Documentation CLIP

1003

Paquetages CLIP

Version	Date	Auteur	Commentaires
1.1	04/09/2008	Vincent Strubel	Ajout de *perl* et des modules associés. Utilitaires *ccsd-sign*.
1.0.2	18/06/2008	Vincent Strubel	Correction des références.
1.0.1	12/06/2008	Vincent Strubel	Ajout de *ccsd-utils* et des fonctions de hachage CCSD du noyau.
1.0	02/06/2008	Vincent Strubel	Version initiale.

Table des matières

Introduction

Le présent document décrit les différents paquetages intégrés au sein des systèmes CLIP. Il n'a pas pour objectif de donner une liste exhaustive de tous les paquetages susceptibles d'être installés sur une configuration CLIP quelconque, mais uniquement d'énumérer et de décrire les paquetages qui résultent de développements spécifiques au projet CLIP (sections 1 et 2), et de détailler les adaptations les plus significatives apportées à des paquetages issus de la distribution *Gentoo Linux* dont CLIP est dérivé (section 3). Cette dernière section se limite aux paquetages pour lesquels des patches spécifiques ont été apportés aux sources, et ne comprend en particulier pas les centaines de paquetages importés dans CLIP depuis *Gentoo* sans modification, ou avec une simple modification du script *ebuild* qui en régit la compilation et l'installation (pour, par exemple, ajouter des fichiers de configuration spécifiques à CLIP, modifier les options de compilation ou d'installation, ou encore adapter les dépendances).

Il est rappelé que les paquetages CLIP se présentent sous la forme d'*ebuilds Gentoo* (paquetages sources), à partir desquels sont produits des paquetages binaires dans un format spécifique, qui seuls sont distribués aux postes clients CLIP. Le format de paquetage binaire CLIP est basé sur le format de paquetage *Debian*, dépourvu de certains mécanismes *Debian* (par exemple, la gestion des fichiers de configuration par *debconf*) et complétés de méta-données spécifiques à CLIP, et d'une double signature CCSD (développeur et validateur) de chaque paquetage. La production de paquetages binaires à partir des paquetages sources est assurée par *portage*, l'application de compilation et d'installation de paquetages *Gentoo*, spécifiquement adaptée pour CLIP. Le détail de cette adaptation est donné dans [3].

Les noms des paquetages sont écrits dans le présent document selon les conventions *Gentoo,* sous la forme <*catégorie*>/<*nom du paquetage*>. Il est par ailleurs rappelé que les paquetages CLIP sont susceptibles d'être intégrés au sein de deux « distributions », « CLIP » pour les paquetages du socle CLIP et des cages CLIP (toujours présentes, quelle que soit la configuration), et « RM » pour les paquetages des cages RM, qui ne sont présentes que dans certaines configurations. La ou les distributions auxquelles s'intègre normalement un paquetage (hors outils de développement) sont systématiquement citées à côté du nom du paquetage.

1 Outils de développement

Les paquetages décrits dans cette section n'ont pas vocation à être déployés sur un poste client CLIP, mais uniquement sur un poste de développement. Les postes de développement CLIP étant essentiellement des distributions *Gentoo* complétées des outils ci-dessous et des bibliothèques nécessaires à la compilation des paquetages CLIP, les outils de développement sont répartis en paquetages suivant le même schéma que les logiciels déployés sur CLIP.

1.1 Outils spécifiques à CLIP

1.1.1 Outils de création de paquetages

clip-dev/ccsd-utils

Installe l'utilitaire *ccsd-hash*, qui permet la création d'empreintes cryptographiques CCSD.

clip-dev/ccsd-sign

Installe les utilitaires *ccsd-sign* et *ccsd-check* de création et de vérification de signatures CCSD, plus optimisés (mais encore non déployés à ce stade) que les utilitaires équivalents de clip-dev/clip-dev-sign et app-clip/clip-sign .

clip-dev/clip-build

Installe l'utilitaire *clip-build*, qui permet de générer un ensemble de paquetages en fonction d'un profil de génération au format XML. Voir aussi [3].

clip-dev/clip-config-template-editor

Editeur graphique de configurations, permettant de visualiser rapidement les paquetages disponibles, leurs dépendances et conflits, et de créer des fichiers XML de description de configurations.

clip-dev/clip-create-config

Utilitaire permettant de générer des paquetages *Debian* de configuration, à partir de fichiers XML de description de configurations produits par *clip-dev/clip-config-template-editor*. Le paquetage installe de plus deux scripts, *conf-sign* et *conf-validate*, permettant, à l'aide de *clip-dev/clip-dev-sign*, d'apposer les signatures développeur et validateur sur un paquetage configuration, après avoir automatiquement vérifié la double signature de chaque paquetage référencé par la configuration.

clip-dev/clip-deb

Installe le script *gencontrol.pl* utilisé par *sys-apps/portage* pour générer le fichier *control* d'un paquetage *Debian* à partir des métadonnées issues de la compilation d'un *ebuild*, ainsi qu'un script *quickdeb* permettant la création d'un paquetage *Debian* à partir des fichiers installés par un *ebuild*. Le

fonctionnement de ce script est détaillé dans [3].

Installe de plus un ensemble de scripts réalisant des opérations de signature développeur / validateur de paquetages sources et binaires, en faisant appel à l'utilitaire à bas niveau *create-sign* fournit par *clip-dev/clip-dev-sign*.

clip-dev/clip-dev-sign

Installe deux utilitaires bas niveau pour la gestion des signatures de paquetages : *create-sign*, qui signe un paquetage *Debian* et insère la signature en fin de paquetage, et *verify-sign*, qui vérifie une telle signature.

clip-dev/clip-devutils

Installe plusieurs scripts utilitaires d'aide au développement CLIP, en particulier *clip-spec-annotate*, qui permet de préprocesser un fichier de configuration XML de *clip-dev/clip-build*, avant de le passer en argument à ce dernier.

1.1.2 Outils d'installation

clip-dev/clip-installer

Ensemble de scripts d'installation d'un système CLIP, réalisant dans cet ordre :

- le formattage des partitions affectées à CLIP (qui doivent avoir été créées au préalable)
- l'installation des paquetages CLIP et éventuellement RM par des appels répétés à *debootstrap*
- une configuration initiale du système, en installant les fichiers de configuration et clés cryptographiques contenues dans une arborescence de configuration, et en créant les comptes utilisateurs définis dans un fichier de configuration de l'installeur.

