ZED!

FOR FILES ON JOURNEY

# CIBLE DE SECURITE CRITERES COMMUNS NIVEAU EAL3+

VERSION Q.2021

Réf. : PX2051296r5

PRIMX

MAKE ENCRYPTION HAPPEN

# Reproduction et droits



**Siège :** 18 rue du Général Mouton-Duvernet 69003 LYON – support@primx.eu

**Direction commerciale :** 21 rue Camille Desmoulins 92100 ISSY-LES-MOULINEAUX – Tél. : +33 (0)1 77 72 64 80 – business@primx.eu

**www.primx.eu**

ZED!
FOR FILES ON JOURNEY

# Sommaire

# Liste des figures

# Liste des tableaux

# 1. INTRODUCTION DE LA CIBLE DE SECURITE

## 1.1. Identification de la cible de sécurité

| | |
|---|---|
| Cible de sécurité : | Zed! Cible de sécurité CC niveau EAL3+ |
| Version de la ST : | PX2051296 v1r4 – Avril 2021 |
| Cible d'évaluation (TOE) : | - Zed! Entreprise édition complète Q.2021.1<br>- Zed ! Entreprise édition complète intégrée dans ZoneCentral Q.2021.1<br>pour les plateformes sous Microsoft Windows 10 versions 1809 LTSC et 20H2 (64 bits).<br>Note : On indiquera simplement Zed ! dans la suite de la cible sauf si une distinction est nécessaire. |
| Niveau EAL : | EAL3 augmenté des composants ALC_FLR.3 et AVA_VAN.3 associé à une expertise de l'implémentation de la cryptographie décrite dans [QUALIF_STD]. |
| Conformité à un PP existant : | Aucune. |
| Référence des CC : | Critères Communs version 3.1 Révision 5,<br>Parties 1 à 3 – Avril 2017 |

## 1.2. Vue d'ensemble de la cible d'évaluation

Zed! est un **produit de sécurité logiciel** permettant de fabriquer des conteneurs de fichiers chiffrés et compressés destinés soit à être archivés soit à être échangés avec des correspondants, en pièces jointes de messages électroniques ou sur des supports variés, comme des clés mémoires USB.

Zed! sera évalué pour une plateforme sous le système d'exploitation Microsoft Windows 10 versions 1809 LTSC et 20H2 (64 bits). Zed ! sera également évalué en tant que produit intégrée dans ZoneCentral Q.2021.1.

## 1.3. Conformité aux Critères Communs

Cette cible de sécurité respecte les exigences des Critères Communs version 3.1 d'avril 2017 :

| | |
|---|---|
| [CC1] | Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model. Version 3.1, Révision 5, Avril 2017. CCMB-2017-04-001. |
| [CC2] | Common Criteria for Information Technology Security Evaluation, Part 2: Security functional requirements. Version 3.1, Révision 5, Avril 2017. CCMB-2017-04-002. |
| [CC3] | Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements. Version 3.1, Révision 5, Avril 2017. CCMB-2017-04-003. |
| [CEM] | Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version 3.1, Révision 5, Avril 2017. CCMB-2017-04-004. |

Tous les composants fonctionnels décrits dans cette cible de sécurité sont issus de la Partie 2 « stricte » des Critères Communs version 3.1 d'avril 2017. Le niveau d'assurance « EAL3 augmenté » retenu est conforme à la Partie 3 « stricte » des Critères Communs version 3.1 révision 5 d'avril 2017. Le niveau d'assurance est un niveau EAL3 augmenté des composants ALC_FLR.3 et AVA_VAN.3.

Toutes les interprétations des Critères Communs parues à la date de démarrage de l'évaluation seront retenues.

## 1.4. Conformité aux référentiels de l'ANSSI

Cette cible de sécurité est conforme aux référentiels de l'ANSSI suivants :

| | |
|---|---|
| [QUALIF_STD] | Processus de qualification d'un produit – version 1.0 du 12 janvier 2017, ANSSI. |
| [CRYPTO_STD] | Guide des mécanismes cryptographiques : Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques - Version 2.04 du 1er janvier 2020, ANSSI. |
| [CLES_STD] | RGS version 2.0 – Annexe B2. Gestion des clés cryptographiques : Règles et recommandations concernant la gestion des clés utilisées dans les mécanismes cryptographiques - version 2.0 du 8 juin 2012, ANSSI |
| [AUTH_STD] | RGS version 1.0 – Annexe B3. Authentification : Règles et recommandations concernant les mécanismes d'authentification - Version 1.0 du 13 janvier 2010, ANSSI. |
| [MOTS_DE_PASSE_STD] | Recommandations de sécurité relatives aux mots de passe Version 1.1 du 5 juin 2012, ANSSI |

# 2. DESCRIPTION DE LA CIBLE D'EVALUATION (TOE)

## 2.1. Présentation de la TOE

### 2.1.1. Description Générale

L'email est l'outil le plus utilisé par les entreprises pour communiquer en interne ou avec leurs partenaires et prestataires. Cette communication au quotidien engendre un échange de documents sensibles quasi systématique. Mais, la plupart des documents échangés sont simplement joints aux emails, ne garantissant aucune confidentialité aux données transmises.

Zed! est un **produit de sécurité** pour postes de travail opérant sous Windows, Linux et Mac (seule la version sous Windows sera évaluée). Zed! se présente comme un produit autonome qui est également intégré dans le produit ZoneCentral de Prim'X. Son rôle est de permettre aux utilisateurs de fabriquer des **conteneurs de fichiers compressés et chiffrés**. Le produit intègre par ailleurs un mécanisme de **contrôle de l'intégrité global** des conteneurs. Ces conteneurs sont destinés à servir d'archive, ou, plus généralement, de pièce-jointe chiffrée dans des courriers électroniques échangés dans une société.

L'ergonomie est similaire aux fichiers «ZIP» standards sous Windows. L'utilisateur peut déposer des fichiers, les renommer, les supprimer, les extraire, etc. Zed! n'a aucune limite de taille de fichier et ne modifie pas l'arborescence des fichiers ou des dossiers qu'il copie.

Le conteneur permet de gérer un stockage chiffré des fichiers, sans modifier leurs caractéristiques (nom, dates, tailles) et de façon la plus transparente possible pour les utilisateurs. Le chiffrement/déchiffrement des fichiers s'effectue en effet lorsque les fichiers sont lu/copiés dans le conteneur et 'à la volée' (sans manipulation particulière de l'utilisateur).

Après avoir fabriqué un conteneur, l'utilisateur peut ajouter des accès pour ses correspondants. Un accès correspond à une **clé d'accès** (une clé cryptographique) que possède un utilisateur. Cette clé peut être soit une clé dérivée d'un mot de passe "d'échange" convenus avec un correspondant (dans ce cas les utilisateurs ne possèdent pas la clé d'accès elle-même mais le mot de passe permettant à Zed! de la calculer), soit une clé RSA hébergée dans un porte-clés comme un fichier de clé (certificat RSA pris dans un fichier certificat ou recherché sur un annuaire LDAP de certificats), une carte à mémoire, un container CSP ou CNG Microsoft Windows (le porte-clés pouvant lui-même être protégé par un code confidentiel). Une clé d'accès permet de retrouver (en les déchiffrant) les informations de chiffrement des fichiers du conteneur. Zed! intègre par ailleurs une fonction très pratique, le 'Carnet de Mots de Passe' qui permet de mémoriser et réutiliser les différents mots de passe des correspondants. Enfin Zed! offre la possibilité de masquer les noms de fichier apparaissant dans le conteneur : le conteneur apparait vide tant qu'une clé d'accès valide n'a pas été renseignée.

Zed! se décline en différents packages :

- **Zed ! Entreprise édition complète**, qui contient le produit complet ;
- ***Zed ! Entreprise édition limitée*** (hors périmètre de la cible de sécurité) qui se présente sous la forme d'un simple exécutable (zedle.exe) et permet aux correspondants de lire le contenu des conteneurs (moyennant la fourniture d'une clé d'accès) et d'en extraire les fichiers. Le correspondant a également le droit de modifier le contenu du conteneur (enlever, ajouter des fichiers) pour pouvoir le renvoyer à l'émetteur d'origine. L'édition limitée ne lui permet pas, cependant, de créer de nouveaux conteneurs ou d'en modifier les accès prévus par le créateur original du conteneur.
- Enfin, **Zed! Entreprise édition complète est également incorporé dans ZoneCentral**.

### 2.1.2. La technologie de Zed!

Le format interne du conteneur permet de contenir des fichiers de toutes tailles, indépendamment les uns des autres, et il gère en interne le nommage de ces fichiers. Le conteneur peut être considéré comme un "dossier virtuel"

contenant des fichiers confidentiels. Le conteneur gère également en interne un fichier de contrôle qui permet de gérer les accès et qui est masqué.

### 2.1.3. Les conteneurs et les accès

Chaque conteneur est défini par certaines caractéristiques de chiffrement (dont font partie les clés de chiffrement des fichiers, les algorithmes, etc.) et par une liste d'accès utilisateurs. La définition des accès est libre, mais le produit est doté de fonctions et de mécanismes d'administration permettant d'imposer certains accès ou certains types d'accès.

Pour pouvoir utiliser un conteneur, un utilisateur doit donc disposer d'une clé d'accès. Il peut s'agir d'une clé RSA hébergée dans un porte-clés comme un fichier de clés, une carte à puce, un container Microsoft CSP ou CNG (le porte-clés intégrant la plupart du temps son propre dispositif d'authentification avec un code confidentiel). Le mot de passe est le plus souvent choisi par l'utilisateur en fonction de la politique de sécurité mise en œuvre.

Une fois l'utilisateur authentifié, la clé d'accès ainsi fournie reste valide tant qu'elle n'a pas été explicitement fermée par l'utilisateur (verrouillage, fermeture de la session ou arrêt du système par exemple).

Lorsque le conteneur a été fabriqué, les fichiers protégés par le conteneur ont été chiffrés avec des clés elles-mêmes chiffrées avec les clés d'accès des utilisateurs. Bien entendu, les clés d'accès elles-mêmes ne figurent pas dans le conteneur.

Zed! propose différents algorithmes et mécanismes de sécurité, tous conformes aux standards en la matière. Il propose deux schémas de gestion de clés d'accès qui peuvent être utilisés en même temps. Un schéma dit « symétrique » basé sur des mots de passe et des clés dérivées de mots de passe (réf. : PKCS#5) et un schéma dit « asymétrique » utilisant des clés RSA (réf. : PKCS#1) embarquées dans des fichiers de clés (réf. : PKCS#12) ou des porte-clés (ref: PKCS#11 et/ou CSP/CNG).

## 2.2. Services d'utilisation et rôles

### 2.2.1. Définition des rôles

Hormis le responsable de la sécurité de l'organisation qui fixe la politique générale de sécurité à appliquer, on distingue 4 rôles mettant en œuvre (directement ou indirectement) les fonctionnalités de la TOE :

- Un rôle opérant uniquement dans l'environnement de la TOE) : **L'administrateur de la sécurité de l'environnement Windows des utilisateurs** (appelé administrateur Windows dans la suite du document) en charge de définir les règles d'usage et de sécurité (les policies), c'est-à-dire le paramétrage de fonctionnement du produit : cette opération de « haut-niveau » est effectuée sous le contrôle de l'administrateur de la sécurité de la TOE qui a étudié les différents paramètres et décidé des valeurs à affecter pour obtenir le comportement souhaité du produit dans le cadre d'utilisation et d'environnement prévu. Les policies sont signées par l'administrateur de la sécurité de la TOE et vérifiées par Zed! avant leur application. Le mécanisme de signature de politiques permet de garantir que seules des politiques validées par l'administrateur puissent être appliquées sur les postes de travail. En environnement Active Directory, un administrateur de domaine, autorisé pourtant à modifier les politiques du domaine, ne pourra pas intervenir sur la configuration du produit : s'il modifie les politiques, la signature deviendra invalide et donc les nouvelles politiques seront refusées sur les postes de travail. Les règles une fois affectées ne changeront ensuite que de façon très exceptionnelle. Il est à noter que ce rôle peut se décliner en plusieurs rôles hiérarchiques correspondant aux différents niveaux des domaines Windows. Dans ce cas les administrateurs Windows des niveaux supérieurs doivent interdire aux administrateurs des sous-niveaux (domaines, contrôleurs de domaines, postes de travail) la modification des « policies » de la TOE qu'ils souhaitent eux-mêmes contrôler.
- Un rôle **administrateur de la sécurité** en charge de l'installation de la TOE, de la signature des politiques, des opérations de recouvrement (et de secours) et de la mise à disposition des clés d'accès et éventuellement des mots de passe. L'administrateur de la sécurité possède également les droits pour gérer les accès.

- Un rôle **administrateur des accès** en charge de la gestion des accès, à savoir l'affectation du premier accès lors de l'initialisation du conteneur (en fait son propre accès) et la création des accès des autres ayant droits. L'administrateur des accès peut déléguer cette fonction à un autre utilisateur en modifiant le rôle de l'accès.
- Un rôle **utilisateur** qui utilise la TOE selon la configuration imposée par les administrateurs (il s'agit en général du correspondant lorsque les Zed! sont échangés).

Il faut noter que, à part la définition des policies, généralement dévolue à un responsable de la sécurité, les autres opérations peuvent être effectuées par différents acteurs en fonction de la confiance, de l'organisation et des moyens de l'entreprise.

### 2.2.2. Services d'administration et d'utilisation d'un conteneur

L'interface d'utilisation des conteneurs chiffrés est tout à fait similaire à l'utilisation des 'dossiers compressés' (.zip) sous Windows (glisser-déplacer, copier-coller …).

Il n'y a pas d'opération d'administration sur un conteneur autres que les gestions d'accès et la signature des politiques en amont. Un conteneur "vit" tant qu'il existe (fichier conteneur non supprimé).

Les opérations d'administration possibles sont:

- Lecture ou modification des policies (administrateur Windows), signature des policies par l'administrateur de sécurité. Attention par défaut les politiques ne sont pas signées, l'opération préalable de signature des politiques et leur intégration dans le package d'installation sont nécessaires pour bénéficier de la fonction de contrôle de signature des politiques.
- L'initialisation du conteneur à la première tentative d'accès par l'administrateur des accès (correspond à la création du premier accès et à l'ajout des accès de recouvrement configurés dans les politiques).
- L'ajout ou la modification d'un accès utilisateur par l'administrateur des accès (l'ajout et la modification impliquent que la clé d'accès du destinataire soit fournie, mot de passe ou clé publique).
- La modification du rôle d'un accès par l'administrateur des accès.
- Le recouvrement par l'administrateur de sécurité.
- Le secours utilisateur (dépannage à distance en cas d'oubli du mot de passe ou perte de la carte à puce par exemple) par l'administrateur de sécurité (hors périmètre de la cible de sécurité).

Note : La visualisation des accès du conteneur ne nécessite pas de rôle administrateur des accès.

Les opérations d'utilisation possibles sont:

- La création d'un nouveau conteneur via le menu « Nouveau » puis « Conteneur Chiffré » de l'explorateur Windows (par l'administrateur des accès). Cette opération ne requiert aucune clé d'accès, elle créé simplement un conteneur vide et sans accès qui est inutilisable tant qu'il n'a pas été initialisé.
- Le renommage ou la suppression d'un conteneur par l'utilisateur (concerne l'environnement Windows et donc ne nécessite pas de clé d'accès).
- L'ajout de fichiers au conteneur par l'utilisateur autorisé (copier/déplacer, insérer, copier/coller, etc.) : les fichiers sont chiffrés quand ils sont mis dans le conteneur, ce qui nécessite d'avoir fourni la clé d'accès du conteneur.
- La visualisation ou l'extraction de fichiers d'un conteneur par l'utilisateur autorisé : toute visualisation ou extraction de fichier nécessite la clé d'accès du conteneur pour que les fichiers soient déchiffrés du conteneur.
- La suppression de fichiers du conteneur par l'utilisateur autorisé (nécessite d'avoir fourni la clé d'accès du conteneur).
- La création et la suppression de dossiers dans le conteneur par l'utilisateur autorisé (nécessite d'avoir fourni la clé d'accès du conteneur).

Un **outil de collecte d'informations** est disponible (si la politique de sécurité le permet) via le menu de la fenêtre "A propos de…". Cet outil permet de générer un rapport de configuration chiffré à transmettre au support technique Prim'X (en pièce jointe d'un courrier électronique par exemple). Les informations collectées et sélectionnées par l'utilisateur sont la configuration des politiques, les applications installées, les fichiers logs. Le rapport est intégré dans un fichier sécurisé basé sur la même technologie que le carnet de mot de passe (avec le support technique comme accès unique) pour transmission via email par exemple.

### 2.2.3. Exemple d'utilisation de Zed!

Il existe différents scénarios de mise en œuvre, mais le principe d'utilisation reste le même pour les utilisateurs.

L'administrateur de la sécurité de l'environnement Windows définit ***les règles d'usage (policies)*** du produit puis l'administrateur de la sécurité les signe avec sa clé de signature privée, ce qui se traduit par une configuration prédéfinie (policy) qui peut être masterisée (personnalisation de l'installation) ou télé-gérée (diffusée, mise à jour) soit par des commandes d'administration fournies par le produit soit par la logistique intégrée des réseaux bureautiques (exemple : contrôleurs de domaines). Ces règles sont généralement établies à « haut niveau » dans l'entreprise par le Responsable de la Sécurité. Parmi ces règles, on trouve, par exemple, l'algorithme de chiffrement à utiliser, le comportement que doit adopter le logiciel dans certains cas, les porte-clés PKCS#11 supportés etc.

Le logiciel, masterisé ou non, est ensuite ***installé*** sur un poste de travail, manuellement ou via les logiciels de télé-installation du marché.

