Dans le cadre de

SECURIDAY 2010

Et sous le thème de

Computer Forensics Investigation

VOUS PRÉSENTE L'ATELIER :

Techniques anti-forensics

Chef Atelier : Sellami Dhia (RT5)

- Ayari Wajdi (RT5)
- Ayari Khouloud (RT4)
- Bejaoui Cyrine (GL3)
- Ben Aissa Sana (RT3)

Présentation de l'atelier et des outils

L'atelier anti-forensics est un atelier dont le but principal est de se focaliser sur les méthodes utilisées par les pirates pour masquer leurs présences ou les modifications qu'ils ont apporté sur une machine cible.

Ainsi les pirates peuvent rendre la tache des investigateurs difficile lors de la collecte des preuves en la retardant ou la rendant impossible à détecter.

Pour ce faire, notre atelier consiste donc à implémenter des techniques ayant pour objectif de limiter les moyens d'enquête. Plusieurs moyens peuvent donc être utilisés à cette fin : détruire, camoufler, modifier des traces, prévenir la création de traces, crypter ou supprimer des données.

Les techniques qui seront présentées dans ce tutoriel sont :

- **La stéganographie :**

La stéganographie est un procédé qui permet de dissimuler des informations à l'intérieur d'une autre source de données.

Un attaquant peut donc cacher des données confidentielles sans qu'on s'en aperçoive.

La stéganographie peut être effectué sur divers support :

- Dans les images
- Dans un texte
- Dans la voix
- Dans une vidéo

- **La cryptographie** :

La cryptographie est l'art de chiffrer les messages de façon à les rendre incompréhensibles. Ce qui fait que le fichier crypté est visible mais il est illisible contrairement à la stéganographie qui consiste à *cacher le* message donc on n'est même pas tenu au courant de son existence.

Par conséquent, on peut détecter le fichier crypté mais sans réussir à assimiler son continu.

Le cryptage est souvent utilisé par les pirates afin de limiter les moyens d'enquêtes des investigateurs. Il est reconnu comme le cauchemar des investigateurs.

- **File extension :**

Consiste à modifier l'extension du fichier ou bien la forme de son contenu.

- **Wiping :**

Wipe est un nom donné à la méthode de suppression en toute sécurité. Dans les systèmes de fichier courant, les fichiers ne sont pas totalement effacés. Un moyen pour faire la suppression, est l'utilisation de Wipe, qui ne fait rien de plus que d'ouvrir le fichier et de l'écraser plusieurs fois avec un contenu pseudo-aléatoire (ou prédéfini).

- **Methode obscurity :**

Une méthode d'obscurité est utilisée pour tenter de masquer la véritable nature ou le sens de certaines données, en changeant le nom ou le contenu du fichier.

- **Methode d'encodage :**

Encodage signifie que le contenu d'un fichier est modifié d'une manière qui peut être facilement inversée. La plus part du temps, un mécanisme d'encodage simple appelé ROT13 est utilisé. (ROT moyens de rotation et 13, les caractères sont mis en rotation 13 fois.)

ROT13 ne laissent pas de signature standard. Il est surtout populaire pour masquer les données contenues dans les clés de registre de Windows.

- **Methode de compression :**

La compression permet au contenu d'un fichier d'être réduit en taille pour le stockage et la transmission. Les algorithmes de compression analysent les fichiers afin de déterminer comment la taille du fichier tel qu'il est stocké peut être réduite. Cette réduction est effectuée par l'analyse de la fréquence des données dans le fichier et en appliquant un algorithme tel que les algorithmes pour gzip, PKZIP et WinZip.

- **Alternate Data Stream :**

ADS signifie Alternate Data Stream (flux de données additionnels). Cette technique ne concerne que les systèmes de fichiers NTFS utilisé sous Windows.
Elle permet de cacher des flux de données dans un fichier totalement légitime. Ces flux peuvent être un simple fichier texte, une image ou bien du code exécutable.

- **Les slack space :**

Lors de la création d'un fichier sur le disque dur, celui-ci alloue un ou plusieurs clusters dédiés au fichier en fonction de sa taille.

Un cluster est la plus petite unité allouable sur un disque dur, il est en fait composé d'un groupe de secteurs (ce nombre dépend de la taille de la partition et du système de fichier utilisé).

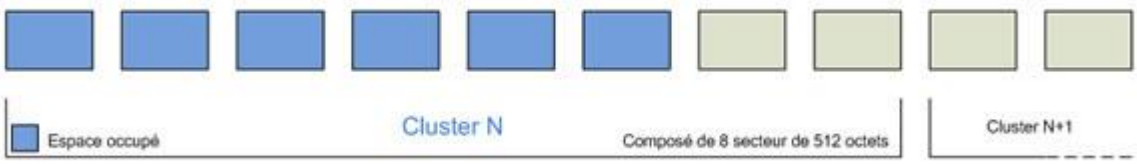
Nous prendrons ici le cas d'un disque dur dont les secteurs ont une taille de 512octets, et dont le cluster fait 4096octets (4096/512 = 8, soit 8 secteurs par cluster).

Ainsi notre système alloue un nombre multiple de 4ko pour chaque fichier que l'on stocke, ce qui, comme vous l'avez peut être deviné apporte quelques inconvénients. Tout d'abord, dans le cas où le fichier a une taille multiple de 4096, son contenu rentrera parfaitement dans le(s) cluster(s) alloué(s). Rien ici ne pose problème.

