

## Récupération d'audio et de vidéos sous Linux (Ubuntu)

=====

Voici quelques outils pour mener à bien le rip d'audio et de vidéos sur Linux.

=====

### 1. Audio rip

=====

Le rip d'audio est une chose aisée sous Linux, et en particulier sous Ubuntu.

Installer d'abord audacity et pavucontrol :

```
$ sudo apt-get audacity pavucontrol
```

- lancez audacity et déclencher l'enregistrement puis faite pause juste après
- appuyez sur alt + F2, tapez pavucontrol et lancez-le
- dans l'onglet enregistrement de pavucontrol, audacity apparait
- choisissez "Monitor of Audio interne Analog Stereo" dans ALSA Capture from / onglet Enregistrement
- lancez votre de site de musique préféré
- pendant la lecture audio sur votre site, enregistrez dans audacity
- exporter en mp3

Cette méthode prenant comme source le son généré par la carte son, les musiques provenant de n'importe quel site sont enregistrables.

Note : une fois pavucontrol configuré comme ci-dessus, il n'est plus nécessaire de le lancer. L'enregistrement via audacity prendra la bonne source directement.

=====

### 2. Vidéo rip

=====

Le rip de vidéos est plus compliqué. Les rips "enregistrement d'écran" sont pour l'instant assez lourds et le résultat n'est pas très bon. Il faut donc faire autrement, et ne pas avoir peur de mettre un peu les mains dans le camboui, c'est l'occasion d'apprendre à se servir de quelques outils et de comprendre le fonctionnement de certains sites.

Le premier réflexe que vous devez avoir est d'examiner la source. Pour cela, il faut regarder dans le code HTML de la page web qui affiche la vidéo. Parfois, c'est un simple lien vers un fichier. Dans ce cas, le fichier est récupérable par la commande wget :

```
$ wget http://site.com/video.flv
```

... le fichier video.flv est téléchargé.

Mais bien souvent, il s'agit de flux, c'est-à-dire de données qui arrivent au fur et à mesure. Il faut les identifier et les réassembler. Le protocole utilisé dépend du site. Ces protocoles ne sont pas du HTTP, mais, principalement dans le cas des vidéos, du RTMP ou du MMS. Comme pour le HTTP cependant, il existe une url pour s'y connecter. Il faut ensuite utiliser le bon outil pour lire le flux et l'enregistrer dans un fichier.

Dans la plupart des cas vous verrez que le player est un programme flash (fichier SWF) auquel est passé un certain nombre de paramètres. Commencez par examiner ces paramètres dans le code source HTML. Votre objectif est de récupérer une url commençant par rtmp:// ou mms://. Si vous ne trouvez rien de probant dans le code source, votre meilleur ami (comme dans bien d'autres cas) est l'extension FireBug de Firefox (<https://addons.mozilla.org/fr/firefox/addon/1843/>). Une fois la vidéo lancée, activez FireBug et regardez les requêtes lancées (onglet Réseau, puis onglet HTML de FireBug) : vous trouverez des informations sur les requêtes passées, leur adresse, et le résultat retourné. Soyez curieux, regardez l'enchaînement des requêtes par rapport au démarrage de la vidéo, inspectez le résultat des requêtes via FireBug, connectez-vous aux url récupérées directement, ...

Une fois l'url du flux récupérée, il faut pouvoir télécharger le fichier vidéo. Voici les outils utilisés pour récupérer ces flux sous forme de fichier :

#### **\* RTMP / flvstreamer**

- installer flvstreamer :  
\$ sudo apt-get install flvstreamer
- puis lancer la commande  
\$ flvstreamer -r rtmp://url/fichier -o fichier.flv  
... le fichier fichier.flv est téléchargé.

Note : en cas de coupure de connexion, vous pouvez reprendre le téléchargement en ajoutant --resume à la ligne de commande.

#### **\* RTMP / rtmpdump**

- \$ rtmpdump -r "urlRTMP" --swfVfy "urlplayer/player.swf" -o fichier.flv  
(notez que dans la plupart des cas l'url du player est facile à trouver dans le code HTML de la page)

#### **\* MMS / mimms**

- installer mimms :  
\$ sudo apt-get install mimms
- puis lancer les commandes :  
\$ mms="url/fichier"  
\$ mimms \$mms  
... le fichier fichier.flv est téléchargé.

