TP2: Sécurité des réseaux

Objectifs

- Découverte réseau
- Capture de trames
- HTTP
- Man In The Middle
- ARP spoofing
- Interception HTTPS

Au cours de ce TP vous allez vous familiariser avec quelques protocoles utilisés pour accéder à des ressources sur internet et avec un certain nombre d'outils d'audit réseau.
Après ce tour d'horizon, vous vous mettrez dans la peau d'un attaquant et tenterez de déjouer une connexion sécurisée afin de voler des identifications.

Préparation du TP

Mise en place

Vous disposez de 2 PCs, l'un sera considéré comme la victime et l'autre comme l'attaquant.

Identifiez l'adresse IP de la victime et créez l'enregistrement « *victime* » dans votre fichier /etc/*hosts*.
Pour vérifier l'opération, vous pouvez lancer la commande : *ping victime*

Introduction à Nmap

Nmap peut détecter les hôtes connectés avec l'option –sP. Une cible peut être une IP ou une plage IP (*ex 192.168.1.0/24*). Par exemple si la plage à scanner pour détecter les hôtes connectés est 192.168.1.0/24 on lancerait nmap de la façon suivante :

nmap –sP 192.168.1.0/24

Filtres Wireshark

- ip.src == 192.168.1.12 : n'affiche que les paquets dont l'IP source est 192.168.1.12
- ip.dst == 192.168.1.1 : n'affiche que les paquets dont l'IP destination est 192.168.1.1
- ip.addr == 192.168.1.1 : n'affiche que les paquets dont la source OU la destination est
- tcp.src == 80 : n'affiche que les paquets dont le port TCP source est 80 (HTTP)
- tcp.dst == 80 : n'affiche que les paquets dont le port TCP destination est 80
- tcp.addr == 80 : n'affiche que les paquets dont le port TCP source OU destination est 80

https://wiki.wireshark.org/DisplayFilters

Découverte réseau

Scannez le réseau local (nmap –sP), puis à l'aide de Wireshark expliquez le fonctionnement de la commande.

On peut également demander à nmap de nous scanner les services disponibles (TCP ou UDP) pour un hôte en particulier ou pour une plage d'hôte. Il y a différentes façons de savoir si un service est disponible, la plus simple étant d'essayer de s'y connecter et de voir ce qui nous répond.
Pour un scan TCP on utilisera le paramètre -sS et pour un scan UDP on utilisera le paramètre -sU. On peut combiner ce paramètre avec le paramètre –p pour ne rechercher qu'un port en particulier.

Scanner les postes du réseau local à la recherche de serveurs web, et ssh en écoute. Listez ensuite tous les ports TCP ouverts sur ces serveurs.

Man-In-The-Middle

L'attaque Man-In-The-Middle (MITM) consiste pour une machine malveillante à se positionner sur le chemin des paquets transitant entre une victime et un serveur distant. Cette position permet à un attaquant malveillant d'intercepter des communications, de les modifier à la volée ou même de les supprimer.

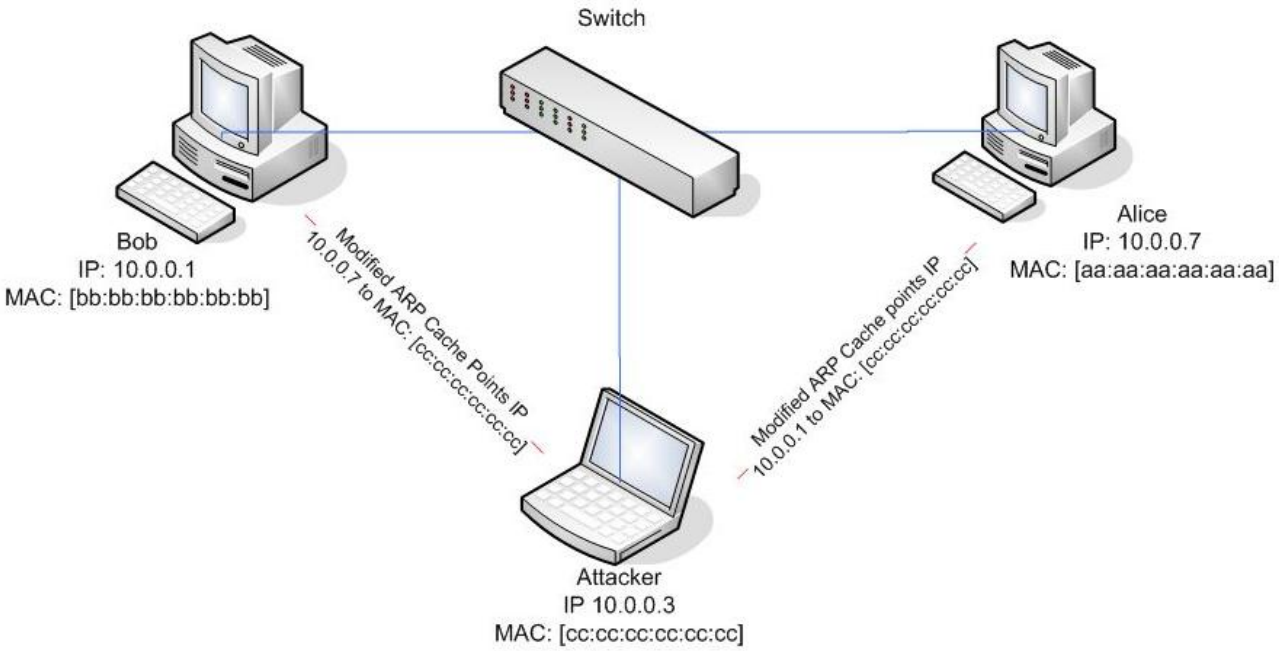
A l'aide de la commande *traceroute*, indiquez quel est le premier point de passage des paquets envoyés par la victime lorsqu'elle navigue sur internet.