Cet utilitaire peut être utilisé aussi bien pour installer CLIP depuis un CD-ROM que depuis une installation Linux préexistante sur le poste. Il permet de plus de spécifier les différents paramètres d'installation qui dépendent de la configuration matérielle du poste (type de noyau à utiliser, résolution de l'écran, type de disque racine, et le cas échéant modules à charger au démarrage et arguments supplémentaires à passer au noyau au démarrage).

clip-dev/clip-livecd

Scripts et fichiers de configuration nécessaire à la création d'un *live CD* d'installation de CLIP. Ces scripts incluent notamment les utilitaires complémentaires à *clip-dev/clip-installer*, permettant de créer une table de partition adaptée avant d'appeler *clip-installer*, et d'installer le chargeur de démarrage après l'appel à *clip-installer*.

clip-dev/debootstrap-clip

Scripts *debootstrap* permettant d'installer les différents compartiments logiciels d'un système CLIP. Le paquetage installe uniquement quatre scripts :

- *clip_core*, permettant l'installation des paquetages primaires CLIP (toutes configurations

confondues)

- *clip_apps*, permettant l'installation des paquetages secondaires CLIP (toutes configurations confondues)
- *rm_core*, pour installer les paquetages primaires d'une cage RM
- *rm_apps*, pour installer les paquetages secondaires d'une cage RM

Ces quelques scripts supportent l'ensemble des configurations CLIP, du fait de l'externalisation dans un fichier de configuration de la liste des paquetages installés par chaque script. Ainsi, un même script, par exemple *clip_core*, peut installer aussi bien le coeur d'un poste CLIP-RM que le coeur d'une passerelle CLIP, en fonction de la liste de paquetages dont le chemin lui est passé par la variable d'environnement *PKGFILE*.

1.1.3 Paquetages virtuels

virtual/clip-devstation

Paquetage virtuel (aucun fichier installé) qui « tire » par ses dépendances un ensemble de paquetages *Gentoo* dont l'installation est requise sur un poste de développement CLIP.

1.2 Outils *gentoo* adaptés

sys-apps/portage

Le système natif d'installation de paquetages *Gentoo* est modifié dans CLIP de manière à supporter la génération de paquetages *Debian*, qu'il s'agisse de paquetages binaires ou de paquetages sources CLIP. Cette génération de paquetages s'accompagne d'un certain nombre de traitement complémentaires eux aussi spécifiques à CLIP, permettant notamment d'assurer la gestion des fichiers de configuration et des fichiers de description d'entrées *veriexec* (cf. [4]) (en utilisant notamment *clip-dev/ccsd-utils* pour la génération d'empreintes CCSD), et d'assurer l'exécutabilité des bibliothèques dynamiques, imposée par le LSM CLIP. Ces différentes adaptations sont détaillées dans [3].

2 Paquetages spécifiques à CLIP

Cette section est consacrée aux paquetages développé spécifiquement pour CLIP, qui n'ont pas d'équivalent en sources publiques.

2.1 Bibliothèques CLIP

clip-dev/ccsd ***(CLIP, RM)***

Bibliothèque cryptographique CCSD (cf. [18]), installée sous la forme d'une bibliothèque dynamique partagée et de fichiers d'en-tête C.

clip-libs/clip-lib ***(CLIP, RM)***

Bibliothèque de fonctions communément utilisées dans de nombreux développement CLIP, dont notamment :

- Gestion des privilèges : réduction de capacités, conservation de certaines capacités à travers un changement d'identité, etc.
- Gestion de *sockets* UNIX : récupération de l'identité de l'interlocuteur, entrées / sorties non bloquantes, etc.
- Gestion de processus : « démonisation » de processus, fermeture des descripteurs de fichiers ouverts.

Ce paquetage inclut une documentation auto-générée par *doxygen*.

clip-libs/clip-libvserver ***(CLIP)***

Bibliothèque réalisant une surcouche d'abstraction des appels systèmes *vserver*, utilisée par tous les développements CLIP qui manipulent des cages *vserver*.

Inclut aussi une fonction de *« proxying »* de terminal de contrôle, permettant de changer de terminal de contrôle avant d'entrer dans une cage *vserver*.

Ce paquetage inclut une documentation auto-générée par *doxygen*.

clip-libs/clip-sub ***(CLIP)***

Fichiers de définition de fonctions *bash* communes, réutilisées principalement par les différents scripts de démarrage et d'ouverture de session. Inclut notamment :

- Des fonctions de gestion des interfaces réseau : activation et configuration.
- Des fonctions de gestion du pare-feu *iptables* : création d'ensemble de règles prédéfinies ou paramétrables.
- Des fonctions de gestion des partitions chiffrées utilisateur : noircissement / dénoircissement de clé de partition à l'aide d'un mot de passe.
- Des fonctions d'import sécurisé de paramètres, permettant de définir des variables

d'environnement *bash* à partir de fichiers de configuration, sans pour autant « sourcer » ces fichiers et en vérifiant au préalable le format des valeurs affectées dans les variables par rapport à une expression régulière.

- Des fonctions de gestion des montages : réalisation d'une série de montages ou de démontages VFS.

clip-libs/libacidfile ***(CLIP, RM)***

Parseur de fichiers de clés ACID pour CCSD. Les fonctions fournies par la bibliothèque permettent la lecture et l'analyse de deux types de fichiers :

- Conteneurs au format KLNDoc v.2, de type « Clé privée ACID » ou « Clé publique ACID », dont peuvent être extrait les différents certificats (d'autorité / d'utilisateur) et clés privées.
- Certificats CCSD au format texte, dont peuvent être extraits les différents champs significatifs (*SubjectName,* dates de validité, etc...).

Ce paquetage inclut une documentation auto-générée par *doxygen*.

2.2 Modules PAM

L'authentification des utilisateurs CLIP est réalisée à l'aide d'une pile de modules PAM (*Pluggable Authentication Modules*), pour la plupart standard, ou du moins disponibles par des sources publiques (par exemple, *pam_tcb*). Deux tels modules sont cependant développés spécifiquement pour CLIP, afin de réaliser des opérations spécifiques à l'ouverture de session.

sys-auth/pam_exec_pwd ***(CLIP)***

Permet d'exécuter une ou plusieurs commandes lors de l'ouverture ou de la fermeture de session PAM. Les commandes peuvent être lancées sous l'identité du processus qui réalise l'authentification (typiquement *root*), ou sous celle de l'utilisateur associé à la session. Les commandes lancées à l'ouverture de session peuvent optionnellement se voire passer le mot de passe de l'utilisateur (qui doit avoir été recueilli au préalable par un autre module PAM), sous la forme d'une variable d'environnement. Le comportement exact est défini par un fichier de configuration, */etc/security/exec.conf*, qui associe des critères de sélection (nom d'utilisateur ou de groupe, ouverture ou fermeture de session) à des commandes à exécuter. Le module est décrit plus en détail dans [11].

sys-auth/pam_jail ***(CLIP)***

Permet d'enfermer un utilisateur dans une cage *vserver* existante, lors de l'ouverture de session. Le comportement exact du module est définit par un fichier de configuration */etc/security/jail.conf*, qui associe des critères de sélection (nom d'utilisateur ou de groupe) à des identifiants *xid* de cages. L'interface *vserver* est réalisée par *clip-libs/clip-libvserver*. Le module est décrit plus en détail dans [5].

