TD

Cryptographie

8 janvier 2023

Ce TP propose d'étudier deux jeux de protocoles applicatifs permettant de sécuriser ou non la communication.

Le premier jeu fait appel aux applications Telnet et SSH permettant d'administrer un serveur à distance. Cette première partie vous propose d'abord d'étudier en quoi le protocole Telnet est vulnérable à partir de l'accès au serveur d'adresse IP 172.31.7.123 joignable à partir du tunnel VPN [1]. Il est ensuite demande de configurer un serveur SSH sur votre Kali de manière à obtenir une solution sécurisée pour le même serveur VPN.

Le second jeu fait appel à la configuration d'un serveur Web à partir d'Apache. Comme précédemment, il est proposé d'étudier l'authentification HTTP Basic et de retrouver les authentifiants en clair. La sécurisation du trafic est obtenue cette fois à partir de certificats et de l'implémentation d'une couche TLS protégeant HTTP.

I. (3 points) **Observation du trafic lors d'une connexion Telnet**

Après avoir monté votre tunnel VPN, connectez vous sur le serveur d'adresse IP 172.31.7.123 à partir de l'outil Telnet. Vous pouvez utiliser les authentifiants `student` ↔ `Enst@!`.

(1) En parallèle, observez le trafic à l'aide de `wireshark`. Du chiffrement est-il mis en place? Est-il possible de récupérer les authentifiants dans `wireshark`?

(2) Également, observer la quanté de paquets échangés au travers de la connexion VPN : que pouvez-vous en déduire quant à la longueur du mot de passe?

II. (8 points) **Installation et configuration d'un serveur SSH**

Contrairement à Telnet, SSH (*Secure SHell*) est un protocol réseau permettant la communication de manière sécurisée, c'est-à-dire imposant chiffrement et authentification, au travers d'un réseau qui n'est pas de confiance comme Internet.

L'implémentation la plus populaire est le programme Open Source `OpenSSH`.

(1) Prise en main : sous Kali, le paquet `openssh-server` est déjà installé.

Créer un compte système sans privilège et en communiquer le mot de passe à votre binôme.

⚠pour cela, la configuration réseau doit effectivement permettre à votre binôme de se connecter à votre machine virtuelle. Il faut paramétrer *Réseau* → Connexion par pont (ou *bridge*) permettant à la machine virtuelle d'avoir un accès direct au réseau de l'hôte. Il faut également regénérer une adresse MAC et une nouvelle adresse IP.

Démarrer le serveur en utilisant sa configuration par défaut et connecter vous sur le serveur de votre binôme.

1. `https://cours.pelicanux.net/static/conf/cours.ovpn`

Quel est le message affiché ? À quoi correspond-il ? Quelle est sa signification ? Quel risque éventuel identifiez-vous ? Quelles étapes doit-on suivre pour s'en prémunir ?

(2) Supprimer à présent les clefs présentent dans le répertoire de configuration de ssh, puis regénérer-les et redémarrer le service ! Quel message d'erreur voit alors votre binôme ? Quel est sa signification ? Quelle attaque peut alors être déjouée ?

(3) Des options en ligne de commandes permettent cependant de ne pas tenir compte de ce message d'avertissement. Quelles sont-elles ? Contre quel type d'attaque ne peut-on plus alors se protéger ? Quells protection continue cependant d'offrir SSH ?

(4) Configurer maintenant votre serveur de manière à interdire l'authentification par mot de passe et imposer l'authentification par clef. En parallèle, générer des clefs et fournisser la partie publique à votre binôme. Configurer votre serveur de manière à autoriser la sienne. Quelles sont les différentes étapes à suivre pour implémenter ce type d'authentification ?

(5) Est-il toujours possible de mener une attaque active de *man-int-the-middle* ? Que se passe-t-il si l'utilisateur comme précédemment ne tient pas compte du message de sécurité ? ⚠ **Question chasseurs de points. Cette question est complexe, en réalité il n'est plus possible de mener une attaque de *man-in-the-middle*, mais pourquoi ? Un gros paquet de points récompensera une explication détaillée.**

III. (9 points) **Installation et configuration d'un serveur Web TLS**

Le but de cette partie consiste à créer un serveur Web et lui permettre d'assurer du chiffrement par certificat.

(1) installer le paquet `apache2` et récupérer la configuration d'apache sur le serveur web mis en place. Choisissez un nom de domaine et communiquer-le à votre binôme. Renseignez vous et votre binôme dans le fichier `/etc/hosts` ce nom de domaine avec le lien avec votre adresse IP. À partir de la commande `htpasswd`, créer un fichier de mot de passe pour que seul lui puisse accéder au contenu. Votre binôme parvient-il à joindre votre serveur ? La connexion est-elle chiffrée ? Qu'observez-vous avec wireshark ? Est-il possible d'obtenir le mot de passe de votre binôme ?

(2) Nous allons maintenant configurer notre serveur pour permettre du chiffrement. La commande suivante permet de générer un certificat et une clef privée :

```
openssl req -x509 \
  -newkey rsa:4096 \
  -keyout key.pem \
  -out cert.pem \
  -days 365
```

Les options de configuration suivantes permettent de configurer `apache` pour prendre en compte ces certificats :

```
SLEngine on
SSLCertificateFile /path/to/you.cert
SSLCertificateKeyFile /path/to/you.key
```

Le chiffrement est-il assuré ? Quel message est envoyé par votre navigateur ? Quelle est sa signification ?

(3) Vous allez à présent créer une autorité de certification et l'utiliser pour signer des certificate requests émises par votre binôme. En réalité, la commande permettant

de générer cette autorité de certification est exactement la même que précédemment! Il est recommandé de la nommer nom_CA.key et nom_CA.pem. Intégrez la partie publique de l'autorité de votre binôme dans votre navigateur!

(4) Créez maintenant votre certificate Signing Request pour votre site Web! Pour cela, créez d'abord une clef publique, puis à partir de celle-ci, le CSR :

```
openssl genrsa -out name.key 2048
openssl req -new -key name.key -out name.csr
```

Fournissez cette CSR à votre binôme pour qu'il l'a signe avec sa CA :

```
openssl x509 -req -in name.csr \
  -CA myCA.pem \
  -CAkey myCA.key \
  -CAcreateserial \
  -out mine.crt \
  -days 365 -sha256
```

Votre nagivateur renvoie-t-il encore des messages d'avertissement? Pourquoi?