UFR de mathématique et d'informatique
Université de Strasbourg

TP n° 5

Objectifs

- Comprendre le déroulement d'un test de pénétration et identifier les différentes phases le composant
- Comprendre le fonctionnement, découvrir les possibilités et mettre en œuvre le Framework de tests de pénétration Metasploit

1. Plateforme de TP

La plateforme utilisée dans le cadre de ce TP est composé de deux machines virtuelles fonctionnant sur VirtualBox placées dans un même réseau (qualifié d'interne selon la terminologie VirtualBox).

- La première est un système Kali Linux (login/mdp : root/toor) qui servira de plateforme d'attaque
- La seconde est un système Ubuntu spécialement vulnérable Metasploitable2 (login/mdp : msfadmin/msfadmin) qui servira de cible aux attaques menées à partir du système Kali.

- Le test de pénétration, CékoiDonc ?

Les entreprises manifestent un intérêt croissant pour les problématiques de sécurité. Nombre d'entre elles mettent en œuvre des démarches certifiantes garantissant la maîtrise des risques de leur système d'information.

La maîtrise des risques passe par l'identification des actifs du système d'information et ensuite des vulnérabilités présentées par ceux-ci. Si certaines vulnérabilités peuvent être envisagées et appréhendées de façon purement théorique, une compréhension technique approfondie de certaines d'entre elles sera nécessaire afin de pouvoir proposer des mesures de sécurité adaptées.

La découverte de vulnérabilités n'est pas chose aisée et nécessite des connaissances informatiques pointues en réseau, système et logiciel ainsi que de disposer d'outils performants difficiles à maîtriser. Ces qualités sont souvent l'apanage de techniciens confirmés dont le degré de compétence est difficile à évaluer. Afin de donner confiance et permettre aux entreprises de comprendre l'action de ces derniers le standard PTES (Penetration Testing Execution Standard) a été défini. Celui-ci est détaillé sur le site suivant : http://www.pentest-standard.org

Il permet ainsi aux entreprises de vérifier que le test mené se déroule de bonne façon et que le pentester agit selon les bonnes pratiques du domaine.

Le test est ainsi décomposé en sept étapes :

- Tout d'abord la phase de **préengagement** qui va permettre au pentester de présenter les modalités de déroulement du test (mode opératoire, résultats recherchés,…)
- Ensuite la phase de **collecte de renseignement**s, qui consistera à récupérer un maximum de renseignements concernant la cible de façon passive (récupération d'information sur les réseaux sociaux, par le biais d'outils comme whois, Shodan, Robtex, etc… sans alerter la cible) ou active (au risque d'éveiller l'attention de la cible) par l'usage d'outils de collecte plus agressifs du style de Nmap.

- Vient alors la phase de **détermination de la menace** qui va consister au préalable à identifier les données et services dignes d'intérêt ainsi que les motivations que pourrait avoir un attaquant à compromettre ceux-ci ainsi que les méthodes auxquelles il pourrait avoir recours pour arriver à ses fins.
- Les menaces étant identifiées on procède alors à **l'analyse des vulnérabilités** en complétant toutes les informations collectées précédemment par des tests de vulnérabilité.
- Les vulnérabilités étant identifiées, on passe alors à la phase d'**exploitation** de celles-ci par l'usage d'outils logiciels ou de scripts produits pour l'occasion permettant souvent d'automatiser l'opération.
- Après avoir compromis la cible vient alors la phase de **post exploitation** ou l'on va s'évertuer de conserver l'accès à celle-ci afin d'y installer des outils ou dérober des données utiles.
- Vient enfin l'ultime étape qui va consister en la **rédaction d'un rapport** complet afin de capitaliser le travail effectué et l'analyser afin de pouvoir mettre en œuvre de mesures de sécurité à même de réduire la vulnérabilité et par conséquent réduire le risque.

2. Vocabulaire du pentester

Le test de pénétration dispose de son propre vocabulaire nécessaire à la compréhension du domaine et notamment utilisé par Metasploit.