Dans le cas contraire, où notre fichier à une taille inférieure à un nombre multiple de 4096, comme dans le schéma ci-dessous, celui-ci n'occupera pas tout l'espace qui lui est dédié. Que contient donc l'espace qu'il reste ?

Celui-ci contiendra bien souvent les données qui occupaient précédemment cet espace et qui n'ont depuis pas été réécrites.

Cela peut donc servir lors de la récupération de données effacées, mais aussi dans la dissimulation de données, puisqu'il nous suffit de placer, à partir de la fin du fichier stocké dans le cluster jusqu'à la fin du cluster, des données que nous voulons cacher.



Pour bien comprendre le principe, on pourrait par exemple prendre un système d'archivage où l'on attribue des casiers en fonctions des domaines. Un casier serait dédié à un domaine x, mais le domaine x pourrait très bien être constitué seulement de quelques dossiers, ces dossiers ne rempliraient pas l'espace offert par le casier en terme de profondeur, l'espace qu'il reste est le slack space dont nous parlions plus haut.

- **Protection par Password :**

Une autre technique qui peut être utilisée par les pirates est celle de la protection par mot de passe, ainsi ses données seront protégées et il y aura pas de moyen pour voir leurs contenus.

Ces techniques seront simulées à l'aide des outils suivant :

- **StegHide :**

StegHide fait partie des outils très puissants de Stéganographie. Il permet de cacher n'importe quel type de fichier dans une image, en le compressant, et en le cryptant.

Attention, l'utilisation s'effectue en ligne de commande, mais est très simple à mettre en œuvre. Les fichiers qui peuvent servir à en cacher d'autres, doivent être des bmp, jpeg, au ou wav. De plus, l'image n'est pas modifiée de manière visible, et seul un examen approfondi pixel par pixel peut éventuellement relever un changement. Autrement dit, le risque de détection est quasiment nul.

Le principe de fonctionnement de StegHide est :

D'abord, les données secrètes sont compressées et cryptés. Ensuite, une séquence de positions de pixels dans le fichier de couverture est établie sur la base d'un générateur de nombres pseudo-aléatoires initialisé avec le mot de passe (les données secrètes seront intégrées dans les pixels à ces postes). De ces postes ceux qui n'ont pas besoin d'être changée (parce qu'ils contiennent déjà la valeur correcte par hasard) sont triés.

Ensuite, un algorithme de théorie des graphes correspondants trouve des paires de positions telles que l'échange de leurs valeurs a pour effet d'incorporer la partie correspondante des données secrètes. Si l'algorithme ne trouve pas de telles paires de plus tous les échanges sont effectivement réalisés. Les pixels aux postes restants (les postes qui ne font pas partie d'un tel couple) sont également modifiés pour contenir les données embarqué (mais cela se fait pas en les écrasant, non pas en les échangeant avec d'autres pixels). Le fait que la plupart des enrobages est fait par l'échange de valeurs de pixels implique que les statistiques du premier ordre (i.e. le nombre de fois où se produit une couleur dans l'image) ne sont pas modifiées.

Pour les fichiers audio, l'algorithme est le même sauf que les échantillons audio sont utilisés au lieu des pixels.

L'astuce est de retirer un bit à chaque octet RVB qui compose chaque pixel de l'image. En effet, en retirant 1 bit, on dégrade l'image, mais ce n'est pas visible à l'œil nu...

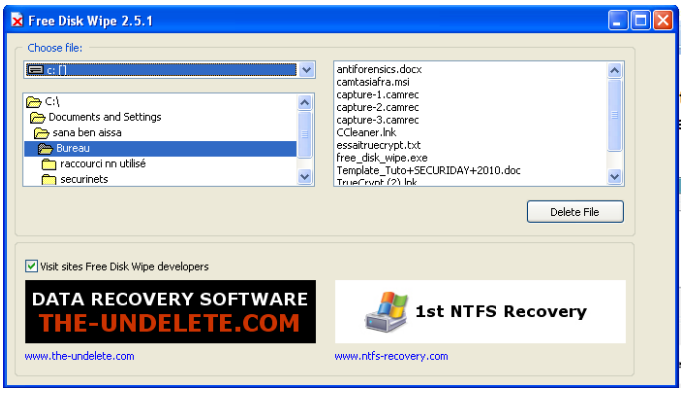
Plus récemment, les messages, transformés en longues suites de bits, sont camouflés parmi les bits d'un autre fichier, image, son, vidéo...

La méthode la plus utilisée consiste à camoufler chaque bit du message dans le dernier bit significatif de chaque point d'une image.

De ce fait, on peut récupérer ce bit à chaque fois et l'utiliser pour stocker les données que l'on souhaite. Nous récupérons donc 1/8e de la taille de l'image pour cacher un document, quel qu'il soit.

Le cryptage s'effectue avec de puissants algorithmes, parmi lesquels le performant et célèbre AES.

- **Free Disk Wipe :**



Lorsque vous effacez un fichier ou un dossier avec une commande régulière "Supprimer" dans Windows, les données réelles ne sont pas rayées du disque dur. En supposant que vous n'avez pas activé la Corbeille, ou que vous supprimez les fichiers de la Corbeille, Windows marque simplement un fichier en tant que «supprimés» dans le système de fichier sans même toucher à son contenu réel. Le contenu du fichier supprimé reste sur le disque dur, à la disposition de toute personne d'être restaurés ou lu.