2.3 Modules PERL

Plusieurs modules *perl* spécifiques à CLIP peuvent être intégrer dans des scripts *perl*, afin notamment de gérer les mises à jour et la journalisation. Ces modules sont à ce stade purement expérimentaux, et ne sont pas déployés sur les postes CLIP. Ils sont utilisés à l'aide de l'interpréteur dev-lang/perl modifié spécifiquement pour CLIP.

dev-perl/CLIP-Conf-Base ***(CLIP, RM)***

Fournit des fonctions de base pour la gestion des options de configuration CLIP, et en particulier pour l'import sécurisé de variables modifiables par l'administrateur local. Ces fonctions sont équivalentes aux fonctions *shell* du même type fournies par clip-libs/clip-sub (*import.sub*). Dépend de dev-perl/CLIP-Logger.

dev-perl/CLIP-Logger ***(CLIP, RM)***

Fournit des fonctions communes de journalisation pour les développements CLIP en *perl*, permettant de journaliser des messages sur la console, à travers *syslog*, ou les deux à la fois.

dev-perl/CLIP-Mount ***(CLIP)***

Fournit des fonctions de gestion des montages VFS, comparables au fonctions *shell* du fichier *mount.sub* de clip-libs/clip-sub. Dépend de dev-perl/CLIP-Logger.

dev-perl/CLIP-Pkg-Base ***(CLIP, RM)***

Fournit les fonctions de base pour la gestion des paquetages de mise à jour CLIP : vérification de signatures, extraction et vérification de méta-données, manipulation des miroirs locaux. Dépend de dev-perl/CLIP-Logger.

dev-perl/CLIP-Pkg-Download ***(CLIP)***

Fournit les fonctions spécifiques au téléchargement des mises à jour CLIP : établissement de listes de paquetages candidats au téléchargement, téléchargement et vérification, gestion du cache de téléchargement. Dépend de dev-perl/CLIP-Pkg-Base.

dev-perl/CLIP-Pkg-Install ***(CLIP, RM)***

Fournit les fonctions spécifiques à l'installation des mises à jour CLIP : établissement de listes de paquetages candidats à l'installation, installation, vérification de l'installation, gestion du cache d'installation et du retour sur erreur. Dépend de dev-perl/CLIP-Pkg-Base.

2.4 Arborescences de fichiers

Un certain nombre de paquetages CLIP, rassemblés dans la catégorie *clip-layout*, sont dédiés à l'installation des arborescences de base des différents compartiments logiciels du système (socle, cages, vues). Ces paquetages n'installent pour la plupart que des répertoires, liens symboliques et fichiers spéciaux associés à des périphériques, et pas d'utilitaires ou bibliothèques.

clip-layout/baselayout-clip ***(CLIP)***

Fournit l'arborescence de répertoires de base du socle CLIP, ainsi que ses fichiers de périphériques. Fournit de plus le système de gestion de scripts de démarrage, et les scripts les plus fondamentaux (*softlevels sysinit* et *boot,* cf. [10]).

clip-layout/baselayout-core-admin ***(CLIP)***

Arborescence non-privée (paquetage secondaire, exposé en lecture-écriture dans la cage $\text{UPDATE}_{\text{clip}}$ et en lecture seule dans $\text{ADMIN}_{\text{clip}}$) de la cage $\text{ADMIN}_{\text{clip}}$.

clip-layout/baselayout-core-audit ***(CLIP)***

Arborescence non-privée (paquetage secondaire, exposé en lecture-écriture dans la cage $\text{UPDATE}_{\text{clip}}$ et en lecture seule dans $\text{AUDIT}_{\text{clip}}$) de la cage $\text{AUDIT}_{\text{clip}}$.

clip-layout/baselayout-core-update ***(CLIP)***

Arborescence non-privée (paquetage primaire, en lecture seule pour $\text{UPDATE}_{\text{clip}}$) de la cage $\text{UPDATE}_{\text{clip}}$.

clip-layout/baselayout-core-user ***(CLIP)***

Arborescence non-privée (paquetage secondaire, exposé en lecture-écriture dans la cage $\text{UPDATE}_{\text{clip}}$ et en lecture seule dans $\text{USER}_{\text{clip}}$) de la cage $\text{USER}_{\text{clip}}$.

clip-layout/baselayout-core-adminpriv ***(CLIP)***

Arborescence privée (paquetage primaire, exposé partiellement en lecture-écriture dans la cage $\text{ADMIN}_{\text{clip}}$ et non exposé dans $\text{UPDATE}_{\text{clip}}$) de la cage $\text{ADMIN}_{\text{clip}}$.

Contient aussi les fichiers *device* de la cage, exposés en lecture-seule dans celle-ci.

clip-layout/baselayout-core-auditpriv ***(CLIP)***

Arborescence privée (paquetage primaire, exposé partiellement en lecture-écriture dans la cage $\text{AUDIT}_{\text{clip}}$ et non exposé dans $\text{UPDATE}_{\text{clip}}$) de la cage $\text{AUDIT}_{\text{clip}}$.

Contient aussi les fichiers *device* de la cage, exposés en lecture-seule dans celle-ci.

clip-layout/baselayout-core-updatepriv ***(CLIP)***

Arborescence privée (paquetage primaire, partiellement exposé en lecture-écriture dans la cage) de la

cage $UPDATE_{clip}$.

Contient aussi les fichiers *device* de la cage, exposés en lecture-seule dans celle-ci.

clip-layout/baselayout-core-userpriv ***(CLIP)***

Arborescence privée (paquetage primaire, exposé partiellement en lecture-écriture dans la cage $USER_{clip}$ et non exposé dans $UPDATE_{clip}$) de la cage $USER_{clip}$.

