

~~CONFIDENTIEL DÉFENSE~~

~~SPECIAL FRANCE~~



Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

PREMIER MINISTRE

Secrétariat général de la
défense et de la sécurité
nationale

Agence nationale de la sécurité
des systèmes d'information

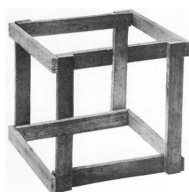
DÉCLASSIFIÉ
par décision n°15699/ANSSI/SDE/ST/LAM
du 18 juillet 2018



DOCUMENTATION CLIP

1102

GÉNÉRATION D'UN SUPPORT D'INSTALLATION CLIP



Ce document est placé sous la « Licence Ouverte », version 2.0 publiée par la mission Etalab

ANSSI, 51 boulevard de la Tour Maubourg, 75700 Paris 07 SP.

~~CONFIDENTIEL DÉFENSE~~

Résumé

Ce document présente la méthode pour générer un support d'installation CLIP (CD-ROM ou périphérique USB) à partir d'un poste de développement. Le support ainsi généré permettra l'installation des versions du système CLIP pour des passerelles (CLIP_GTW) ou pour des postes clients (CLIP_RM).

DOCUMENT DE TRAVAIL

HISTORIQUE

Révision	Date	Auteur	Commentaire
1.4.1	21/08/2017	Nicolas Godinho	Procédure de génération <i>ex nihilo</i> d'un installateur CLIP et mise à jour de la procédure de création d'un support d'installation USB
1.4	23/07/2013	Benjamin Morin	Révision complète du document pour actualisation de la procédure de génération.
1.3.1	18/09/2008	Vincent Strubel	Correction de la référence CLIP_DCS_12007.
1.3	29/08/2008	Olivier Levillain	Mise à jour de la génération des miroirs et ajout de la génération des clés amorçables.
1.2	04/08/2008	Olivier Levillain	Simplification de la procédure de génération.
1.1.1	30/07/2008	Vincent Strubel	Convention plus lisible pour les références.
1.1	27/06/2008	Vincent Strubel	Compléments sur la génération des miroirs et sur le partage des rôles.
1.0	26/06/2008	Olivier Levillain	Version initiale.

Table des matières

1	Introduction	5
2	Génération des miroirs	5
2.1	Suivi de version des paquetages et structure des miroirs	5
2.2	Génération des miroirs	6
3	Génération du <i>squashfs</i> de l'installateur	7
3.1	Environnement de l'installateur	7
3.2	Génération de l'installateur <i>ex nihilo</i>	7
3.3	Génération de l'installateur à partir d'un installateur existant	8
3.3.1	Outils de modification d'un installateur	8
3.3.2	Configuration de l'environnement de l'installateur	8
3.4	Mise à jour de l'installateur	8
3.5	Génération de l'image et écriture sur un support	9
4	Génération d'une clé USB amorçable	9
	Références	10

1 Introduction

Les principaux éléments qui constituent un support d'installation CLIP sont les suivants :

- Les miroirs de paquetages CLIP, nécessaires à l'installation d'un système CLIP (client ou passerelle). Les miroirs se trouvent dans le répertoire `mirrors/` du support. La section 2 décrit les étapes nécessaires à leur constitution ;
- L'image du système de fichier de l'installateur, qui contient l'ensemble des utilitaires d'installation de CLIP. Cette image se présente sous la forme du fichier `image.squashfs` situé à la racine du support d'installation ; elle est projetée en mémoire dans un `tmpfs` au cours du démarrage de l'installateur et constitue la racine de son système de fichier. La section 3 décrit les étapes nécessaires à sa constitution ;
- Le chargeur d'amorce et l'image du noyau du système d'exploitation de l'installateur (fichiers `syslinux.cfg`, `vmlinuz-clip`, `initrd.img`, etc.) ;
- Optionnellement, les profils de configuration de clients ou de passerelles, qui contiennent en particulier le matériel cryptographique IPsec des postes. Les profils se trouvent dans le répertoire (`config/`) du support. Le format des fichiers de configuration ne sont pas décrits dans ce document.