Exploit = moyen par lequel un attaquant tire partie d'un défaut d'un système, d'une application ou d'un service pour produire un résultat à son avantage no prévu par le développeur (dépassement de tampon, injection sql, etc…)

Payload = code que l'on veut faire exécuter par le système ciblé

type singles (autonome, dispose de tout le nécessaire pour mener à terme l'exploit sur la cible)

type stagers (permet d'établir une connexion entre le système attaquant et le système cible à exploiter ensuite avec un autre module)

type stages (une fois la connexion établie est chargé sur la cible. Contient le shell nécessaire à l'exécution de l'exploit)

Shellcode = suite d'instructions véhiculées par un payload et exécutées du la cible lors de l'exploitation

Reverse shell = payload qui initialise une connexion depuis la cible vers l'attaquant

Bind shell = payload qui attache un interpréteur de commandes à l'écoute d'un port sur la machine cible permettant notamment à l'attaquant de s'y connecter

Listener = composant qui attend une connexion entrante (souvent sur la machine de l'attaquant afin de permettre au reverse shell s'exécutant sur la cible d'établir la connexion

Encoder = programme permettant de chiffrer un payload afin de le protéger de l'action des anti-virus et des IDS

Nops = fichier de bourrage permettant de donner une taille appropriée à un payload

3. Metasploit

Metasploit est un Framework de tests de pénétration développé par la société Rapid7 et disponible sous plusieurs versions dont une gratuite.

C'est un outil à multiples facettes qui peut-être configuré et étendu avec d'autres produits (ex : Nmap, Nessus, Shodan, Armitage, etc...).

Il peut être utilisé pour mener à bien différentes étapes d'un test de pénétration : collecte d'informations (passive et active), analyse de vulnérabilités, exploitation de vulnérabilités, post exploitation et enfin production de rapport (dans sa version payante).

Une formation gratuite à son utilisation est disponible ici : https://www.offensive-security.com

a) Architecture de Metasploit

Sur le système Kali Linux utilisé pour ce TP les fichiers du Framework Metasploit sont présents dans le répertoire /usr/share/metasploit-framework/

Le répertoire précédent contient plusieurs sous-répertoires d'importance :

- /usr/share/metasploit-framework/data qui stocke des fichiers éditables utilisés pour certaines exploits (ex : dictionnaire de mots de passe,..) ;
- /usr/share/metasploit-framework/documentation qui contient la documentation d'utilisation de Metasploit ;
- /usr/share/metasploit-framework/lib qui contient les fichiers assurant le fonctionnement du framework ;
- /usr/share/metasploit-framework/modules/exploit qui contient les modules à charger pour effectuer des exploits ;
- /usr/share/metasploit-framework/modules/auxiliary qui contient les modules permettant d'effectuer des actions diverses et variées (scanners, fuzzers, sniffers, etc...) ;
- /usr/share/metasploit-framework/modules/payloads qui contient les payloads utilisable lors de l'exploit ;
- /usr/share/metasploit-framework/modules/encoders qui contient les encoders permettant de masquer les payloads aux anti-virus et ids ;
- /usr/share/metasploit-framework/modules/nops qui contient les nops permettant de donner un taille acceptable au payloads ;

b) Les interfaces d'utilisation de Metasploit

- Msfcli : interface en ligne de commande utilisée pour automatiser l'usage de Metasploit pour des attaques programmées sans interaction avec un utilisateur. Exécution grâce à la commande « msfcli ».

- Armitage : interface graphique permettant un usage assisté de Metasploit. Très utile pour observer le déroulement d'exploits, de scan, etc… et ainsi apprendre par l'observation. Exécution grâce à la commande « armitage ».

- Msfconsole : interface principale d'utilisation de Metasploit qui permet une utilisation raisonnée, progressive et complète de Metasploit étape par étape en contrôle total des actions effectuées. Exécution grâce à la commande « msfconsole».

c) Usage de Msfconsole

Msfconsole se charge via la commande msfconsole (ex : # msfconsole)

S'offre alors à nous une aide précieuse via la commande help qui va nous indiquer les commandes propres à msfconsole qui vont nous permettre d'interagir avec Metasploit (ex : msf> help)

Commandes classiques pour l'usage de msfconsole :