Contient aussi les fichiers *device* de la cage, exposés en lecture-seule dans celle-ci.

clip-layout/baselayout-viewer ***(CLIP)***

Arborescence de base des vues visionneuses de cages RM (sous-arborescence de la cage $USER_{clip}$ dans un poste CLIP-RM). Un seul paquetage fournit l'arborescence de toutes les vues visionneuses, grâce éventuellement à l'utilisation de l'option *CLIP_VROOTS* (cf. [3]).

clip-layout/baselayout-rm ***(CLIP)***

Arborescence de base des cages RM, comportant aussi les arborescences privées des différentes vues de ces cages. Un seul paquetage fournit l'arborescence de toutes les cages RM, grâce éventuellement à l'utilisation de l'option *CLIP_VROOTS* (cf. [3]).

clip-layout/baselayout-admin

Arborescence de base non-privée (exposée en lecture seule dans la vue, et en lecture-écriture dans la vue UPDATE) de la vue ADMIN d'une cage RM.

clip-layout/baselayout-audit ***(RM)***

Arborescence de base non-privée (exposée en lecture seule dans la vue, et en lecture-écriture dans la vue UPDATE) de la vue AUDIT d'une cage RM.

clip-layout/baselayout-update ***(RM)***

Arborescence non-privée (en lecture seule) de la vue UPDATE d'une cage RM.

clip-layout/baselayout-user ***(RM)***

Arborescence de base non-privée (exposée en lecture seule dans la vue, et en lecture-écriture dans la vue UPDATE) de la vue USER d'une cage RM.

clip-layout/clip-release ***(CLIP)***

Installe un unique fichier, */etc/shared/clip-release*, contenant le type de système CLIP et le numéro de version principal, afin de permettre son affichage lors du démarrage.

clip-layout/rm-devices ***(CLIP)***

Fichiers *devices* exposés par CLIP dans les cages RM. Les *devices* des différentes vues sont distingués

par des répertoires distincts, chacun étant monté en *bind* sur le */dev* d'une unique vue. Un seul paquetage fournit l'arborescence de toutes les cages RM, grâce éventuellement à l'utilisation de l'option *CLIP_VROOTS* (cf. [3]).

2.5 Utilitaires

app-clip/chroot-launch ***(CLIP)***

Utilitaire privilégié permettant à un utilisateur non privilégié de lancer une commande *chrootée* dans un chemin (partiellement défini lors de la compilation du paquetage). Permet le lancement des visionneuses VNC dans les vues visionneuses d'un système CLIP-RM.

app-clip/clip-backup-clt ***(CLIP)***

Utilitaire *backupclt* permettant à un utilisateur de la cage $ADMIN_{clip}$ de dialoguer avec le démon *backupsrv* du socle (*app-clip/clip-backup-srv*) pour configurer les options de sauvegarde / restauration du système.

app-clip/clip-backup-srv ***(CLIP)***

Scripts de sauvegarde / restauration du système, exécutés dans le socle au démarrage du système (cf. [10] et [15]). Ce paquetage installe aussi un démon *backupsrv*, et le script de démarrage qui le lance dans le socle. Le démon peut être contacté depuis la cage $ADMIN_{clip}$ (à l'aide de *app-clip/clip-backup-clt*) pour configurer les opérations de sauvegarde / restauration du système à réaliser au prochain démarrage.

app-clip/clip-data-backup-clt ***(CLIP)***

Utilitaire *databackupclt* permettant à un utilisateur de la cage $ADMIN_{clip}$ de dialoguer avec le démon *databackupsrv* du socle (*app-clip/clip-data-backup-srv*) pour configurer les options de sauvegarde / restauration des données.

app-clip/clip-data-backup-srv ***(CLIP)***

Scripts de sauvegarde / restauration des données, exécutés dans le socle au démarrage du système (cf. [10] et [15]). Ce paquetage installe aussi un démon *databackupsrv*, et le script de démarrage qui le lance dans le socle. Le démon peut être contacté depuis la cage $ADMIN_{clip}$ (à l'aide de *app-clip/clip-data-backup-clt*) pour configurer les opérations de sauvegarde / restauration des données à réaliser au prochain démarrage.

app-clip/clip-download ***(CLIP)***

Installe le script de téléchargement de mises à jour CLIP, *clip-download*, le script de démarrage qui le lance dans $UPDATE_{clip}$ (éventuellement, cf. [10]) au démarrage, ainsi que les entrées *crontab* permettant de le lancer régulièrement après cela. Les doubles signatures de paquetages sont systématiquement vérifiées après téléchargement à l'aide des outils de *app-clip/clip-sign* . Installe de

plus le démon *downloadmaster*, et les scripts de démarrage qui le lancent dans la cage $UPDATE_{clip}$ au démarrage de cette dernière. Ce démon permet, à l'aide du client *app-clip/downloadrequest,* de piloter les téléchargements (lancement, verrouillage et déverrouillage) depuis la cage $ADMIN_{clip}$.

app-clip/clip-generic-net ***(CLIP)***

Scripts de configuration réseau génériques, communs à toutes les configurations de postes CLIP. Ces scripts réalisent au démarrage (en s'appuyant au besoin sur les fonctions définies par *clip-libs/clip-sub*) les opérations de configuration suivantes :

- Configuration du filtrage réseau *netfilter.*
- Configuration initiale des politiques de sécurité IPsec et lancement du démon de définition de politiques (*net-firewall/racoon2*).
- Configuration des interfaces réseau.
- Lancement avec un fichier de configuration adapté du démon IKEv2 (*net-firewall/racoon2*).

Ces différentes opérations sont réalisées en important de manière sécurisée les paramètres de configuration réseau modifiables par l'administrateur local. De plus, les scripts génériques peuvent être complétés, en fonction du type de poste (client, passerelle, CLIP-RM, etc) par des fichiers de configuration et la définition de fonction « *hooks* » apportées par d'autres paquetages, par exemple *app-clip/clip-net*, *app-clip/clip-single-net* ou *app-clip/clip-gtw-net*.

Le détail des opérations de configuration réseau est donné dans [13].

app-clip/clip-gtw-net ***(CLIP)***

Fichiers de configuration et scripts spécifiques à la configuration réseau d'une passerelle CLIP, installés en complément de *app-clip/clip-generic-net* . Ce paquetage convient aussi bien à une passerelle de type UPDATE qu'à une passerelle de type RM, les différences de configuration étant ajustées par un script *postinst* en fonction de la valeur de la variable d'environnement *GTW_TYPE*.

app-clip/clip-install-clip ***(CLIP)***

Installe les scripts, scripts de démarrage et fichiers de configuration nécessaires à l'installation des mises à jour CLIP (paquetages primaires et secondaires), ainsi que les scripts de démarrage lançant dans le socle les mises à jour des éventuelles cages RM. Les opérations d'installation proprement dites sont réalisées par le script *clip-install* de *app-clip/clip-install-common*.

app-clip/clip-install-common ***(CLIP, RM)***

Installe le script *clip-install*, réalisant l'installation des mises à jour de paquetages primaires et secondaires pour CLIP et RM, appelé respectivement par les scripts des paquetages *app-clip/clip-install-clip* et *app-clip/clip-install-rm*. Ce script vérifie systématiquement les doubles signatures de paquetages à l'aide de *app-clip/clip-sign* .

app-clip/clip-install-rm ***(RM)***

Installe les scripts et fichiers de configuration nécessaires à l'installation des mises à jour RM (paquetages secondaires). Les opérations d'installation proprement dites sont réalisées par le script *clip-*

install de *app-clip/clip-install-common*.