Par ailleurs, il est à la charge de l'administrateur de la sécurité de ***définir (fournir) les clés d'accès*** des utilisateurs (issues d'une PKI, par exemple). Zed! supporte différents scénarios de gestion de clés, mais n'en fournit pas l'infrastructure. Si une PKI est en place, il sait en utiliser les éléments (clés RSA, porte-clés, certificats), si elle n'est que partiellement installée, ou s'il n'y en a pas, il sait également utiliser des accès par mots de passe.

Lorsqu'un utilisateur crée un nouveau conteneur, il en devient l'administrateur des accès, affectant les accès et les rôles de ses correspondants (par défaut le rôle est simple « utilisateur »).
Ensuite, seuls les utilisateurs disposant de clés d'accès valides pour le conteneur chiffré pourront lire ou écrire des fichiers dans celui-ci. A la première tentative d'accès à un fichier chiffré dans le conteneur, Zed! demande à l'utilisateur une clé d'accès permettant de déchiffrer le fichier (en pratique, le schéma est plus complexe, et cette clé d'accès permet de déchiffrer des clés intermédiaires qui elles-mêmes chiffrent les fichiers). Si l'utilisateur peut la fournir, alors le fichier peut être déchiffré (ou chiffré, s'il s'agit d'une création ou d'une écriture). Sinon, l'utilisateur se voit refuser l'accès avec le message « Accès non autorisé». L'authentification de l'utilisateur auprès du conteneur reste effective jusqu'à la mise en veille ou le verrouillage de la session Windows, la clôture de la session Windows ou l'arrêt du poste. Comme pour les archives zip, pour « ouvrir » (lire) un fichier d'un conteneur, il doit d'abord être extrait. L'opération « ouvrir » automatise l'extraction (dans le dossier temp de l'utilisateur) et l'active. Quand le conteneur est refermé, le fichier est chiffré.

## 2.3. Périmètre et architecture de la cible d'évaluation

### 2.3.1. Les composants de Zed!

La figure 1 présente l'architecture du produit : le périmètre de la TOE est délimité par des pointillés. Les principaux composants sont listés ci-dessous.

«**Interface Explorer**» permet de gérer les menus et la vue graphique.

« **ZDU** » (ou « **ZCU** » dans ZoneCentral) est un «daemon» utilisateur instancié pour chaque session utilisateur Windows et qui référence les clés utilisateur saisies via l'entrée d'un mot de passe, l'interface PKCS#11, le CSP ou le CNG.

Le service « **ZEP** » (ou « **ZCP** » dans ZoneCentral) contrôle la signature des politiques.

«**Moteur Zed**», coordonne les différents traitements;

Le « **driver crypto** » qui est le centre cryptographique de Zed! ou de ZoneCentral : il gère les clés de conteneur et exécute les opérations de calcul associées. Les clés ne sortent jamais de son enceinte, sauf lorsque le produit est configuré pour utiliser des porte-clés (comme des extensions PKCS#11 pour des cartes à puce ou des CSPs/CNGs). Cette implémentation de la cryptographie en mode kernel du système renforce le niveau de protection global car c'est un emplacement très difficilement accessible aux logiciels 'pirates'.

Le « **driver clavier** » qui est un filtre de saisie clavier dans Zed ! ou ZoneCentral : il intercepte à très bas niveau les mots de passe et codes confidentiels saisis de façon à ce que leur valeur reste confinée le plus bas possible dans le système. Ils sont ensuite utilisés par le driver cryptographique, ou remis aux moteurs externes (CSP/PKCS#11). Cela ne concerne QUE les mots de passe gérés par Zed!, c'est-à-dire ceux qui conditionne les accès aux fichiers chiffrés. Cette implémentation renforce également la protection de ces données sensibles, qui ne remontent pas au niveau applicatif du système, source régulière et préférée des logiciels 'pirates'.

Remarque :

- Les composants Interface Explorer et Moteur Zed sont les mêmes dans Zed ! Entreprise et ZoneCentral
- Les composants ZDU, ZEP, Driver crypto et Driver clavier de Zed ! Entreprise ont leur équivalent dans ZoneCentral (avec les mêmes fonctions et les mêmes interfaces).

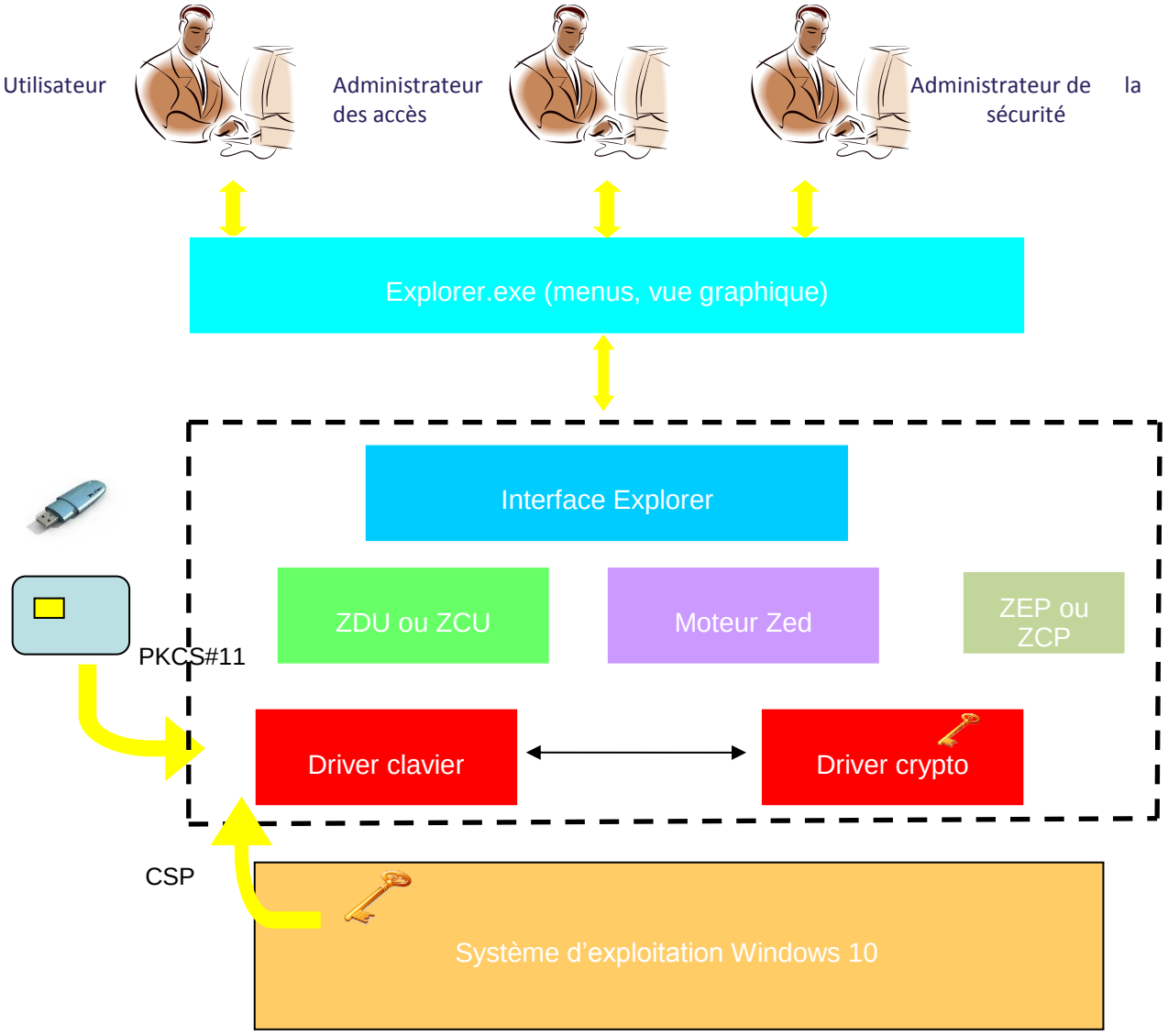


**Figure 1 – Architecture de Zed!**

## 2.3.2. Périmètre de la TOE

### 2.3.2.1. Périmètre logique

La TOE est constitué de Zed! Entreprise édition complète et Zed ! Entreprise édition complète intégrée dans ZoneCentral. L'édition limitée ne fait pas partie du périmètre de l'évaluation.

Seules les versions Zed! Entreprise édition complète Q.2021.1 et ZoneCentral Q.2021.1 configurées avec les politiques de sécurité activées ci-dessous sont déclarées conformes (les deux produits partagent les mêmes politiques pour Zed !). Toutes les politiques non indiquées sont configurées avec leur valeur par défaut.

**Mode Active directory**

- La politique P131 (accès obligatoires) doit être configuré avec l'accès de recouvrement de l'administrateur.

Note : Au moment d'effectuer le recouvrement, les politiques P269 - Ouverture au moyen de clés de recouvrement, P198 - Affichage des accès de recouvrement et P199 - Affichage des accès obligatoires devront être activées. Ces politiques ne sont normalement pas activées dans un environnement de production (notamment pour masquer les accès de recouvrement) et ne font donc pas partie du périmètre de test de l'évaluation).

- La politique P730 (seuil d'acceptation des mots de passe) doit être configurée à 100% et la politique P732 (longueur des mots de passe) doit être configurée à 12.
- La politique P339 (options du rapport de configuration) doit être configurée à « 0 » qui permet la collecte (valeur par défaut), il faut également renseigner le mot NONE dans le nom de la valeur de la politique P137 (accès imposés lors du chiffrement des informations collectées).
- La politique P381 (mécanisme de chiffrement pour les conteneurs chiffrés) doit être configurée à « CTS » (valeur par défaut).
- La politique P382 (autoriser l'utilisation du jeu d'instructions AES-NI) doit être configurée à « Non ».
- La politique P399 (version du format des conteneurs et messages chiffrés) doit être configurée à « Version 2 ».
- La politique P383 (mode de chiffrement RSA) doit être configurée à « PKCS#1 v2.2 avec utilisation de SHA-256 ».
- La politique P386 (mécanisme de signature) doit être configurée à « PKCS#1 v2.2 PSS».
- La politique P387 (mécanisme de dérivation (mot de passe)) doit être configurée à « SHA256-PBKDF2 » ou « SHA512-PBKDF2 ».
- La politique P233 (masquage des noms de fichiers et de dossiers des conteneurs chiffrés) doit être configurée à « Toujours masquer ».

Pour Zed! Entreprise édition complète Q.2021.1 intégrée dans ZoneCentral Q.2021.1 :

En plus des politiques ci-dessus (P131, P730 et P732, P339 et P137, P381, P382, P399, P383, P386, P387 et P233), il faut ajouter les politiques ZoneCentral suivantes (qui s'appliquent à la création de la liste d'accès de l'utilisateur) :

- La politique P702 (durée de validité des mots de passe) doit être configurée à une valeur inférieure à 90 jours.
- La politique P710 (seuil d'acceptation des mots de passe) doit être configurée à 100% et la politique P712 (longueur des mots de passe) doit être configurée à 12.

**Mode Alternatif :**

Pour Zed! Entreprise édition complète Q.2021.1 :

- Toutes les politiques ci-dessus (P131, P730 et P732, P339 et P137, P381, P382, P399, P383, P386, P387 et P233) sont à configurer dans le fichier de configuration des politiques dédié au mode alternatif.

En dehors de ce fichier, il faut configurer la suivante dans les politiques Active Directory:

- La politique P070 (configuration alternative des politiques) doit être configurée en indiquant le chemin du fichier de configuration alternative des politiques

Pour Zed! Entreprise édition complète Q.2021.1 intégrée dans ZoneCentral Q.2021.1 :

En plus des politiques ci-dessus (P131, P730 et P732, P339 et P137, P381, P382, P399, P383, P386, P387 et P233), il faut ajouter les politiques ZoneCentral suivantes dans le fichier de configuration des politiques dédié au mode alternatif (qui s'appliquent à la création de la liste d'accès de l'utilisateur) :

- La politique P702 (durée de validité des mots de passe) doit être configurée à une valeur inférieure à 90 jours.
- La politique P710 (seuil d'acceptation des mots de passe) doit être configurée à 100% et la politique P712 (longueur des mots de passe) doit être configurée à 12.

En dehors de ce fichier, il faut configurer la suivante dans les politiques Active Directory:

- La politique P070 (configuration alternative des politiques) doit être configurée en indiquant le chemin du fichier de configuration alternative des politiques

Le périmètre d'évaluation est constitué de l'ensemble des composants du logiciel Zed! hormis :

- l'outil GPOSign.exe permettant à l'administrateur de sécurité de signer les politiques ainsi que la génération de la clé de signature. Par contre la vérification de la signature des politiques par Zed! fait bien partie du périmètre de la TOE.
- L'utilisation du mode SSO (Single Sign On) qui permet d'ouvrir automatiquement les conteneurs chiffrés lorsque la session Windows est ouverte (mais reporte le niveau de sécurité à celui de Windows ou du composant SSO tiers).
- La suppression d'accès car cette opération ne renouvelant pas la clé de chiffrement du conteneur, il est préférable de créer un autre conteneur.
- Le secours utilisateur distant.

#### 2.3.2.2. Périmètre physique

Zed! sera évalué sur une plate-forme PC sous le système d'exploitation Windows 10 versions 1809 LTSC et 20H2 (64 bits) de Microsoft.

L'utilisation avec les différentes clés d'accès sera évalué (mot de passe et clé RSA). En particulier, le dialogue PKCS#11 entre la TOE et les porte-clés utilisateurs, le dialogue PKCS#12 entre la TOE et les fichiers de clés seront également évalués.

Les éléments suivants sont hors évaluation :

- Les systèmes d'exploitation Windows ;
- Les portes clés utilisés (comme les porte-clés de type Token USB, les fichiers de clés ou les containers CSP/CNG).

Le logiciel Zed! utilise des clés utilisateurs (les «clés d'accès») qui peuvent être fournies par l'environnement (clés RSA dans des porte-clés utilisateur).

Zed ! Entreprise est téléchargée depuis le site Web de l'éditeur à partir d'un compte privé sur le site client (client.primx.eu/Software/Download) ou le site partenaire (partner.primx.eu/Software/Download). Le programme d'installation est signé par Prim'X avec la technologie Authenticode. La valeur de la signature peut être comparé à celle indiquée pour le package à la page « Signature » du site Web. Le programme installe l'outil de signature des politiques (hors périmètre) et tous les guides du produit au format pdf à savoir le guide d'installation, le guide utilisateur, le manuel des politiques et le guide de signature des politiques. Les guides sont également téléchargeables à partir du compte client ou partenaire.

ZoneCentral est téléchargé selon le même principe que Zed et installe le guide des conteneurs chiffrés ainsi que les documents et outils d'administration communs à ZoneCentral et Zed : outil de signature des politiques (hors périmètre), manuel des politiques et guide de signature des politiques.

#### 2.3.2.3. Plate-forme de tests pour l'évaluation de la TOE

Pour l'évaluation du produit Zed!, les plates-formes ci-dessous devront être mise en place par l'évaluateur (le terme PC désigne en fait une machine virtuelle associée). Le type physique de porte-clés (carte à puce ou clé USB) étant transparent pour Zed! (seul le dialogue PKCS#11 est important), les tests de l'évaluateur s'effectueront avec un seul type de porte-clés.

On activera les politiques de sécurité conformément au périmètre logique défini ci-dessus. En mode active directory, la fonction de contrôle de signature des politiques nécessite une installation de la TOE avec un package d'installation spécialement préparé en se référant à la documentation de la fonction.

Plate-forme 1 sous Windows 10 64 bits version 1809 LTSC avec Zed ! Entreprise édition complète version Q.2021.1 en modes d'authentification par token USB et fichier de clés:

- Un contrôleur de domaine (Windows Server 2019)

- 2 PC sous Windows 10 version 1809 LTSC 64 bits avec Zed ! Entreprise édition complète version Q.2021.1

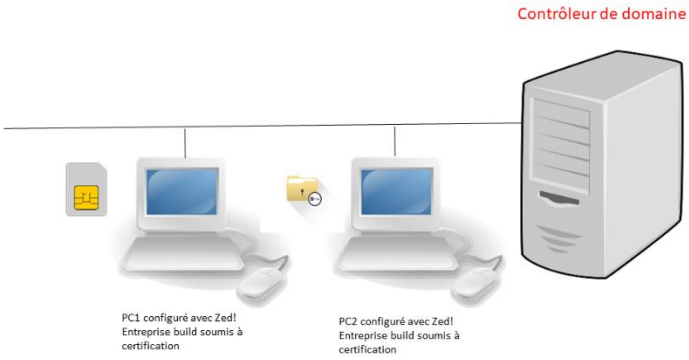


**Figure 2 – Plate-forme de tests n°1**

Plate-forme 2 sous Windows 10 version 20H2 64 bits avec Zed ! Entreprise édition complète version Q.2021.1 et l'édition complète intégrée dans ZoneCentral version Q.2021.1 en mode d'authentification par CSP et mot de passe respectivement:

- Un contrôleur de domaine (Windows Server 2019)
- Un PC sous Windows 10 20H2 64 bits avec Zed ! Entreprise édition complète version Q.2021.1
- Un PC sous Windows 10 64 bits avec Zed! version Q2021.1 édition complète intégrée dans ZoneCentral version Q.2021.1

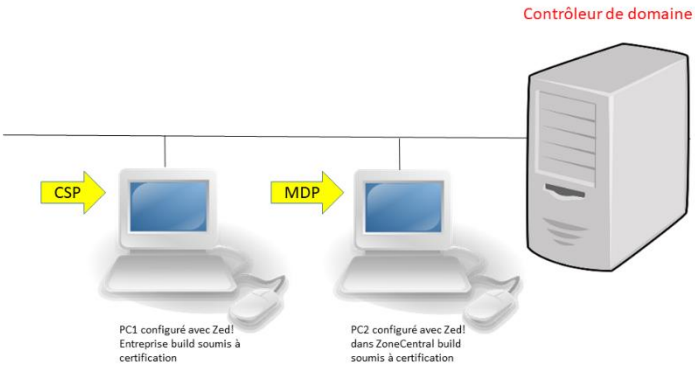


**Figure 3 – Plate-forme de tests n°2**

# 3. DÉFINITION DU PROBLÈME DE SÉCURITÉ

## 3.1. Les biens sensibles

### 3.1.1. Biens sensibles de l'utilisateur

#### 3.1.1.1. Clés d'accès : D. AUTH_USER

Pour ouvrir (lecture, remplissage, gestion des accès) les conteneurs chiffrés, Zed! met en œuvre les clés d'accès des utilisateurs. En fonction des cas de figure, il peut être amené à manipuler directement soit le mot de passe utilisateur, soit la clé d'accès elle-même, soit le code confidentiel de protection de la clé d'accès.