- back = retour en arrière vers le contexte précédent ;
- banner = affiche la bannière très originale et divertissante de Metasploit ;
- cd = change de répertoire courant ;
- color = permet d'activer/désactiver l'affichage coloré des informations ;
- connect = permet de se connecter à une machine distante qui écoute (fonctionnement identique à netcat)
- exit = permet de quitter msfconsole
- quit = permet de quitter msfconsole
- search = permet de chercher des modules par leur nom
- grep = applique la commande filtre grep à la sortie d'une autre commande
- help = affiche le menu d'aide
- info = affiche les informations spécifiques à d'un module que l'on vient de charger
- jobs = permet de voir les jobs en exécution via Metasploit (ex : listener qu'on a lancé)
- kill = permet de tuer un job
- load = permet de charger un plugin de Metasploit
- set = permet d'affecter une valeur à une variable propre au contexte dans lequel on se trouve (msfconsole, un module, …)
- setg = permet d'affecter une valeur à une variable globale à tous les contextes
- show = permet d'afficher tous les modules, encoders, nops, exploits, payloads, auxiliary, post, plugins disponible dans Metasploit
- unload = permet de décharger un plugin
- unset = permet de supprimer la valeur affectée à une variable propre au contexte dans lequel on se trouve
- unsetg = permet de supprimer la valeur affectée à une variable globale
- use = charge un module grâce à son nom (+ chemin complet)

4. Travail à réaliser afin de prendre en main Msfconsole

Exercice 1 (CHARGEMENT ET USAGE D'UN MODULE DE SCAN TCP) :

Charger le module auxiliaire de scan de port tcp
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Affecter la valeur 1-1024 à la variable correspondant aux ports que l'on désire scanner
Lancer le scan

Quels sont les ports TCP ouverts sur la machine cible ?

Exercice 2 (CHARGEMENT ET USAGE D'UN MODULE DE SCAN UDP) :

Charger le module auxiliaire de scan de port udp_sweep
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Lancer le scan

Quels sont les ports UDP ouverts sur la machine cible ?

Exercice 3 (RECHERCHE D'UNE VERSION DE SERVICE FTP) :

Charger le module auxiliaire de scan ftp_version
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Lancer le scan

Quel est le service ftp qui s'exécute sur la cible ?

Exercice 4 (RECHERCHE D'UN LOGIN ET MOT DE PASSE FTP) :

Charger le module auxiliaire de scan ftp ftp_login
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Ajouter le login et mot de passe msfadmin msfadmin au fichier /usr/share/metasploit-framwork/data/wordlists/tftp.txt
Affecter le fichier /usr/share/metasploit-framwork/data/wordlists/tftp.txt à la variable appropriée
Lancer le scan

Vérifier que la tentative pour le login/mot de passe rajouté fonctionne.

Exercice 5 (RECHERCHE D'UN SERVICE SAMBA) :

Charger le module auxiliaire de scan smb_version
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Lancer le scan

Quel est le service samba qui s'exécute sur la cible ?

Exercice 6 (RECHERCHE D'UN SERVICE WEB) :

Charger le module auxiliaire de scan http_version
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Lancer le scan

Quel est le service web qui s'exécute sur la cible ?

Exercice 7 (RECHERCHE D'UN LOGIN ET MOT DE PASSE SSH) :

Charger le module auxiliaire de scan ssh ssh_login
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Ajouter le login et mot de passe msfadmin msfadmin au fichier /usr/share/metasploit-framwork/data/wordlists/unix_users.txt
Affecter le fichier /usr/share/metasploit-framwork/data/wordlists/unix_users.txt à la variable appropriée
Lancer le scan

Vérifier que la tentative pour le login/mot de passe rajouté fonctionne.

Exercice 8 (EXPLOITATION D'UNE VULNÉRABILITÉ FTP) :

Grâce à nmap identifier les ports ouverts sur la machine cible
Identifier la version du service FTP qui tourne sur la machine cible
Démarrer Msfconsole
Chercher un exploit qui soit lié au service ftp identifié sur la machine cible
Charger l'exploit correspondant à ce service
Afficher les variables disponibles pour le paramétrage de ce module
Affecter la valeur 192.168.100.2 à la variable correspondant à l'adresse de la cible
Lancer l'attaque

Grâce aux commandes id et uname que pouvez déduire ?

Exercice 9 (UTILISATION D'ARMITAGE) :

Utiliser Armitage afin de découvrir les vulnérabilités présentées par la machine cible.
Tenter de la compromettre en utilisant les menus disponibles.