app-clip/clip-mdadm ***(CLIP)***

Installe, sur une configuration intégrant des disques RAID, les scripts de reconstruction de synchronisation RAID, et le démon *mdadmd* qui, exécuté dans le socle, permet de lancer une telle synchronisation depuis la cage $ADMIN_{clip}$.

app-clip/clip-mdadm-clt ***(CLIP)***

Installe, sur une configuration matérielle intégrant des disques RAID, le client *mdadmclt*, qui permet, en se connectant au démon *mdadmd* de *app-clip/clip-mdadm*, de commander depuis la cage $ADMIN_{clip}$, une synchronisation RAID.

app-clip/clip-net ***(CLIP)***

Fichiers de configuration et scripts spécifiques à la configuration réseau d'un client CLIP-RM, installés en complément de *app-clip/clip-generic-net* .

app-clip/clip-sign ***(CLIP, RM)***

Installe l'utilitaire *verify-sign-full* permettant la vérification de la double signature d'un paquetage, et appelé aussi bien après le téléchargement d'un nouveau paquetage (par *app-clip/clip-download*) qu'avant son installation (par *app-clip/clip-install-common*).

app-clip/clip-single-net ***(CLIP)***

Fichiers de configuration et scripts spécifiques à la configuration réseau d'un client CLIP-single, installés en complément de *app-clip/clip-generic-net* .

app-clip/clip-update-user-data ***(CLIP, RM)***

Installe le script *clip-update-user-data*, qui peut être invoqué par exemple lors de l'ouverture de session, pour exécuter successivement tous les scripts installés dans */usr/local/bin/clip-update-user-data-scripts/*. Ces scripts peuvent être installés par des paquetages, afin de procéder à la mise à jour de données utilisateur (par exemple, fichiers de configuration de *$HOME/.kde/*, suite à une mise à jour KDE).

app-clip/clip-usb-clt ***(CLIP)***

Installe le client de gestion des supports amovibles, *usbclt*, qui permet de se connecter au démon *usbadmind* (*app-clip/clip-usbkeys*) pour réaliser les opérations de gestion supportées par ce dernier (cf. [17]). Installe aussi le client graphique *usbmenu*, permettant de lancer ces opérations à partir d'un client graphique durant les sessions $ADMIN_{clip}$ et $AUDIT_{clip}$ (les autres sessions lancent ces commandes depuis le menu *x11-misc/fbpanel* (CLIP-RM), ou à l'aide d'entrées créées dans le menu KDE par *clip-data/kde-config-clip* (CLIP-single).

app-clip/clip-usbkeys ***(CLIP)***

Installe les scripts de gestion des supports amovibles sécurisés (cf. [17]), et le démon *usbadmind* qui, lancé au démarrage dans le socle, permet de lancer ces commandes depuis une cage CLIP, à l'aide des clients de *app-clip/clip-usb-clt*.

app-clip/clip-user-mount ***(CLIP)***

Installe les scripts lancés par *sys-auth/pam_exec_pwd* à l'ouverture et à la fermeture de sessions utilisateur CLIP, pour monter (respectivement démonter) les partitions utilisateur chiffrées et temporaires, et réaliser certaines opérations de nettoyage de fichiers (fichiers *Xauthority* par exemple).

app-clip/clip-useradmin ***(CLIP)***

Installe les scripts de gestion de comptes utilisateurs, et le démon *useradmind* qui, lancé dans le socle, permet aux utilisateurs des cages CLIP de réaliser ces opérations de gestion (à l'aide des clients de *app-clip/clip-userclt*).

app-clip/clip-userclt ***(CLIP)***

Client de gestion de comptes utilisateurs, qui permet, en se connectant au démon *usbadmind* de *app-clip/clip-useradmin,* de lancer les opérations de gestion de comptes utilisateurs.

app-clip/clip-vserver ***(CLIP)***

Installe les fichiers de configuration des cages RM (et des cages SECURE_UPDATE_RM), et les scripts de démarrage qui permettent de lancer ces cages et les vues visionneuses associées (cf. [10]).

app-clip/core-services ***(CLIP)***

Installe les fichiers de configuration des cages CLIP, et les scripts de démarrage qui lancent ces cages (ou du moins, les préconfigurent, dans le cas des cages X11 et $USER_{clip}$, qui sont en réalité lancées par *x11-apps/xdm,* cf. [10]).

app-clip/downloadrequest ***(CLIP)***

Installe le client de pilotage des mises à jour, qui permet de lancer, depuis la cage $ADMIN_{clip}$, des commandes de téléchargement transmises au démon *downloadmaster* installé par *app-clip/clip-download*.

app-clip/install-ccsd ***(CLIP)***

Utilitaire simple et privilégié, permettant, au sein de la cage ADMINclip, d'installer des clés privées et publiques CCSD pour IKEv2 dans les répertoires de configuration de *net-firewall/racoon2,* avec des permissions adaptées (notamment, *root* comme propriétaire).

app-clip/jailmaster ***(CLIP)***

Installe *jailmaster*, le démon maître des cages RM, qui lance les services des vues AUDIT, ADMIN, UPDATE, avant de se mettre en attente de connexions utilisateur dans la vue USER. Le paquetage est

installé dans le Socle CLIP, mais exécuté dans les cages RM. Un unique paquetage *Debian* installe une copie de *jailmaster* à la racine de chaque cage RM, grâce à l'option *CLIP_VROOTS* de *sys-apps/portage* (cf. [3]).

app-clip/jailrequest ***(CLIP)***

Client lancé dans la cage $USER_{clip}$, pour se connecter à un démon *app-clip/jailmaster* et lancer une session USER dans une cage RM.

app-clip/rm-sessions ***(RM)***

Installe les scripts de lancement de sessions USER (sessions USER ou ADMIN-RM) dans les cages RM, appelés par *app-clip/jailmaster* sur réception d'une connexion *app-clip/jailrequest*.

app-clip/verictl ***(CLIP, RM)***

Installe l'utilitaire *verictl*, permettant la configuration du sous-système *veriexec* du noyau, dont le fonctionnement est détaillé dans [4]. Installe aussi, dans le cas d'un paquetage généré pour CLIP et non RM, le script de démarrage *veriexec*, qui permet d'initialiser ce sous-système (cf. [10]).

app-clip/vsctl ***(CLIP)***

Installe les utilitaires *vsctl, nsmount* et *vsattr*, qui permettent la configuration et la manipulation de cages *vserver*, et dont le fonctionnement est détaillé dans [5].