#### **\* MMS / msdl**

```
- installer msdl
  $ wget http://downloads.sourceforge.net/msdl/msdl-1.2.3.tar.gz
  $ tar xvzf msdl-1.2.3.tar.gz
  $ cd msdl-1.2.3/
  $ ./configure
  $ make
  $ sudo make install
- puis lancer les commandes :
  $ msdl <mms://url>
  ... le fichier est téléchargé.
```

### **\* Brute-force / get-flash-videos**

C'est un autre outil pour récupérer des vidéos gérées par des lecteurs flash :

<http://code.google.com/p/get-flash-videos/>  
(onglet download)

puis :

```
$ get_flash_videos urlsite
```

C'est un outil qui fonctionne par brute-force, c'est-à-dire qu'il tente par essai/erreur des adresses de récupération. Son efficacité est donc assez aléatoire.

### **\* Analyse de flux / tcpdump**

Si vous n'arrivez pas à trouver l'adresse du flux, un autre outil, très pro, peut vous être utile. Il s'agit de tcpdump, qui est capable de dumper l'intégralité des données arrivant sur votre poste, en fonction de l'adresse, du port, ...

Dans le cas d'un flux RTMP (port 1935), il suffit de lancer la commande :

```
$ sudo tcpdump -i eth0 -p -s 0 -w file.tcpdump -v tcp src port 1935
```

... puis lancez la vidéo. Les données entrantes vont être enregistrées dans le fichier file.tcpdump. Après une vingtaine de secondes, interrompez tcpdump (CTRL+C).

Faites ensuite une recherche dans ce fichier, avec des mot-clés pertinents (player, rtmp, tmp, mp4, flv, config, http, ...) :

```
$ sudo tcpdump -A -n -s 0 -r file.tcpdump | grep motclé
```

... avec un peu de chance, vous obtiendrez l'url du flux (que vous utiliserez ensuite avec les outils décrits ci-dessus)

### **\* Techniques d'obfuscation**

Il n'est pas rare que le player flash et ses paramètres n'apparaissent pas directement et clairement dans le code HTML, et que leur code HTML soient générés à partir d'un script javascript. On parle d'obfuscation du code. Dites-vous d'abord

que puisque votre navigateur est capable d'aller jusqu'à la vidéo, les informations nécessaires pour trouver le chemin sont forcément transmises jusqu'à votre poste. Elles sont côté client.

L'extension WebDeveloper de Firefox (<https://addons.mozilla.org/en-US/firefox/addon/60/>) est de manière générale très pratique. Elle vous permettra entre autre de voir le code HTML généré plutôt que le code HTML brut de départ (menu Voir Source / Voir le code source généré). Il est possible que cela suffise. Dans le cas contraire, repérez le code javascript obfusqué (en gros c'est un code bourré de caractères hexa sans queue ni tête) et utilisez un désobfuscateur javascript (<http://www.dcode.fr/javascript-unobfuscator-desobfuscation>) et un beautifier de code, pour le rendre plus lisible (<http://www.javascriptbeautifier.com>). Un code obfusqué est un code qui cache quelque chose. Sûrement ce que vous cherchez ;). Analysez le résultat (l'écriture de code HTML par javascript est fait par la commande document.write, mais un peu de recherche sur la syntaxe de javascript n'est pas un luxe - c'est un langage plutôt simple et il existe pleins de tutos à ce sujet). Et n'oubliez pas FireBug, qui peut vous rendre service aussi dans ce cas.

### **\* Un exemple**

Voici un exemple d'exploitation de ces outils sur le site de Arte.

Allez sur le site et attendez qu'il se connecte à une vidéo. Une fois la vidéo démarrée, affichez le code source et cherchez la balise <div id="playerContainer">. Un peu en-dessous de cette balise se trouve des balises <param> associées à une balise <object>. Il s'agit d'une vidéo lue par un player flash. Remarquez le lien indiqué dans la balise <param name="movie">. Des paramètres sont envoyés au player SWF. On peut penser qu'il s'agit d'un lecteur générique qui lit ces paramètres pour savoir où trouver la vidéo. C'est en général souvent le cas. Si on est capable de lire ces paramètres, alors on aura sûrement l'url du flux.

