[Forensic] Artefacts

Introduction

De nombreux artefacts sont consultables sur les systèmes Linux pour effectuer une analyse forensique.

```
remnux@remnux:~$ cat /etc/os-release
NAME="Ubuntu"
VERSION="14.04.2 LTS, Trusty Tahr"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 14.04.2 LTS"
VERSION_ID="14.04"
HOME_URL="http://www.ubuntu.com/"
SUPPORT_URL="http://help.ubuntu.com/"
BUG_REPORT_URL="http://bugs.launchpad.net/ubuntu/"
```

Artefacts

Liste d'artefacts

Artefacts	Descriptions
/etc/*-release	Informations sur l'OS et les numéros de version
/etc/issue	Informations sur l'OS et les numéros de version
/etc/issue.net	Informations sur l'OS et les numéros de version
/etc/timezone	Fuseau horaire
/etc/localtime	Fuseau horaire

/etc/passwd	Comptes utilisateurs
/etc/group	Groupes et membres
/etc/shadow	Mots de passe hashés des comptes utilisateurs
/etc/sudoers | /etc/sudoers.d	Politiques sudo
/etc/fstab	Points de montages automatiques
/etc/hostname	Nom d'hôte de la machine
/etc/hosts	Résolution des noms de domaine
/etc/network/interfaces	Configuration réseau
/etc/resolv.conf	Configuration DNS
/var/lib/networkmanager/internal	Dernière IP attribuée au système
/var/lib/networkmanager/NetworkManager.state	Etat actuel du réseau, du wifi et de l'accès à internet sur le système
/var/lib/networkmanager/seen-bssids	Enregistre les BSSID wifi vus, mais pas nécessairement connectés
/var/lib/networkmanager/timestamps	Enregistre les baux DHCP
/var/lib/dpkg/status	Journaux d'évènements du gestionnaire de paquets DPKG (Debian / Ubuntu)
/var/lib/dpkg/status	Journaux d'évènements du gestionnaire de paquets RPM (Redhat)
/var/lib/pacman/local	Journaux d'évènements du gestionnaire de paquets PACMAN (Arch Linux)
/var/log/apt/history.log	Journaux d'évènements de ce qui a été installé avec le gestionnaire de paquets APT (Debian / Ubuntu)
/var/log/apt/term.log	Enregistre les sorties de terminal des commandes d'installations avec APT (Debian / Ubuntu)
/var/log/yum.log*	Contient les dates d'installation des paquets (Redhat)
/var/log/dnf.log*	Contient les dates d'installation des paquets dans un format difficile à lire (Redhat)

/var/log/dpkg.log*	Journaux d'évènements pour les paquets installés manuellement avec le gestionnaire de paquets DPKG (Debian / Ubuntu)
/var/log/auth.log	Connexions utilisateurs.
/var/log/btmp	Connexions échouées des utilisateurs.
/var/log/faillog	Connexions échouées des utilisateurs.
/var/log/dmesg	Périphériques matériels détectés par le kernel au démarrage.
/var/log/journal/*	Logs systèmes et services (remplaçant de syslog)
/var/log/lastlog	Dernières connexions pour chaque utilisateur.
/var/log/syslog	Logs systèmes et services.
/var/log/wtmp | /var/log/utmp	Connexions réussies des utilisateurs.

Date d'installation du système

La méthode la plus sûre est de chercher la date de création du système de fichiers, en utilisant la commande suivante pour **EXT4** :

```
tune2fs –l /dev/sdb2 | grep -i "created"
```

Et la commande suivante pour **BTRFS** :

```
btrfs subvol show /mnt/evidence/| grep -i "creation time"
```

Dernières extinctions

```
last -f /var/log/wtmp | grep shutdown
```

$PATH

Le $PATH contient les chemins vers tous les binaires exécutables. Vous pouvez afficher les chemins (*chroot* requis) :

```
echo $PATH
```

Vous pouvez aussi récupérer tous les binaires (applications), avec la commande suivante (*chroot* requis) :

```
for i in $(echo $PATH | tr -s ":" "\n"); do find $i/ -type f; done > apps.txt
```

Démarrages automatiques

Systemd

Vous pouvez consulter tous les fichiers **.service** ou **.target** ou **.socket** contenus dans les répertoires suivants :

- **/etc/systemd/system/**
- **/usr/lib/systemd/system**
- **/lib/systemd/system/**

> Vous pouvez consulter les logs systemd dans le fichier **/var/log/syslog** .

Init

Sur les vieux systèmes, init était utilisé et non systemd. Les scripts exécutés au démarrage avec init sont situés dans **/etc/init.d** .