2.6 Fichiers de configuration

clip-data/kde-config-clip ***(CLIP)***

Installe des fichiers de configuration spécifiques à CLIP pour un environnement KDE installé dans la cage $USER_{clip}$. Ces fichiers permettent de pré-personnaliser les environnements de bureau des utilisateurs (activation par défaut de la langue française, double-clic pour activer les icônes, menus, icônes et barre de tâches adaptés).

clip-data/kde-config-rm ***(RM)***

Installe des fichiers de configuration spécifiques à CLIP pour un environnement KDE installé dans la vue USER d'une cage RM. Ces fichiers permettent de pré-personnaliser les environnements de bureau des utilisateurs (activation par défaut de la langue française, double-clic pour activer les icônes, menus, icônes et barre de tâches adaptés).

clip-data/splash-theme-clip ***(CLIP)***

Installe un thème spécifique à CLIP pour l'écran graphique de démarrage fournit par *media-gfx/splashutils*.

2.7 Patches noyau

Le noyau CLIP est installé par un paquetage source unique, *sys-kernel/clip-kernel*, qui permet de générer plusieurs paquetages binaires compilés avec des configurations différentes selon la définition des variables *USE* et *DEB_NAME_SUFFIX* (cf. [3]). Ces paquetages ont tous en commun un ensemble de *patches*, pour la plupart spécifiques à CLIP.

clip-lsm

Ajoute au noyau le sous-système CLIP-LSM, qui apporte un ensemble de mécanismes de sécurité complémentaires, comme la vérification d'empreintes cryptographiques d'exécutables, la gestion de niveau de privilèges hétérogènes, et des contrôles d'accès spécifiques pour les opérations réseau et les montages VFS. Ces différents mécanismes sont détaillés dans [4]. Le *patch* est suivi en version comme un module indépendant, incluant une documentation auto-générée par *doxygen*.

clip-patches

Ce module du suivi de version rassemble un ensemble de *patches* appliqués au noyau CLIP, dont un *patch* combiné *Vserver* + *Grsecurity* (cf. [5] et [6]), disponible par des sources publiques, ainsi que plusieurs *patches* spécifiques à CLIP, assurant notamment :

- Le durcissement de certains mécanismes *Vserver* et *Grsecurity* (par exemple, *GRKERNSEC_TPE* ou *GRKERNSEC_IO*, cf. [6]).
- La désactivation de plusieurs variables *sysctl,* inutiles et potentiellement nocives sous CLIP.
- Le montage en lecture seule du système de fichiers « anonyme » qui supporte les appels systèmes *eventfd / signalfd*, et qui, automatiquement partagé entre toutes les cages, créerait sinon un canal de communication entre celles-ci (cf. [5]).
- Le support d'une option *O_MAYEXEC* dans les arguments passés à l'appel *sys_open()*, qui fait échouer l'ouverture d'un fichier si l'accès à celui-ci se fait par un montage VFS portant l'option *MS_NOEXEC*. Cette option permet à des interpréteurs modifiés pour CLIP (cf. par exemple *app-shells/bash* ou *sys-apps/busybox*) de vérifier qu'il n'exécutent pas par erreur des scripts situés sur un montage *noexec* (ce qui contribue au maintien du principe de séparation « W^X » sur les fichiers, cf. [2])

kccsd

Ce *patch* réalise l'intégration des primitives cryptographiques symétriques et des fonctions de hachage de CCSD ([18]) dans l'API cryptographique du noyau (API *crypto_tfm*), et introduit les quelques modifications nécessaires à leur support par la couche IPsec. Le détail de ces modifications est donné dans [8]. Le *patch* est suivi en version comme un module indépendant, incluant une documentation auto-générée par *doxygen*.

2.8 Paquetages virtuels

CLIP intègre quelques paquetages virtuels spécifiques, qui permettent de simplifier la gestion de

dépendances.

virtual/clip-net-virtual ***(CLIP)***

Permet d'exprimer une dépendance commune vis-à-vis desn des paquetages de configuration réseau spécifiques à une configuration de poste CLIP : *app-clip/clip-net*, *app-clip/clip-gtw-net* et *app-clip/clip-single-net* .

virtual/x11-protos ***(CLIP, RM)***

Permet de satisfaire des dépendances (introduites par des erreurs dans les paquetages *Gentoo*) vis-à-vis des paquetages de la catégorie *x11-protos/*, qui ne sont jamais déployé sur les postes CLIP dans la mesure où ils ne fournissent que des fichiers d'en-tête C.

3 Paquetages *gentoo* modifiés

Cette section liste les principales modifications de paquetages *Gentoo* pour leur intégration dans CLIP. Il est rappelé que les paquetages qui sont simplement modifiés pour en changer les options de configuration ou d'installation, ou encore installer des fichiers de configuration ou des *maintainer-scripts* (cf. [3]) spécifiques à CLIP, ne sont pas décrits dans le présent document.

app-admin/syslog-ng ***(CLIP, RM)***

Démon de collecte de journaux, modifié spécifiquement dans CLIP pour s'interfacer avec *clip-libs/clip-libvserver*. Le patch ajoute une nouvelle option *-x <xid>* à la ligne de commande du démon. Lorsque cette option est passée, le démon s'enferme dans la cage *vserver* de *xid <xid>,* qui doit avoir été créée et configurée par ailleurs. L'enfermement dans la cage n'est réalisé qu'après l'ouverture des *sockets* de collecte de journaux, qui peuvent donc rester en dehors de la cage.

app-arch/dpkg ***(CLIP, RM)***

Utilitaire d'installation et de manipulation des paquetages *Debian,* modifié dans CLIP pour utiliser le répertoire */var/pkg/lib/dpkg* (partagé entre le Socle et $\text{UPDATE}_{\text{clip}}$) plutôt que */var/lib/dpkg* pour le stockage des méta-données de paquetages installés, et ne pas réaliser avant l'installation d'un paquetage les tests de présence de certains utilitaires absents d'une installation CLIP standard (notamment *ldconfig, start-stop-daemon* ou *update-alternatives*[1]).

app-shells/bash ***(CLIP)***

Shell standard sous Linux, modifié spécifiquement dans CLIP pour mettre en oeuvre l'option d'ouverture de fichiers *O_MAYEXEC* ajoutée au noyau par *clip-patches*. Ainsi, le *bash* modifié de CLIP refuse d'exécuter ou de « sourcer » un fichier stocké sur un montage monté avec l'option *noexec*.

dev-lang/perl ***(CLIP, RM)***

Interpréteur *perl*, modifié dans CLIP de manière à mettre en oeuvre l'option d'ouverture de fichiers *O_MAYEXEC* ajoutée au noyau par *clip-patches,* et à ne pas accepter l'interprétation de code *perl* passé sur la ligne de commande (option *-e*).

kde-base/kcontrol ***(CLIP, RM)***

Centre de configuration KDE, modifié dans CLIP de manière à supprimer les modules de configuration qui nécessitent la mise en oeuvre de privilèges *root* (qui ne sont jamais disponibles dans CLIP).