- Accès par mot de passe : Zed! gère la saisie du mot de passe, sa transformation (dérivation) en clé d'accès puis le déchiffrement de la clé de chiffrement et déchiffrement des fichiers du conteneur par cette clé d'accès. La politique de complexité des mots de passe est configurable par les politiques de sécurité ;
- Accès par clé RSA hébergée dans un fichier de clés en utilisant le mécanisme PKCS#12 : Zed! gère la saisie du code confidentiel du fichier de clés, lit et déchiffre le fichier de clés avec ce code confidentiel, obtient la clé d'accès RSA et effectue le déchiffrement de la clé de chiffrement et déchiffrement des fichiers du conteneur par cette clé d'accès;
- Accès par clé RSA hébergée dans un token logique accédé au travers d'un composant externe PKCS#11 (ce composant pouvant piloter une carte à mémoire, un token USB ou tout autre dispositif hardware ou software) : Zed! gère la saisie du code confidentiel du token logique, le remet au composant externe pour le déverrouiller. Zed! fournit également au composant externe la clé de chiffrement des fichiers chiffrés par sa clé publique. Le composant déchiffre la clé de chiffrement avec sa clé privée puis la transmet à Zed! qui peut alors effectuer le déchiffrement des fichiers;
- Accès par clé RSA hébergée dans un token logique accédé au travers d'un composant externe CSP ou CNG (ce composant pouvant piloter une carte à mémoire, un token USB ou tout autre dispositif hardware ou software) : Zed! ne gère pas la saisie du code confidentiel du token logique, c'est le composant externe qui le fait spontanément avec ses propres moyens. Zed! fournit au composant externe la clé de chiffrement des fichiers chiffrés par sa clé publique. Le composant déchiffre la clé de chiffrement avec sa clé privée puis la transmet à Zed! qui peut alors effectuer le déchiffrement des fichiers.

En fonction de ces cas, donc, Zed! manipule comme biens sensibles un mot de passe ou code confidentiel (en saisie), et une clé d'accès cryptographique. Dans les cas 1 et 2, il manipule les deux éléments, dans le cas 3, il ne manipule que le premier, dans le cas 4, il n'en manipule aucun.

Il faut noter que Zed! ne génère PAS les clés d'accès des utilisateurs : quand il s'agit de clés RSA, quel que soit le porte-clés qui les héberge et le module qui les traite, elles sont toujours générées par un outil externe à Zed! (en général une PKI), de même que le porte-clés éventuel et le code confidentiel de protection. Quand il s'agit de mots de passe, c'est l'administrateur de la sécurité ou le premier utilisateur (administrateur des accès) qui le choisissent. L'utilisateur et son environnement (règles et procédures internes, établies par le Responsable de la Sécurité) sont responsables de la qualité de ces clés, de la protection du porte-clés et de leur bonne utilisation.

Plutôt que de définir directement les accès utilisateurs dans un conteneur, il est possible (c'est même le comportement par défaut) de passer par un maillon intermédiaire, la **liste d'accès**. Une liste d'accès regroupe l'accès utilisateurs, l'accès de secours et les accès de recouvrement et le conteneur fait ensuite référence à cette liste. Cela permet notamment d'utiliser une même liste d'accès pour tous les conteneurs (unicité de gestion), et de regrouper les listes d'accès au même endroit (centralisation).

#### 3.1.1.2. Le carnet de mot de passe : D.CARNET_MOT_DE_PASSE

Le carnet de mot de passe est stocké dans le profil Windows de l'utilisateur. Il contient l'enregistrement de tous les mots de passe définis pour les correspondants. Les entrées peuvent être enregistrées lorsque l'utilisateur reçoit un

conteneur d'un destinataire pour la première fois et qu'il renseigne le mot de passe qui lui a été fourni ou lorsque l'utilisateur ajoute un mot de passe pour un destinataire.

Toutes les entrées sont chiffrées par la clé personnelle de l'utilisateur qui doit donc être fournie pour ouvrir et utiliser le carnet.

#### 3.1.1.3. Bi clé de signature : D.ID_ADMIN

Les politiques de sécurité sont signées par l'administrateur de la sécurité et vérifiées par Zed! avant leur application. Le bi-clé de signature dont surtout la clé privée de l'administrateur fait donc partie des biens sensibles de cet utilisateur particulier. La clé publique doit également être contrôlée par Zed ! au moment de son utilisation.

#### 3.1.1.4. Fichiers chiffrés : D.DONNEES_UTILISATEUR

Zed! permet de conserver sous forme chiffrée les fichiers et dossiers (possibilité également de chiffrer leurs noms). Les biens sensibles sont donc les fichiers et dossiers utilisateurs, de tous types, stockés dans les conteneurs.

Les fichiers ainsi chiffrés dans les conteneurs sont des biens sensibles de l'utilisateur protégés par la TOE (qui doit conserver leur image stockée chiffrée sans copie en clair) tant qu'ils demeurent dans leur conteneur. Le contrôle d'intégrité des fichiers est assuré par le contrôle d'intégrité globale du conteneur avant ouverture.

Les conteneurs peuvent posséder une image de 'fond' (watermark), par défaut un cadenas vert. Il est possible de définir soi-même l'image de fond d'un conteneur donné, et de faire en sorte que cette image soit « embarquée » dans le conteneur. Cette image fait donc partie des données utilisateur à protéger en intégrité.

### 3.1.2. Biens sensibles de la TOE

#### 3.1.2.1. Le conteneur considéré dans son ensemble : D. CONT

Le conteneur et tout ce qu'il contient (fichiers utilisateur, fichiers techniques, structure) forme un ensemble qui ne doit pas être détourné pour servir de vecteur d'attaque en ajoutant des données illicites transférables sur la station du destinataire. Un mécanisme de vérification d'intégrité globale est donc mis en place au sein du conteneur, ce mécanisme est implémenté lors de la demande d'ouverture du conteneur, avant toute opération, qui est refusée en cas d'atteinte à l'intégrité.

#### 3.1.2.2. La clé symétrique de chiffrement de fichiers : D.CLE_FIC

Les fichiers du conteneur sont chiffrés par une clé de chiffrement générée lors de l'initialisation du conteneur. A chaque fichier est associé un vecteur d'initialisation spécifique. Ces biens sont stockés chiffrés par les accès dans le fichier de contrôle du conteneur.

#### 3.1.2.3. Les programmes : D.PROGRAMMES

Pour assurer son fonctionnement, la TOE met en œuvre ses ***programmes*** (exécutables, bibliothèques dynamiques). La sécurité en intégrité de ces programmes repose sur l'environnement : il faut être administrateur Windows pour les modifier. Ces programmes sont également signés (système authenticode Windows).

#### 3.1.2.4. La configuration : D.CONFIGURATION

Pour assurer son fonctionnement, la TOE met en œuvre des ***policies*** :

- Soit par l'intermédiaire des « Group Policies » qui sont des fonctions de gestion centralisée de la famille Microsoft Windows permettant la gestion des ordinateurs et des utilisateurs dans un environnement Active Directory.
- Soit en utilisant un «mode alternatif», permettant de définir la configuration désirée au sein d'un ou plusieurs fichiers accessibles sur un simple partage de fichiers.

La sécurité en intégrité de ces policies est assurée :

- Par l'environnement (i.e. le système des policies sous Windows) : il faut être l'administrateur Windows de plus haut niveau pour les modifier (si un domaine Windows définit une valeur pour un paramètre, alors un administrateur local au poste ne pourra pas la modifier).

- Par le produit dans la mesure où les politiques sont signées par l'administrateur de la sécurité et vérifiées par Zed! avant d'être appliquées.

#### 3.1.2.5. Le fichier de contrôle : D.FICHIER_CONTROLE

Ce fichier contient le libellé du conteneur, un identifiant unique, quelques informations de gestion, et les 'wrappings' d'accès, c'est-à-dire les clés de chiffrement du conteneur chiffrées par les clés d'accès des utilisateurs habilités. Le contrôle d'intégrité du fichier de contrôle est assuré par le contrôle d'intégrité globale du conteneur avant ouverture.

#### 3.1.2.6. Le fichier catalogue : D.FICHIER_CATALOGUE

Ce fichier contient la liste des fichiers applicatifs du conteneur, avec leur position dans l'arborescence, les tailles originales, les horodatages, etc. Le contrôle d'intégrité du fichier catalogue est assuré par le contrôle d'intégrité globale du conteneur avant ouverture.

#### 3.1.2.7. Synthèse des biens sensibles

Le tableau ci-dessous résume la liste des biens sensibles protégés par Zed! et indique la nature de la sensibilité associée. Les qualificatifs « forte » et « faible » de la sensibilité font référence au degré de protection vis-à-vis du potentiel d'attaque visé dans la cible (chapitre 3.3). Une sensibilité forte impose un niveau de protection résistant à l'attaque correspondante pour le niveau visé (divulgation du bien, atteinte à l'intégrité non détectée), une sensibilité faible indique que le bien n'a pas à être protégé au degré visé. Par exemple la divulgation des politiques apporte peu d'information intéressante à un éventuel attaquant (configuration générale du produit) mais la modification des politiques doit être contrôlée sous peine d'atteinte à la sécurité du produit (ajout d'un accès de recouvrement par exemple).

*Remarque : de façon générale, l'intégrité n'est pas l'objectif premier de Zed!. Le rôle principal du produit est de gérer la confidentialité des biens sensibles qui lui sont confiés. Cependant Zed! intègre un mécanisme de contrôle de l'intégrité globale du conteneur avant qu'il ne soit ouvert par l'utilisateur. La mise en œuvre de ces contrôles permet de détecter des altérations qui seraient nuisibles au bon fonctionnement de Zed! ou qui pourraient induire un défaut dans son objectif de confidentialité.*

| Biens sensibles | Confidentialité | Intégrité |
|---|---|---|
| *Biens sensibles de l'utilisateur* | | |
| Eléments des clés d'accès manipulés par Zed! : cas des mots de passe ou codes confidentiels éventuels. (D.AUTH_USER) | Forte | NA |
| Eléments des clés d'accès manipulés par Zed! : cas des clés d'accès elle-même si elles sont directement utilisées par Zed! (D.AUTH_USER) | Forte | Forte |
| Mots de passe des correspondants stockés dans le carnet de mot de passe (D.CARNET_MOT_DE_PASSE) | Forte | *Faible.* |
| Bi clé de signature (D.ID_ADMIN) | Forte (clé privée) | Forte |
| Fichiers et dossiers de l'utilisateur stockés dans les conteneurs (D.DONNEES_UTILISATEUR) | Forte | Forte (toute erreur doit être signalée et empêcher le déchiffrement des données) |
| *Biens sensibles de la TOE* | | |
| Le conteneur pris dans son ensemble (D. CONT) | *Faible* | Forte |
| Clé de chiffrement de fichiers (D.CLE_FIC) | Forte | Forte |

| Biens sensibles | Confidentialité | Intégrité |
|---|---|---|
| Fichier de contrôle du conteneur (D.FICHIER_CONTROLE) | *Faible* | Forte |
| Fichier catalogue (D.FICHIER_CATALOGUE) | *Faible* | Forte |
| Configuration de Zed! (D.CONFIGURATION) | *Faible* | Forte |
| Programmes de Zed! (D.PROGRAMMES) | *Faible* | Forte |

**Tableau 1 : Synthèse des biens sensibles**

## 3.2. Hypothèses

Pour Zed!, nommée la TOE dans les paragraphes suivants, les hypothèses suivantes sur l'environnement d'utilisation seront prises en compte pour l'évaluation du niveau de confiance offert aux utilisateurs :

| | |
|---|---|
| **H.NON_OBSERV** | L'environnement physique de la TOE permet aux utilisateurs d'entrer leur mot de passe ou code PIN sans être observable directement et sans que cela puisse être intercepté (clavier sans fil…) par d'autres utilisateurs ou attaquants potentiels. |
| **H.POSTE_SUR** | L'environnement opérationnel ne permet pas à un attaquant d'accéder au disque (physiquement ou par le réseau) lorsque des données sensibles sont accessibles à un utilisateur légitime sur l'équipement. L'équipement doit donc apporter des protections efficaces contre l'accès illicite distant (pare-feu correctement configuré, antivirus et anti logiciels espions avec bases de données à jour etc.). |
| **H.CONFIANCE_ADMIN** | L'administrateur de la sécurité et l'administrateur des accès sont des personnes de confiance. Les administrateurs Windows sont des personnes de confiance en charge de la configuration (avec des valeurs sures) des « policies ». Ils sont formés à l'utilisation de la TOE tout comme les utilisateurs.<br><br>Les administrateurs de la sécurité doivent garantir (ensemble si ils appartiennent à des organisations différentes) l'utilisation de politiques de sécurité conformes au niveau recherché (force des mots de passe notamment). |
| **H.CONSERVATION_CLES** | Les utilisateurs sont chargés de la conservation dans un lieu sûr et de la non divulgation des clés d'accès qui leurs ont été transmises par l'administrateur de la sécurité. L'administrateur de la sécurité est également chargé de conservation dans un lieu sûr et de la non divulgation des clés de recouvrement et de son bi-clé de signature. |
| **H.ACCES** | L'administrateur de la TOE est chargé de la création des listes d'accès et de leur stockage sur un partage dédié bénéficiant des mêmes mesures de sécurité que les postes utilisateur et dont l'accès en écriture est uniquement autorisé à l'administrateur de la TOE. Il est également chargé, lors de la fourniture des clés d'accès possédant un certificat X509, de vérifier la validité de ces certificats et leur adéquation avec l'usage qui en est fait par la TOE. |
| **H.ENV_PROTECT_TOE** | L'environnement technique de la TOE assure l'intégrité des composantes de la TOE. L'administration et la mise à jour de la TOE sont assurées par des personnes formées et habilitées. |
| **H.FIDELE_ENV** | L'environnement d'exécution fournit à la TOE une date et une heure exacte pour assurer les fonctions d'horodatage. |
| **H.ENV_ALEA** | L'environnement d'exécution fournit à la TOE des mécanismes (événements partiellement imprédictible) pour produire les aléas nécessaires à la génération des secrets. |
| **H.CRYPTO_EXT** | Les clés d'accès générées ou stockées à l'extérieur de la TOE doivent être conformes aux documents [CRYPTO_STD] et [CLES_STD] pour le niveau standard. |

## 3.3. Menaces [contre les biens sensibles de la TOE]

*Il s'agit ici des menaces portant sur les biens sensibles de la TOE elle-même. Celles qui concernent les biens des utilisateurs sont couvertes par les Politiques de Sécurité Organisationnelles (services du produit) décrites plus loin.*

De manière générale, l'attaquant est un individu qui n'a pas accès aux données du conteneur (il n'a pas le droit d'en connaitre), il peut être interne ou externe à la société et agir selon différents canaux :

- L'attaquant accède aux données de l'utilisateur légitime par vol ou accès illégitime au poste de travail. Par hypothèse, on considère que d'autres moyens sont utilisés pour protéger les données qui peuvent se retrouver sur le disque utilisateur après extraction du conteneur (chiffrement du disque par exemple). Par contre l'agent menaçant considéré ne doit pas pouvoir ouvrir des conteneurs assurant des fonctions de stockage sécurisé sur un poste par exemple ou accéder à des données sensibles qui auraient été temporairement stockées dans l'environnement d'exploitation pendant l'utilisation licite de Zed!. Il ne doit pas non plus affaiblir la protection des conteneurs en modifiant les politiques de sécurité.
- L'attaquant intercepte les données sur le réseau (sans accès physique au poste de travail). Par exemple, il intercepte un courrier échangé contenant un conteneur en pièce jointe afin de l'attaquer en installant ou pas le produit Zed! ou ZoneCentral.

L'attaquant considéré est doté d'un potentiel d'attaque « enhanced-basic » au sens des Critères Communs.

| | |
|---|---|
| **M.DETOURN_COMPOSANT** | En possession d'un conteneur intercepté, un attaquant se procure le produit Zed! Entreprise (ou ZoneCentral) et met en œuvre, éventuellement à bas niveau, les composants internes de la TOE, pour contourner certaines fonctions de sécurité en provoquant ou profitant d'un dysfonctionnement. L'attaquant peut pour cela effectuer du «reverse-ingeniering» sur le programme, développer des programmes d'appel des fonctions internes de la TOE, ou s'aider d'un debugger. Le bien impacté est le programme de la TOE (confidentialité et intégrité) et les données utilisateur (confidentialité).<br><br>Il ne doit pas pouvoir, avec ces moyens, réussir à «pénétrer» le conteneur dans lequel il n'aurait pas normalement accès. |
| **M.POLITIQUE_SECU_INT** | Un attaquant signe des politiques à la place de l'administrateur de sécurité (politiques domaines ou politiques locales si accès au poste). Elle peut par exemple configurer son propre accès de recouvrement qui sera automatiquement ajouté lorsque l'utilisateur créera ses prochains conteneurs. Le bien impacté est la configuration (intégrité) et indirectement les données utilisateur (confidentialité). |
| **M.CARNET_CONF** | Un attaquant récupère le carnet de mot de passe stocké sur le poste de l'utilisateur pour tenter de retrouver les mots de passe des correspondants. Les biens impactés sont donc les mots de passe des correspondants et par suite les données utilisateur stockées dans les conteneurs possédant ces accès (tous en confidentialité).<br><br>Par exemple, l'attaquant tente de déchiffrer les mots de passe contenus dans le carnet afin de les utiliser pour ouvrir les anciens ou futurs conteneurs échangés. |
| **M.FIC_CONTROLE_CONF** | Un attaquant récupère le fichier de contrôle d'un conteneur pour |

tenter de retrouver des informations protégées. Les biens impactés sont donc les clés de chiffrement et les données utilisateur (tous en confidentialité).