Les méthodes de récupération des données des laboratoires de récupération, sont basées sur la résonance électromagnétique résiduelle qui est toujours présent sur le disque, même après que vous écrivez tous les zéros sur le contenu du fichier.

Si vous voulez vous débarrasser d'informations sensibles, vous devez alors remplacer le contenu du fichier sur la surface du disque dur, sinon, n'importe qui peut y accéder ultérieurement .Cela dit, écraser le contenu d'un fichier n'est pas suffisant pour garantir la sécurité.

Après la génération d'une séquence aléatoire de l'information et de l'écriture de cette séquence sur le contenu du fichier original, il est pratiquement impossible de récupérer le contenu d'origine du fichier, même dans un laboratoire propre. Afin de rendre la récupération, même théoriquement impossible, un processus militaire certifié conforme de la destruction des données peut être appliquée. La norme militaire spécifie l'utilisation d'une séquence cryptographique forte de nombres aléatoires qui est écrit sur le contenu original du fichier n'est pas une fois mais trois fois de suite. Ce processus de destruction de données de niveau militaire garantit l'impossibilité de récupération de données même si un Etat étranger met toutes ses ressources pour analyser votre disque dur!

Ce logiciel permet alors de nettoyer en toute sécurité toutes les traces des informations sensibles de votre disque dur tout à fait gratuitement ! Les fichiers supprimés sont alors complètement irrécupérables.

- **TrueCrypt :**

TrueCrypt est un logiciel de chiffrement, libre, gratuit et fonctionnant sous Windows. Sa force réside dans sa simplicité d'utilisation et son efficacité. Il permet de créer des volumes logiques virtuels qui prennent la forme d'un fichier ordinaire. Les fichiers qui seront stockés dans ce volume seront totalement cryptés.

Les algorithmes de chiffrement utilisé dans cet outil sont présentés dans le tableau ci-dessous.

Algorithm	Designer(s)	Key Size (Bits)	Block Size (Bits)	Mode of Operation
AES	J. Daemen, V. Rijmen	256	128	XTS
Serpent	R. Anderson, E. Biham, L. Knudsen	256	128	XTS
Twofish	B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson	256	128	XTS
AES-Twofish		256; 256	128	XTS
AES-Twofish-Serpent		256; 256; 256	128	XTS
Serpent-AES		256; 256	128	XTS
Serpent-Twofish-AES		256; 256; 256	128	XTS
Twofish-Serpent		256; 256	128	XTS

- **WinSesame :**

WinSesame est un logiciel professionnel de la sécurité vous permettant de:

  - Protéger un fichier ou un dossier avec un mot de passe.
  - Faire des fichiers ou des dossiers confidentiels.
  - Interdire l'ouverture d'un fichier ou un dossier.
  - Protéger les données sur le disque dur externe, USB, cdrom, réseau.
  - Créer un coffre-fort virtuel.
  - Faire un contrôle parental puissant.
  - Transfert de données confidentielles par e-mail.
  - Protéger les fichiers et dossiers avec un keyfile.
  - Nettoyez les disques contre la récupération de données résiduelles.
  - Fermer automatiquement tous les dossiers et fichiers protégés ouverts.

Facile à utiliser grâce à un menu principal qui contient toutes les fonctions du programme dans le menu contextuel. Le programme est flexible et s'adapte facilement vos besoins.

2. Environnement logiciel

- Free Disk Wipe (Windows)

Ce logiciel est téléchargeable de l'adresse suivante :

(http://www.the-undelete.com/wipe_remove_delete_erase.php).

- StegHide (Ubuntu)

On peut trouver les fichiers sources de cet outil à cette adresse :

(http://sourceforge.net/projects/steghide/files/steghide/0.5.1/steghide-0.5.1.tar.bz2/download).

- T r u e C r y p t (Windows)

Le lien qui permet de télécharger ce logiciel est :

(http://www.truecrypt.org/downloads).

- Change Extension (Windows)

(http://www.01net.com/telecharger/windows/Utilitaire/manipulation_de_fichier/fiches/12988.html).

- WinSesame (Windows)
  (http://www.aragonsoft.com/en/winsesame/telechargement_en.php

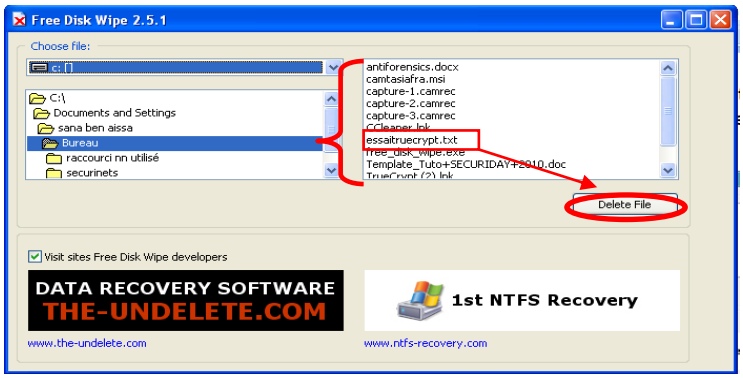
3. Installation et configuration

- **Free Disk Wipe:**

Installation:

Ce logiciel ne nécessite pas de grandes connaissances pour l'installation. C'est un fichier exécutable sous Windows donc il suffit de le lancer directement, en cliquant, sur le fichier avec l'extension « .exe ».

