Phase 1 : La reconnaissance

- Active ou passive
- Récupérer des informations avant de passer à l'attaque :
- whois
- Google hacking :
  - https://www.google.com/advanced_search?hl=fr
  - https://www.exploit-db.com/google-hacking-database
  - https://images.google.com/
- Exif pour extraire les métadonnées d'une image
- Chercher des informations sur une photo : https://tineye.com/
- Facebook : Très gos réseau social avec énormément d'informations

- OSINT:
  - OSINT Framework : un répertoire de sites web d'outils de découverte et de collecte de données pour presque tous les types de sources ou de plates-formes.
  - Babel X Ce système de recherche international utilise l'IA pour franchir les barrières linguistiques pour n'importe quel terme de recherche. Il s'agit d'un service basé sur le cloud.
  - Google Dorks : méthode de collecte de données OSINT utilisant des requêtes de recherche Google intelligentes avec des arguments avancés.
  - Shodan : un moteur de recherche pour les appareils en ligne et un moyen d'obtenir des informations sur les faiblesses qu'ils peuvent présenter :
    Shodan: https://www.shodan.io/explore
    Filtres: https://github.com/JavierOlmedo/shodan-filters
  - Maltego : Outil OSINT permettant de collecter des informations et de les rassembler en vue d'une analyse graphique des corrélations.

- Metasploit : un puissant outil de test de pénétration qui peut trouver des vulnérabilités de réseau et même être utilisé pour les exploiter.

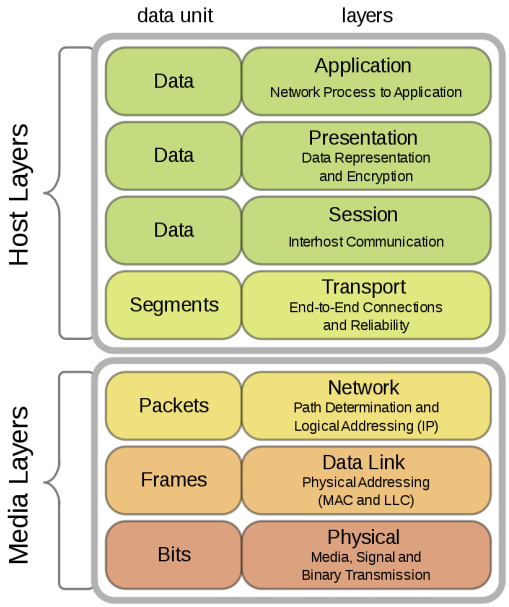
- Recon-ng : un outil de reconnaissance web open-source développé en Python et qui continue à se développer au fur et à mesure que les développeurs contribuent à ses capacités.

- Aircrack-ng : un outil de test et de craquage de la sécurité des réseaux wifi qui peut être utilisé à la fois de manière défensive et offensive pour trouver des réseaux compromis.

- Nslookup et Dig sur Linux
- Etape la plus facile, mais la plus longue

Phase 2 : Le balayage réseau

- Récupérer des détails précis sur les systèmes
- Connaître le Modèle OSI :



- Scanning réseau et différents outils :

- NMAP : Scanner et explorer des réseaux, et ainsi détecter les hôtes et les services qui y sont connectés.

Commandes basique :

- “nmap -sn 192.168.1.0/24” permet d’effectuer un balayage réseau et enverra des requêtes "ping" à toutes les adresses IP du réseau 192.168.1.0/24 et affichera une liste des hôtes actifs sur le réseau

- "nmap -sP 192.168.1.10" permet d'effectuer une analyse de ping pour vérifier si l'hôte ayant l'adresse IP "192.168.1.10" est actif et disponible sur le réseau.

- nmap <cible> : scanne une cible spécifique
- nmap -sS <cible> : scanne la cible avec une technique de scan SYN et détermine tous les ports ouverts.
- nmap -sV <cible> : scanne la cible et détermine les versions des services en cours d'exécution
- nmap -A <cible> : scanne la cible avec des options avancées telles que la détection de système d'exploitation, la détection de version et la détection de scripts
- nmap -p <port> <cible> : scanne un port spécifique sur la cible
- nmap -F <cible> : scanne rapidement la cible en utilisant un nombre limité de ports couramment utilisés
- nmap -O <cible> : tente de déterminer le système d'exploitation en cours d'exécution sur la cible

Protection : pare feu avec ufw et iptable

- Nikto : Scanneur de vulnérabilité web

- - OWASP : Découvrir des vulnérabilités WEB
- - Système de détection d’intrusion (IDS) : Snort

- Ports ouverts donc surement des **vulnérabilités présentes. On pourra donc tenter d’exploit ces ports ouverts dans la phase 3.**

Phase 3 : Gagner l'accès

- Étape clé du test d’intrusion
- On accède au système
- Faiblesse exploitée

- Diverses façon de gagner l'accès :

- Exploitation directe d'une vulnérabilité logicielle (Metasploit par exemple) :

Commandes de base :

- Pour voir la liste des modules disponibles : `show modules`
- Pour sélectionner un module : `use <nom_du_module>`
- Par exemple : use auxiliary/scanner/ftp/ftp_version qui va nous dire s'il y'a un exploit sur le ftp de metasploitable (par exemple)
- Pour voir les options disponibles pour un module : `show options`
- Pour définir une option : `set <nom_de_l'option> <valeur>`
- Pour voir les payloads disponibles une fois l'exploit réussi : `show payloads`
- Pour lancer une attaque : `exploit` ou `run`
- Pour chercher un exploit : `search`

Environnement de test d'intrusion
Décliné en plusieurs versions
Metasploit : utiliser des exploits et des payloads pour tester la sécurité d'un système

Architecture :

Rex (bibliothèque qui gère les tâches courantes (protocoles, encodage)
Msf::Core : Framework avec ses APIs
Msf::Base : APIs simplifiées pour utiliser le framework

Interfaces :

MsfConsole : accès au Framework via la console
Interface Web : accès au framework via une page web (outdated)

- Exploitation d'une faiblesse quelconque (cracking de mdp, élévation de privilèges)
- Utilisation **de logiciels espions** (keyloggers par exemple)
- Exploitation de la faille humaine (ingénierie sociale par exemple)

- Maintenir l'accès et se cacher :

- Utilisation de programmes sur le système attaqué (Post-exploitation)
- Utilisation de portes dérobées pour se facilité l'accès

- Suppression de fichiers logs/sauvegardes pour couvrir les traces

- Éléments de prévention :

- Utiliser une **politique de mot de passe** stricte
- Installer des **outils de sécurité** (antivirus, IDS,…) et les **mettre à jour**
- Faire une veille constante (vérifier les modifications sur le système)
- Faire des **sauvegardes** régulières
- Rester **vigilant**

Phase 4 : Maintenir l'acccès

- On se facilite un accès futur (le cas des backdoors)

Phase 5 : Couvrir les traces

- Destruction des preuves
- Suppression des fichiers logs