Par exemple, l'attaquant tente de retrouver des informations protégées (telles que les clés de chiffrement) à partir des fichiers chiffrées du conteneur et du fichier de contrôle de la TOE ou bien essaye de déchiffrer directement les informations contenues dans le fichier de contrôle.

**M.FIC_CONTROLE_INT**

Un attaquant récupère le fichier de contrôle d'un conteneur (stocké dans celui-ci) et le modifie afin de s'ajouter parmi les accès autorisés à ouvrir le conteneur (l'attaquant peut se positionner entre les deux correspondants par exemple). Le bien impacté est donc le fichier de contrôle du conteneur (intégrité) et indirectement les données utilisateur (confidentialité).

L'attaquant peut ainsi intercepter et lire les fichiers échangés entre les correspondants légitimes ou envoyer le conteneur à un correspondant (en usurpant l'identité d'un utilisateur légitime) dans le but de lui faire envoyer des fichiers sensibles.

**M.FIC_CATALOGUE_INT**

Un attaquant récupère le fichier catalogue d'un conteneur (stocké dans celui-ci) et le modifie pour attaquer l'arborescence du conteneur. Le bien impacté est donc le fichier catalogue du conteneur et les données utilisateur (tous en intégrité).

L'attaquant peut ainsi intercepter les conteneurs échangés et faire disparaitre un ou plusieurs des fichiers du conteneur sans que le destinataire ne s'en aperçoive.

**M.DETOURNEMENT_CONTENEUR**

Un attaquant positionné sur le réseau intercepte un conteneur pour le détourner et l'utiliser en tant que vecteur d'attaque (insertion d'un programme malveillant par exemple). L'objectif de l'attaquant n'est alors pas seulement l'accès aux biens sensibles présents dans le conteneur mais peut être aussi la compromission du poste du destinataire (par exemple).

Le bien impacté est donc le conteneur dans sa globalité (intégrité) et indirectement les données utilisateur présentes sur le poste (confidentialité et possiblement intégrité si le programme malveillant est un ransomware par exemple).

| Biens sensibles | Menaces |
|---|---|
| D.DONNEES_UTILISATEUR | M.DETOURN_COMPOSANT<br>M.POLITIQUE_SECU_INT<br>M.FIC_CONTROLE_CONF<br>M.FIC_CONTROLE_INT<br>M.FIC_CATALOGUE_INT<br>M.DETOURNEMENT_CONTENEUR |
| D.CARNET_MOT_DE_PASSE | M.CARNET_CONF |
| D. CONT | M.DETOURNEMENT_CONTENEUR |

| | |
|---|---|
| D.CLE_FIC | M.FIC_CONTROLE_CONF |
| D. FICHIER_CONTROLE | M.FIC_CONTROLE_INT |
| D.FICHIER_CATALOGUE | M.FIC_CATALOGUE_INT |
| D.CONFIGURATION | M.POLITIQUE_SECU_INT |
| D.PROGRAMMES | M.DETOURN_COMPOSANT |

**Tableau 2 Association biens sensibles vers menaces**

Note :

- D.AUTH_USER est couvert par H.NON_OBSERV et H.CONSERVATION_CLES
- D.ID_ADMIN est couvert par l'hypothèse H.CONSERVATION_CLES

## 3.4. Politiques de sécurité organisationnelles

**OSP.CONFIDENTIALITE** La TOE doit offrir un service de protection en confidentialité (chiffrement), automatique et systématique, du stockage et de la transmission (pièce jointe de courrier électronique) des fichiers sensibles des utilisateurs.

**OSP.INTEGRITE** La TOE doit offrir un service de contrôle de l'intégrité (scellement), automatique et systématique, du conteneur.

**OSP.ACCES** La TOE doit permettre aux utilisateurs de fournir une clé d'accès permettant d'accéder aux fichiers sensibles du conteneur auquel ils désirent accéder. S'ils ne peuvent fournir une clé d'accès valide pour le conteneur, l'accès doit être rejeté.

**OSP.RECOUVREMENT** La TOE doit offrir un service de recouvrement des fichiers sensibles des utilisateurs par l'emploi de clés d'accès de recouvrement gérées par l'administrateur de la sécurité. Ces clés sont systématiquement et automatiquement affectées lors de l'initialisation des conteneurs.

**OSP.COLLECTE** La TOE doit offrir un service de collecte d'information dans un fichier protégé pour les opérations de support. Les informations collectées sont sélectionnables parmi les logs, la configuration, les applications installées etc.

**OSP.ADMIN_ACCES** La TOE doit offrir un service de gestion des accès.

**OSP.VERIF_POLICIES** La TOE doit offrir un service (transparent pour l'utilisateur) de vérification de la signature des politiques de sécurité par la clé privée de l'administrateur de sécurité. L'application de toute nouvelle politique est conditionnée par le succès de cette vérification.

**OSP.CRYPTO** Le référentiel de l'ANSSI ([CRYPTO_STD], [CLES_STD] et [AUTH_STD]) défini pour le niveau de résistance standard doit être suivi pour la génération d'aléa, la gestion des clés et pour les mécanismes cryptographiques et d'authentification utilisés dans la TOE.

# 4. OBJECTIFS DE SÉCURITÉ

## 4.1. Objectifs de sécurité pour la TOE

### 4.1.1. Contrôle d'accès

| | |
|---|---|
| **O.AUTH** | La TOE doit permettre d'identifier et authentifier tout utilisateur. Pour cela, la TOE ne doit autoriser l'accès à un fichier d'un conteneur qu'après mise à disposition d'une clé d'accès valide pour ce conteneur. |
| **O.ROLES** | La TOE doit gérer trois rôles d'utilisateurs pour un conteneur chiffrée : un rôle 'utilisateur normal' ou plus simplement 'utilisateur' (utilisation des fichiers du conteneur sous condition de mise à disposition d'une clé d'accès valide), un rôle 'administrateur des accès' en charge de l'initialisation du container avec affectation des accès et un rôle 'administrateur de la sécurité' (installation, signature des politiques, recouvrement en plus de la gestion des accès). |

### 4.1.2. Cryptographie

| | |
|---|---|
| **O.CHIFFREMENT** | La TOE doit chiffrer et déchiffrer les données sensibles par l'emploi de clés cryptographiques. La TOE doit utiliser des clés différentes pour protéger les différents conteneurs ainsi que des vecteurs d'initialisation différents pour chaque fichier d'un conteneur. La TOE doit générer ces clés de chiffrement conformément aux exigences pour le niveau de robustesse standard des référentiels cryptographiques [CRYPTO_STD] et [CLES_STD] de l'ANSSI. |
| **O.SCELLEMENT** | La TOE doit pouvoir contrôler l'intégrité des conteneurs par l'emploi de clés cryptographiques différentes pour chaque conteneur. La TOE doit générer ces clés de scellement conformément aux exigences pour le niveau de robustesse standard des référentiels cryptographiques [CRYPTO_STD] et [CLES_STD] de l'ANSSI. |
| **O.EFFACEMENT_CLES** | La TOE doit assurer le nettoyage des traces de données sensibles (clés de chiffrement des fichiers, éléments permettant de retrouver les clés d'accès) dans la mémoire (RAM) dès la fin des opérations réalisées par la TOE. |
| **O.ALGO_STD** | La TOE doit produire des aléas et fournir un choix d'algorithmes cryptographiques et de tailles de clés conformes aux standards de ce domaine, prévus dans [CRYPTO_STD] et complétés par [CLES_STD]. |

### 4.1.3. Gestion des conteneurs

| | |
|---|---|
| **O.ADM_ACCES** | La TOE doit offrir une interface à l'administrateur de la sécurité et l'administrateur des accès permettant de visualiser les accès et gérer les clés d'accès aux conteneurs. L'utilisateur peut seulement visualiser les accès. |
| **O.COMPATIBILITE** | La TOE doit assurer l'ouverture sécurisée des conteneurs créés avec une ancienne version de Zed! et dont les éléments faisaient l'objet d'un contrôle d'intégrité par partie. |
| **O.RECOUVREMENT** | La TOE doit permettre d'affecter des clés d'accès de recouvrement. |
| **O.COLLECTE** | La TOE doit permettre de collecter de manière sécurisée des informations utiles aux opérations de support. |

### 4.1.4. Protections lors de l'exécution

| | |
|---|---|
| **O.INT_POLICIES** | La TOE doit vérifier la signature de toutes nouvelles politiques de sécurité à appliquer. En cas d'échec lors de la vérification, les politiques appliquées restent inchangées. |

## 4.2. Objectifs de sécurité pour l'environnement

### 4.2.1. Pendant l'utilisation

| | |
|---|---|
| **OE.NON_OBSERV** | L'utilisateur ne doit accéder à ses données sensibles que lorsqu'il se trouve dans un environnement de confiance (lorsqu'il se trouve seul ou avec des personnes ayant le besoin d'en connaître). Les mots de passe partagés entre les correspondants doivent être échangés à travers des canaux organisationnels protégés. |
| **OE.ENV_OPERATIONNEL** | Lorsque l'utilisateur est authentifié, l'environnement opérationnel doit assurer la confidentialité des données sensibles et des données d'authentification.<br><br>Note d'application :<br>L'équipement doit apporter des protections efficaces contre l'écoute illicite et la transmission non autorisée de données (pare-feu correctement configuré, antivirus avec base de données à jour, « anti-spyware », etc.).<br>Les applications installées sur l'équipement ne doivent pas perturber le bon fonctionnement de la TOE. |
| **OE.HORODATAGE** | L'environnement d'exploitation fournit à la TOE un horodatage de qualité pour lui permettre d'assurer correctement les fonctions nécessitant une date et une heure exacte (traçage des événements de sécurité notamment). |
| **OE.ENV_ALEA** | L'environnement d'exploitation fournit à la TOE des données lui permettant de mettre en œuvre des mécanismes pour fournir les aléas nécessaires à la génération des secrets. |

### 4.2.2. Formation des utilisateurs et des administrateurs

| | |
|---|---|
| **OE.FORMATION** | L'administrateur des accès et les utilisateurs de la TOE doivent être formés à l'utilisation de la TOE et sensibilisés à la sécurité informatique (ceci prend en compte la sensibilisation sur la qualité des clés d'accès et de leur support lorsqu'elles sont hébergées par un porte-clés).<br>L'administrateur de la sécurité doit recevoir une formation adaptée à cette fonction. |
| **OE.CRYPTO_EXT** | L'administrateur de la sécurité doit être sensibilisé sur la qualité des clés d'accès qu'il apporte à la TOE afin que ces clés soient conformes à l'état de l'art dans leur implémentation. Il doit également être sensibilisé à la qualité du support de ces clés lorsqu'elles sont hébergées par un porte-clés externe. |
| **OE.CONSERV_CLES** | Les utilisateurs doivent conserver, dans un lieu sûr, les clés d'accès qui leurs ont été transmises par un administrateur de la sécurité et empêcher leur divulgation. L'administrateur de la sécurité doit |

conserver ses clés de recouvrement et son bi-clé de signature dans un lieu sûr et empêcher leur divulgation.

### 4.2.3. Administration

| | |
|---|---|
| **OE.CONFIANCE_ADMIN** | L'administrateur des accès et l'administrateur de la sécurité doivent être des personnes de confiance. Les administrateurs Windows sont des personnes de confiance en charge de la configuration (avec des valeurs sures) des « policies ».<br><br>Si les correspondants appartiennent à des entités gérés par des administrateurs de la sécurité différents, ceux-ci doivent garantir ensemble l'utilisation de politiques de sécurité conformes au niveau recherché (force des mots de passe notamment). |
| **OE.ACCES** | L'administrateur de la TOE est chargé de la création des listes d'accès et de leur stockage sur un partage dédié bénéficiant des mêmes mesures de sécurité que les postes utilisateur et dont l'accès en écriture est uniquement autorisé à l'administrateur de la TOE. Il doit, lors de la fourniture des clés d'accès possédant un certificat X509, vérifier la validité de ces certificats et leur adéquation avec l'usage qui en est fait par la TOE. Cette exigence s'applique en particulier aux certificats racines dits «authenticode» à partir desquels la vérification d'intégrité de la TOE peut être effectuée. |
| **OE.ENV_PROTECT_TOE** | L'environnement technique de la TOE assure l'intégrité des composantes de la TOE et notamment ses programmes. L'administration et la mise à jour de la TOE sont assurées par les administrateurs habilitées. |

# 5. EXIGENCES DE SÉCURITÉ DES TI

## 5.1. Exigences de sécurité de la TOE

Dans cette section, les exigences de sécurité de la TOE ont été traduites en français afin d'améliorer leur compréhension. Le texte officiel servant de référence se trouve dans l'annexe A. Dans le texte français, toutes les opérations sur les composants (assignation, sélection, itération et raffinement) sont représentées par des caractères en italiques (et en caractères gras pour la partie servant de référence).

### 5.1.1. Exigences fonctionnelles de sécurité de la TOE

Les composants fonctionnels sélectionnés pour répondre aux objectifs de sécurité de la TOE sont les suivants :

| Composants CC retenus | |
|---|---|
| FCS_CKM.1 | Génération de clés cryptographiques |
| FCS_CKM.3 | Accès aux clés cryptographiques |
| FCS_CKM.4 | Destruction de clés cryptographiques |
| FCS_COP.1 | Opération cryptographique |
| FDP_ACC.1 | Contrôle d'accès partiel |
| FDP_ACF.1 | Contrôle d'accès basé sur les attributs de sécurité |
| FDP_ITC.1 | Importation depuis une zone hors du contrôle de la TSF |
| FDP_RIP.1 | Protection d'une partie des informations résiduelles |
| FDP_SDI.2 | Contrôle de l'intégrité des données stockées et action |
| FIA_AFL.1 | Gestion d'une défaillance de l'authentification |
| FIA_UAU.2 | Authentification d'un utilisateur préalablement à toute action |
| FIA_UID.2 | Identification d'un utilisateur préalablement à toute action |
| FMT_MOF.1 | Administration des fonctions de la TSF |
| FMT_MSA.1 | Gestion des attributs de sécurité |
| FMT_MSA.2 | Attributs de sécurité sûrs |
| FMT_MSA.3 | Initialisation statique d'attribut |
| FMT_MTD.1 | Gestion des données de la TSF |
| FMT_SMF.1 | Spécification des fonctions d'administration |
| FMT_SMR.1 | Rôles de sécurité |
| FPT_TST.1 | Tests de la TSF |

**Tableau 3 : Exigences fonctionnelles de sécurité pour la TOE**

#### 5.1.1.1. Introduction

Les exigences fonctionnelles de sécurité (SFR) font référence aux sujets suivants:

- Administrateur de la sécurité, administrateur des accès et utilisateur de la TOE avec comme attributs de sécurité leur rôle et leur clé d'accès permettant ou non d'effectuer les opérations sur les conteneurs.

Les exigences fonctionnelles de sécurité (SFR) font référence aux objets suivants:

- Conteneurs manipulés par les utilisateurs de la TOE et qui contiennent les données sensibles des utilisateurs (fichiers, clés),

Les exigences fonctionnelles de sécurité (SFR) font référence aux opérations suivantes:

- Gestion des conteneurs (politiques, recouvrement, gestion des accès, collecte d'information)
- Utilisation des conteneurs

#### 5.1.1.2. Classe FCS : Support Cryptographique

| | |
|---|---|
| **FCS_CKM** | **Gestion des clés cryptographiques** |
| FCS_CKM.1 | Génération des clés cryptographiques |
| FCS_CKM.1.1 | La TSF doit générer les clés cryptographiques conformément à un algorithme de génération de clés cryptographiques spécifié parmi les suivants<br>- *génération de nombres pseudo-aléatoires utilisés pour la génération des clés de chiffrement et des clés RSA de listes d'accès en utilisant les générateurs Hash_DRBG, HMAC_DRBG ou CTR_DRBG décrit dans la publication « Recommendation for Random Number Generation Using Deterministic Random Bit Generators » (référence SP 800-90A révision 1) du NIST ;*<br>- *diversification de clés PKCS#5 à partir des mots de passe*<br>- *Génération de mot de passe par le carnet de mot de passe pour en dériver une clé*<br>et à des tailles de clés cryptographiques *de 128, 192 et 256 bits pour les clés symétriques et de 2048 à 4096 bits pour les clés asymétriques* qui satisfont *aux exigences cryptographiques de l'ANSSI définies dans [CRYPTO_STD] et [CLES_STD].* |
| FCS_CKM.3 | Accès aux clés cryptographiques |
| FCS_CKM.3.1 | La TSF doit réaliser *l'utilisation de clés* conformément à une méthode d'accès aux clés cryptographiques spécifiée *par utilisation du driver clavier et déchiffrement (déwrapping) des clés par la clé d'accès.* |
| FCS_CKM.4 | Destruction de clés cryptographiques |
| FCS_CKM.4.1 | La TSF doit détruire les clés cryptographiques conformément à une méthode de destruction spécifiée de clés cryptographiques *par réécriture de motifs composés de zéros suivi par une opération de lecture-vérification. Si la lecture-vérification de la réécriture échoue, le processus doit être répété.* |
| **FCS_COP** | **Opération cryptographique** |
| FCS_COP.1 | Opération cryptographique |
| FCS_COP.1.1 | La TSF doit exécuter *le hachage, le calcul et la vérification d'intégrité, le chiffrement, le déchiffrement, la vérification de la signature des politiques de sécurité, la génération de clés,* |

| | |
|---|---|
| | *le wrapping et dewrapping de clés et la dérivation de clés* conformément à un algorithme cryptographique spécifié *HMAC, SHA-256 et SHA-512, RSA PKCS#1 v2.2, AES mode CBC* et avec des tailles de clés cryptographiques *de 128, 192 et 256 bits pour les clés symétriques et de 2048 bits à 4096 bits pour les clés asymétriques* qui satisfont à ce qui suit: exigences cryptographique de l'*ANSSI définies dans [CRYPTO_STD] et [CLES_STD].* |