Cron

Les tâches crontab peuvent être vérifiées dans les répertoires suivants :

- **/etc/crontab/**
- **/var/spool/cron/crontabs/**

De plus, les tâches propres aux utilisateurs peuvent être consultées avec la commande suivante (*chroot* requis) :

```
crontab -l
```

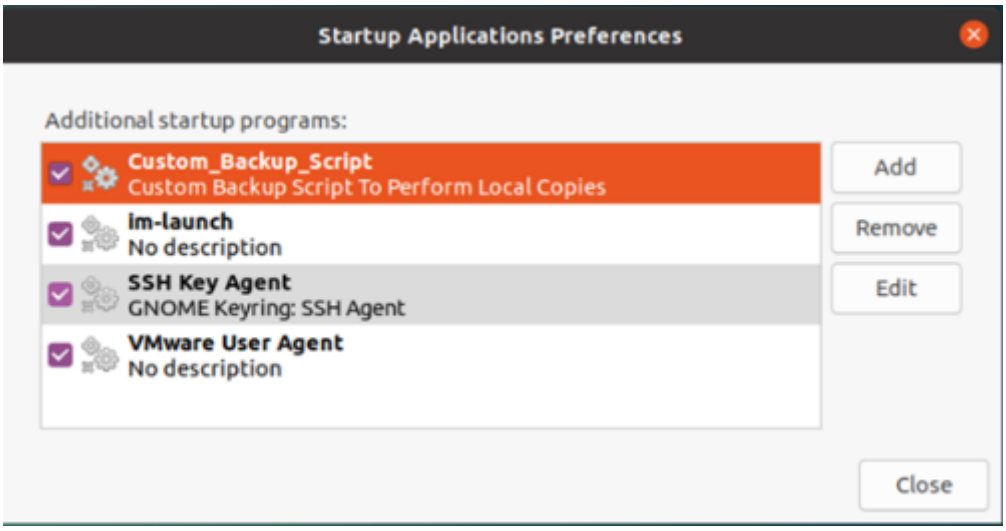
Profils shells

Ces fichiers de profil sont en réalité des scripts bash qui s'exécutent au démarrage de la session de l'utilisateur, ils sont exécutés dans l'ordre suivant :

- **/etc/profile**
- **~/.bash_profile**
- **~/.bash_login**
- **~/.profile**

GUI Startup Manager

Sur les environnements de bureau traditionnels (Gnome, KDE, XFCE), il est possible de configurer des applications de démarrage :



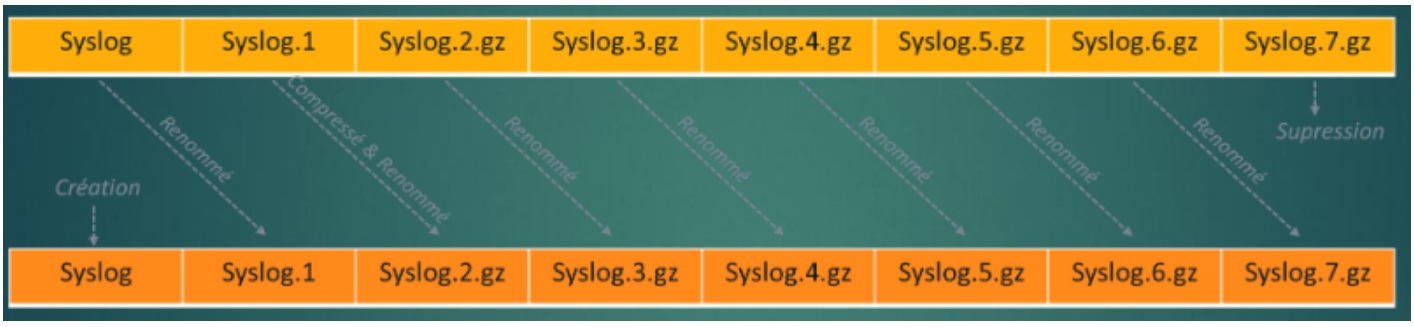
> Vous pouvez les retrouver à l'emplacement **~/.config/autostart/** .

Logrotate

La configuration de Logrotate est disponible via le fichier **/etc/logrotate.conf** ou dans le dossier **/etc/logrotate.d/** . Voici la configuration par défaut :

- weekly : rotation hebdomadaire.
- rotate 4 : conserve 4 cycle de rotation.
- create : un nouveau fichier vide après chaque rotation.

