Résumé du cours "Introduction à la Sécurité des systèmes d'information - Tests de pénétration"

1. Bases du hacking et tests de pénétration

Définitions fondamentales

- **Hacking/Piratage informatique** : Manipulation des ordinateurs pour qu'ils fassent quelque chose pour laquelle ils n'ont pas été conçus
- **Piratage éthique** : Manipulation d'ordinateurs avec autorisation et/ou dans le but d'améliorer la sécurité
- **Piratage non éthique** : Manipulation d'ordinateurs sans autorisation

Tests de pénétration

- Application pratique du piratage éthique
- Objectif : Identifier et exploiter les failles de sécurité dans un environnement cible
- Méthodologie : Penser comme un criminel informatique pour anticiper les attaques
- Démarche professionnelle et sécurisée, avec documentation des vulnérabilités et évaluation des risques
- **Règle essentielle** : Les tests sont limités aux systèmes explicitement autorisés par écrit

Motivations pour un test de pénétration

- Découvrir les vulnérabilités avant les criminels
- Aider les organisations à comprendre et gérer leurs risques
- Prioriser les ressources pour atténuer les risques les plus élevés
- Produire des résultats plus concrets que les audits de sécurité passifs

Portée des tests

- Serveurs et applications réseau

- Systèmes clients et locaux
- Facteur humain (ingénierie sociale)
- Sécurité physique (portes, serrures)
- Cryptographie

2. Risques et défis professionnels

Risques professionnels

- Intrusion dans des systèmes non autorisés
- Dommages aux systèmes testés (temps d'arrêt, perte de données)
- Complications légales et réputation

Exemple concret : l'affaire Coalfire (2019-2020)

- Deux pentesters arrêtés au palais de justice de l'Iowa
- Accusations initiales : cambriolage et possession d'outils de cambriolage
- Problèmes identifiés :
    - Désaccord entre administrations sur l'autorisation des tests
    - Documentation insuffisante sur les activités autorisées
    - Ambiguïté sur les méthodes permises (tests physiques, crochetage)
- Résolution : Abandons des charges après 12 heures d'emprisonnement

3. Préparation et cadrage du test

Documents essentiels avant le test

- **Livrables** : Ce qui sera fourni au client
- **Règles d'engagement** : Définition du cadre et des méthodes
- **Document de cadrage** : Définition précise de la portée
- **Accord de non-divulgation** : Protection des informations
- **Limitation de responsabilité** : Protection juridique
- **"Carte de sortie de prison gratuite"** : Document attestant l'autorisation

Règles d'engagement - Éléments clés

1. **Programmation des tests**
    - Dates de début et fin
    - Horaires (24/7 ou heures non-ouvrables)
    - Briefings réguliers (quotidiens ou hebdomadaires)
2. **Coordonnées et communication**
    - Contacts disponibles 24/7
    - Procédures en cas de dommages
    - Procédures en cas de détection d'attaque réelle
    - Communication avec l'équipe informatique du client
3. **Approches de test**
    - **Boîte noire** : Sans connaissance préalable du réseau (comme un attaquant réel)
    - **Boîte blanche** : Avec connaissance préalable (tests plus efficaces et moins risqués)
4. **Gestion des données sensibles**
    - Procédures pour les informations personnelles identifiables (IPI)
    - Observation et documentation

Champ d'application - Définition précise

1. **Préoccupations de sécurité spécifiques**
    - Fuites de données, pannes, menaces persistantes
2. **Perspective du test**
    - Attaquant externe ou interne
3. **Systèmes concernés**
    - Systèmes à tester (noms d'hôtes, adresses IP)
    - Systèmes à exclure (trop critiques ou fragiles)
    - Équipements tiers
4. **Environnement cible**
    - Environnement de test (plus sûr mais moins réaliste)
    - Environnement de production (plus risqué mais plus pertinent)
5. **Méthodes autorisées**
    - Balayage des ports, analyses ping
    - Analyse de vulnérabilités
    - Exploitation