#### 5.1.1.3. Classe FDP : Protection des données de l'utilisateur

| | |
|---|---|
| **FDP_ACC** | **Politique de contrôle d'accès** |
| FDP_ACC.1 | Contrôle d'accès partiel |
| FDP_ACC.1.1 | La TSF doit appliquer la politique *SFP.ACCESS_OBJ* aux :<br>*Sujets: Administrateurs et utilisateurs de la TOE*<br>*Objets: Conteneurs contenant les fichiers utilisateur et le fichier de contrôle.*<br>*Opérations : Gestion des conteneurs et utilisation.* |
| **FDP_ACF** | **Fonctions de contrôle d'accès** |
| FDP_ACF.1 | Contrôle d'accès basé sur les attributs de sécurité |
| FDP_ACF.1.1 | La TSF doit appliquer la politique *SFP.ACCESS_OBJ* aux objets en fonction des :<br>*Sujets: Administrateurs et utilisateurs de la TOE*<br>*Attributs de sécurité : Clés d'accès utilisateur permettant ou non d'ouvrir le conteneur et rôle.* |
| FDP_ACF.1.2 | La TSF doit appliquer les règles suivantes pour déterminer si une opération entre des sujets contrôlés et des objets contrôlés est autorisée :<br>*Objet : Conteneur*<br>*Opération: Gestion des conteneurs et utilisation*<br>*Règle : authentification réussie après mise à disposition de la clé d'accès associée au conteneur concerné avec accès à la gestion du conteneur uniquement pour le rôle administrateur des accès (gestion des accès) et administrateur de la sécurité (toute opération de gestion).* |
| FDP_ACF.1.3 | La TSF doit autoriser explicitement l'accès de sujets à des objets en fonction des règles complémentaires suivantes : *Aucune*. |
| FDP_ACF.1.4 | La TSF doit refuser explicitement l'accès de sujets à des objets en fonction des règles complémentaires suivantes : *Aucune*. |
| **FDP_ITC** | **Importation depuis une zone hors du contrôle de la TSF** |
| FDP_ITC.1 | Importation de données utilisateur sans attributs de sécurité |
| FDP_ITC.1.1 | La TSF doit appliquer *la politique de sécurité des fonctions (SFP) SFP.ACCESS_OBJ* lors de l'importation de données utilisateur, contrôlées par les SFP, en provenance de l'extérieur de la TOE. |
| FDP_ITC.1.2 | La TSF doit ignorer tout attribut de sécurité associé aux données utilisateur lorsqu'elles sont importées depuis l'extérieur de la TOE. |
| FDP_ITC.1.3 | La TSF doit appliquer les règles suivantes lors de l'importation des données utilisateur contrôlées par la SFP en provenance de l'extérieur de la TOE : *Aucune* |
| **FDP_RIP** | **Protection des informations résiduelles** |
| FDP_RIP.1 | Protection d'une partie des informations résiduelles |
| FDP_RIP.1.1 | La TSF doit garantir que toute information contenue précédemment dans une ressource est rendue inaccessible lors de *la désallocation de la ressource* des objets *clés de chiffrement* |

| | |
|---|---|
| | *des conteneurs et clés d'accès.* |
| **FDP_SDI** | **Intégrité des données stockées** |
| FDP_SDI.2 | Contrôle de l'intégrité des données stockées et action à entreprendre |
| FDP_SDI.2.1 | La TSF doit contrôler les données de l'utilisateur stockées au sein de conteneurs contrôlés par la TSF à la recherche des *erreurs d'intégrité* sur tous les objets, en fonction des attributs suivants *HMAC global ou par partie du conteneur.* |
| FDP_SDI.2.2 | En cas de détection d'une erreur d'intégrité, la TSF doit *afficher un message d'erreur et interdire l'ouverture du conteneur*.<br>*Raffinement non éditorial :*<br>*La version Q.2021.1 effectue le contrôle d'intégrité sur la globalité du conteneur, les anciennes versions de Zed ! effectuent le contrôle d'intégrité sur les différentes parties du conteneur.*<br>*Le contrôle du HMAC global s'effectue au moment de l'ouverture du conteneur. Si le conteneur contient une protection en intégrité par partie, les fichiers techniques et le watermark sont contrôlés au moment de l'ouverture du conteneur, les fichiers utilisateurs sont contrôlés lorsqu'ils sont ouverts.* |

#### 5.1.1.4. Classe FIA : Identification et authentification

| **FIA_AFL** | **Défaillances de l'authentification** |
|---|---|
| FIA_AFL.1 | Gestion d'une défaillance de l'authentification |
| FIA_AFL.1.1 | La TSF doit détecter le fait que *cinq* tentatives d'authentification infructueuse ont eu lieu en relation avec *l'ouverture d'un conteneur.*<br>Note : il a été nécessaire d'effectuer un raffinement éditorial afin de rendre le texte correct. |
| FIA_AFL.1.2 | Quand le nombre spécifié de tentatives d'authentification infructueuses a été atteint ou dépassé, la TSF doit *temporiser l'accès à ce conteneur.* |

| **FIA_UAU** | **Authentification de l'utilisateur** |
|---|---|
| FIA_UAU.2 | Authentification d'un utilisateur préalablement à toute action |
| FIA_UAU.2.1 | La TSF doit exiger que chaque utilisateur soit authentifié avec succès avant d'autoriser toute autre action transitant par la TSF pour le compte de cet utilisateur. |

| **FIA_UID** | **Identification de l'utilisateur** |
|---|---|
| FIA_UID.2 | Identification d'un utilisateur préalablement à toute action |
| FIA_UID.2.1 | La TSF doit exiger que chaque utilisateur soit identifié avec succès avant d'autoriser toute autre action transitant par la TSF pour le compte de cet utilisateur. |

#### 5.1.1.5. Classe FMT : Administration de la sécurité

| **FMT_MOF** | **Administration des fonctions de la TSF** |
|---|---|
| FMT_MOF.1 | Administration du comportement des fonctions de sécurité |
| FMT_MOF.1.1 | La TSF doit restreindre l'aptitude d'*activer ou désactiver* les fonctions de *collecte d'information et de recouvrement* à *l'administrateur de la sécurité.* |

| **FMT_MSA** | **Administration des attributs de sécurité** |
|---|---|

| | |
|---|---|
| FMT_MSA.1 | Gestion des attributs de sécurité |
| FMT_MSA.1.1 | La TSF doit mettre en œuvre la *politique SFP.ACCESS_OBJ* pour restreindre aux *administrateurs des accès et à l'administrateur de la sécurité* la possibilité de *changer la valeur par défaut, modifier ou supprimer* les attributs de sécurité *clés d'accès et rôles*. |
| FMT_MSA.2 | Attributs de sécurité sûrs |
| FMT_MSA.2.1 | La TSF doit garantir que seules des valeurs sûres sont acceptées pour *les clés d'accès.* |
| FMT_MSA.3 | Initialisation statique d'attribut |
| FMT_MSA.3.1 | La TSF doit mettre en œuvre *la politique SFP.ACCESS_OBJ* afin de fournir des valeurs par défaut *restrictives* pour les attributs de sécurité qui sont utilisés pour appliquer la SFP. |
| FMT_MSA.3.2 | La TSF doit permettre à *l'administrateur des accès et à l'administrateur de sécurité* de spécifier des valeurs initiales alternatives pour remplacer les valeurs par défaut lorsqu'un objet ou une information est créé. |
| **FMT_MTD** | **Gestion des données de la TSF** |
| FMT_MTD.1 | Gestion des données de la TSF |
| FMT_MTD.1.1 | La TSF doit restreindre, à l'*administrateur de la sécurité*, la possibilité de *changer la valeur par défaut, modifier ou supprimer* les *stratégies de sécurité*.<br>*Raffinement non éditorial :*<br>*Ce composant est implémenté par la signature des politiques de sécurité*. |
| **FMT_SMF** | **Spécification des fonctions d'administration** |
| FMT_SMF.1 | Spécification des fonctions d'administration |
| FMT_SMF.1.1 | La TSF doit être capable d'exécuter les fonctions d'administration suivantes :<br>- *Les fonctions de gestion des accès*<br>- *La fonction de recouvrement*<br>- *La fonction de collecte d'information pour le support* |
| **FMT_SMR** | **Rôle pour l'administration de la sécurité** |
| FMT_SMR.1 | Rôles de sécurité |
| FMT_SMR.1.1 | La TSF doit tenir à jour les rôles *administrateur de la sécurité, administrateur des accès et utilisateur de la TOE*. |
| FMT_SMR.1.2 | La TSF doit être capable d'associer les utilisateurs aux rôles |

#### 5.1.1.6. Class FPT: Protection de la TSF

| | |
|---|---|
| **FPT_TST.1** | **Auto tests de la TSF** |
| FPT_TST.1 | Tests de la TSF |
| FPT_TST.1.1 | La TSF doit exécuter une suite d'auto tests *lors du démarrage initial, périodiquement en utilisation courante* pour démontrer que la TSF fonctionne correctement. |

| | |
|---|---|
| FPT_TST.1.2 | La TSF doit fournir aux utilisateurs autorisés la possibilité de vérifier l'intégrité *d'aucune partie des données de la TSF.* |
| FPT_TST.1.3 | La TSF doit fournir aux utilisateurs autorisés la possibilité de vérifier l'intégrité *d'aucune partie de la TSF.* |

### 5.1.2. Exigences d'assurance de sécurité de la TOE

Comme indiqué au paragraphe 3.3, la TOE doit être résistante aux attaques de pénétration effectuées par un attaquant ayant un potentiel d'attaque « enhanced-basic ».

Le niveau d'assurance visé par la TOE est le niveau :

**EAL3 augmenté des composants ALC_FLR.3 et AVA_VAN.3 associé à une expertise de l'implémentation de la cryptographie décrite dans [QUALIF_STD].**

Ce qui correspond à la sélection des composants d'assurance suivants :

| Composant | | Commentaire |
|---|---|---|
| ADV_ARC.1 | Security architecture description | EAL3 |
| ADV_FSP.3 | Functional specification with complete summary | EAL3 |
| ADV_TDS.2 | Architectural design | EAL3 |
| AGD_OPE.1 | Operational user guidance | EAL3 |
| AGD_PRE.1 | Preparative procedures | EAL3 |
| ALC_CMC.3 | Authorisation controls | EAL3 |
| ALC_CMS.3 | Implementation representation CM coverage | EAL3 |
| ALC_DEL.1 | Delivery procedures | EAL3 |
| ALC_DVS.1 | Identification of security measures | EAL3 |
| ALC_FLR.3 | Systematic flaw remediation | + |
| ALC_LCD.1 | Developer defined life-cycle model | EAL3 |
| ASE_CCL.1 | Conformance claims | EAL3 |
| ASE_ECD.1 | Extended components definition | EAL3 |
| ASE_INT.1 | ST introduction | EAL3 |
| ASE_OBJ.2 | Security objectives | EAL3 |
| ASE_REQ.2 | Security requirements | EAL3 |
| ASE_SPD.1 | Security problem definition | EAL3 |
| ASE_TSS.1 | TOE summary specification | EAL3 |
| ATE_COV.2 | Analysis of coverage | EAL3 |
| ATE_DPT.1 | Testing: basic design | EAL3 |
| ATE_FUN.1 | Functional testing | EAL3 |
| ATE_IND.2 | Independent testing - sample | EAL3 |
| AVA_VAN.3 | Focused vulnerability analysis | + |

**Tableau 4 : Composants d'assurance de sécurité**

Ce niveau d'assurance respecte les dépendances entre les composants d'assurance CC mentionnés dans la Partie 3 des Critères Communs.

# 6. SPÉCIFICATIONS GLOBALES DE LA TOE

Les fonctions de sécurité réalisées par la TOE sont décrites dans ce chapitre.

**F.CONTROLE_ACCES** **Contrôle d'accès au conteneur**

Cette fonction de sécurité constitue l'interface réalisant le contrôle d'accès obligatoire pour ouvrir le conteneur contrôlé par la TOE. La TSF autorise ou refuse l'accès à un conteneur chiffré sur la base de la vérification d'un couple identifiant/authentifiant fourni par l'utilisateur de la TOE. Une temporisation est appliquée après cinq échecs d'authentification consécutifs.

**F.ENTREE_SECURISEE** **Entrée sécurisée**

Cette fonction de sécurité recouvre la communication sécurisée de données fournies en entrée de la TOE en utilisant pour cela des fonctions de chiffrement et déchiffrement de clé de conteneur et le driver clavier quand il s'agit d'entrer un mot de passe ou un code de fichier de clés.

**F.CONFIGURATION_TOE** **Modification de la configuration de la TOE**

Cette fonction de sécurité couvre l'ensemble des opérations de configuration de la TOE (initialisation et modification) et assure que seules des valeurs sûres de paramètres de configuration peuvent être utilisées. Les données de configuration concernent les «« policies »» de Windows (Group Policies ou mode alternatif) qui sont signées par l'administrateur de sécurité et exploitées par la TOE après vérification de leur signature. Ces données définissent notamment les types d'accès supportés, les algorithmes utilisés (AES 256 bits par défaut), la force des mots de passe, le contrôle des certificats. Si la vérification est correcte, le poste est mis en conformité avec les nouvelles politiques.

**F.GESTION_CLES_ACCES** **Gestion des clés**

Cette fonction de sécurité gère les attributs de sécurité que sont les clés d'accès des utilisateurs et les rôles (utilisateur, administrateur de la sécurité ou administrateur des accès) qui leur sont associés. Un accès correspond à une clé d'accès (une clé cryptographique) que possède un utilisateur et permettant d'obtenir les éléments de chiffrement / déchiffrement du conteneur ainsi que l'ouverture du carnet de mot de passe. Si ces éléments sont extraits pour effectuer des opérations de gestion des accès, la clé d'accès présentée doit être associée au rôle administrateur des accès (ou administrateur de la sécurité). Cette fonction gère également l'accès de recouvrement qui est un accès particulier.

Enfin la fonction F.GESTION_CLES_ACCES réalise également les opérations de génération des clés RSA des listes d'accès et d'ajout des clés d'accès ainsi que les opérations d'accès à ces clés (par l'intermédiaire des tokens pkcs#11 notamment). Elle assure le nettoyage de ces clés en mémoire après un verrouillage de session ou une mise en veille, une fermeture de session ou une fermeture du poste.

**F.OPERATIONS_CRYPTO** **Implémentation des opérations cryptographiques**

Cette fonction de sécurité couvre la génération des clés de chiffrement et de scellement, le chiffrement et déchiffrement des fichiers du conteneur, les opérations liées au contrôle de l'intégrité des conteneurs ainsi que l'ensemble des opérations cryptographiques mises au service des autres fonctions de sécurité.

La fonction effectue des auto tests au démarrage et de manière périodique pour vérifier le bon fonctionnement des algorithmes et du générateur aléatoire.

# 7. ANNONCES DE CONFORMITE A UN PP

Cette cible de sécurité ne déclare aucune conformité à un Profil de Protection.

# 8. ARGUMENTAIRE

## 8.1. Argumentaire pour les objectifs de sécurité

Cette section présente les liens de couverture entre les objectifs de sécurité et les éléments qui constituent la définition de l'environnement de la TOE (hypothèses, politiques de l'organisation et menaces).

