Le but de ce projet est d'écrire un TP pour les deuxième année qui prendra la forme d'une enquête informatique (*digital forencics*). Dans le scénario, une entreprise se serait fait pirater, et le but du TP serait de découvrir, via un certain nombre d'indices, comment le pirate s'est introduit dans le réseau, ce qu'il a fait sur les ordinateurs et quelles données il a volées. Le tout en utilisant uniquement les connaissances d'un étudiant de deuxième année, auxquelles on ajoutera quelques compétences supplémentaires en sécu mais qui resteront simples à comprendre.

Contexte

Le réseau de l'entreprise est composé de :

* Un serveur Web (Apache+PHP sous Debian Stretch) nommé www. Il est public (accessible depuis internet) et héberge le site Web de l'entreprise. Sur ce site se trouvent deux pages PHP particulièrement importantes : ping.php et logs.php.
* Un poste de travail (sous Debian Stretch). Il n'est pas accessible depuis internet, mais en revanche il peut établir des connexions vers internet. Il est accessible en SSH depuis le réseau de l'entreprise. Il contient un certain nombre de données privées et sensibles de l'entreprise : base de données de clients, plans techniques de produits en cours de développement, liste des pots de vin distribués aux décideurs.

Les pages PHP :

* ping.php est un *looking glass* rudimentaire : il permet de lancer un ping depuis www vers une adresse IPv4 ou un nom de domaine indiqué dans une zone de texte de la page Web
* logs.php permet d'afficher les logs du serveur Web, seulement si l'utilisateur a donné le bon mot de passe

Scénario de l'attaque

Un pirate, depuis internet, a réussi à s'introduire dans le réseau de l'entreprise. Le déroulement de son attaque est le suivant :

A1) Le pirate découvre l'existence des pages ping.php et logs.php

A2) Il se sert d'une faille dans la page ping.php pour afficher le contenu du fichier logs.php

A3) Il récupère le mot de passe de logs.php et le login de l'admin de www

A4) Il se connecte en SSH sur www en utilisant le login et mot de passe précédent

A5) Il passe root sur www via une élévation de privilège

A6) Il découvre l'existence du poste de travail via un scan réseau depuis www

A7) Il découvre le mot de passe du poste de travail via une attaque de type brute-force SSH (TODO : comment découvre-t-il le login ?)

A8) Depuis le poste de travail, il exfiltre les fichiers draft\_iphone30.docx et client\_database.sql en se connectant sur un serveur FTP qu'il contrôle sur internet

A9) Il crée un script exfiltration.sh qui contient une liste de commandes scp pour copier les fichiers pots\_de\_vin\*.xlsx vers un serveur qu'il contrôle sur internet. Il exécute ce script puis le supprime

A10) Il ajoute une tâche cron qui ouvre une connexion TCP sur le port 4242 d'un serveur qu'il contrôle sur internet, tous les jours à 3h00

Sous projets

Le projet sera découpé en deux parties indépendantes et complémentaires. Chaque stagiaire va travailler sur une partie, indépendamment de l'autre. Je me chargerai de fusionner les travaux dans un sujet de TP final.

Sous-projet 1 : étapes A1 à A5 du scénario.

Sous-projet 2 : étapes A6 à A10 du scénario.

Étapes du projet

1. Suivre le MOOC de l'ANSSI : https://secnumacademie.gouv.fr/
2. Monter une maquette complète reprenant tous les éléments du scénario (pirate, www, poste, ftp-pirate, ssh-pirate, cc-pirate) en utilisant des VM VirtualBox, avec un plan d'adressage cohérent.
3. Sur la maquette, créer et configurer tous les éléments nécessaires au déroulement du scénario de l'attaque : config des serveurs web/SSH/FTP, création des utilisateurs et mot de passe, création des scripts PHP, création des fichiers exfiltrés par le pirate, etc.
4. Tester le scénario d'attaque du pirate. Si besoin, corriger la maquette et recommencer le scénario du début, et ainsi de suite jusqu'à ce que le scénario se déroule parfaitement.
5. Quand le scénario d'attaque est parfaitement rôdé, le reproduire une énième fois, et en parallèle, procéder à la collecte des traces qui serviront d'indices pour les étudiants réalisant le TP.
6. Écrire le sujet de TP. Ce dernier sera accompagné de tous les indices collectés précédemment, mais pas de la maquette (? à décider plus tard)

Indices

Au début du TP, les étudiants réalisant le TP n'auront pas connaissance du scénario de l'attaque (étapes A1 à A10). Ils seront amenés à découvrir ces étapes par eux-mêmes au fur et à mesure de l'avancement de la séance de TP. Les traces collectées seront les seuls indices fournis aux étudiants pour mener leur enquête. Pour chaque étape du scénario (A1 à A10), au moins une trace sera fournie :

A1) Logs du serveur Web (access.log)

A2) Logs du serveur Web (access.log) et contenu de ping.php

A3) Contenu de logs.php (TODO il vaudrait mieux leur donner accès à www)

A4) Logs de connexion à www (auth.log)

A5) history ?

A6) Historique des commandes, liste des paquetages installés, trace réseau (fichier .pcap lisible avec Wireshark)

A7) Logs de connexion au poste de travail (auth.log)

A8) Historique des commandes sur le poste de travail, trace réseau (fichier .pcap) permettant de reconstituer la connexion FTP. Comme ce protocole est en clair, on peut voir tous les fichiers envoyés/reçus

A9) Historique des commandes sur le poste de travail, trace réseau. Comme SFTP est chiffré, impossible de savoir quels fichiers ont été envoyés. Seule solution, voir le contenu de exfiltration.sh mais le pirate l'a effacé ! Utiliser les outils de sleuthkit pour retrouver le contenu du fichier effacé

A10) Trace réseau sur plusieurs jours montrant une tentative de connexion quotidienne, crontab

Vulnérabilités et persistances

Liste des vulnérabilités exploitées par le pirate pour réaliser son attaque.

Mauvaise validation de l'entrée utilisateur

ping.php utilise la fonction system() pour lancer la commande système ping avec le contenu de la variable target. Cette dernière est passée en paramètre du script via un formulaire HTML qui forme une requête GET. La variable target est censée contenir une adresse IPv4 ou un nom de domaine, mais …

Réutilisation de mot de passe

Le secret qui protège l'accès à logs.php est codé en dur dans le script, et les commentaires dans le code PHP indiquent le login de l'admin. Le mot de passe SSH du serveur Web www est le même que ce secret.

Mot de passe faible

Le mot de passe de poste fait partie des '500 worst passwords ever'

Élévation de privilège

sudo (tout simplement : admin peut faire sudo ; ou bien user standard mais peut faire sudo x et faille sudo cf root-me) XXX à préciser

Persistance

crontab pour lancer un *reverse shell* en utilisant netcat en mode client