Utilisation :



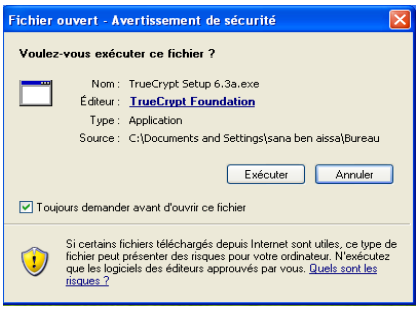
L'utilisation de cet outils est simple .Il suffit de :

Télécharger l'outil
Choisir un disque dur
Choisissez un dossier
Choisir un fichier et le supprimer avec le bouton 'Delete file'.

- **T r u e C r y p t :**

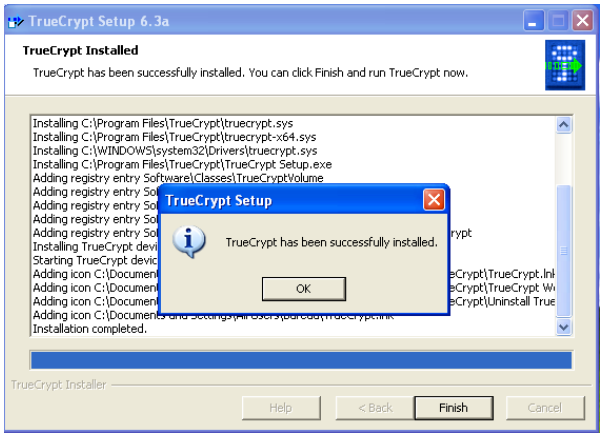
Installation:

Après avoir téléchargé le logiciel, on l'exécute et on obtient l'interface suivante:



On clique sur exécuter pour que l'installation commence. Son installation ne diffère pas sur l'installation de « FREE DISK WIPE».

L'installation se fait en suivant toutes les étapes avec le bouton « NEXT ». La bonne installation du logiciel est confirmée avec le message suivant.



Utilisation :

Dès son lancement, le logiciel ouvre un assistant proposant trois possibilités : créer un disque virtuel chiffré sous forme de fichier, chiffrer un volume sur une partition non système ou chiffrer la partition système ou le volume système entier.

Le choix de l'opération à réaliser dépendra des informations que vous souhaitez protéger : si vous cherchez simplement à protéger certains fichiers sur un disque, la création d'un volume virtuel pourra être idéale.

Le logiciel permet de déterminer facilement la taille du volume, le mot de passe associé, et de monter le volume ainsi créé pour y glisser les fichiers de votre choix.

Simple et efficace. Les autres options sont à préconiser si vous souhaitez chiffrer tout un volume, tel qu'un disque externe ou une clé USB, ou carrément le volume système.

La première étape consiste à choisir le type de volume : standard ou caché (hidden volume). La seconde option est un type particulier dont le principe est de créer un volume caché à l'intérieur du volume virtuel, en fournissant deux mots de passe : un pour le volume « extérieur » et un pour le volume caché.

Le logiciel permet ensuite de choisir l'algorithme de chiffrement. Trois algorithmes sont disponibles : AES, Serpent et Twofish. En outre, plusieurs combinaisons de ces trois algorithmes peuvent être utilisées

(AES-Twofish, AES-Twofish-Serpent, Serpent-AES, Serpent-Twofish-AES et Twofish-Serpent).

Cette étape permet également de choisir l'algorithme de « hash » parmi les trois disponibles (RIPEMD-160, SHA-512 et Whirlpool).

Les étapes suivantes consistent à choisir la taille du volume dans le cas de la création d'un volume virtuel, puis la définition du mot de passe. Lors de cette étape, il est également possible d'utiliser un fichier comme clé de déchiffrement. N'importe quel type de fichier peut être utilisé mais il est également possible d'en générer un. Dans les deux cas, la vigilance est de mise : pour un mot de passe, il faut veiller à ce que celui ci soit suffisamment complexe pour ne pas être trouvé trop facilement, mais il convient également de pouvoir le mémoriser, sans quoi le volume virtuel deviendrait inutilisable. De même, si vous utilisez un fichier comme clé, celui ci ne doit pas être perdu, ce qui semble évident, mais il ne doit pas non plus subir la moindre modification.

Il est recommandé d'entrer un mot de passe de longueurs supérieure a 20 pour qu'il ne soit pas possible de le cracker avec les techniques brute force.

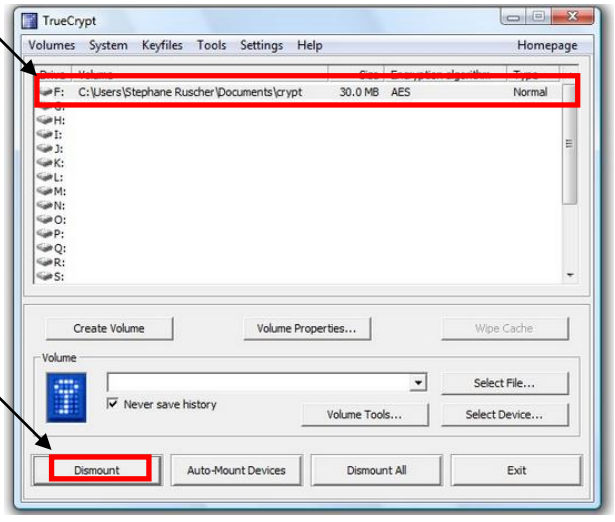
Enfin, la création du volume proprement dite propose de choisir le format du volume (FAT ou NTFS, le premier choix étant recommandé) et de passer au formatage.