### 8.1.1. Hypothèses

Le tableau ci-dessous présente la couverture des hypothèses retenues par les objectifs de sécurité :

| | | OE.NON_OBSERV | OE.ENV_OPERATIONNEL | OE.HORODATAGE | OE.CONFIANCE_ADMIN | OE.CONSERV_CLES | OE.ACCES | OE.ENV_PROTECT_TOE | OE.FORMATION | OE.ENV_ALEA | OE.CRYPTO_EXT |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Hypothèses | H.NON_OBSERV | X | | | | | | | | | |
| | H.POSTE_SUR | | X | | | | | | | | |
| | H.CONFIANCE_ADMIN | | | | X | | | | X | | |
| | H.CONSERVATION_CLES | | | | | X | | | X | | |
| | H.ACCES | | | | | | X | | | | |
| | H.ENV_PROTECT_TOE | | | | | | | X | X | | |
| | H.FIDELE_ENV | | | X | | | | | | | |
| | H.ENV_ALEA | | | | | | | | | X | |
| | H.CRYPTO_EXT | | | | | | | | | | X |

**Tableau 5 : Couverture des hypothèses par les objectifs de sécurité**

| | |
|---|---|
| **H.NON_OBSERV** | L'environnement physique de la TOE permet aux utilisateurs d'entrer leur mot de passe ou code PIN sans être observable directement et sans que cela puisse être intercepté (clavier sans fil…) par d'autres utilisateurs ou attaquants potentiels. |

L'objectif OE.NON_OBSERV couvre directement cette hypothèse en mettant à disposition de l'utilisateur un environnement physique adéquat.

| | |
|---|---|
| **H.POSTE_SUR** | L'environnement opérationnel ne permet pas à un attaquant d'accéder au disque (physiquement ou par le réseau) lorsque des données sensibles sont accessibles à un utilisateur légitime sur son poste. Le poste utilisateur doit donc apporter des protections efficaces contre l'accès illicite distant (pare-feu correctement configuré, antivirus et anti logiciels espions avec bases de données à jour etc.). |

L'objectif OE.ENV_OPERATIONNEL couvre directement cette hypothèse en mettant à disposition de l'utilisateur un environnement opérationnel adéquat.

| | |
|---|---|
| **H.CONFIANCE_ADMIN** | L'administrateur de la sécurité et l'administrateur des accès sont des personnes de confiance. Les administrateurs Windows sont des personnes de confiance en charge de la configuration (avec des valeurs sures) des « policies ». Ils sont formés à l'utilisation de la TOE tout comme les utilisateurs.<br><br>Les administrateurs de la sécurité doivent garantir (ensemble si ils appartiennent à des organisations différentes) l'utilisation de politiques de sécurité conformes au niveau recherché (force des mots de passe notamment). |

Les objectifs OE.CONFIANCE_ADMIN et OE.FORMATION couvrent directement cette hypothèse en employant des personnes de confiance et en leur apportant la formation nécessaire.

| | |
|---|---|
| **H.CONSERVATION_CLES** | Les utilisateurs sont chargés de la conservation dans un lieu sûr et de la non divulgation des clés d'accès qui leurs ont été transmises par l'administrateur de sécurité. L'administrateur de la sécurité est également chargé de conservation dans un lieu sûr et de la non divulgation des clés de recouvrement et de son bi-clé de signature. |

Les objectifs OE.CONSERV_CLES et OE.FORMATION couvrent cette hypothèse en responsabilisant et en sensibilisant les utilisateurs et l'administrateur.

| | |
|---|---|
| **H.ACCES** | L'administrateur de la TOE est chargé de la création des listes d'accès et de leur stockage sur un partage dédié bénéficiant des mêmes mesures de sécurité que les postes utilisateur et dont l'accès en écriture est uniquement autorisé à l'administrateur de la TOE. Il est également chargé, lors de la fourniture des clés d'accès possédant un certificat X509, de vérifier la validité de ces certificats et leur adéquation avec l'usage qui en est fait par la TOE. |

L'objectif OE.ACCES couvre directement cette hypothèse.

| | |
|---|---|
| **H.ENV_PROTECT_TOE** | L'environnement technique de la TOE assure l'intégrité des composantes de |

| | |
|---|---|
| | la TOE. L'administration et la mise à jour de la TOE sont assurées par des personnes formées et habilitées. |

Les objectifs OE.ENV_PROTECT_TOE et OE.FORMATION couvrent cette hypothèse en assurant l'intégrité des programmes de la TOE et en formant les administrateurs.

| | |
|---|---|
| **H.FIDELE_ENV** | L'environnement d'exécution fournit à la TOE une date et une heure exacte pour assurer les fonctions d'horodatage. |

L'objectif OE.HORODATAGE couvre directement cette hypothèse.

| | |
|---|---|
| **H.ENV_ALEA** | L'environnement d'exécution fournit à la TOE des mécanismes (événements partiellement imprédictible) pour produire les aléas nécessaires à la génération des secrets. |

L'objectif OE.ENV_ALEA couvre directement cette hypothèse.

| | |
|---|---|
| **H.CRYPTO_EXT** | Les clés d'accès générées ou stockées à l'extérieur de la TOE doivent être conformes aux documents [CRYPTO_STD] et [CLES_STD] pour le niveau standard. |

L'objectif OE.CRYPTO_EXT couvre directement cette hypothèse.

### 8.1.2. Menaces

Le tableau ci-dessous présente les liens de couverture entre les objectifs de sécurité et les menaces retenues :

| | | O.AUTH | O.ROLES | O.CHIFFREMENT | O.SCELLEMENT | O.EFFACEMENT_CLES | O.ALGO_STD | O.ADM_ACCES | O.COMPATIBILITE | O.RECOUVREMENT | O.COLLECTE | O.INT_POLICIES | OE.CONFIANCE_ADMIN | OE.CONSERV_CLES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Menaces | M.DETOURN_COMPOSANT | X | | | | X | X | | | | | | | |
| | M.POLITIQUE_SECU_INT | | | | | | | | | | | X | X | X |
| | M.CARNET_CONF | X | | | | | X | | | | | | | |
| | M.FIC_CONTROLE_CONF | X | | X | | | X | | | | | | | |
| | M.FIC_CONTROLE_INT | X | | X | X | | X | | | | | | | |
| | M.FIC_CATALOGUE_INT | X | | X | X | | X | | | | | | | |
| | M.DETOURNEMENT_CONTENEUR | X | | | X | | X | | | | | | | |

**Tableau 6 : Couverture des menaces par les objectifs de sécurité**

---

**M.DETOURN_COMPOSANT**

En possession d'un conteneur intercepté, un attaquant se procure le produit Zed! Entreprise (ou ZoneCentral) et met en œuvre, éventuellement à bas niveau, les composants internes de la TOE, pour contourner certaines fonctions de sécurité en provoquant ou profitant d'un dysfonctionnement. L'attaquant peut pour cela effectuer du «reverse-ingeniering» sur le programme, développer des programmes d'appel des fonctions internes de la TOE, ou s'aider d'un debugger. Le bien impacté est le programme de la TOE (confidentialité et intégrité) et les données utilisateur (confidentialité).

Il ne doit pas pouvoir, avec ces moyens, réussir à «pénétrer» le conteneur dans laquelle il n'aurait pas normalement accès.

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait qu'avant toute opération sur la TOE, une authentification est nécessaire (O.AUTH),

➔ Pour se protéger la TOE doit :

- Garantir le fait qu'il n'est pas possible, cryptographiquement, de retrouver les clés de chiffrement du conteneur sans fournir une clé d'accès valide : le détournement d'un composant (i.e. sa mise en œuvre de façon détournée ou non prévue) ne peut pas permettre de franchir cette barrière (O.AUTH et O.ALGO_STD),
- Garantir le fait qu'un composant détourné ne conserve pas de résidus de clé permettant de présenter un chemin pour une attaque (O.EFFACEMENT_CLES).

➔ Pour limiter l'impact de la menace, la TOE doit :

Rien

| | |
|---|---|
| **M.POLITIQUE_SECU_INT** | Un attaquant signe des politiques à la place de l'administrateur de sécurité (politiques domaines ou politiques locales si accès au poste). Elle peut par exemple configurer son propre accès de recouvrement qui sera automatiquement ajouté lorsque l'utilisateur créera ses prochains conteneurs. Le bien impacté est la configuration (intégrité) et indirectement les données utilisateur (confidentialité). |

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait que les administrateurs (dont les administrateurs Windows) sont des personnes de confiance (OE.CONFIANCE_ADMIN) ;
- Garantir le fait que l'administrateur de sécurité conserve sa clé privée de signature dans un lieu sûr (OE.CONSERV_CLES)

➔ Pour se protéger la TOE doit :

- Garantir le fait qu'il n'est pas possible d'appliquer des politiques de sécurité (et donc modifier le fichier des politiques) sans qu'elles soient signées par le responsable de sécurité (O.INT_POLICIES)

➔ Pour limiter l'impact de la menace, la TOE doit :

Rien

| | |
|---|---|
| **M.CARNET_CONF** | Un attaquant récupère le carnet de mot de passe stocké sur le poste de l'utilisateur pour tenter de retrouver les mots de passe des correspondants. Les biens impactés sont donc les mots de passe des correspondants et par suite les données utilisateur stockées dans les conteneurs possédant ces accès (tous en confidentialité).<br><br>Par exemple, l'attaquant tente de déchiffrer les mots de passe contenus dans le carnet afin de les utiliser pour ouvrir les anciens et futurs conteneurs échangés. |

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait qu'avant toute opération sur la TOE, une authentification est nécessaire (O.AUTH),

➔ Pour se protéger la TOE doit :

- Garantir le fait qu'il n'est pas possible, cryptographiquement, de retrouver les mots de passe du carnet de mot de passe sans fournir une clé d'accès valide, et par le fait que cet objectif prévoit que le carnet de mot de passe de la TOE respecte également ce principe (O.AUTH et O.ALGO_STD).

➔ Pour limiter l'impact de la menace, la TOE doit :

- Rien

| | |
|---|---|
| **M.FIC_CONTROLE_CONF** | Un attaquant récupère le fichier de contrôle d'un conteneur pour tenter de retrouver des informations protégées. Les biens impactés sont donc les clés de chiffrement et les données utilisateur (tous en confidentialité).<br><br>Par exemple, l'attaquant tente de retrouver des informations protégées (telles que les clés de chiffrement) à partir des fichiers |

| | |
|---|---|
| | chiffrées du conteneur et du fichier de contrôle de la TOE ou bien essaye de déchiffrer directement les informations contenues dans le fichier de contrôle. |

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait qu'avant toute opération sur la TOE, une authentification est nécessaire (O.AUTH),

➔ Pour se protéger la TOE doit :

- Garantir le fait qu'il n'est pas possible, cryptographiquement, de retrouver les clés de chiffrement du conteneur sans fournir une clé d'accès valide, et par le fait que cet objectif prévoit que le fichier de contrôle de la TOE respecte également ce principe (O.AUTH et O.ALGO_STD).

➔ Pour limiter l'impact de la menace, la TOE doit :

- Garantir le fait que les fichiers internes des différents conteneurs sont rendus «cryptographiquement différents» par l'utilisation d'aléas ne permettant pas de tirer des enseignements d'un fichier de contrôle pour en attaquer un autre (O.CHIFFREMENT et O.ALGO_STD).

| | |
|---|---|
| **M.FIC_CONTROLE_INT** | Un attaquant récupère le fichier de contrôle d'un conteneur (stocké dans celui-ci) et le modifie afin de s'ajouter parmi les accès autorisés à ouvrir le conteneur (l'attaquant peut se positionner entre les deux correspondants par exemple). Le bien impacté est donc le fichier de contrôle du conteneur (intégrité) et indirectement les données utilisateur (confidentialité).<br><br>L'attaquant peut ainsi intercepter et lire les fichiers échangés entre les correspondants légitimes ou envoyer le conteneur à un correspondant (en usurpant l'identité d'un utilisateur légitime) dans le but de lui faire envoyer des fichiers sensibles. |

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait qu'avant toute opération sur la TOE, une authentification est nécessaire (O.AUTH),

➔ Pour se protéger la TOE doit :

- Garantir le fait qu'il n'est pas possible, cryptographiquement, de retrouver les clés de chiffrement du conteneur sans fournir une clé d'accès valide : le détournement du fichier de contrôle (i.e. sa mise en œuvre de façon détournée ou non prévue) est interdit par cette barrière (O.AUTH et O.ALGO_STD),
- Garantir le fait que toute atteinte portée à l'intégrité du fichier de contrôle sera détectée et portée à connaissance de l'utilisateur qui ouvrira le conteneur (O.SCELLEMENT).

➔ Pour limiter l'impact de la menace, la TOE doit :

- Garantir le fait que les fichiers internes des différents conteneurs sont rendus «cryptographiquement différents» par l'utilisation d'aléas ne permettant pas de tirer des enseignements d'un fichier de contrôle pour en attaquer un autre (O.CHIFFREMENT et O.ALGO_STD).

| | |
|---|---|
| **M.FIC_CATALOGUE_INT** | Un attaquant récupère le fichier catalogue d'un conteneur (stocké dans celui-ci) et le modifie pour attaquer l'arborescence du conteneur. Le bien impacté est donc le fichier catalogue du conteneur et les données utilisateur (tous en intégrité).<br><br>L'attaquant peut ainsi intercepter les conteneurs échangés et faire disparaitre un ou plusieurs des fichiers du conteneur sans que le destinataire ne s'en aperçoive. |

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait qu'avant toute opération sur la TOE, une authentification est nécessaire (O.AUTH),

➔ Pour se protéger la TOE doit :

- Garantir le fait qu'il n'est pas possible, cryptographiquement, de retrouver les clés de chiffrement du conteneur sans fournir une clé d'accès valide : le détournement du fichier catalogue (i.e. sa mise en œuvre de façon détournée ou non prévue) est interdit par cette barrière (O.AUTH et O.ALGO_STD),
- Garantir le fait que toute atteinte portée à l'intégrité du fichier catalogue sera détectée et portée à connaissance de l'utilisateur qui ouvrira le conteneur (O.SCELLEMENT).

➔ Pour limiter l'impact de la menace, la TOE doit :

- Garantir le fait que les fichiers internes des différents conteneurs sont rendus «cryptographiquement différents» par l'utilisation d'aléas ne permettant pas de tirer des enseignements d'un fichier catalogue pour en attaquer un autre (O.CHIFFREMENT et O.ALGO_STD).

---

| | |
|---|---|
| **M.DETOURNEMENT_CONTENEUR** | Un attaquant positionné sur le réseau intercepte un conteneur pour le détourner et l'utiliser en tant que vecteur d'attaque (insertion d'un programme malveillant par exemple). L'objectif de l'attaquant n'est alors pas seulement l'accès aux biens sensibles présents dans le conteneur mais peut être aussi la compromission du poste du destinataire (par exemple).<br><br>Le bien impacté est donc le conteneur dans sa globalité (intégrité) et indirectement les données utilisateur présentes sur le poste (confidentialité et possiblement intégrité si le programme malveillant est un ransomware par exemple). |

➔ Pour prévenir cette menace, la TOE doit :

- Garantir le fait qu'avant toute opération sur la TOE, une authentification est nécessaire (O.AUTH),

➔ Pour se protéger la TOE doit :

- Garantir le fait que toute atteinte portée à l'intégrité du conteneur sera détectée et portée à connaissance de l'utilisateur qui ouvrira le conteneur (O.SCELLEMENT).
- Garantir le fait qu'il n'est pas possible, cryptographiquement, de contourner les mécanismes de contrôle d'intégrité (O.ALGO_STD).

➔ Pour limiter l'impact de la menace, la TOE doit :

- rien

### 8.1.3. Politiques de sécurité de l'organisation

Le tableau ci-dessous présente les liens de couverture entre les objectifs de sécurité et les politiques de sécurité de l'organisation retenues :

| | | O.AUTH | O.ROLES | O.CHIFFREMENT | O.SCELLEMENT | O.EFFACEMENT_CLES | O.ALGO_STD | O.ADM_ACCES | O.COMPATIBILITE | O.RECOUVREMENT | O.COLLECTE | O.INT_POLICIES |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Politiques de sécurité de l'organisation | **OSP.CONFIDENTIALITE** | | | X | X | X | X | | | | | |
| | **OSP.INTEGRITE** | | | | X | X | X | | X | | | |
| | **OSP.ACCES** | X | | | X | X | | | | | | |
| | **OSP.RECOUVREMENT** | X | X | | | X | | | | X | | |
| | **OSP.COLLECTE** | X | X | | | X | | | | | X | |
| | **OSP.ADMIN_ACCES** | X | X | | | X | | X | | | | |
| | **OSP.VERIF_POLICIES** | | | | | | | | | | | X |
| | **OSP.CRYPTO** | | | | | | X | | | | | |

**Tableau 7 : Couverture des politiques de sécurité de l'organisation par les objectifs de sécurité**

---

**OSP.CONFIDENTIALITE** La TOE doit offrir un service de protection en confidentialité (chiffrement), automatique et systématique, du stockage et de la transmission (pièce jointe de courrier électronique) des fichiers sensibles des utilisateurs.

Note: cette politique concerne la création initiale du conteneur, et le fait qu'une fois le conteneur créée, tout fichier déposé dans celui-ci est stocké chiffré. Cette politique ne concerne pas les accès au conteneur, qui relèvent de OSP.ACCES.

➔ Pour couvrir cette politique, la TOE :

- Génère, lors du chiffrement, des aléas pour la création des clés de chiffrement du conteneur (O.ALGO_STD) ;
- Chiffre les fichiers dans le conteneur (O.CHIFFREMENT) ;
- Ne déchiffre pas les fichiers si une erreur d'intégrité est détectée (O.SCELLEMENT) ;

➔ Pour garantir la mise en œuvre de la politique, la TOE :

- Efface les traces mémoire liées aux clés de chiffrement des fichiers (O.EFFACEMENT_CLES) ;

---

**OSP.INTEGRITE** La TOE doit offrir un service de contrôle de l'intégrité (scellement), automatique et systématique, du conteneur

➔ Pour couvrir cette politique, la TOE :

- Génère, lors du chiffrement, des aléas pour la création des clés de scellement du conteneur (O.ALGO_STD) ;
- Calcul un sceau et un HMAC à l'ouverture du conteneur et interdit l'ouverture de celui-ci en cas d'échec de vérification (O.SCELLEMENT). S'il s'agit d'un conteneur généré avec une ancienne version, la TOE avertit l'utilisateur et procède à la même vérification pour les différentes parties du conteneur (O.COMPATIBILITE).

➔ Pour garantir la mise en œuvre de la politique, la TOE :

- Efface les traces mémoire liées aux clés de scellement des fichiers (O.EFFACEMENT_CLES) ;

---

**OSP.ACCES** La TOE doit permettre aux utilisateurs de fournir une clé d'accès permettant d'accéder aux fichiers sensibles du conteneur auquel ils désirent accéder. S'ils ne peuvent fournir une clé d'accès valide pour le conteneur, l'accès doit être rejeté.

Note: cette politique ne concerne pas la gestion des accès (création assurée par OSP.ADMIN_ACCES), mais l'utilisation d'un accès.

➔ Pour couvrir cette politique, la TOE :

- Demande une authentification avant toute manipulation du fichier d'un conteneur (O.AUTH);
- Calcul un HMAC du fichier de contrôle (O.SCELLEMENT) contenant la clé de conteneur chiffrée par les accès utilisateur et interdit l'ouverture du conteneur en cas d'échec de vérification.

➔ Pour garantir la mise en œuvre de la politique, la TOE :

- Fait en sorte que seule une clé d'accès valide puisse permettre de retrouver les clés de chiffrement du conteneur, et que les fichiers ou informations internes de la TOE ne permettent pas de faire autrement (O.AUTH).
- Efface les traces mémoire liées aux éventuels calculs cryptographiques intermédiaires (dérivation de mots de passe) ou au transport des valeurs de clés de chiffrement lorsqu'elles sont calculées par un dispositif cryptographique externe (token) (O.EFFACEMENT_CLES) ;

---

**OSP.RECOUVREMENT** La TOE doit offrir un service de recouvrement des fichiers sensibles des utilisateurs par l'emploi de clés d'accès de recouvrement gérées par l'administrateur de la sécurité. Ces clés sont systématiquement et automatiquement affectées lors de l'initialisation des conteneurs.