Les opérations décrites dans le présent document sont réalisées par un utilisateur du réseau de développement CLIP ayant le profil Développeur [CLIP-DCS-120], à l'exception de la génération des miroirs, qui est accomplie par un utilisateur de profil Validateur. Par ailleurs, un Validateur intervient aussi au terme de la génération du support d'installation, pour vérifier ce dernier avant de le transmettre aux personnels chargés de l'installation de postes CLIP [CLIP 2001].

On suppose que l'utilisateur dispose d'une station de développement CLIP, c'est-à-dire d'une machine (physique ou logique, voir [CLIP 1103]) avec une distribution Linux Gentoo installée, dont les arbres portage sont ceux du projets CLIP. La station doit en particulier posséder l'overlay `clip-overlay-dev`.

De plus, on suppose avoir accès, par exemple par le réseau, à un miroir de paquetages CLIP signés pour les distributions CLIP-RM et CLIP-GTW, comme indiqué en 2.1. On suppose enfin que les paramètres USE de portage contiennent le flag `clip-devstation`. La majorité des opérations nécessitant les privilèges root, nous supposons dans tout ce document que l'utilisateur a l'identité root.

2 Génération des miroirs

2.1 Suivi de version des paquetages et structure des miroirs

Pour mémoire, les paquetages CLIP sont susceptibles d'être intégrés au sein de deux « distributions » :

- la distribution « CLIP » [CLIP 1304], qui comprend les paquetages du socle système et des cages CLIP communes à tous les types de systèmes CLIP (CLIP-RM et CLIP-GTW) ;
- la distribution « RM » [CLIP 1401], qui comprend les paquetages des cages RM, qui ne sont présentes que dans certains types de systèmes (notamment, CLIP-RM).

Les paquetages doublement signés d'un type de système CLIP donné sont suivis en version au sein d'un *repository* dédié du serveur de gestion de version subversion ; ce *repository* est typiquement nommé `clip-<type>-dpkg` (par exemple, `clip-gtw-dpkg` pour CLIP-GTW). On trouve au sein de chaque sous-répertoire consacré à une version particulière du système (par exemple, `clip-gtw-dpkg/pkg/` ou `clip-gtw-dpkg/branches/stable-4.3.6/`), les répertoires de chaque distribution qui compose le système en question (dans le cas du type CLIP-RM, on trouve ainsi les répertoires `clip` et `rm`).

Les miroirs de paquetages destinés à l'installation d'un système CLIP sont générés à partir des paquetages extraits du serveur subversion, ce qui assure le lien entre cette gestion de configuration et les supports d'installation générés.

Deux miroirs sont créés par distribution présente sur un système CLIP : un pour les paquetages essentiels (appelés primaires), et un pour les paquetages secondaires. On créera ainsi deux miroirs propres aux paquetages CLIP d'un poste CLIP-GTW, et quatre miroirs pour les paquetages d'un poste CLIP-RM (deux pour les paquetages de la distribution CLIP - primaires et secondaires - et deux pour les paquetages de la distribution RM - primaires et secondaires -).

2.2 Génération des miroirs

Le script `get-mirrors.sh` du paquet `clip-livecd` automatise les opérations de récupération des paquetages nécessaires à la création du miroir de la distribution `<distri>` du système de type `<type>`. Ces opérations sont les suivantes :

1. création d'un répertoire temporaire `mirrors` ;
2. export subversion (`svn export`) dans `mirrors/clip-<type>-dpkg/<distri>/<conf>/pool` du paquetage `clip-<type>-dpkg/<distri>/<conf>_<version>.deb` où `<conf>` sera par exemple `rm-apps-conf` pour les paquetages secondaires de la distribution RM du système CLIP-RM, et `version` sera de la forme `4.3.6-r6_i386` ;
3. parcours des dépendances du paquetage décrivant la configuration, et export dans le même sous-répertoire `pool` des autres paquetages ;
4. création de l'index du miroir à l'aide de `dpkg-scanpackages` pour créer le fichier `mirrors/clip-<type>-dpkg/<distri>/<conf>/dists/clip/main/binary-i386/Packages.gz`

La génération d'un support pour l'installation de postes clients et passerelles nécessite ainsi six invocations du script `get-mirrors.sh`, avec comme paramètres :

- l'adresse du dépôt subversion utilisé ;
- le nom du sous-répertoire dans le dépôt concernant le système utilisé ;
- le nom du paquetage contenant la configuration, avec le numéro de version ;
- le nom de la distribution (`clip` ou `rm`).