ARP Cache Poisoning

Si l'attaquant ne se trouve pas naturellement sur le chemin, il est cependant possible sur un réseau lo cal de forcer à être un intermédiaire grâce à l'attaque dite de « ARP cache poisoning ».
Cette technique consiste à corrompre le cache ARP de deux machines A et B du chemin pour :

- Faire passer l'attaquant pour A auprès de B
- Faire passer l'attaquant pour B auprès de A

Le schéma suivant résume l'attaque :



Si la victime navigue sur internet, pouvons-nous modifier le cache ARP du serveur web ?

De quelle machine pouvons-nous corrompre le cache ARP pour intercepter la navigation internet de la victime ?

Nous allons utiliser l'outil mitm_arp (fourni avec le TP, ***python mitm_arp -h*** pour voir l'aide de l'outil) pour corrompre le cache ARP de la victime et de la cible.

A l'aide la commande : mitm_arp et de l'option –t (cible à déterminer) essayez d'intercepter la navigation. A l'aide de wireshark expliquez comment cela fonctionne.

La commande **arp -a** sous Linux et Windows permet de consulter le cache ARP de la machine. En utilisant cette commande, vérifiez sur la machine victime que l'adresse MAC associée à l'adresse IP de la gateway correspond bien à l'adresse MAC de l'attaquant.

Si on essaye de se connecter à des sites WEB depuis la machine de la victime cela ne fonctionne pas. Sur l'ordinateur de l'attaquant, tapez les commandes suivantes :

echo 1 > /proc/sys/net/ipv4/ip_forward

echo 0 | tee /proc/sys/net/ipv4/conf//send_redirects*

Que fait la première commande ?

Lancez un ping depuis le poste victime vers la passerelle. Vérifiez que le ping est bien intercepté par le poste attaquant. Vérifiez que vous voyez bien passer tous les échanges entre la victime et son site internet. L'attaque est-elle visible par la victime ?

Exploiter le Man-In-The-Middle

Laissez tourner l'ARP spoofing entre la victime et la passerelle. Vous avez accès au trafic sortant de la victime. Tout le trafic chiffré est pour l'instant inaccessible pour l'attaquant (par exemple le SSL lorsque la victime navigue sur des sites en HTTPs). Nous verrons dans la suite de se TP comment faire pour déjouer le chiffrement entre la victime et sa destination.

Mais pour le moment, tout le trafic en clair sortant de la victime est directement accessible pour l'attaquant. Par exemple un attaquant peut récupérer les credentials d'un switch/routeur administré avec TELNET. De la même manière il peut récupérer des credentials FTP, HTTP, SMTP, etc.

Dans ce chapitre, nous allons nous concentrer sur le protocole HTTP.

Pour cet illustrer cet exercice, nous allons utiliser le site **www.charentelibre.fr**, accessible entièrement en HTTP.

1) le protocole HTTP

a) Sur le poste victime, connectez-vous avec un navigateur sur la page du site : www.charentelibre.fr

Analysez les requêtes HTTP avec Wireshark. Pouvez-vous distinguer plusieurs parties différentes dans une requête HTTP ? Combien de requêtes ont été effectuées ? Essayez de les expliquer.