Le logiciel vous propose à cette étape de renforcer la sécurité de la clé de chiffrement au moyen d'un générateur de nombre pseudo aléatoire.
Si vous créez un volume caché, il vous faudra répéter toutes ces opérations pour celui-ci, et fournir un mot de passe différent de celui que vous avez défini pour le volume « externe ».

Une fois le volume créé, vous pouvez y accéder en relançant TrueCrypt. L'interface principale du logiciel permet de monter un volume créé, et d'afficher la liste des volumes déjà montés.

Volume monté

Pour démonter le volume



L'interface principale du logiciel

Pour récupérer vos fichiers il suffit de démonter le volume créé, contenant l'information recherché.

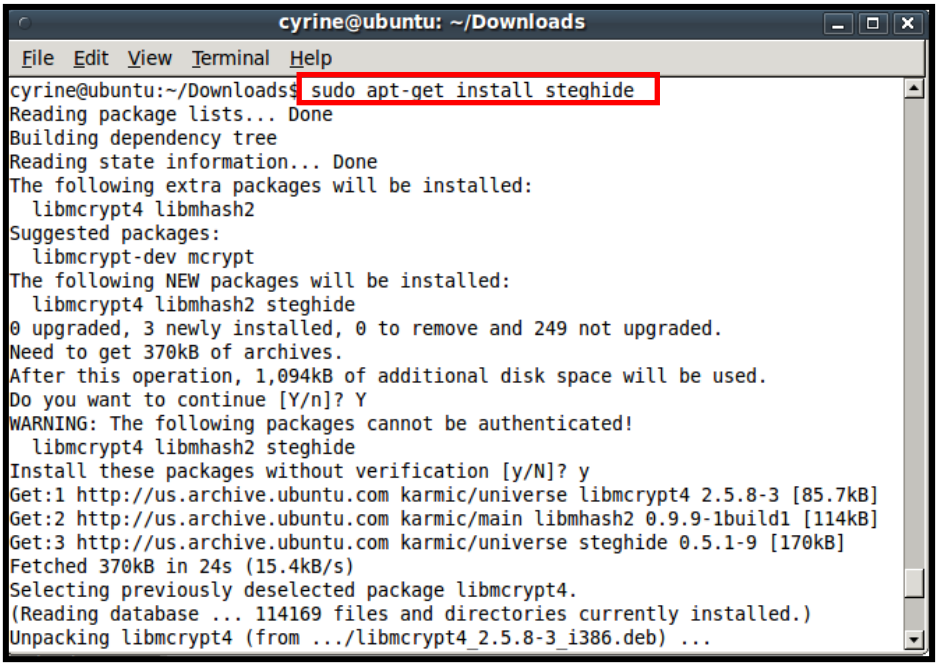
- **Steghide :**

Installation:

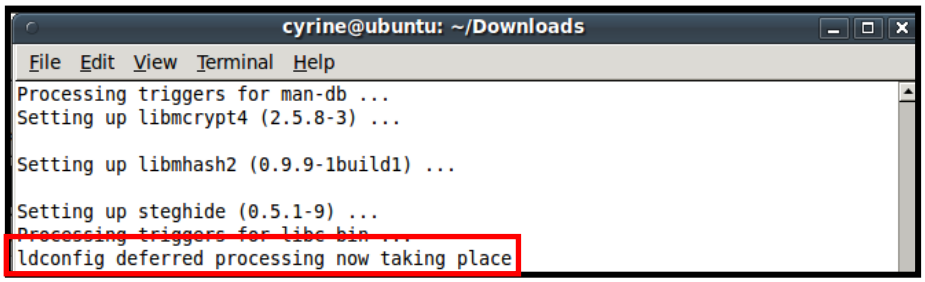
Vous pouvez facilement installer StegHide avec le gestionnaire de paquets APT :

Ouvrez un terminal sous Ubuntu en cliquant sur le menu **Applications,** sur **Accessoires** puis sur **terminal** :

Ensuite, tapez la commande suivante :



StegHide est alors prêt à être utilisé.



Utilisation :

StegHide fonctionne en ligne de commandes et c'est facile à manipuler.

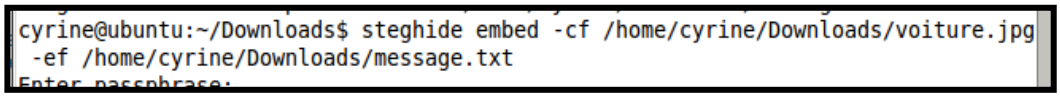
- **Cacher un élément dans une image :**

Ouvrez un terminal sous Ubuntu en cliquant sur le menu **Applications,** sur **Accessoires** puis sur **terminal.**

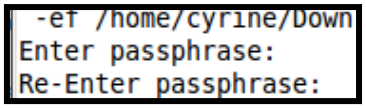
Saisissez alors la commande :

steghide embed –cf/home/cyrine/Downloads/voiture.jpg –ef /home/cyrine/Downloads/message.txt