[1] Ces utilitaires sont appelés par un nombre important de *maintainer-scripts* de la distribution *Debian*. Ils ne sont en revanche pas installés sur CLIP, soit parce qu'ils apportent des fonctionnalités spécifiques qui ne sont pas mises en oeuvre dans CLIP (cas de *update-alternatives*, pour lequel aucun équivalent n'existe dans CLIP à ce jour, où de *ldconfig*, dont l'usage est remplacé par l'adaptation des *RPATH ELF* des exécutables et bibliothèques – cf. [3]), soit parce qu'ils ne sont présents que dans certains compartiments (cas de *start-stop-daemon*, qui n'est pas installé dans une cage RM). Les *maintainer-scripts* CLIP étant écrits spécifiquement pour CLIP, ces restrictions sont bien prises en compte et les tests de présence par *dpkg* sont superflus et potentiellement problématiques.

kde-base/kdebase-startkde ***(CLIP, RM)***

Scripts de démarrage de l'environnement KDE, modifié dans CLIP de manière à :

- Ne pas lancer l'assistant de personnalisation *kpersonnalizer*
- Ne pas « sourcer » des fichiers de configuration sous contrôle de l'utilisateur (typiquement, dans *$HOME/.kde/*), opération qui romprait le principe de « W^X sur les fichiers » (cf. [2]) et n'est pas permise par *app-shells/bash* et *sys-apps/busybox* du fait de leur utilisation de l'option d'ouverture *O_MAYEXEC* et du montage *noexec* des partitions utilisateur.

net-firewall/racoon2 ***(CLIP)***

Démon de négociation de clés IKEv2, qui n'est pas intégré dans la distribution *Gentoo*, mais est disponible en sources publiques. Le démon est intégré dans CLIP, avec plusieurs modifications :

- Utilisation de la bibliothèque *clip-dev/ccsd* pour la négociation de clés authentifiée, en utilisant des clés privées et publiques au format ACID, parsées à l'aide de la bibliothèque *clip-libs/libacidfile*.
- Support des algorithmes de chiffrement et d'authentification IPsec basé sur les algorithmes CCSD, qui sont intégrés au noyau par le patch *kccsd*.
- Réduction au strict nécessaire des privilèges détenus par le démon, après son démarrage, à l'aide de la bibliothèque *clip-libs/clip-lib*.
- Meilleur support des fonctionnalités de répondeur anonyme (mode « IP_RW » de *racoon2*, avec génération à la volée des politiques aussi bien que des associations de sécurité) pour les passerelles.

net-misc/tightvnc ***(CLIP, RM)***

Client et serveur VNC, utilisés dans CLIP pour l'affichage des bureaux des cages RM. Le démon serveur est adapté spécifiquement dans CLIP de manière à :

- Attendre des connexions VNC sur une *socket* UNIX, locale, plutôt que sur une *socket* réseau.
- N'accepter qu'une seule connexion sur cette *socket*, et se terminer immédiatement lors de la déconnexion du client.

Le client est de plus modifié pour lui aussi utiliser une *socket* UNIX plutôt que locale. Un drapeau USE spécifique (cf. [3]) permet de générer, selon la distribution cible, un paquetage *Debian tightvnc* contenant uniquement le serveur (RM) ou uniquement le client (CLIP).

sys-libs/gcc-lib ***(CLIP, RM)***

Ce paquetage n'existe pas dans la distribution *Gentoo*, mais est dérivé du paquetage *sys-devel/gcc*. Il réalise une compilation partielle des sources de *gcc* de manière à n'installer que les bibliothèques de support des environnement d'exécution C (*libgcc_s*) et C++ (*libstdc++*). Les sources utilisées sont celles de *Gentoo Hardened* (c'est-à-dire incorporant le support de *Propolice/SSP,* cf. [3]), sans modification spécifique à CLIP.

sys-libs/glibc ***(CLIP, RM)***

Bibliothèque standard C. La version déployée dans CLIP combine les patches *hardened* de la distribution *Gentoo* (support de *Propolice/SSP* en particulier, cf. [3]), avec l'intégration des fonctions *blowfish* nécessaires au mécanisme de hachage de mot de passe *bcrypt* (cf. [11]), qui est fourni par un patch dérivé de celui utilisé par la distribution *OpenWall Linux*.

sys-apps/apt ***(CLIP, RM)***

Utilitaire de gestion de paquetages *Debian,* modifié dans CLIP pour utiliser les répertoires */var/pkg/lib/* et */var/pkg/cache* (partagés entre le Socle et $\text{UPDATE}_{\text{clip}}$) plutôt que */var/lib* et */var/cache* pour le stockage des méta-données de paquetages installés, et pour corriger quelques problèmes de la méthode de téléchargement *https*.

sys-apps/busybox ***(CLIP, RM)***

« Boîte à outils » fournissant des versions simples de la plupart des commandes UNIX, utilisée dans CLIP pour fournir les commandes de bases des différentes cages et vues. Le *shell* intégré à *busybox* est modifié de manière spécifique à CLIP pour mettre en oeuvre l'option d'ouverture de fichiers *O_MAYEXEC* ajoutée au noyau par *clip-patches*. Ainsi, le *shell busybox* modifié de CLIP refuse d'exécuter ou de « sourcer » un fichier stocké sur un montage monté avec l'option *noexec.*

Par ailleurs, l'*ebuild busybox* est modifié dans CLIP de manière à générer différents paquetages *Debian*, par exemple *busybox-update* ou *busybox-audit*, en fonction de la valeur de la variable d'environnement *DEB_NAME_SUFFIX* (cf. [3]). Ces différents paquetages sont générés avec des configurations adaptées permettant de n'inclure dans chaque cage ou vue que les utilitaires nécessaires à son fonctionnement.

sys-apps/shadow ***(CLIP)***

Ensemble d'utilitaires permettant la gestion des comptes utilisateurs, modifié dans CLIP par l'intégration de trois *patches* issus de la distribution *OpenWall Linux* et modifiés spécifiquement pour les adapter à une version plus récente de *shadow*. Ces *patches* apportent les fonctionnalités suivantes :

- Support des fonctions de hachage de mot de passe *bcrypt* (cf. [11]), par ailleurs fournies par *sys-libs/glibc*.
- Support du mécanisme d'authentification *tcb* (cf. [11]), qui est par ailleurs fourni par le paquetage *sys-apps/tcb* (importé de *Gentoo* sans modification majeure).
- Longueur maximale des noms de groupes configurable.

sys-apps/sysvinit ***(CLIP)***

Démon *init* du système, adapté spécifiquement dans CLIP de manière à remonter son masque de capacités POSIX héritables de 0 à *CAP_INIT_SET* (toutes les capacités sauf *CAP_SETPCAP*) lors du passage en *runlevel 1* ou *6* (arrêt ou redémarrage). Cette modification permet de transmettre une capacité héritable aux utilitaires invoqués dans la séquence d'arrêt, leur permettant d'activer des capacités héritables attribuées par *veriexec* (cf. [4]) et ainsi de disposer des quelques privilèges supplémentaires nécessaire à l'arrêt correct du système.