Même question pour les réponses HTTP. Expliquez à quoi correspond le statut de la réponse HTTP. Essayez de les distinguer rapidement en grandes familles.

Sur le poste de la victime, connectez-vous au site web de gameblog en passant par Google. Analysez les entêtes HTTP de la requête vers le site web de gameblog.
Expliquez à quoi sert le champ Referer dans les en-têtes HTTP.

Expliquez à quoi correspond le champ User-Agent dans les en-têtes HTTP. Peut-il être modifié ?

Commentez l'impact de ces métadonnées HTTP quand vous naviguez sur internet.

Sur le poste de la victime, téléchargez un pdf : http://packetlife.net/media/library/1/BGP.pdf
Sur le poste de l'attaquant, extraire le PDF téléchargé par la victime avec Wireshark en sélectionnant le contenu dans la réponse HTTP, puis en exporter le contenu (File-->Export-->Selected Packet Bytes)

Pouvez-vous ouvrir le PDF ? Commentez

2) Interception de mots de passe

Dans cette partie, nous allons voir comment récupérer les credentials d'un utilisateur lors d'une connexion en HTTP. Puis, trouver d'autres moyens pour voler l'identité d'un utilisateur.

Sur le poste victime, créez un compte utilisateur sur le site www.charentelibre.fr vous pouvez utiliser une @mail temporaire (par exemple sur https://temp-mail.org/)

Sur le poste victime connectez-vous à l'aide de vos identifiants. Côté attaquant, retrouvez les traces de la connexion et récupérez les identifiants de l'utilisateur. Quelle méthode HTTP a été utilisée pour envoyer les credentials au serveur? Circulent-ils en clair ?

Sur le poste victime, à l'aide de votre navigateur, postez un commentaire sur un article du site.

Côté attaquant, analysez dans Wireshark la requête POST qui a permis d'envoyer le commentaire sur l'article que vous avez choisi. Notamment le contenu du commentaire et des champs du header HTTP.

Le but de la prochaine attaque va être de poster un commentaire à la place de l'utilisateur précédent sans s'être authentifié/connaître les credentials de l'utilisateur.

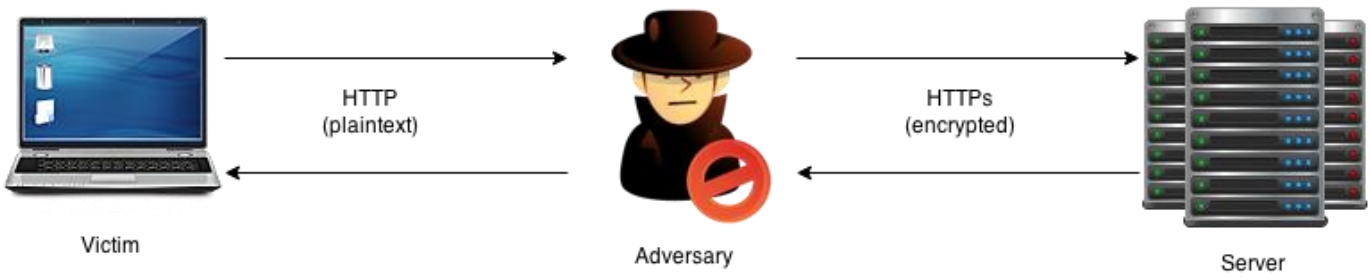
Pour cela, nous allons utiliser l'outil **curl**, un client HTTP (comme Firefox ou Chrome) en ligne de commande.

Voici la syntaxe à utiliser:

curl -X POST --data {contenu} **--header "Cookie:**{cookies de session}**"** {URL}

En vous aidant du POST précédent récupéré depuis Wireshark, forgez une requête valide et postez un commentaire à la place de la victime. Les outils de développeur de Firefox vous permettront d'obtenir un modèle de requête *curl* légitime (onglet réseau)

Sur le poste victime, rafraichissez la page et vérifiez qu'un nouveau commentaire soit apparu. Que s'est-il passé ? Selon vous quel est le rôle des cookies? Essayez d'expliquer comment l'attaquant a pu se faire passer pour l'utilisateur victime sans avoir envoyé son login/mot de passe.