Afin d'alléger les lignes de commandes ci-dessous, l'alias `gm` et la variable d'environnement `REPO` sont définis pour désigner respectivement le script `/opt/clip-livecd/get-mirrors.sh` et l'adresse du serveur subversion. On suppose que les miroirs sont créés dans le répertoire `clip-installer` :

```
# REPO="https ://clip.ssi.gouv.fr/"
# alias gm=/opt/clip-livecd/get-mirrors.sh
# pwd
<chemin-vers-clip-installer>/
# mkdir mirrors
# gm -r $REPO -s clip4-rm-dpkg -R ./mirrors/clip4-rm-dpkg -D clip -d clip-core-conf_4.3.7-r6_i386.deb
* Building mirror for clip-core-conf 4.3.7-r6...
# gm -r $REPO -s clip4-rm-dpkg -R ./mirrors/clip4-rm-dpkg -D clip -d clip-apps-conf_4.3.7-r7_i386.deb
* Building mirror for clip-apps-conf 4.3.7-r7
# gm -r $REPO -s clip4-rm-dpkg -R ./mirrors/clip4-rm-dpkg -D rm -d rm-core-conf_4.3.6-r6_i386.deb
* Building mirror for rm-core-conf 4.3.6-r6
# gm -r $REPO -s clip4-rm-dpkg -R ./mirrors/clip4-rm-dpkg -D rm -d rm-apps-conf_4.3.7-r8_i386.deb
* Building mirror for rm-apps-conf 4.3.7-r8
# gm -r $REPO -s clip4-gtw-dpkg -R ./mirrors/clip4-gtw-dpkg -D clip -d clip-core-conf_4.3.6-r29_i386.deb
* Building mirror for clip-core-conf 4.3.6-r29
# gm -r $REPO -s clip4-gtw-dpkg -R ./mirrors/clip4-gtw-dpkg -D clip -d clip-apps-conf_4.3.6-r25_i386.deb
* Building mirror for clip-apps-conf 4.3.7-r25
```

Si le développeur dispose d'un *repository* local (par exemple, issu d'un *checkout* du *repository* <https://clip.ssi.gouv.fr/clip4-gtw-dpkg/branches/stable-4.3.6/> dans le répertoire `/repositories/clip-pkg/`), les arguments `-r` et `-s`, qui désignent respectivement le nom du serveur de gestion de versions et le nom du *repository* sur ce serveur, peuvent être remplacés par l'argument `-p` suivi du chemin local où se trouve le *repository* :

```
# alias gm=/opt/clip-livecd/get-mirrors.sh
# pwd
<chemin-vers-clip-installer>/clip-installer
# mkdir mirrors
# gm -p /repositories/clip-pkg/stable-4.3.6 -R ./mirrors/clip4-gtw-dpkg -D clip -d clip-core-conf_4.3.6-r29_i386.deb
* Building mirror for clip-core-conf 4.3.6-r29
# gm -p /repositories/clip-pkg/stable-4.3.6 -R ./mirrors/clip4-gtw-dpkg -D clip -d clip-apps-conf_4.3.6-r25_i386.deb
* Building mirror for clip-apps-conf 4.3.7-r25
...
```

3 Génération du *squashfs* de l'installeur

La génération du *squashfs* de l'installeur est nécessaire si des modifications sont apportées à l'installeur CLIP ; elle est facultative si seuls les miroirs de paquetages ont changé d'une version à l'autre du système à installer.

3.1 Environnement de l'installeur

Sur un poste de développement CLIP, l'environnement de l'installeur réside dans un *chroot*. On suppose dans la suite que le répertoire qui contient la racine du système de fichier de l'installeur existe et s'appelle `clip-installer/rootfs`.