    - Pivotage
    - Tests physiques
    - Ingénierie sociale
    - Considérations sur les tests de déni de service

Protection légale

- Accord de limitation de responsabilité (plafonnement des dommages)
- Clauses de propriété intellectuelle
- Connaissance des lois applicables

4. Méthodologie des tests de pénétration

Étapes d'une attaque

1. Reconnaissance
2. Scanning
3. Exploitation
4. Maintien de l'accès
5. Couverture des traces

Étapes d'un test de pénétration

1. Reconnaissance
2. Scanning
3. Exploitation (+ documentation)

5. Phase de reconnaissance

Objectifs

- Rassembler des informations sur la cible à partir de sources publiques
- Comprendre l'entreprise, son secteur, ses installations, sa direction
- Identifier les systèmes potentiellement intéressants
- Découvrir les sous-réseaux IP appartenant à la cible

Méthodes

- Recherche "techno-littéraire" (sans interaction technique)
- Utilisation de Google, LinkedIn, bases de données publiques
- Documentation des employés (particulièrement les administrateurs)
- Identification des technologies utilisées

Ressources spécifiques

- **Penetration Testing Framework** : Liste de contrôle détaillée
- **OSINT Framework** (Open-Source Intelligence) : Collecte d'informations à partir de sources gratuites
- **Google Dorks** : Requêtes Google avancées pour identifier des systèmes vulnérables

Documentation

- **Inventaire des systèmes** : Méthode organisée pour documenter les découvertes
- Outils de documentation automatique : Dradis, MagicTree

6. Phase de scanning (balayage)

Objectifs

- Sondage actif du réseau
- Cartographie des adresses, topologie, systèmes d'exploitation
- Identification des services et ports ouverts
- Détection des vulnérabilités

Types d'analyses réseau

1. **Balayages/traces**
    - Envoi de sondes limitées pour identifier les systèmes actifs
    - Déduction de la topologie réseau
2. **Balayage des ports**
    - Détection des ports TCP/UDP ouverts
    - Types de scan TCP :

        - **Connect scan (-sT)** : Utilisable sans privilèges root, mais moins discret
        - **SYN scan (-sS)** : "Balayage semi-ouvert", plus discret et efficace
    - États possibles des ports TCP : Open, Closed, Filtered
    - Scan UDP (-sU) : Plus complexe car réponses moins prévisibles
3. **Prise d'empreintes digitales**
    - Identification du système d'exploitation (-O)
    - Détection des versions de services (-sV)
4. **Analyse complète** : Option -A pour combiner toutes les méthodes

Outils spécifiques

- **Nmap** : Outil principal de découverte réseau et audit
- **ZMap** : Scanner de réseau ultra-rapide pour les grands réseaux

Défis du scanning

- Compromis entre vitesse et profondeur d'analyse
- Gestion des pertes de paquets
- Adaptation aux grandes infrastructures

7. Analyse de vulnérabilité

Méthodes d'analyse

- Vérification des versions logicielles contre les bases de vulnérabilités
- Analyse du protocole utilisé
- Examen du comportement du programme
- Tentatives d'exploitation (plus risquées mais plus concluantes)

Scanners de vulnérabilité

- **Architecture** : Moteur d'analyse + plugins pour chaque vulnérabilité
- **Options de déploiement** : Externe (vue d'attaquant) ou interne (plus complet)
- **Solutions disponibles** :
    - **Commercial** : Nessus (standard industriel)
    - **Gratuit** : OpenVAS (150 000+ tests)

8. Phase d'exploitation

Objectifs de l'exploitation

- Exécuter des commandes arbitraires
- Modifier des paramètres système
- Copier des fichiers vers/depuis la cible
- Élever les privilèges

Avantages et risques pour les pentesters

- **Avantages** : Réduction des faux positifs, meilleure évaluation des impacts
- **Risques** : Crash système, perte de données, exposition à des données sensibles