Contourner le chiffrement

Nous avons vu précédemment comment utiliser le MITM pour récupérer des données, identifiants, métadonnées, credentials si les communications ne sont pas chiffrées. Cela peut parfois suffire à attaquant pour retrouver les informations qu'il recherche ou se propager plus loin encore dans le SI. Néanmoins, de nos jours les communications sont très souvent chiffrées, notamment avec TLS (anciennement SSL) pour la navigation WEB. Dans cette partie, nous allons voir quelques méthodes pour contourner le chiffrement et récupérer le contenu des communications.

Contourner une communication HTTP sécurisée avec TLS

Une faiblesse du HTTPs (HTTP + TLS) réside dans le fait que pour accéder à une page chiffrée il faut généralement d'abord passer par une page non chiffrée. Par exemple, sur certains sites, pour accéder à la version sécurisée (port 443), il faut y être redirigé depuis la page en clair non sécurisé sur le port 80.

Cela permet de piéger les utilisateurs qui ne rajoutent pas le 's' à la suite de http dans la barre d'adresse (une page sécurisée commence par https:// et se connecte sur le port 443, tandis qu'une page non sécurisée commence par http:// et se connecte sur le port 80)

Un attaquant peut exploiter ce comportement en modifiant à la volée toutes les réponses du serveur pour enlever le 's' de chaque chaque lien hypertexte de type "https://" avant de les relayer à la victime. L'attaquant communique donc de façon sécurisé avec le serveur mais la victime communique en clair avec l'attaquant.

L'outil que nous allons utiliser pour accomplir cette attaque s'appelle SSLStrip. SSLStrip agit comme un proxy HTTP vers HTTPs, il transfère les requêtes qu'il reçoit vers le véritable serveur et relaye ses réponses vers la victime.

Par défaut, SSLStrip écoute sur le port 10000. Il va falloir détourner le trafic entrant sur la machine de l'attaquant et à destination des port 80 et 443 sur le port d'écoute de SSLStrip (dans notre cas 10000).

Pour cela nous allons devoir faire du NAT. Utilisez la commande suivante:

iptables -t nat -A PREROUTING -p tcp --destination-port 80 -j REDIRECT --to-port 10000

iptables -t nat -A PREROUTING -p tcp --destination-port 443 -j REDIRECT --to-port 10000

Sur le poste attaquant lancer SSLStrip (avec les options -f et -w, lancez sslstrip -h pour voir l'aide)

Sur le poste victime, consultez des sites Web normalement chiffrés (étudiants bordeaux, facebook, twitter, instagram, etc.) directement en HTTP (ex: http://www.facebook.com) ou depuis un moteur de recherche en HTTP (bing, google, yahoo) puis connectez-vous à l'un de ces sites. **La connexion est-elle sécurisée? Récupérer les credentials sur le poste de l'attaquant.**

RQ: assurez-vous de ne pas avoir contacté auparavant la version HTTPs du site que vous ciblez, cela risque de rendre SSLStrip inefficace (cf. Exercice suivant)

Bonus 1

Depuis 2010, une nouvelle fonctionnalité a été ajoutée à TLS, le HSTS qui permet de forcer un navigateur Web compatible (c'est le cas avec Firefox et Chrome) à communiquer avec un serveur en HTTPs quoiqu'il arrive. En conséquence, il suffit qu'un utilisateur ait consulté au moins une fois la page HTTPs d'un serveur Web utilisant HSTS pour que SSLStrip soit inefficace sur ce site.

Néanmoins, certains mécanismes permettent de contourner en partie le HSTS, renseignez-vous sur la combinaison SSLStrip2 + dns2proxy pour déjouer le HSTS. Réalisez l'attaque en vous aidant de ces deux outils **SSLStrip2** et **dns2proxy**

Bonus 2

En utilisant l'outil BURP en tant que proxy transparent (Démarrer un listener http et un listener HTTPS, redirigez le Traffic à l'aide de règles de NAT, …), vous pouvez mettre en place une autre « solution » afin de déchiffrer le TLS. La victime peut-elle se rendre compte de l'interception ?

Bonus 3

Vous allez tester une nouvelle méthode pour faire du MITM. Arrêter le spoof ARP et renseignez-vous sur les messages ICMP Redirect.

A l'aide de l'outil hping, forgez une requête de ce type et vérifier la table de routage de la victime. Vous pouvez par exemple rediriger les requêtes DNS de la victime vers votre machine.

Attention, selon l'OS que vous utilisez pour la victime, il peut y avoir des protections contre ce type d'attaques, et dans ce cas il peut être nécessaire de modifier des paramètres du noyau.