Voici un schéma explicatif du fonctionnement de Logrotate :



> Une tâche cron exécute quotidiennement logrotate, celle-ci est disponible à cet emplacement **/etc/cron.daily/logrotate** .

>

> Les status et horodatages des rotations sont disponible à cet emplacement **/var/lib/logrotate/status** .

Journaux d'évènements

> Les logs au format texte sont en train d'être remplacés par des fichiers de base de données sur les systèmes Linux, ce qui les rend plus difficile à consulter.

Vous pouvez afficher les logins réussis avec l'horodatage pour les utilisateurs du système avec la commande suivante :

```
lastlog |pr -2 -t -s | column -t
```

Vous pouvez afficher les dernières connexions de l'utilisateur courant avec la commande suivante :

```
last -F
```

Vous pouvez consulter les logs d'un système distant avec **journalctl** :

```
journalctl --directory=/mnt/evidence/var/log/journal
```

Voici quelques options pour journalctl :

Commandes	Descriptions
journalctl -o short	Affichage de l'horodatage par défaut [ex: Aug 03 02:43:12]
journalctl -o short-full	Affichage de l'horodatage au format ANSI [ex: Wed 2022-08-03 02:43:12]
journalctl -o short-iso	Affichage de l'horodatage au format ISO [ex:2022-08-03T02:43:12-0700]
journalctl -o short-unix	Affichage de l'horodatage au format Integer [ex: 1659519792.935000]
journalctl -o verbose	Affiche le détail complets des champs
journalctl --no-hostname	Enlève le champ du nom d'hôte pour améliorer la lisibilité
journalctl -a	Affiche l'ensemble des champs des journaux
journalctl -r	Affiche les logs en sens inverse
journalctl --list-boots	Liste les démarrages
journalctl -u <SERVICE>.service	Information sur un service
journalctl _UID=<UID>	Informations sur l'utilisateur (avec son id)
journalctl -k | grep -i USB	Informations kernel sur les périphériques USB

Home

Les répertoires homes des utilisateurs contiennent généralement les artefacts les plus intéressants. Par défaut ils sont situés dans :

- **/home/<USER>** : Pour les utilisateurs
- **/root** : Pour le compte root

Voici des artefacts contenus dans les répertoires home qui pourraient vous intéresser :

Artefacts	Descriptions
.bashrc	Script exécuté à chaque nouvelle session shell. Il contient généralement les configuration du shell, des fonctions, des variables et les alias.
.bash_logout	Script exécuté à chaque fermeture du shell.

Bash_history

Ce fichier est présent dans le home de l'utilisateur et stocke les commande exécutées par celui-ci.

Voici quelques éléments à prendre en considération:

- Il n’enregistre pas les horodatages par défaut (**$HISTTIMEFORMAT**).
- Il peut être manipulé / modifié / supprimé.
- Il ne contient pas les commandes des shells en cours.
- Il peut être situé n’importe ou sur le système (**$HISTFILE**).
- En ajoutant un espace devant la commande, la commande n'est pas enregistrée (**$HISTCONTROL**).
- Le fichier a une taille maximum et un nombre de ligne limité. (**$HISTFILESIZE**) et (**$HISTSIZE**).

Éléments récents

Sur un système avec un environnement de bureau, vous pouvez retrouver l'équivalent des jumplists pour windows via le fichier **~/.local/share/recently-used.xbel** :



Il s'agit d'un fichier “xml” contenant le nom des fichiers accédés avec l'interface graphique. Il inclue d'importantes informations telles que :

- Le nom du programme.
- La date et heure.
- L'emplacement et le nom du fichier.

Corbeille

Sur un système avec un environnement de bureau, vous pouvez retrouver la corbeille à l'emplacement **~/.local/share/Trash** . Cependant, seulement les fichiers supprimés avec l'interface graphique seront présents.

Deux sous répertoires sont à étudier :

- **files/** : Contenant le fichier original.
- **info/** : Contenant les informations de suppression (date et chemin).

Navigateurs internet

Voici les emplacements des profiles utilisateurs selon les navigateurs :

Navigateur	Chemin
Firefox	~/.mozilla/Firefox/
Chrome	~/.config/google-chrome/
Opera	~/.config/opera/
Vivaldi	~/.config/vivaldi/

Timeline

Pour générer une timeline sur les évènements passés sur un système de fichiers, plusieurs solutions s'offrent à vous :

- La super timeline plaso avec **log2timeline** (lente à générer mais très performante).
- La timeline de **sleuthkit** (rapide à générer).