Pour mettre en œuvre la politique, la TOE :

➔ Pour couvrir cette politique, la TOE :

- Permet d'affecter des clés d'accès de recouvrement au conteneur (O.RECOUVREMENT).
- Demande une authentification pour accéder à la gestion des clés de recouvrement (O.AUTH) ;
- N'autorise que l'administrateur de la sécurité à effectuer les opérations de recouvrement (O.ROLES)

➔ Pour garantir la mise en œuvre de la politique, la TOE :

- Fait en sorte que seule une clé d'accès valide puisse permettre de retrouver les clés de chiffrement d'un conteneur, et que les fichiers ou informations internes de la TOE ne permettent pas de faire autrement (O.AUTH) ;
- Efface les traces mémoire liées aux éventuels calculs cryptographiques intermédiaires (dérivation de mots de passe) ou au transport des valeurs de clés de chiffrement lorsqu'elles sont calculées par un dispositif cryptographique externe (token) (O.EFFACEMENT_CLES).

---

**OSP.COLLECTE** La TOE doit offrir un service de collecte d'information dans un fichier protégé pour les opérations de support. Les informations collectées sont sélectionnables

parmi les logs, la configuration, les applications installées etc.

Pour mettre en œuvre la politique, la TOE :

➔ Pour couvrir cette politique, la TOE :

- Offre une interface à l'administrateur lui permettant de collecter les informations (O.COLLECTE)
- Demande une authentification pour accéder au fichier protégé (O.AUTH) ;
- Permet à l'administrateur d'activer ou de désactiver la fonction (O.ROLES)

➔ Pour garantir la mise en œuvre de la politique, la TOE :

- Efface les traces mémoires des clés d'accès manipulées (O.EFFACEMENT_CLES).

---

**OSP.ADMIN_ACCES** La TOE doit offrir un service de gestion des accès.

➔ Pour couvrir cette politique, la TOE :

- Demande une authentification avant de permettre la gestion des accès au conteneur chiffré (O.AUTH) ;
- Contrôle que seul un utilisateur disposant du rôle administrateur (des accès ou de la sécurité) a le droit de gérer les accès (O.ROLES) ;
- Offre une interface à l'administrateur lui permettant de visualiser les accès (possible également pour l'utilisateur) et gérer les clés d'accès au conteneur (O.ADM_ACCES).

➔ Pour garantir la mise en œuvre de la politique, la TOE :

- Efface les traces mémoires des clés d'accès manipulées (O.EFFACEMENT_CLES).

---

**OSP.VERIF_POLICIES** La TOE doit offrir un service (transparent pour l'utilisateur) de vérification de la signature des politiques de sécurité par la clé privée de l'administrateur de sécurité. L'application de toute nouvelle politique est conditionnée par le succès de cette vérification.

➔ Pour couvrir cette politique, la TOE :

- Vérifie la signature des nouvelles politiques de sécurité appliquées et refuse leur application si la signature est incorrecte (O.INT_POLICIES).

---

**OSP.CRYPTO** Le référentiel de l'ANSSI ([CRYPTO_STD], [CLES_STD] et [AUTH_STD]) défini pour le niveau de résistance standard doit être suivi pour la génération d'aléa, la gestion des clés et pour les mécanismes cryptographiques et d'authentification utilisés dans la TOE.

➔ Pour couvrir cette politique, la TOE :

- Fournit un choix d'algorithmes cryptographiques et de tailles de clés conformes aux standards de ce domaine, prévus dans [CRYPTO_STD] et complétés par [CLES_STD] pour la production d'aléa et la gestion des clés (O.ALGO_STD).

### 8.1.4. Synthèse sur la couverture des objectifs

Le tableau ci-dessous présente la synthèse des liens de couverture entre les objectifs de sécurité, les hypothèses, menaces et politiques de sécurité de l'organisation retenues.

| | | H.NON_OBSERV | H.POSTE_SUR | H.CONFIANCE_ADMIN | H.CONSERVATION_CLES | H.ACCES | H.ENV_PROTECT_TOE | H.FIDELE_ENV | H.ENV_ALEA | H.CRYPTO_EXT | M.DETOURN_COMPOSANT | M.POLITIQUE_SECU_INT | M.CARNET_CONF | M.FIC_CONTROLE_CONF | M.FIC_CONTROLE_INT | M.FIC_CATALOGUE_INT | M.DETOURNEMENT_CONTENEUR | OSP.CONFIDENTIALITE | OSP.INTEGRITE | OSP.ACCES | OSP.RECOUVREMENT | OSP.COLLECTE | OSP.ADMIN_ACCES | OSP.VERIF_POLICIES | OSP.CRYPTO |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Objectifs de sécurité | OE.NON_OBSERV | X | | | | | | | | | | | | | | | | | | | | | | | |
| | OE.ENV_OPERATIONNEL | | X | | | | | | | | | | | | | | | | | | | | | | |
| | OE.HORODATAGE | | | | | | | X | | | | | | | | | | | | | | | | | |
| | OE.CONFIANCE_ADMIN | | | X | | | | | | | | X | | | | | | | | | | | | | |
| | OE.CONSERV_CLES | | | | X | | | | | | | X | | | | | | | | | | | | | |
| | OE.ACCES | | | | | X | | | | | | | | | | | | | | | | | | | |
| | OE.ENV_PROTECT_TOE | | | | | | X | | | | | | | | | | | | | | | | | | |
| | OE.FORMATION | | | X | X | | X | | | | | | | | | | | | | | | | | | |
| | OE.ENV_ALEA | | | | | | | | X | | | | | | | | | | | | | | | | |
| | OE.CRYPTO_EXT | | | | | | | | | X | | | | | | | | | | | | | | | |
| | O.AUTH | | | | | | | | | | X | | X | X | X | X | X | | | X | X | X | X | | |
| | O.ROLES | | | | | | | | | | | | | | | | | | | | X | X | X | | |
| | O.CHIFFREMENT | | | | | | | | | | | | | X | X | X | | X | | | | | | | |
| | O.SCELLEMENT | | | | | | | | | | | | | | X | X | X | X | X | X | | | | | |
| | O.EFFACEMENT_CLES | | | | | | | | | | X | | | | | | | X | X | X | X | X | X | | |
| | O.ALGO_STD | | | | | | | | | | X | | X | X | X | X | | X | X | | | | | | X |
| | O.ADM_ACCES | | | | | | | | | | | | | | | | | | | | | | X | | |
| | O.COMPATIBILITE | | | | | | | | | | | | | | | | | | X | | | | | | |
| | O.RECOUVREMENT | | | | | | | | | | | | | | | | | | | | X | | | | |
| | O.COLLECTE | | | | | | | | | | | | | | | | | | | | | X | | | |
| | O.INT_POLICIES | | | | | | | | | | | X | | | | | | | | | | | | X | |

**Tableau 8 : Couverture des objectifs de sécurité par les hypothèses, menaces et politiques de sécurité de l'organisation**

# 8.2. Argumentaire pour les exigences de sécurité

## 8.2.1. Dépendances entre exigences fonctionnelles de sécurité

Le tableau ci-dessous présente la couverture des dépendances entre les composants fonctionnels sélectionnés :

| Composant | Dépendances | Dépendances satisfaites |
|---|---|---|
| FCS_CKM.1 | [FCS_CKM.2 ou FCS_COP.1], FCS_CKM.4 | FCS_COP.1, FCS_CKM.4 |
| FCS_CKM.3 | [FDP_ITC.1 ou FDP_ITC.2 ou FCS_CKM.1], FCS_CKM.4 | FDP_ITC.1, FCS_CKM.1, FCS_CKM.4 |
| FCS_CKM.4 | [FDP_ITC.1 ou FDP_ITC.2 ou FCS_CKM.1] | FDP_ITC.1, FCS_CKM.1 |
| FCS_COP.1 | [FDP_ITC.1 ou FDP_ITC.2 ou FCS_CKM.1], FCS_CKM.4 | FDP_ITC.1, FCS_CKM.1, FCS_CKM.4 |
| FDP_ACC.1 | FDP_ACF.1 | FDP_ACF.1 |
| FDP_ACF.1 | FDP_ACC.1, FMT_MSA.3 | FDP_ACC.1, FMT_MSA.3 |
| FDP_ITC.1 | [FDP_ACC.1 ou FDP_IFC.1], FMT_MSA.3 | FDP_ACC.1, FMT_MSA.3 |
| FDP_RIP.1 | Aucune | Aucune |
| FDP_SDI.2 | Aucune | Aucune |
| FIA_AFL.1 | FIA_UAU.1 | FIA_UAU.2 |
| FIA_UAU.2 | FIA_UID.1 | FIA_UID.2 |
| FIA_UID.2 | Aucune | Aucune |
| FMT_MOF.1 | FMT_SMF.1, FMT_SMR.1 | FMT_SMF.1, FMT_SMR.1 |
| FMT_MSA.1 | [FDP_ACC.1 ou FDP_IFC.1], FMT_SMF.1, FMT_SMR.1 | FDP_ACC.1, FMT_SMF.1, FMT_SMR.1 |
| FMT_MSA.2 | [FDP_ACC.1 ou FDP_IFC.1], FMT_MSA.1, FMT_SMR.1 | FDP_ACC.1, FMT_MSA.1, FMT_SMR.1 |
| FMT_MSA.3 | FMT_MSA.1, FMT_SMR.1 | FMT_MSA.1, FMT_SMR.1 |
| FMT_MTD.1 | FMT_SMR.1, FMT_SMF.1 | FMT_SMR.1, FMT_SMF.1 |
| FMT_SMF.1 | Aucune | Aucune |
| FMT_SMR.1 | FIA_UID.1 | FIA_UID.2 |
| FPT_TST.1 | Aucune | Aucune |

**Tableau 9 : Satisfaction des dépendances entre exigences fonctionnelles de sécurité**

### 8.2.2. Dépendances entre exigences d'assurance de sécurité

Le tableau ci-dessous présente la couverture des dépendances entre les composants d'assurance sélectionnés :

| Composant | Dépendances | Dépendances satisfaites |
|---|---|---|
| ADV_ARC.1 | ADV_FSP.1, ADV_TDS.1 | ADV_FSP.3, ADV_TDS.2 |
| ADV_FSP.3 | ADV_TDS.1 | ADV_TDS.2 |
| ADV_TDS.2 | ADV_FSP.3 | ADV_FSP.3 |
| AGD_OPE.1 | ADV_FSP.1 | ADV_FSP.3 |
| AGD_PRE.1 | Aucune | Aucune |
| ALC_CMC.3 | ALC_CMS.1, ALC_DVS.1, ALC_LCD.1 | ALC_CMS.3, ALC_DVS.1, ALC_LCD.1 |
| ALC_CMS.3 | Aucune | Aucune |
| ALC_DEL.1 | Aucune | Aucune |
| ALC_DVS.1 | Aucune | Aucune |
| ALC_FLR.3 | Aucune | Aucune |
| ALC_LCD.1 | Aucune | Aucune |
| ASE_CCL.1 | ASE_INT.1, ASE_ECD.1, ASE_REQ.1 | ASE_INT.1, ASE_ECD.1, ASE_REQ.2 |
| ASE_ECD.1 | Aucune | Aucune |
| ASE_INT.1 | Aucune | Aucune |
| ASE_OBJ.2 | ASE_SPD.1 | ASE_SPD.1 |
| ASE_REQ.2 | ASE_OBJ.2, ASE_ECD.1 | ASE_OBJ.2, ASE_ECD.1 |
| ASE_SPD.1 | Aucune | Aucune |
| ASE_TSS.1 | ASE_INT.1, ASE_REQ.1, ADV_FSP.1 | ASE_INT.1, ASE_REQ.2, ADV_FSP.3 |
| ATE_COV.2 | ADV_FSP.2, ATE_FUN.1 | ADV_FSP.3, ATE_FUN.1 |
| ATE_DPT.1 | ADV_ARC.1, ADV_TDS.2, ATE_FUN.1 | ADV_ARC.1, ADV_TDS.2, ATE_FUN.1 |
| ATE_FUN.1 | ATE_COV.1 | ATE_COV.2 |
| ATE_IND.2 | ADV_FSP.2, AGD_OPE.1, AGD_PRE.1, ATE_COV.1, ATE_FUN.1 | ADV_FSP.3, AGD_OPE.1, AGD_PRE.1, ATE_COV.2, ATE_FUN.1 |
| AVA_VAN.3 | ADV_ARC.1, ADV_FSP.4*, ADV_TDS.3*, ADV_IMP.1*, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1 | ADV_ARC.1, ADV_FSP.3, AGD_OPE.1, AGD_PRE.1, ATE_DPT.1 |

**Tableau 10 : Satisfaction des dépendances entre exigences d'assurance de sécurité**

### 8.2.3. Argumentaire pour les dépendances non satisfaites

*La dépendance AVA_VAN.3 avec ADV_FSP.4, ADV_IMP.1 et ADV_TDS.3 ne sont pas satisfaites par construction du paquet d'assurance de la qualification de niveau standard défini par l'ANSSI.

### 8.2.4. Argumentaire de couverture des objectifs de sécurité par les exigences fonctionnelles

Les tableaux ci-dessous présentent la couverture des composants fonctionnels sélectionnés par les objectifs de sécurité :

| Objectifs de sécurité de la TOE | FCS_CKM.1 | FCS_CKM.3 | FCS_CKM.4 | FCS_COP.1 | FDP_ACC.1 | FDP_ACF.1 | FDP_ITC.1 | FDP_RIP.1 | FDP_SDI.2 | FIA_AFL.1 | FIA_UAU.2 | FIA_UID.2 | FMT_MOF.1 | FMT_MSA.1 | FMT_MSA.2 | FMT_MSA.3 | FMT_MTD.1 | FMT_SMF.1 | FMT_SMR.1 | FPT_TST.1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **O.AUTH** | | | | | x | x | x | | | x | x | x | | | | | | | | |
| **O.ROLES** | | | | | x | x | | | | | | | x | | | | x | x | x | |
| **O.CHIFFREMENT** | x | x | | x | | | | | | | | | | | | | | | | |
| **O.SCELLEMENT** | x | x | | x | | | | | x | | | | | | | | | | | |
| **O.EFFACEMENT_CLES** | | | | | | | | x | | | | | | | | | | | | |
| **O.ALGO_STD** | x | x | x | x | | | | | | | | | | | | | | | | x |
| **O.ADM_ACCES** | | | | | | | | | | | | | | x | x | x | | x | x | |
| **O.COMPATIBILITE** | | | | x | | | | | x | | | | | | | | | | | |
| **O.RECOUVREMENT** | | | | | | | | | | | | | x | | | | x | x | x | |
| **O.COLLECTE** | | | | | | | | | | | | | x | | | | x | x | x | |
| **O.INT_POLICIES** | | | | x | | | | | | | | | | | | | x | | | |

**Tableau 11 : Couverture des objectifs de sécurité par les exigences fonctionnelles de sécurité**

#### 8.2.4.1. Contrôle d'accès

| | |
|---|---|
| **O.AUTH** | La TOE doit permettre d'identifier et authentifier tout utilisateur. Pour cela, la TOE ne doit autoriser l'accès à un fichier d'un conteneur qu'après mise à disposition d'une clé d'accès valide pour ce conteneur. |

Afin de remplir cet objectif :

- La TOE identifie et authentifie chaque utilisateur avant d'ouvrir les conteneurs chiffrés (FIA_UID.2 et FIA_UAU.2) et applique une règle de ralentissement d'affichage de la mire d'authentification à un utilisateur, suite à plusieurs essais d'authentification infructueux (FIA_AFL.1).
- Pour que la TOE donne l'accès au conteneur chiffré, l'utilisateur doit présenter sa clé d'accès (token USB par exemple) en vue de son authentification (FDP_ITC.1).
- La TOE applique ensuite une politique de contrôle d'accès au conteneur (FDP_ACC.1) basé sur les attributs de sécurité (FDP_ACF.1).

| | |
|---|---|
| **O.ROLES** | La TOE doit gérer trois rôles d'utilisateurs pour un conteneur chiffrée : un rôle 'utilisateur normal' ou plus simplement 'utilisateur' (utilisation des fichiers du conteneur sous condition de mise à disposition d'une clé d'accès valide), un rôle 'administrateur des accès' en charge de l'initialisation du container avec affectation des accès et un rôle 'administrateur de la sécurité' (installation, signature des politiques, recouvrement en plus de la gestion des accès). |

Afin de remplir cet objectif :

- La TOE doit gérer et distinguer les rôles administrateur de la sécurité, administrateur des accès et utilisateur de la TOE (FMT_SMR.1).
- La TOE permet aussi de contrôler l'accès des utilisateurs aux conteneurs et aux opérations sur ces conteneurs (FDP_ACC.1), et de restreindre l'accès aux seuls utilisateurs possédant la clé d'accès associée (FDP_ACF.1).
- Enfin, la TOE doit permettre de restreindre aux administrateurs certaines fonctions d'administration de la sécurité (FMT_SMF.1) et la gestion des « policies » (FMT_MTD.1) ainsi que la possibilité de pouvoir utiliser ou non l'accès de recouvrement (FMT_MOF.1).