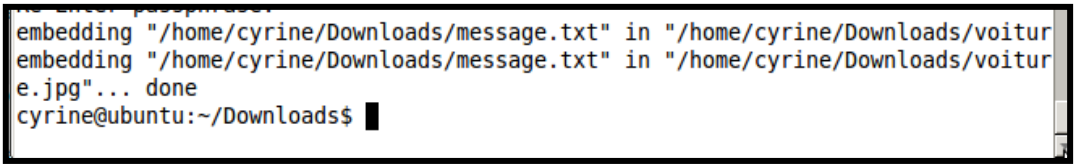
en remplaçant **/home/cyrine/Downloads/ voiture.jpg** par le chemin complet du fichier image que vous utilisez comme cachette et **/home/cyrine/Downloads/message.txt** par le chemin complet du fichier à cacher.



Le fichier caché étant crypté, saisissez un mot ou phrase secrète. Elle serait nécessaire pour récupérer le fichier. Appuyez sur Entrée et saisissez à nouveau le mot de passe.



Le fichier est alors camouflé dans l'image.



Vous pouvez alors envoyer l'image à un ami ou la mettre à sa disposition. Aux yeux de tous l'image est affichée normalement et sans aucune modification.

- **Récupérer l'élément caché sous Linux :**

Ouvrez un terminal sous Ubuntu en cliquant sur le menu **Applications,** sur **Accessoires** puis sur **terminal.**

Saisissez alors la commande :

steghide extract –sf /home/cyrine/Downloads/voiture.jpg –ef en remplaçant **/home/cyrine/Downloads/ voiture.jpg** par le chemin complet du fichier image que vous utilisez comme cachette

```
cyrine@ubuntu:~/Downloads$ steghide extract -sf /home/cyrine/Downloads/Winter.jp
g
```

Saisissez le mot ou la phrase secrète utiliser pour crypter le fichier caché et appuyez sur Entrée.

```
cyrine@ubuntu:~/Downloads$
g
Enter passphrase:
```

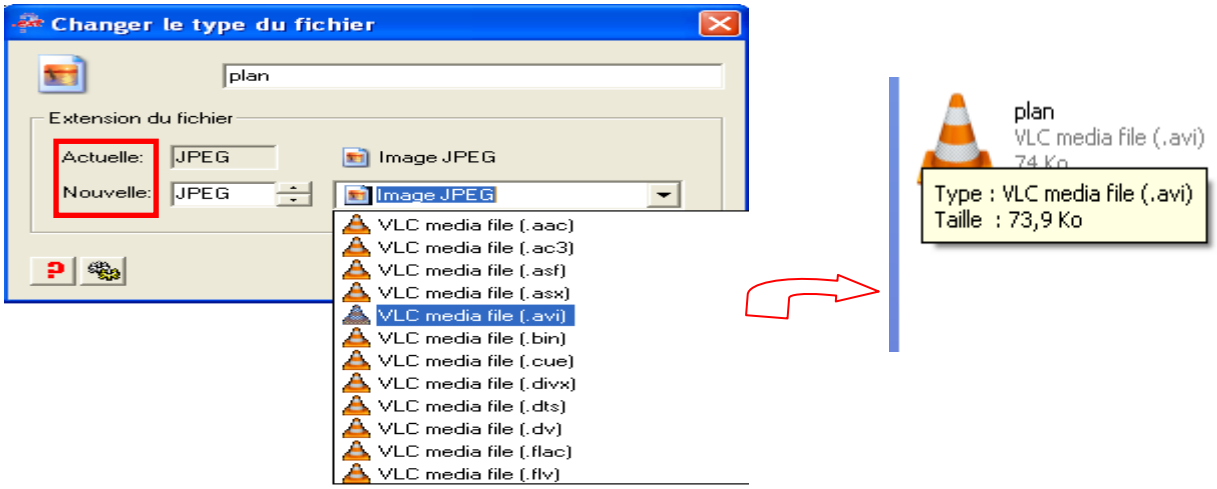
Le fichier caché est alors extrait. Et vous pouvez l'ouvrir normalement.

```
wrote extracted data to "secret.txt".
cyrine@ubuntu:~/Downloads$
```

- **Change Extension :**

Ici on va essayer de simuler la technique de changement d'extension de fichier.

Un pirate a des images qui peut l'incriminer lors d'une investigation il change son extension en utilisant un outil comme « Change Extension » :



- **Alternate Data Streams :**

Un pirate peut exécuter une application sans que son exécutable ne soit visible, ou bien il veut garder un document caché et dissimulé avec un autre fichier normal, il fait recourt à ADS (Alternate Data Streams).

La création des ADS se fait très facilement, l'exemple ci-dessous illustre la création d'un fichier rk.exe comme étant un ADS du dossier C:\Windows\ :

```
C:\> type rk.exe > C:\Windows:rk.exe
```

Pour créer un ADS, il suffit donc d'ajouter au nom d'un répertoire ou d'un fichier un « : » puis le nom de notre flux.