Sleuthkit

Pour générer une timeline avec sleuthkit, commencez par créer un bodyfile :

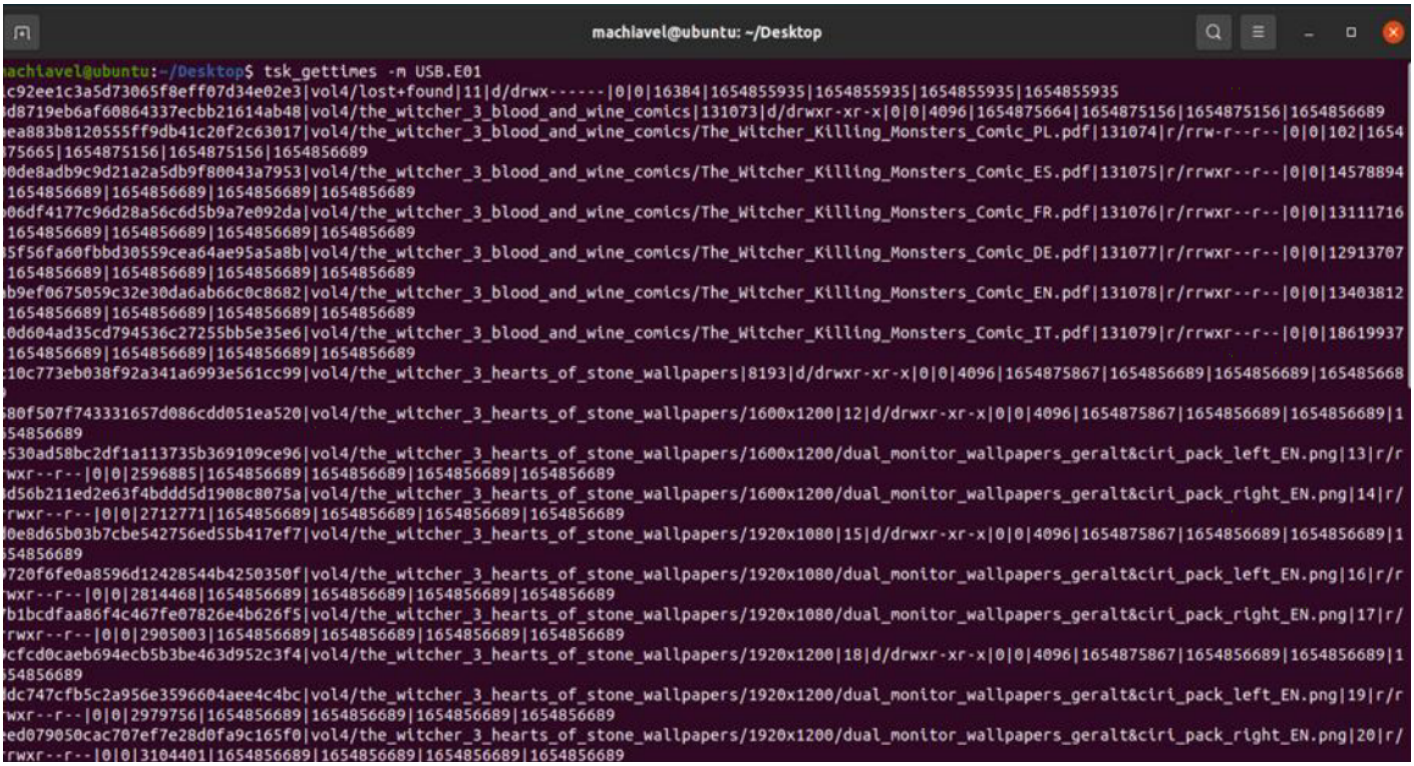
```
tsk_gettimes -m image.E01 > bodyfile
```

Chaque ligne contient les informations suivantes :
MD5|nom|inode|mode_en_chaîne|UID|GID|taille|atime|mtime|ctime|crtime .

Voici le détail des informations :

- **MD5** : MD5 du fichier.
- **nom** : Chemin complet du fichier.
- **inode** : Identifiant unique du fichier dans le système de fichiers.
- **mode** : Permissions UNIX.
- **UID** : Identifiant de l'utilisateur propriétaire du fichier.
- **GID** : Identifiant du groupe propriétaire du fichier.
- **taille** : Taille du fichier.
- **atime** : Heure du dernier accès au fichier.
- **mtime** : Heure de la dernière modification du fichier.
- **ctime** : Heure du dernier changement de métadonnée du fichier.
- **crtime** : Heure de création du fichier.

