public Key Fingerprints

The security of the SSH protocols relies on the verification of public host keys. Since public keys tend to be very large, it is difficult for a human to verify an entire host key. Even with a Public Key Infrastructure (PKI) in place, it is useful to have a standard for exchanging short fingerprints of public keys.

This section formally describes the method of generating public key fingerprints that is in common use in the SSH community.

The fingerprint of a public key consists of the output of the MD5 message-digest algorithm [RFC1321]. The input to the algorithm is the public key data as specified by [RFC4253]. (This is the same data that is base64 encoded to form the body of the public key file.)

http://www.cisd.ethz.ch/services/SVN/SSR

Action Avec le compte root, copier la clé dans le dossier personnel
```
cp /etc/ssh/ssh_host_rsa_key.pub /root/
```

Mettre cette clé au format rfc4716 (option -e)
```
ssh-keygen -vef ssh_host_rsa_key.pub
```

Rediriger la sortie dans un fichier
```
ssh-keygen -vef ssh_host_rsa_key.pub > key.txt
```

Récupérer cette clé sur PC-Windows avec WinSCP

Cryptool : ouvrir ce fichier pour ne conserver que la valeur de cette clé

Cryptool : décoder en base64 :
Indiv. Procedures – Tools – Codes - Base64 Encode/Decode - Base64Decode

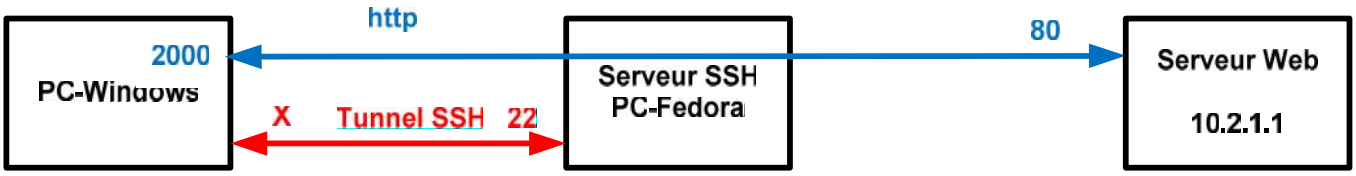
Cryptool : calcul du MD5
Indiv. Procedures - Hash - MD5

Comparer le résultat avec celui du §1.1

Q_2e Voir corrigé du §2.2

3	Tunnel SSH	30 min

Objectif Dans l'exemple ci-dessous, le flux http destiné au serveur 10.2.1.1 est protégé par un tunnel SSH du PC-Windows au serveur SSH (qui fait office de Proxy)



But 3.1 **Etablir une connexion HTTP sans tunnel**

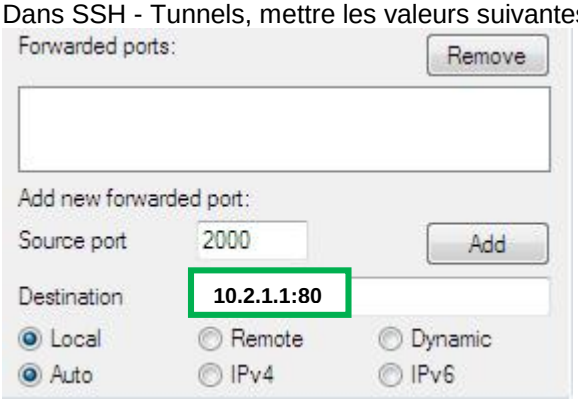
Action Sur le **PC-Windows**, ouvrir IE (Internet Explorer) puis typer **http://10.2.1.1/ip.php**

Remarque Ce script PHP retourne l'adresse IP du client http

```
<?php
echo "<h3>Votre IP: ".$_SERVER["REMOTE_ADDR"]."</h3>";
?>
```

But 3.2 **Créer un tunnel SSH encapsulant du HTTP**

Action Sur le **PC-Windows**, ouvrir putty (Bureau)
Entrer l'adresse IP
Dans SSH - Tunnels, mettre les valeurs suivantes :



Appuyer sur le bouton **Add** puis **Open**

Ouvrir à nouveau IE puis typer **http://localhost:2000/ip.php**

Q_3a Comment fonctionne le client putty ?

Q_3b Quelle est à présent l'adresse IP et pourquoi ?

But 3.3 **Analyser une capture Wireshark**

Infos **Client = 10.1.40.88** **Serveur_SSH = 10.1.40.124** **Serveur_Web = 10.1.1.2**
Acquisition **Tunnel.pcap** effectuée sur le serveur SSH

Q_3c L'échange entre Serveur_SSH et Serveur_Web est-il chiffré ?
Appliquer un filtre d'affichage

Q_3d Quels sont les ports TCP utilisés entre Serveur_SSH et Serveur_Web pour le flux http ?
Appliquer un filtre d'affichage

Q_3e Etudier les ports TCP utilisés entre Client et Serveur_SSH
Appliquer un filtre d'affichage
Utiliser l'onglet Statistics – Conversation

Q_3f Le port 2000 utilisé au §3.2 est-il visible dans l'acquisition Wireshark ? Sinon pourquoi ?

Q_3g Déterminer le schéma bloc Client – Proxy – Webserver
Représenter la couche applicative http, la couche SSH et la redirection (port forwarding)

4	Analyse Wireshark et logs	20 min

Objectif Comprendre les différentes étapes du protocole SSH lors de l'établissement d'une connexion à partir de :

- **Putty.pcap** = Acquisition effectuée sur le client SSH (IP=10.2.3.65)
- **Log_Putty.pdf** = Logs produits par Putty

Q_4a Quelle l'adresse IP du serveur ?

Q_4b Quelle est la chaîne d'identification du serveur ?

Q_4c Quelle est la chaîne d'identification du client ?

Q_4d Quel est l'algorithme utilisé pour le chiffrement des données ?

Q_4e Quel est l'algorithme utilisé pour le contrôle d'intégrité ?

Q_4f Dans quel paquet se trouve la valeur e ?

Q_4g Dans quel paquet se trouve la valeur f ?

Q_4h Dans quel paquet se trouve la clé publique du serveur ?

Q_4i Quelle est l'utilité de la signature dans le paquet 8 ?

Q_4j Quelle est l'invite de commande (prompt) obtenue par le client ?

5	En réserve	10 min

Objectif Mettre en place une connexion automatique (passwordless) avec le serveur basée sur une authentification par clé publique