Travaux pratiques

John The Ripper

Objectifs

Il existe encore beaucoup de monde qui utilise des protocoles durant lesquels des empreintes de mot de passe sont envoyées en clair. Un attaquant peut donc utiliser un casseur de mot de passe pour faire de la recherche exhaustive sur ces empreintes. L'attaquant peut même aller plus loin en utilisant un *crawler* pour exploiter vos données personnelles et essayer de se mettre dans votre tête pour casser vos mots de passe.

Keywords and phrases Cassage de mot de passe
Digital Object Identifier TP5

1 Casseur d'empreinte

1.1 Fonctionnement de JtR

Les spécialistes de sécurité informatique parlent de casseur de mot de passe mais il s'agit plutôt de casseur d'empreinte. A partir d'un ensemble d'empreintes cibles, un casseur d'empreinte va essayer de trouver un antécédent en énumérant les valeurs d'un dictionnaire. C'est essentiellement un outil capable de faire des attaques de recherche exhaustive contre des fonctions de hachage cryptographique. L'architecture de tous les casseurs de mot de passe est toujours la même. On trouve deux programmes. Le premier moteur a pour fonction de produire des valeurs à tester. Le deuxième programme a pour but d'effectuer les opérations cryptographiques afin de créer des empreintes à tester avec les empreintes cibles.
Il existe trois grands logiciels pour casser des mots de passe :

- John The Ripper (l'outil historique),
- Hashcat (l'outil qui exploite les GPU),
- Ophcrack.

Nous allons travailler avec John The Ripper (JtR) qui est disponible sur Kali Linux. En cas de souci vous disposez de toute la documentation nécessaire sur:

http://www.openwall.com/john/doc/.

Il existe plusieurs versions de JtR: une gratuite, une payante et une communautaire (`jumbo`). Nous allons utiliser la version gratuite pré-compilée de Kali. Idéalement, vous compileriez la version communautaire pour l'optimiser pour votre processeur.

1.1.1 Utilisation basique

Pour utiliser `john` vous avez besoin deux choses: un fichier contenant des empreintes et connaître le format des empreintes.
Pour tester `john`, nous allons créer un fichier contenant une empreinte SHA256.

```
echo -n "toto" | sha256sum | gawk '{print $1}' > digest
```

Si vous voulez connaître les options disponibles avec `john`, vous disposez de l'option `-list=`. La commande suivante nous donne tous les formats supportés:

```
john --list=formats
john --list=formats | grep -i sha
```

La deuxième commande nous donne uniquement ce qui concerne SHA. Le format à utiliser est `Raw-SHA256`.
Utiliser `john` est simple maintenant:

```
john --single --format=Raw-SHA256 digest
```

`john` écrit tous les résultats d'une session dans un seul et unique fichier. Pour le consulter, il suffit d'utiliser la commande suivante:

```
john --show --format=Raw-SHA256 digest
```

Dans cet exemple, `john` n'arrive pas à retrouver le mot de passe associé à l'empreinte.

Exercice 1: Mise en application

Question 1 Quel est le format des empreintes contenues dans le fichier `EX1` ?

Question 2 Que faut-il faire en premier pour que `john` se mette au travail ?

Question 3 Compléter le fichier `format.bash` pour vous aider à trouver automatiquement le format associé à une empreinte.

1.1.2 Utilisation en mode wordlist

Nous pouvez maintenant passer au mode `wordlist` ou dictionnaire Nous allons commencer simplement en créant un dictionnaire très simple.

```
echo "toto" > dictionnaire
```

```
john --wordlist=dictionnaire --format=Raw-SHA256 digest
```

Si tout ce passe bien vous devriez voir un résultat positive.

```
john --show --format=Raw-SHA256 digest
```

Notre dictionnaire est pour l'instant très basique. Si la personne n'a pas choisi exactement `toto` comme mot de passe, notre dictionnaire ne sert à rien. On peut le voir en regardant les mots de passe que `john` a testés grâce à l'option `-stdout`.

```
john --wordlist=dictionnaire --stdout
```

`john` permet de faire du *mangling*: on applique des règles sur les mots du dictionnaire pour les dériver.

```
john --wordlist=dictionnaire --rules --stdout
```

Grâce à l'option `list` on peut voir toutes les règles de *mangling*.

```
john --list=rules
```

Par défaut, les règles appliquées sont celles désignées `wordlist` mais on peut lui indiquer d'en utiliser d'autres.

```
john --wordlist=dictionnaire --rules:all  --stdout
```

Exercice 2: We will Rockyou

`Kali Linux` dispose d'un certain nombre de *wordlist*. Vous pouvez en trouver d'autres sur https://wiki.skullsecurity.org/index.php?title=Passwords ou sur https://crackstation.net/.
⚠ Certains fichiers de *wordlist* sont énormes.

Question 4 Télécharger le dictionnaire `rockyou.txt` et tenter de casser les empreintes contenues dans `EX2`.

Question 5 Modifier le fichier `/etc/john/john.conf` afin de creer votre propre ensemble de règle de *mangling* qui s'appellera `Queenrule`. Votre derivation devra concaténer une année entre 1900 et 2016 (http://www.openwall.com/john/doc/RULES.shtml) à un mot de dictionnaire.

1.1.3 Autres modes

Le principal mode de `john` s'appelle `incremental`. C'est ce mode qui permet de faire de la recherche exhaustive.

```
john --incremental --format=Raw-SHA256 digest
```

Attention ce mode ne termine jamais par défaut ! Il est intéressant de voir que l'on peut spécifier quel jeu de caractères le mode `incremental` utilise. La commande suivante permet de lui faire générer des candidats contenant uniquement des minuscules.

```
john --incremental:Lower --format=Raw-SHA256 digest
```

Il existe d'autres modes et d'autres fonctionnalités à vous de les découvrir. Vous disposer maintenant de la connaissance des fonctions de base de `john`.

Exercice 3: Utilisation avancée

Question 6 `john` dispose d'un mode `prince` et d'un mode `markov` essayez de comprendre comment ces deux modes fonctionnent (avec l'option `-stdout` par exemple).

Question 7 A vous de jouer pour le fichier `EX3`.

2 Crawler

Il est intéressant de connaître les données personnelles de votre cible pour spécialiser le dictionnaire avant d'utiliser `john`. Utiliser un `crawler` ou robot d'indexation va vous permettre de cibler une page web de votre cible et de récupérer tous les mots qui y sont contenus. Un *crawler* est un outil qui permet d'explorer automatiquement Internet. À partir d'une *url* de base, un *crawler* va extraire tous les liens contenus et les explorer.
Si on ne donne aucune limite un *crawler* peut se mettre à explorer tout Internet. Utiliser un crawler peut entraîner des problèmes juridiques. Les sites Internet ont des conditions d'utilisation et des fichiers `robot.txt` qui définissent comment un robot d'indexation doit se comporter. Si vous ne respectez pas ces fichiers vous risquez des sanctions.
Nous allons utiliser `Cewl` durant le TP. C'est un *crawler* assez simple. Voici comment l'utiliser pour extraire des mots de la page de Jean Louis Roch.

```
cewl http://moais.imag.fr/membres/jean-louis.roch/ > dict.roch
```

On peut contrôler la profondeur de l'exploration avec l'option `-depth x` avec `x` la profondeur que l'on autorise.

Exercice 4: Carte blanche

Question 8 Vous avez carte blanche pour casser les empreintes du fichier `EX4`.