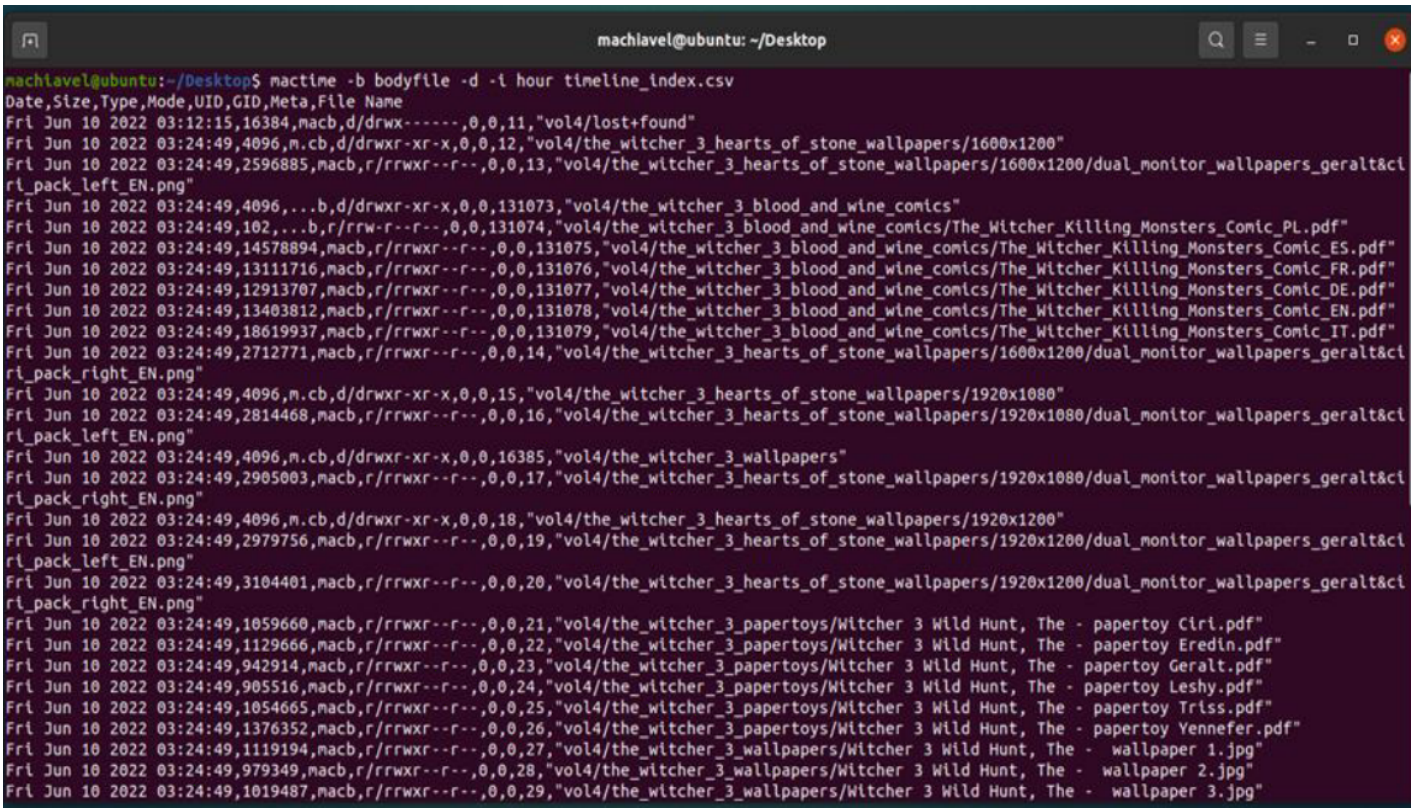
Voici un exemple de ce à quoi ressemble le bodyfile :



Ensuite, il vous faut convertir votre timeline en CSV :

```
mactime -b bodyfile > timeline.csv
```

Voici à quoi ressemble le fichier CSV généré :



Plaso

Tout d'abord créez votre bodyfile :

```
log2timeline.py --storage_file <bodyfile> <image.E01>
```

> Le fichier de sortie est une base de donnée SQLite.

Puis transformez votre bodyfile en timeline :

```
psort -w <timeline> <bodyfile>
```

Visualiseurs

Une fois la timeline générée, il faut utiliser des outils de visualisation pour effectuer une analyse. Pour cela, vous pouvez utiliser :

- **Timesketch** (https://timesketch.org/)
- **Timeline Explorer** (Eric Zimmerman)
- **Glogg** (http://glogg.bonnefon.org/index.html)

Revision #7
Created 2025-03-08 09:39:48 UTC by Elieroc
Updated 2025-07-03 13:24:27 UTC by Elieroc