Types d'exploits

1. **Exploits de serveurs**
    - Ciblage d'applications serveur vulnérables
    - Accès direct via le réseau
2. **Exploits sur le client**
    - Ciblage d'applications client (PDF, Word, navigateur)
    - Nécessite une action de l'utilisateur
    - Contourne souvent les pare-feu restrictifs
    - Exécution au niveau de privilège de l'utilisateur
3. **Exploits d'élévation des privilèges**
    - Passage d'utilisateur standard à administrateur/root
    - Exemples récents : sudo (CVE-2021-3156), polkit (CVE-2021-4034)

Framework Metasploit

- **Définition** : Plateforme pour la recherche de vulnérabilités et développement d'exploits
- **Interfaces** : msfconsole (CLI), msfrpcd (RPC), msfvenom (empaquetage), interface web
- **Modules principaux** :

1. **Exploits** : Code exploitant une vulnérabilité
2. **Charges utiles (Payloads)** : Code exécuté après l'exploit
    - Singles : Charges autonomes
    - Stagers : Première partie (communication)
    - Stages : Seconde partie (fonctionnalité)
3. **Encodeurs** : Reformatage d'exploits pour éviter la détection
4. **Auxiliaire** : Outils complémentaires (scan, DoS)

Je vais résumer l'ensemble du contenu du cours sur la post-exploitation et les tests de pénétration. Je m'assurerai d'inclure toutes les notions importantes.

Shell vs Terminal

- **Shell** : Offre uniquement entrée/sortie standard
- **Terminal** : Plus complet, inclut:
    - Jeux de caractères
    - Taille de fenêtre réglable
    - Sortie en couleur
    - Capacité de redessiner/effacer l'écran
    - Gestion des séquences de contrôle spéciales
- **Problèmes avec un simple shell**:
    - Ne fonctionne pas bien avec des commandes comme `top`, `vi`, `emacs`, `more`
    - Problèmes avec les invites de `sudo`, `su`, `ssh`, `telnet`
    - Gestion des caractères de contrôle (ex: CTRL-C) peut être problématique
- **Vérification du type d'accès**:
    - Commande `tty` sous Linux:
        - "not a tty" = shell simple
        - "/dev/XXXX" = terminal complet
- **Solutions si on n'a qu'un shell**:
    - Utiliser des commandes alternatives adaptées au shell
    - Chercher d'autres moyens d'accès
    - Démarrer un serveur SSH/Telnet
    - Obtenir des identifiants pour se connecter normalement

Inspection Post-exploitation

- **Considérations**:
  - Vérifier si les règles d'engagement permettent l'inspection du système
  - Déterminer si on peut transférer des fichiers depuis/vers le système exploité
- **Inspection réseau**:
  - Vérifier le cache ARP pour identifier les communications récentes
  - Rechercher d'autres réseaux accessibles (configuration réseau et table de routage)
- **Inspection des applications**:
  - Inventorier les logiciels installés
  - Extraire des données par application:
    - Serveur DNS: fichiers de zone
    - Serveur Web: scripts et bases de données
    - Serveur de messagerie: comptes emails

Transfert de fichiers

- **Vers la cible**:
  - FTP, SCP, NFS, SMB, Meterpreter (si le pare-feu le permet)
- **Depuis la cible**:
  - HTTP/HTTPS (plus susceptibles d'être autorisés par le pare-feu)

Mots de passe

- **Emplacements des mots de passe hachés**:
  - Linux: `/etc/shadow`
  - Windows: Base de données SAM (Security Account Manager)
- **Autres éléments à rechercher**:
  - Clés cryptographiques (SSH, PGP)
  - Identifiants Microsoft Credential Manager
  - Scripts avec mots de passe codés en dur
  - Profils de clients sans fil (clés pré-partagées)