Activez FireBug et allez dans l'onglet "Réseau" puis "HTML". Après une rapide analyse des requêtes, on trouve un GET qui récupère un fichier XML (je vous laisse découvrir lequel). Les fichiers XML sont très intéressants car ils stockent en général des paramètres - et c'est bien ce que nous cherchons. Dans ce fichier XML (onglet "Réponse"), nous voyons les balises suivantes :

```
[...]
<videos>
  <video lang="de" ref="http://videos.arte.tv/de/do_delegate/videos/karambolage-3524864,view,asPlayerXml.xml"/>
  <video lang="fr" ref="http://videos.arte.tv/fr/do_delegate/videos/karambolage-3524866,view,asPlayerXml.xml"/>
</videos>
[...]
```

Cela ressemble fort à deux nouveaux fichiers XML de paramétrage (un par langue). Ouvrons le fichier XML fr en allant à l'url : [http://videos.arte.tv/fr/do\\_delegate/videos/karambolage-3524866,view,asPlayerXml.xml](http://videos.arte.tv/fr/do_delegate/videos/karambolage-3524866,view,asPlayerXml.xml)

Nous voyons :

```
[...]
<urls>
```

```

        <url quality="sd">
            rtmp://artestras.fcod.llnwd.net/a3903/o35/MP4:geo/videothek/EUR_DE_FR/arteprod/A7_SGT_ENC_06_042975-015-
A_PG_MQ_FR?h=a9bc4a1cae38b836592221460e992fbf
        </url>
        <url quality="hd">
            rtmp://artestras.fcod.llnwd.net/a3903/o35/MP4:geo/videothek/EUR_DE_FR/arteprod/A7_SGT_ENC_08_042975-015-
A_PG_HQ_FR?h=4da891f1cbcc4c322483fc06a5ad5824
        </url>
</urls>
[...]
```

Bingo ! L'adresse du flux pour cette vidéo (en qualité HD) est donc :

```
rtmp://artestras.fcod.llnwd.net/a3903/o35/MP4:geo/videothek/EUR_DE_FR/arteprod/A7_SGT_ENC_08_042975-015-A_PG_HQ_FR?
h=4da891f1cbcc4c322483fc06a5ad5824
```

(Notez le hash en paramètre de l'url. Cela signifie que la connexion à cette url est identifiée. Par chance nous avons l'information complète, adresse + hash, mais un système de protection plus perfectionné n'aurait pas laissé le hash dans le XML : il l'aurait transmis via un autre paramétrage, éventuellement obfusqué, le tout assemblé par le player swf. Cela aurait un peu plus compliqué notre tâche, d'autant plus si ce hash avait été calculé dans le player lui-même en fonction d'une donnée synchronisée avec le serveur. Il aurait alors fallu identifier et récupérer cette donnée puis décompiler le player SWF pour obtenir la méthode de calcul. Plus de recherche montre que ce hash ne fournit pas une unicité à la validité de la requête. La protection aurait pu être plus élaborée. Pour les plus curieux, vous trouverez un décompilateur flash ici : <http://www.showmycode.com/>).

Il n'y a plus qu'à utiliser flvstreamer pour lire les données RTMP et les sauver dans un fichier :

```
$ flvstreamer -r "rtmp://artestras.fcod.llnwd.net/a3903/o35/MP4:geo/videothek/EUR_DE_FR/arteprod/A7_SGT_ENC_08_042975-
015-A_PG_HQ_FR?h=4da891f1cbcc4c322483fc06a5ad5824" -o fichier.flv
... et le fichier est téléchargé !
```

Note1 : il s'agit d'un exemple didactique, car dans le cas de Arte, il existe un outil de récupération automatique des vidéos pour Ubuntu : [http://doc.ubuntu-fr.org/arte\\_7\\_recorder](http://doc.ubuntu-fr.org/arte_7_recorder)

Note2 : tout cela est-il légal ? Autant qu'un magnétoscope ;)

=====

Il se peut que des coquilles se soient glissées dans ce tuto. Merci de m'en informer pour que je corrige ;) Have fun !

```

.-.-.
: .--':_;      :_;
: :      .-...-- .- .-.-. .-.-. .-.-.
: :_ : : : ..': : ' .; ; ' ..' .; :
'. _.' :_; :_; :_; \_ _/_ _.' \_ _.'
      c.soleares at gmail.com
```