Manipulations d'un installeur au sein d'un LXC

Si le SDK CLIP avec lequel le générateur est préparé se trouve lui-même dans un conteneur LXC, la configuration de ce dernier doit généralement être adaptée aux besoins des utilitaires de génération. Il est par exemple nécessaire d'autoriser des opérations *chroot* au sein du LXC et d'y exposer le périphérique physique (clé USB `/dev/sdb` par exemple) sur lequel le support d'installation est créé.

L'environnement de génération d'un installeur CLIP est dérivé d'un système Gentoo, modifié avec les paquetages et outils spécifiques à CLIP. Deux options s'offrent à nous afin de générer cet installeur CLIP :

- la première est de générer *ex nihilo* l'installeur à partir d'un *stage3* Gentoo Hardened ;
- la seconde est de partir d'un *squashfs* existant (présent sur le support d'installation fourni au même titre que les miroirs et le SDK).

3.2 Génération de l'installeur *ex nihilo*



`portage-overlay-clip/clip-dev/clip-devutils`

La commande suivante permet la création *ex nihilo* d'un `rootfs` d'installeur CLIP dans un répertoire donné. Compte tenu du nombre de paquets composant l'installeur CLIP, cette procédure peut s'avérer longue bien qu'entièrement automatique.

```
# mkdir /opt/clip-installer/squashfs-rootfs
# clip-installer-bootstrap
```

3.3 Génération de l'installateur à partir d'un installateur existant

La commande suivante permet de décompresser l'image `image.squashfs` d'un installateur existant dans un répertoire `clip-installer/rootfs` (par défaut, cette commande décompresse l'image dans un répertoire `squashfs-root`) :

```
# pwd
/path/to/clip-installer
# unsquashfs -d ./rootfs image.squashfs
```

3.3.1 Outils de modification d'un installateur

Le paquetage `clip-livecd` contient les outils (présents sous `/opt/clip-livecd`) nécessaires à la génération de l'installateur. Ce paquet doit être installé sur le poste de développement :

```
# emerge -av clip-livecd
```

Le script `/opt/clip-livecd/enter-loop.sh` permet en particulier « d'entrer » dans l'environnement de l'installateur pour y réaliser les modifications. Le script prend en argument le répertoire contenant la racine du système de fichier de l'installateur :

```
# pwd
<chemin-vers-clip-installer>/clip-installer/rootfs
# /opt/clip-livecd/enter-loop.sh .
```

Entre autres opérations, le script `enter-loop.sh` réalise des montage *bind* de certains répertoires du poste de développement dans l'environnement de l'installateur (arbre portage de CLIP-dev ainsi que les répertoires `/dev`, `/proc`, etc.). Il « source » pour ce faire le fichier `/etc/clip-build.conf` du poste de développement.

3.3.2 Configuration de l'environnement de l'installateur

Les valeurs par défaut de la majorité des paramètres de compilation des paquetages Gentoo de l'environnement de l'installateur sont renseignées et versionnées dans les fichiers du répertoire portage-overlay-clip/profiles/clip-livecd/x86/, qui hérite lui-même du profil `clip-dev`, défini dans `portage-overlay-clip/profiles/clip-dev/x86/` (cf. fichier parent du profil `clip-livecd`).

La configuration de l'environnement de l'installateur se résume donc la plupart du temps :

- à renseigner la valeur de la variable d'environnement `CLIP_BASE` dans le fichier `<rootfs>/etc/make.conf`, qui doit pointer vers le répertoire contenant les arbres portage et les paquetages binaires. Le script `enter-loop.sh` (*bind*-)monte automatiquement l'emplacement pointé par `CLIP_BASE` du SDK vers le répertoire `<rootfs>/opt/clip-int/`. La variable `CLIP_BASE` de l'environnement de l'installateur doit donc pointer vers `/opt/clip-int`.
- à remplacer le répertoire `<rootfs>/etc/make.profile` par un lien symbolique vers `/opt/clip-int/portage-overlay-clip/profiles/clip-livecd/x86`.

Le cas échéant, on pourra également renseigner les valeurs des variables `PORTDIR` et `PORTDIR_OVERLAY` dans le fichier `<rootfs>/etc/make.conf` afin qu'ils pointent vers des sous-répertoires portage (`/opt/clip-int/`).