Méthodes d'obtention de mots de passe

- **Attaque en ligne**:
  - Envoi direct des tentatives à la cible
  - Avantages: fonctionne quand aucune autre option n'est disponible
  - Inconvénients: lent, risque de blocage de comptes, détectable
- **Attaque hors ligne**:
  - Travail sur les hachages récupérés
  - Avantages: plus rapide, parallélisable, pas de blocage de comptes, moins détectable

Craquage de mots de passe

- **Éléments nécessaires**:
  - Liste de mots de passe
  - Ensemble de permutations
- **Facteurs influençant la vitesse**:
  - Taille de la liste
  - Parallélisme disponible (GPU)
  - Complexité de l'algorithme de hachage
- **Ressources pour les listes**:
  - Kali Linux: `/usr/share/metasploit-framework/data/wordlists/` et `/usr/share/wordlists/`
  - En ligne: crackstation.net (15 Go)
- **Optimisation des listes**:
  - Adapter aux politiques de mots de passe de l'entreprise
  - Créer des dictionnaires personnalisés (outil CeWL)
- **Algorithmes de hachage Linux**:
  - 1 - MD5 (le plus ancien, plus rapide à craquer)
  - 2 - Blowfish
  - 5 - SHA-256
  - 6 - SHA-512 (le plus récent, plus lent à craquer)
- **Optimisation du matériel**:
  - Utiliser des serveurs avec GPU ou ASICS

    - Location de ressources cloud

Importance des mots de passe récupérés

- Peuvent être réutilisés sur plusieurs systèmes
- Commencer le craquage immédiatement, même si le système est peu intéressant
- Après le test: documenter le temps nécessaire pour craquer chaque mot de passe
- Ne pas conserver de copies des mots de passe après la fin du pentest

Meterpreter

- **Définition**: Metasploit Interpreter
- **Caractéristiques**:
    - Interface de ligne de commande cohérente pour diverses cibles
    - Plateformes supportées: Windows, Linux, OS X, Python, PHP, Java, iOS, Android
- **Avantages en matière de furtivité**:
    - Réside entièrement en mémoire (pas de fichiers sur disque)
    - Pas de nouveaux processus
    - Injection dans un processus existant avec possibilité de migration
    - Communication cryptée (TLS)
- **Commandes de navigation**:
    - `cd`, `lcd`, `pwd`, `lpwd`, `ls`, `cat`
    - `download`/`upload` pour transférer des fichiers
    - `edit` pour modifier des fichiers sur la cible
- **Recherche de fichiers**:
    - `search`
- **Gestion des processus**:
    - `getpid`: obtenir le PID de Meterpreter
    - `getuid`: obtenir l'UID de l'utilisateur
    - `ps`: lister les processus
    - `kill`: terminer un processus
    - `execute`: lancer un autre processus
    - `migrate`: déplacer Meterpreter vers un autre processus

- **Fonctions avancées** (potentiellement hors cadre légal):
    - Capture d'écran
    - Désactivation du clavier/souris
    - Enregistrement des frappes
    - Capture webcam/microphone

Techniques de pivot

- **"Nesting"**:
    - Installation d'outils sur le système pivot
    - Connexion via bureau à distance ou SSH
    - Équivalent à "entrer directement" dans le système
    - Vérifier si l'installation de logiciels est autorisée
- **"Live Off the Land"**:
    - Ne rien installer sur la machine pivot
    - Utiliser des logiciels en mémoire (ex: Meterpreter)
    - Exécuter des commandes avec les logiciels déjà installés
    - Acheminer le trafic réseau à travers le pivot
    - Avantages: plus facile à nettoyer, plus difficile à détecter

Ce résumé couvre l'ensemble des notions présentées dans le cours sur la post-exploitation et les tests de pénétration, notamment les différences entre shell et terminal, les techniques d'inspection post-exploitation, les méthodes d'obtention et de craquage de mots de passe, l'utilisation de Meterpreter et les techniques de pivot.