Maintenant nous pouvons le lancer de la façon suivante:

```
C:\> start C:\Windows:rk.exe
```

Tel qu'évoqué précédemment, un ADS peut aussi être contenu dans un simple fichier, mais rien ne change sur le plan manipulation :

```
C:\> md5sum c:\windows\notepad.exe

\ac58d8a9201d6bd43b8417f5478e3ef1 *c:\\windows\\notepad.exe

C:\> echo Exemple de texte dans un ADS >
C:\windows\notepad.exe:secret

C:\> more <c:\windows\notepad.exe:secret

Exemple de texte dans un ADS

C:\> md5sum c:\windows\notepad.exe

\ac58d8a9201d6bd43b8417f5478e3ef1 *c:\\windows\\notepad.exe
```

Nous pouvons constater que leur création ne nécessite aucun privilège et que leur présence est invisible pour l'utilisateur puisque même le hash MD5 est identique.

Sur le plan de la furtivité, certains anti-virus par exemple, ne prennent pas en charge les ADS lors du scan, ce qui permet la dissimulation de données malicieuses.

- **Les slack space :**

Voici un exemple de l'exploitation de cette technique par l'utilisation de l'outil bmap sous linux:

```
pirate@mini-moi:~$ stat /etc/passwd
  File: `/etc/passwd'
  Size: 1623      Blocks: 8    IO Block: 4096  fichier régulier
Device: 801h/2049d Inode: 3402101     Links: 1
Access: (0644/-rw-r--r--) Uid: (    0/    root)  Gid: (    0/    root)
Access: 2009-08-02 23:04:41.000000000 +0200
Modify: 2009-06-25 08:40:06.000000000 +0200
Change: 2009-06-25 08:40:06.000000000 +0200
pirate@mini-moi:~$ md5sum /etc/passwd
6d7017c1fd892e4b5a33989d33803df1  /etc/passwd
pirate@mini-moi:~$ cat secret
Ceci est un exemple de message secret!!
pirate@mini-moi:~$ sudo bmap --mode slack /etc/passwd
getting from block 13599420
file size was: 1623
slack size: 2473
block size: 4096
pirate@mini-moi:~$ cat secret |sudo bmap --mode putslack /etc/passwd
stuffing block 13599420
file size was: 1623
slack size: 2473
block size: 4096
pirate@mini-moi:~$ sudo bmap --mode slack /etc/passwd
```

```
getting from block 13599420
file size was: 1623
slack size: 2473
block size: 4096
Ceci est un exemple de message secret!!
pirate@mini-moi:~$ stat /etc/passwd
  File: `/etc/passwd'
  Size: 1623     Blocks: 8     IO Block: 4096    fichier régulier
Device: 801h/2049d Inode: 3402101      Links: 1
Access:(0644/-rw-r--r--) Uid: (    0/    root)   Gid:  (    0/    root)
Access: 2009-08-02 23:04:41.000000000 +0200
Modify: 2009-06-25 08:40:06.000000000 +0200
Change: 2009-06-25 08:40:06.000000000 +0200
pirate@mini-moi:~$ md5sum /etc/passwd
6d7017c1fd892e4b5a33989d33803df1  /etc/passwd
```

Comme nous pouvons le voir, cette méthode possède plusieurs avantages :

- Elle est simple à mettre en place
- Furtive (peut l'être encore plus en chiffrant les données que l'ont stocke et en les découpant sur plusieurs slack space différents)
- Ne modifie en rien le fichier « hôte » (la somme MD5 et les dates de modification/accès ne sont pas modifiées)

Malgré tout, comme toutes les méthodes, elle possède un inconvénient majeur : si le fichier est modifié, il y a des risques que le slack space soit réduit, ce qui pourra provoquer la corruption des données que l'on souhaite dissimuler. Pour éviter que cela se produise, il est préférable de stocker nos informations dans des fichiers dont le contenu à très peu de chance d'être modifié.

Une variante de la technique du slack space, serait d'utiliser ce que l'on appelle « unallocated space » (l'espace non alloué), c'est à dire l'espace qui n'est pas ou plus utilisé par un fichier.

Lors de la suppression d'un fichier, par exemple, l'espace occupé par le fichier est juste marqué comme non alloué, il n'est pas pour autant physiquement effacer. On peut donc écrire des données dans ces espaces.

Mais encore une fois le placement de données y est risqué, puisque la zone peut très bien être allouée par le système de fichier, mais ceci peut cependant être « contourné ».

Il est en effet possible d'indiquer les secteurs où l'on cache des données, comme étant défectueux, ainsi le système de fichier n'allouera pas ces secteurs pour y stocker des données.

- **WinSesame:**

Pour verrouiller un fichier ou un dossier, cliquez sur le verrouillage d'un dossier ou verrouillage d'une icône de fichier dans le menu principal du programme. Pour verrouiller un dossier, vous pouvez aussi cliquer à droite sur l'icône du dossier et sélectionnez "protection WinSesame" dans le menu contextuel.

Entrez et confirmez le mot de passe de votre choix pour protéger ce document (la longueur n'est pas limitée). Si le document a déjà été verrouillé le mot de passe qui lui est associés se souviendra automatiquement pour éliminer tout risque d'erreur.

Toutes les options de verrouillage peuvent être adaptés à ce dossier, ou les laissé à la valeur par défaut (vous pouvez définir dans les options du programme).

Vous pouvez choisir d'afficher le mot de passe en clair, si ce n'est pas un risque et dans ce cas activer la confirmation automatique.

Vous pouvez suivre l'état d'avancement de l'opération avec les progrès 3 bars qui vous montrent l'état d'avancement.

