Network sniffing

Projet

Dans ce projet, vous allez découvrir et utiliser un analyseur réseau appelé **Wireshark**. Wireshark permet de choisir une interface réseau et d'écouter les trames qui circulent sur le réseau.
Il est possible de filtrer ces trames avec différentes options que vous allez être amenés à découvrir pendant ce projet.
Il est aussi possible d'intégrer les captures dans des scripts bash avec la commande *tshark.*

Ce projet est une initiation concrète au modèle **OSI** et aux principaux protocoles réseau. Vous réaliserez une documentation qui vous servira de référence pour laa suite des projets réseaux.

Réalisez une documentation en présentant rapidement wireshark et en répondant aux questions en italiques.

Quelle est la différence entre une trame et un paquet ? Qu'est-ce que le format *pcap/pcapng ?*

De préférence, sous Linux, installez Wireshark. Lancez-le en root.

Choisissez l'interface qui connectée à Alcasar, écoutez ce qu'il s'y passe et capturez des paquets suivants :

- ➔ Paquets ARP
- ➔ Paquets UDP
- ➔ Paquets TCP

Avec Wireshark, désencapsuler les trames pour retrouver les différentes couches du modèle OSI.

Quelles sont les adresses MAC sources, les IP sources et les adresses MAC destinations, les IP destinations des données capturées ?

Référencez d'autres trames ou paquets circulants sur le réseau. Identifiez leurs protocoles et leur fonction.

Puis, cherchez les spécifications du format des messages ARP/UDP/TCP et faites correspondre les captures en hexadécimal.
Pour le TCP, essayez de trouver les paquets correspondants aux étapes de connexion entre votre hôte et un serveur. (SYN ACK FIN ...).

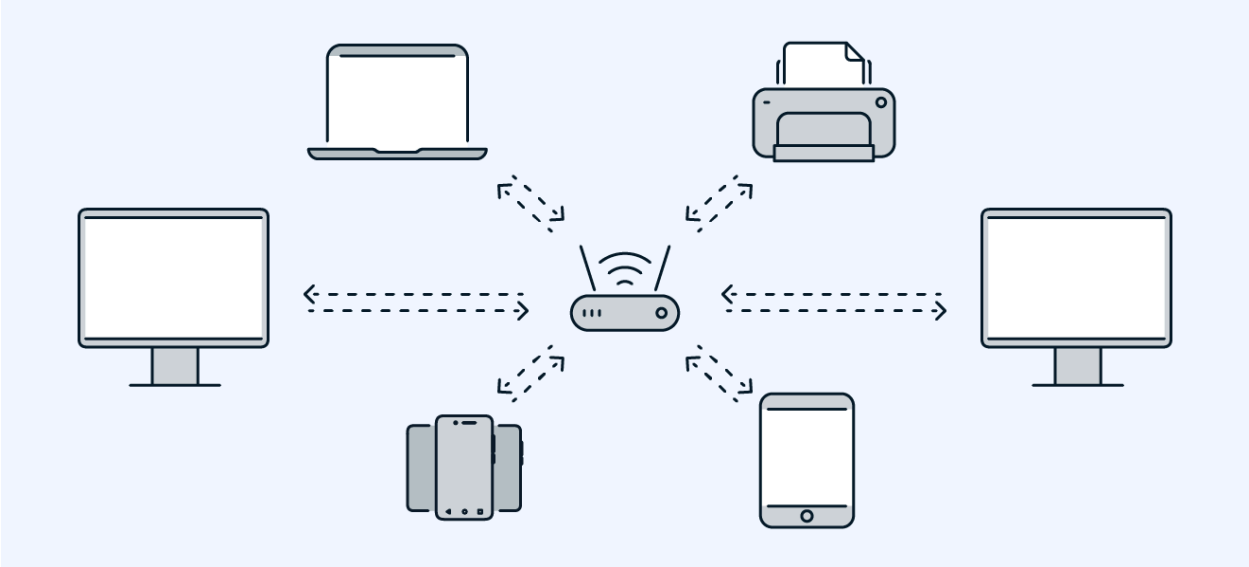
Décrivez le mécanisme de connexion avec un diagramme.

Comme vous pouvez le constater, lorsqu'on écoute un réseau, beaucoup de messages circulent. Trouver le message qui nous intéresse revient parfois à

chercher une aiguille dans une botte de foin !
Il est donc fondamental d'apprendre à utiliser les filtres de Wireshark.

Documentez-vous sur le sujet et faites quelques tests pour n'afficher que les trames qui vous intéressent.

Partie 2



Sur un réseau local (VM serveur et VM client en NAT par exemple) arrangez-vous pour installer les services qui vous permettront d'écouter *quelques-uns* des protocoles suivants :

- ➔ Message
- ➔ DHCP
- ➔ Message DNS
- ➔ Message mDNS ? (qu'est-ce que c'est ?)
- ➔ Message SSL
- ➔ Message FTP
- ➔ Message SMB
- ➔ Messages HTTPS
- ➔ Message TLSv1.2

Capturez à présent les paquets en utilisant des filtres Wireshark. Faites une sauvegarde uniquement des paquets pertinents dans un fichier au format Wireshark (cf ci-dessus).

Interprétez les paquets capturés avec les spécifications des protocoles.

En écoutant des échanges FTP sans TLS, que remarquez-vous dans les paquets ? Est-il possible de récupérer des données sensibles de connexion ? En est-il de même avec les échanges SSL ?

Partie 3

À présent, vous allez utiliser votre terminal et des scripts pour écouter le réseau et extraire les données.

La commande s'appelle ***tshark***, installez le paquet nécessaire.

À présent, reprenez la ***Partie 2*** et trouvez la commande qui vous permettra d'écouter et de capturer les paquets de quelques-uns de protocoles listés.

Expliquez les différentes options que vous utilisez dans votre documentation.

Notez qu'il est possible de filtrer vos captures de plusieurs manières :

- ➔ Directement avec des options de tshark
- ➔ En redirigeant son résultat

Compétences visées

- ➔ Administrer et sécuriser les infrastructures réseaux
- ➔ Participer à la mesure et à l'analyse du niveau de sécurité de l'infrastructure
- ➔ Participer à la détection et au traitement des incidents de sécurité
- ➔ Participer à l'élaboration et à la mise en œuvre de la politique de sécurité

Rendu

Le projet est à rendre sur votre github à l'adresse suivante :
https://github.com/prenom-nom/wireshark.
Votre travail sera évalué en présentation sans support à l'équipe

Base de connaissances

- ➔ Wireshark
- ➔ Tshark filitrage
- ➔ Hex Packet Decoder : Interprétation des trames réseau par couches et protocoles avec lectures de la data incluse dans chacune des couches.
- ➔ Base64 En/Decoder
- ➔ MD5 En/Decoder
- ➔ BIN to HEX converter