Par ailleurs, le fichier *inittab* déployé par ce paquetage est adapté dans CLIP de manière à introduire un *softlevel* supplémentaire, *nonetwork*, dans la séquence de démarrage (cf. [10]), et à ne pas lancer, sauf dans des configurations de test, d'utilitaires d'ouverture de session console *agetty*.

x11-apps/xauth ***(CLIP)***

Utilitaire de gestion des *Cookies* d'authentification auprès du serveur X11, adapté de manière spécifique à CLIP pour ajouter un argument supplémentaire aux commandes *xauth generate*, permettant de générer un *Cookie* pour un domaine non privilégié spécifique (cf. [12]).

x11-apps/xdm ***(CLIP)***

Démon d'authentification X11, modifié dans CLIP pour s'interfacer avec la bibliothèque *clip-libs/clip-libvserver*, de manière à pouvoir lancer ses deux fils, le serveur X11 d'une part et le client X11 de lancement de session d'autre part, enfermés dans des cages *vserver* distinctes. Ces cages sont configurées par XDM lui-même, avec des paramètres (*xid,* racine, etc.) ajoutés aux ressources *Xresources* de l'application XDM (et normalement définis dans le fichier */usr/local/etc/X11/xdm/xdm-setup*).

Le paquetage intègre de plus un ensemble de fichiers de configuration spécifiques à CLIP, qui permettent notamment d'ajouter une fenêtre de dialogue *Xmessage* à l'écran d'ouverture de session, permettant d'arrêter ou de redémarrer le système, sans authentification.

x11-base/xorg-server ***(CLIP)***

Le serveur X11 est modifié dans CLIP par l'ajout de deux *patches* spécifiques, qui apportent les fonctionnalités suivantes :

- Réduction des privilèges nécessaires au fonctionnement du serveur X11, en désactivant les accès directs au matériel en mode *Framebuffer*. Cette modification permet de lancer un serveur X11 malgré les restrictions d'accès au matériel imposées par Grsecurity (cf. [6]), notamment l'interdiction de tout appel *iopl()*/*ioperm()*.
- Support de plusieurs domaines de sécurité (au sens de l'extension *Xsecurity*) non privilégiés, cloisonnés entre-eux et vis-à-vis du domaine privilégié, ce qui permet d'assurer le cloisonnement graphique entre cages RM au sein des systèmes CLIP-RM.

Ces deux modifications sont décrites plus en détail dans [12].

x11-libs/libxext ***(CLIP)***

Bibliothèque d'extensions *X11*, modifiée dans CLIP par un *patch* spécifique qui apporte le support de plusieurs domaines *Xsecurity* non privilégiés et cloisonnés entre eux (cf. [12]).

x11-misc/fbpanel ***(CLIP)***

Barre de menu graphique minimaliste, intégrée au bureau $USER_{clip}$ dans les configurations CLIP-RM. Le code source est modifié de manière à ajouter une localisation en langue française, ainsi que les *plugins* suivants (développés spécifiquement, mais fortement inspirés de *plugins* existants) :

- moniteur de charge batterie

- bouton lanceur de cage RM (qui n'est actif que lorsque la cage n'est pas active)
- barre de tâches multiniveau, affichant automatiquement chaque entrée d'une couleur spécifique, pour représenter le domaine de sécurité (*Xsecurity* étendu, cf. [12]) auquel appartient le client correspondant.

Le menu *fbpanel* est par ailleurs adapté dans CLIP de manière à proposer les opérations dédiées à la session $USER_{clip}$: changement de mot de passe, verrouillage de session, gestion des supports amovibles sécurisés RM_H et RM_B.

x11-wm/openbox ***(CLIP)***

Gestionnaire de fenêtres minimaliste, intégré au bureau $USER_{clip}$ dans les configurations CLIP-RM et adapté de manière spécifique pour prendre en compte le domaine *Xsecurity* (cf. [12]) auquel appartiennent les différents clients graphiques, et le représenter explicitement par :

- des bandeaux de couleur
- une chaîne de texte ajoutée en préfixe du nom du client

Openbox est par ailleurs configuré dans CLIP-RM de manière à désactiver entièrement le redimmensionnement et le déplacement des fenêtres, pour éviter notamment que de tels opérations ne permettent de masquer les bandeaux de couleur représentant le niveau de sécurité d'une fenêtre.

Annexe A Références

[1] Documentation CLIP – 1001 - Périmètre fonctionnel CLIP

[2] Documentation CLIP – 1002 – Architecture de sécurité

[3] Documentation CLIP – 1101 – Génération de paquetages (1.4 ou ultérieure)
[4] Documentation CLIP – 1201 – Patch CLIP LSM (1.6 ou ultérieure)
[5] Documentation CLIP – 1202 – Patch Vserver (1.0.2 ou ultérieure)
[6] Documentation CLIP – 1203 – Patch Grsecurity (1.0 ou ultérieure)

[7] Documentation CLIP – 1204 – Privilèges Linux

[8] Documentation CLIP – 1205 – Implémentation CCSD en couche noyau

[9] Documentation CLIP – 1206 – Génération de nombres aléatoires

[10] Documentation CLIP – 1301 – Séquences de démarrage et d'arrêt (1.0 ou ultérieure)

[11] Documentation CLIP – 1302 – Fonctions d'authentification CLIP

[12] Documentation CLIP – 1303 – X11 et cloisonnement graphique (1.0 ou ultérieure)

[13] Documentation CLIP – 1501 – Configuration réseau

[14] Spécification fonctionnelle des outils de mise à jour, CLIP-ST-13000-006-DCS (Ed0 Rev4 ou ultérieure)
[15] Document de conception de l'étude sur le retour à une configuration antérieure, CLIP-DC-15000-094-DCS (Ed0 Rev 3 ou ultérieure)
[16] Document de conception de l'étude sur les paramètres contrôlables par l'administrateur, CLIP-DC-15000-093-DCS (Ed0 Rev1 ou ultérieure)

[17] Document de conception de l'étude sur les supports amovibles, CLIP-DC-15000-088-DCS (Ed0 Rev1 ou ultérieure)

[18] Couche Cryptographique pour la Sécurité de Défense – Document d'Interface Client version 3.2