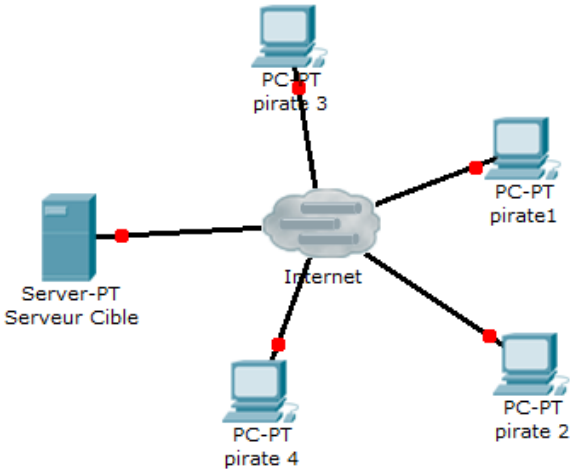
L'icône du dossier ou fichier est modifié pour indiquer que ce document est verrouillé.

Pour ouvrir un document protégé, double cliquez dessus ou cliquez sur l'ouverture d'un fichier verrouillé ou un dossier icône dans le menu principal et sélectionnez le document.

Entrez le mot de passe. Différentes options sont disponibles dans cette fenêtre (ouverture automatique, le stockage du mot de passe à la refermeture, fermeture automatique).

4. scénario de test

Dans cette partie on va monter l'utilité de nos outils dans un cas réel. On suppose qu'il y a sur le réseau des pirates qui se connaissent. Et que l'un d'eux (pirate 1) a besoin de l'aide pour faire tomber une machine cible. Voici le schéma du réseau.



Dans ce cas le pirate 1 demande de l'aide a travers l'envoi d'une image à ces amis pirates, dans laquelle il a insérer toute les informations qu'il a récolté sur sa machine cible afin qu'ils puissent l'aider à réaliser un DOS. Pour ce qui précède notre pirate a utilisé l'outil STEGHIDE.

Les figures ci-après montre le la procédure :

- L'image utilisée pour l'envoi du texte

- Le contenu du fichier texte à camoufler :

text.txt

```
cible: 175.168.2.10 : serveur Web
type d'attaque: DOS
Date: 28/04/2010
Heure: 20h00
```

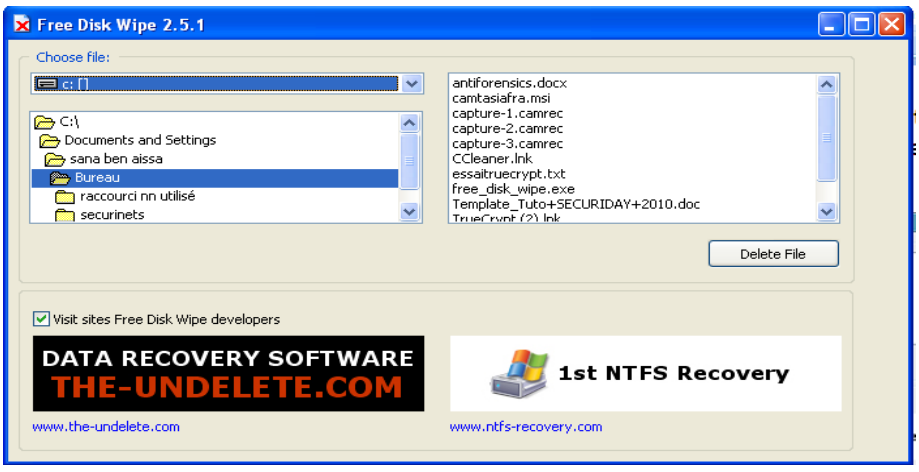
- La commande lancée par le pirate 1 pour camoufler le texte dans l'image et le protéger par un mot de passe est illustrée dans cette image :

```
pirate1@ubuntu:~$ steghide embed -cf /home/cyrine /Desktop/ImgHote.jpg -ef /home/
pirate1/Desktop/text.txt
Enter passphrase:
Re-Enter passphrase:
embedding "/home/pirate1/Desktop/text.txt" in "/home/pirate1/Desktop/ImgHote.jpg"
embedding "/home/pirate1/Desktop/text.txt" in "/home/pirate1/Desktop/ImgHote.jpg"
.. done
```

Après la réception de l'image les amis de notre pirate, connaissant déjà le mot de passe, extraire le texte de l'image.

```
pirate2@ubuntu:~$ steghide extract -sf /home/pirate2/Desktop/ImgHote.jpg
Enter passphrase:
wrote extracted data to "text.txt".
pirate2@ubuntu:~$
```

Après avoir réalisé son attaque le pirate doit effacer tout document qui permet de le lier à cet incident pour qu'en cas ou les investigateurs remontent à lui ils ne pourront trouver aucune preuve contre lui. Dans ce cadre c'est l'outil Free Disk Wipe qui le permette.



5. Conclusion

Ce tutoriel est une présentation de l'anti-forencics et de quelques outils qui le permette. En effet l'anti-forencics est un ensemble de mesure et de techniques utilisés par un internaute pour essayer d'arrêter un processus d'enquête numérique. Inversement, notre atelier a le but de vous sensibiliser des risques que chacun de nous leurs est exposés chaque jour. Les outils décrits dans ce tutoriel n'ont pas toujours un but maléfique, ils peuvent être utilisés dans la sécurité de nos données personnelles.