3.4 Mise à jour de l'installateur

La mise à jour d'un installateur (ou sa création) peut être réalisée via celle de certains de ses paquetages, notamment `baselayout`, `sysvinit`, `clip-installer` et `clip-livecd`, qui tireront les autres paquetages par le jeu des dépendances (notamment le noyau et l'image `initrd.img`). Les fichiers de configuration sont ensuite mis à jour avec les nouvelles versions :


```
# /opt/clip-livecd/enter-loop.sh clip-installer/rootfs
# emerge -av --newuse baselayout sysvinit clip-installer clip-livecd
# etc-update
```

3.5 Génération de l'image et écriture sur un support

La génération du fichier `image.squashfs` à partir du `rootfs` est réalisée par la commande `prepare-media.sh`. Cette commande génère également le fichier `initrd` du système et ajoute optionnellement des fichiers complémentaires sur le support d'installation final.

À partir du répertoire de travail `clip-installer/`, qui contient le système de fichier racine `rootfs/` de l'installateur préparé selon les instructions précédentes, la création du support d'installation dans le répertoire temporaire `tmp/` se lance de la façon suivante :

```
# pwd
<chemin-vers-clip-installer>/clip-installer
# mkdir tmp
# /opt/clip-livecd/prepare-media.sh -B tmp/ rootfs/
```

Le contenu du répertoire temporaire peut ensuite être copié sur un support USB amorçable (cf. section 4 ci-dessous).

Les paquetages CLIP doivent être présents dans le répertoire `mirrors/` du support d'installation. Il suffit donc de copier le répertoire `mirrors/` constitué selon la procédure décrite en section 2 pour compléter la création du support d'installation.

La commande `prepare-media.sh` peut également générer une image iso amorçable pouvant être ensuite gravée. L'option `-i` doit être utilisée à cet effet (l'option `-o` permet de préciser le nom de l'image, par défaut `livecd.iso`) :


```
# /opt/clip-livecd/prepare-media.sh -i -o clipinstall.iso -B tmp/ rootfs/
```

L'option `-a` permet pour sa part de préciser un nom de répertoire dont le contenu est copié récursivement à la racine du support d'installation final. Ceci est notamment utile dans le cas de la génération d'un CD-ROM pour inclure les miroirs de paquetages ne pouvant être ajoutés ou modifiés ultérieurement, contrairement à un support USB amorçable. On peut également ajouter les configurations des clients à installer (répertoire `config/`) :

```
# /opt/clip-livecd/prepare-media.sh -i -o clipinstall.iso -a extra/ -B tmp/ rootfs/
```

Le répertoire temporaire `tmp/` peut être réutilisé pour générer une nouvelle image de support d'installation, en ne mettant à jour que le système, ou au contraire en ne mettant à jour que les miroirs. Un support amovible USB peut également être monté directement sous le répertoire `tmp/` pour que la génération du support d'installation soit faite directement.

4 Génération d'une clé USB amorçable

 `portage-overlay-clip/clip-dev/clip-livecd`

Cette section décrit la procédure nécessaire à la création finale d'un support d'installation USB amorçable. Cette manipulation n'est à faire que la première fois, avant que le contenu du support d'installation ne soit copié ou généré sur le support.

Le script `prepare-key.sh` se charge de formater convenablement le support d'installation USB.

```
# /opt/clip-livecd/sbin-scripts/prepare-key.sh -b tmp/ -d /dev/sdb
```

L'exposition des *block devices* correspondant à la clé USB ainsi qu'à sa première partition (par exemple, `/dev/sdb` et `/dev/sdb1`) au sein du SDK lorsque celui est conteneurisé dans un environnement LXC ou Docker est laissé à la charge du lecteur.

Références

- [CLIP 1103] Documentation CLIP, 1103, *Environnement de développement*, ANSSI.
- [CLIP 1304] Documentation CLIP, 1304, *Cages CLIP*, ANSSI.
- [CLIP 1401] Documentation CLIP, 1401, *Cages RM*, ANSSI.
- [CLIP 2001] Documentation CLIP, 2001, *Procédure d'installation*, ANSSI.
- [CLIP-DCS-120] *Règles et procédures de développement CLIP*, CLIP_MAP-12000-007-DCS.