#### 8.2.4.2. Cryptographie

| | |
|---|---|
| **O.CHIFFREMENT** | La TOE doit chiffrer et déchiffrer les données sensibles par l'emploi de clés cryptographiques. La TOE doit utiliser des clés différentes pour protéger les différents conteneurs ainsi que des vecteurs d'initialisation différents pour chaque fichier d'un conteneur. La TOE doit générer ces clés de chiffrement conformément aux exigences pour le niveau de robustesse standard des référentiels cryptographiques [CRYPTO_STD] et [CLES_STD] de l'ANSSI. |

Afin de remplir cet objectif:

- Pour chiffrer les fichiers présents dans un conteneur, la TOE doit tout d'abord être capable de générer les clés cryptographiques (FCS_CKM.1) et y accéder de manière sécurisée (FCS_CKM.3), afin de les utiliser pour réaliser les opérations cryptographiques selon différents algorithmes (FCS_COP.1).

| | |
|---|---|
| **O.SCELLEMENT** | La TOE doit pouvoir contrôler l'intégrité des conteneurs par l'emploi de clés cryptographiques différentes pour chaque conteneur. La TOE doit générer ces clés de scellement conformément aux exigences pour le niveau de robustesse standard des référentiels cryptographiques [CRYPTO_STD] et [CLES_STD] de l'ANSSI. |

Afin de remplir cet objectif :

- Pour calculer le sceau global (HMAC) d'un conteneur, la TOE doit tout d'abord être capable de générer les clés cryptographiques (FCS_CKM.1) et y accéder de manière sécurisée (FCS_CKM.3), afin de les utiliser pour réaliser les opérations cryptographiques (FCS_COP.1).
- La TOE contrôle l'intégrité du conteneur et retourne une indication en cas de détection d'erreur avec interdiction d'ouvrir le conteneur (FDP_SDI.2).

| | |
|---|---|
| **O.EFFACEMENT_CLES** | La TOE doit assurer le nettoyage des traces de données sensibles (clés de chiffrement des fichiers, éléments permettant de retrouver les clés d'accès) dans la mémoire (RAM) dès la fin des opérations réalisées par la TOE. |

Afin de remplir cet objectif :

- La TOE permet un nettoyage totalement sécurisé des clés dans la mémoire (RAM) (FDP_RIP.1).

| | |
|---|---|
| **O.ALGO_STD** | La TOE doit produire des aléas et fournir un choix d'algorithmes cryptographiques et de tailles de clés conformes aux standards de ce domaine, prévus dans [CRYPTO_STD] et complétés par [CLES_STD]. |

Afin de remplir cet objectif :

- La TOE doit être capable de fournir un choix d'algorithmes de génération (FCS_CKM.1), d'accès (FCS_CKM.3) et de destruction (FCS_CKM.4) de clés cryptographiques.
- Elle doit aussi permettre d'exécuter des opérations cryptographiques conformément à des algorithmes et tailles de clés cryptographique spécifiés (FCS_COP.1).
- La TOE doit exécuter des tests pour vérifier le bon fonctionnement des algorithmes cryptographiques (FPT_TST.1).

#### 8.2.4.3. Gestion

| | |
|---|---|
| **O.ADM_ACCES** | La TOE doit offrir une interface à l'administrateur de la sécurité et l'administrateur des accès permettant de visualiser les accès et gérer les clés d'accès aux conteneurs. L'utilisateur peut seulement visualiser les accès. |

Afin de remplir cet objectif :

- La TOE offre des fonctions de gestions des accès (FMT_SMF.1).
- La TOE limite les accès à ces fonctions de gestion en fonction du rôle associé aux utilisateurs (FMT_SMR.1).
- La TOE assure que seul l'administrateur des accès et l'administrateur de la sécurité peuvent gérer les attributs de sécurité des objets stockés : clés et rôles (FMT_MSA.1).
- La TOE garantie, de plus, que seuls des valeurs sûres sont acceptées pour les attributs de sécurité, en contrôlant la force des mots de passe par exemple (FMT_MSA.2).
- L'administrateur (des accès ou de la sécurité) peut aussi définir les données d'initialisation des attributs (tel que le rôle initialisé par défaut à « utilisateur ») (FMT_MSA.3).

| | |
|---|---|
| **O.COMPATIBILITE** | La TOE doit assurer l'ouverture sécurisée des conteneurs créés avec une ancienne version de Zed! et dont les éléments faisaient l'objet d'un contrôle d'intégrité par partie. |

Afin de remplir cet objectif :

- En cas d'ouverture d'un conteneur proposant la vérification d'intégrité par partie, la TOE doit vérifier les sceaux (HMAC) des fichiers, du watermark, du fichier de contrôle et du catalogue présents dans le conteneur (FCS_COP.1)
- La TOE retourne une indication en cas de détection d'erreur avec interdiction d'ouvrir le conteneur (FDP_SDI.2).
- La TOE avertit ouvre le conteneur en cas de succès (FDP_SDI.2).

| | |
|---|---|
| **O.RECOUVREMENT** | La TOE doit permettre d'affecter des clés d'accès de recouvrement. |

Afin de remplir cet objectif :

- La TOE doit permettre de restreindre à l'administrateur de la sécurité (FMT_MOF.1 associé à FMT_SMR.1) l'activation ou la désactivation de la fonction de recouvrement (FMT_SMF.1).
- La fonction de recouvrement est configurée dans les policies signées par l'administrateur de la sécurité (FMT_MTD.1).

| | |
|---|---|
| **O.COLLECTE** | La TOE doit permettre de collecter de manière sécurisée des informations utiles aux opérations de support. |

Afin de remplir cet objectif :

- La TOE doit permettre de restreindre à l'administrateur de la sécurité (FMT_MOF.1 associé à FMT_SMR.1) l'activation ou la désactivation de la fonction de collecte d'information (FMT_SMF.1).
- La fonction de collecte d'information est configurée dans les policies signées par l'administrateur de la sécurité (FMT_MTD.1).

| | |
|---|---|
| **O.INT_POLICIES** | La TOE doit vérifier la signature de toutes nouvelles politiques de sécurité à appliquer. En cas d'échec lors de la vérification, les politiques appliquées restent inchangées. |

Afin de remplir cet objectif :

- La TOE doit permettre d'exécuter des opérations de vérification de signature conformément aux algorithmes et tailles de clés cryptographique spécifiés (FCS_COP.1).
- La TOE doit vérifier que la signature utilisée a bien été effectuée par l'administrateur de la sécurité qui est seul autorisé à modifier les politiques de sécurité (FMT_MTD.1).

### 8.2.5. Pertinence du niveau d'assurance

Le niveau d'assurance EAL3 augmenté des composants ALC_FLR.3 et AVA_VAN.3 associé à une expertise de l'implémentation de la cryptographie a été choisi pour assurer la conformité au processus de qualification de niveau standard défini par l'ANSSI dans [QUALIF_STD]. Ce niveau d'assurance impose:

- Des tests indépendants effectués par l'évaluateur (l'utilisateur final est alors assuré que les fonctions de sécurité de la TOE sont implémentées comme spécifié)
- Une analyse de vulnérabilité indépendante effectuée par l'évaluateur qui considèrera un niveau d'attaquant correspondant au niveau élémentaire renforcé ou inférieur (l'utilisateur final est alors assuré que la TOE est résistante à des attaques de pénétration effectuées par des attaquants possédant un faible potentiel d'attaque).

- L'évaluation de l'architecture de sécurité et de l'architecture logiciel incluant l'analyse de l'implémentation (fonctions cryptographiques seulement) pour vérifier qu'il n'y a pas de défaut de sécurité

- De bonnes pratiques en matière de développement de la partie cryptographique (l'utilisateur final est alors assuré que le produit a été correctement et sécuritairement conçu et développé).

- De bonnes pratiques en matière de maintenance et support aux utilisateurs assurant que toutes les anomalies identifiées seront corrigées et rapportées aux utilisateurs du produit considéré qui pourraient être affectés par cette anomalie.

## 8.3. Argumentaire pour les spécifications globales de la TOE

Le tableau ci-dessous justifie la nécessité des fonctions de sécurité de la TOE par rapport aux composants fonctionnels CC sélectionnés :

| Exigences fonctionnelles de sécurité pour la TOE | | F.CONTROLE_ACCES | F.ENTREE_SECURISEE | F.CONFIGURATION_TOE | F.GESTION_CLES_ACCES | F.OPERATIONS_CRYPTO |
|---|---|---|---|---|---|---|
| FCS_CKM.1 | Génération de clés cryptographiques | | | | X | X |
| FCS_CKM.3 | Accès aux clés cryptographiques | | X | | | |
| FCS_CKM.4 | Destruction de clés cryptographiques | | | | X | |
| FCS_COP.1 | Opération cryptographique | X | X | X | X | X |
| FDP_ACC.1 | Contrôle d'accès partiel | X | | | X | |
| FDP_ACF.1 | Contrôle d'accès basé sur les attributs de sécurité | X | | | X | |
| FDP_ITC.1 | Importation depuis une zone hors du contrôle de la TSF | | X | | | |
| FDP_RIP.1 | Protection d'une partie des informations résiduelles | | | | X | |
| FDP_SDI.2 | Contrôle de l'intégrité des données stockées et action à entreprendre | | | | | X |
| FIA_AFL.1 | Défaillance de l'authentification | X | | | | |
| FIA_UAU.2 | Authentification d'un utilisateur préalablement à toute action | X | X | X | | |
| FIA_UID.2 | Identification d'un utilisateur préalablement à toute action | X | X | X | | |
| FMT_MOF.1 | Administration des fonctions de la TSF | X | | X | | |
| FMT_MSA.1 | Gestion des attributs de sécurité | | | | X | |
| FMT_MSA.2 | Attributs de sécurité sûrs | | | X | | |
| FMT_MSA.3 | Initialisation statique d'attribut | | | | X | |
| FMT_MTD.1 | Gestion des données de la TSF | | | X | | |
| FMT_SMF.1 | Spécification des fonctions d'administration | | | X | X | X |
| FMT_SMR.1 | Rôles de sécurité | | | | X | |
| FPT_TST.1 | Tests de la TSF | | | | | X |

**Tableau 12 : Couverture des exigences fonctionnelles par les spécifications globales de la TOE**

**FCS_CKM.1 Génération de clés cryptographiques**

A chaque conteneur est associé une clé de chiffrement et déchiffrement des fichiers ainsi qu'une clé de scellement. Ces clés sont tirées lors de l'initialisation du conteneur. Elles répondent aux critères de longueurs de clés configurées dans les policies. Par défaut, ce sont des clés AES de 256 bits.

Le format de certaines clés d'accès utilisateur (liste d'accès personnelle) peut également faire l'objet d'un chiffrement intermédiaire par un bi clé RSA générée par la TOE.

Le carnet de mot de passe propose de créer un mot de passe fort pour votre correspondant. Comme tout accès par mot de passe, ce mot de passe est ensuite diversifié pour générer la clé d'accès.

La fonction de sécurité F.GESTION_CLES_ACCES implémente la génération des clés RSA et F.OPERATIONS_CRYPTO la génération des clés AES et des mots de passe (par le carnet de mot de passe).

**FCS_CKM.3 Accès aux clés cryptographiques**

L'accès aux clés cryptographiques gérées par la TOE est implémenté par la fonction de sécurité F.ENTREE_SECURISEE pour la protection des données en entrée.

Cette fonction est utilisée lors de toute authentification utilisateur.

**FCS_CKM.4 Destruction de clés cryptographiques**

Lorsqu'un conteneur est supprimé, les clés cryptographiques relatives au conteneur sont détruites. De même lorsqu'un accès est supprimé, la clé d'accès correspondante est détruite.

La fonction de sécurité F.GESTION_CLES_ACCES implémente cette exigence fonctionnelle.

**FCS_COP.1 Opération cryptographique**

La TOE effectue les opérations cryptographiques suivantes :

- Récupère une clé d'accès avant de pouvoir créer une clé de conteneur et chiffrer le conteneur à sa création,
- Récupère une clé d'accès pour déchiffrer la clé de conteneur avant de pouvoir créer un nouvel accès (il y a donc ensuite chiffrement de la clé de conteneur par la clé associée à ce nouvel accès),
- Récupère une clé d'accès avant de pouvoir déchiffrer la clé de conteneur, afin de pouvoir chiffrer ou déchiffrer les fichiers du conteneur,
- Récupère une clé d'accès avant de pouvoir déchiffrer la clé de scellement, afin de pouvoir sceller ou vérifier les données sensibles,
- Récupère un mot de passe afin d'en dériver une clé d'accès qui va chiffrer ou déchiffrer la clé de conteneur.
- Transmet la clé de conteneur chiffrée au porte-clés puis récupère la clé de conteneur déchiffrée par le porte-clés afin de pouvoir chiffrer ou déchiffrer les fichiers du conteneur.
- Vérifie la signature des politiques avec le certificat de l'administrateur de sécurité
- Effectue des tests au démarrage des programmes et périodiquement pour vérifier le bon fonctionnement du générateur aléatoire, des algorithmes et l'intégrité du code.

La fonction de sécurité F.OPERATIONS_CRYPTO, implémentent les opérations de chiffrement et déchiffrement des conteneurs, de contrôle de l'intégrité, des auto tests ainsi que les opérations cryptographiques mises au service des autres fonctions.

Les fonctions F.GESTION_CLES_ACCES (création de la clé d'accès) et F.CONTROLE_ACCES (vérification de la clé d'accès) utilisent les fonctions de dérivation des clés à partir des mots de passe.

La fonction F.ENTREE_SECURISEE utilise des fonctions de wrapping pour assurer le transfert sécurisé des clés entre la TOE et les porte-clés physique.

La fonction F.CONFIGURATION_TOE intervient pour la configuration des types d'accès et algorithmes supportés.

---

**FDP_ACC.1 Contrôle d'accès partiel**

Afin d'utiliser un conteneur géré par la TOE, l'utilisateur doit impérativement présenter une clé d'accès valide, associée au conteneur concerné. Cette exigence de sécurité est implémentée dans la TOE par les fonctions de sécurité

- F.GESTION_CLES_ACCES pour la configuration des accès au conteneur par l'administrateur
- F.CONTROLE_ACCES pour le contrôle d'accès au conteneur

---

**FDP_ACF.1 Contrôle d'accès basé sur les attributs de sécurité**

Afin d'utiliser un conteneur géré par la TOE, l'utilisateur doit présenter une clé d'accès valide, associée au conteneur concerné. Pour pouvoir mettre en place ce fonctionnement :

- des droits et des rôles sont associés aux utilisateurs (F.GESTION_CLES_ACCES),
- et l'accès aux conteneurs est donc contrôlé (F.CONTROLE_ACCES)

---

**FDP_ITC.1 Importation depuis une zone hors du contrôle de la TSF**

Des données nécessaires au bon fonctionnement de la TOE sont importées depuis l'extérieur de la TSF comme les clés d'accès ou les mots de passe saisis par l'utilisateur. Ce ne sont que des données, aucun attribut de sécurité n'est importé.

La fonction de sécurité F.ENTREE_SECURISEE implémente la communication de clés d'accès fournies en entrée vers la TOE, et couvre donc cette exigence.

---

**FDP_RIP.1 Protection d'une partie des informations résiduelles**

Cette exigence fonctionnelle est mise en œuvre par la fonction de sécurité F.GESTION_CLES_ACCES qui gère l'effacement sécurisé des clés en mémoire.

---

**FDP_SDI.2 Contrôle de l'intégrité des données stockées et action à entreprendre**

Cette exigence fonctionnelle est mise en œuvre par la fonction de sécurité F.OPERATIONS_CRYPTO qui implémentent toutes les fonctions nécessaires au contrôle de l'intégrité des conteneurs.

---

**FIA_AFL.1 Gestion d'une défaillance de l'authentification**

Le nombre maximum d'essai de mots de passe ou de code confidentiel autorisés lors de l'ouverture d'un conteneur est fixé à cinq. Passé ce nombre, la demande d'ouverture est rejetée et l'utilisateur doit recommencer sa demande d'authentification (ce qui le ralenti entre ses différentes séquences d'essais).

La fonction de sécurité F.CONTROLE_ACCES couvre cette fonctionnalité.

---

**FIA_UAU.2 Authentification d'un utilisateur préalablement à toute action**

Aucune ouverture de conteneur n'est possible sans une phase préalable d'authentification et d'identification de l'utilisateur. Pour chaque authentification, les utilisateurs doivent présenter une clé d'accès valide.

Cette exigence fonctionnelle est implémentée par :

- F.CONFIGURATION_TOE pour la configuration des accès autorisés au conteneur (type d'accès, force des mots de passe, type de certificat …).
- F.CONTROLE_ACCES pour le contrôle d'accès au conteneur
- F.ENTREE_SECURISEE pour sécuriser la communication des données fournies en entrée vers la TOE.

---

**FIA_UID.2 Identification d'un utilisateur préalablement à toute action**

Aucune ouverture de conteneur n'est possible sans une phase préalable d'authentification et d'identification de l'utilisateur. Pour chaque authentification, les utilisateurs doivent présenter une clé d'accès valide.

Cette exigence fonctionnelle est implémentée par :

- F.CONFIGURATION_TOE pour la configuration des accès autorisés au conteneur (type d'accès, type de certificat …).
- F.CONTROLE_ACCES pour le contrôle d'accès au conteneur
- F.ENTREE_SECURISEE pour sécuriser la communication des données fournies en entrée vers la TOE.

---

**FMT_MOF.1 Administration des fonctions de la TSF**

Seul l'administrateur de la sécurité peut activer ou désactiver les fonctions de collecte d'information et de recouvrement.

La fonction de sécurité F.CONFIGURATION_TOE associée à F.CONTROLE_ACCES (pour l'authentification recouvrement) implémente cette exigence. Une politique de sécurité spécifique (signée par l'administrateur de la sécurité) permet d'activer et désactiver la possibilité d'utiliser l'accès de recouvrement ainsi que la fonction de collecte d'information.

---

**FMT_MSA.1 Gestion des attributs de sécurité**

Seuls l'administrateur des accès et l'administrateur de la sécurité ont la possibilité de changer la valeur par défaut, modifier ou supprimer les attributs de sécurité « clés d'accès et rôle ».

Cet attribut de sécurité est stocké dans le fichier de contrôle, lui-même masqué par Zed!.

La fonction de sécurité F.GESTION_CLES_ACCES implémente cette exigence.

---

**FMT_MSA.2 Attributs de sécurité sûrs**

La fonction de sécurité F.CONFIGURATION_TOE (force des mots de passe, contrôle des certificats par exemple) permet de garantir que les attributs de sécurité « clé d'accès » sont sûrs.

**FMT_MSA.3 Initialisation statique d'attribut**

La TSF permet à l'administrateur des accès ou à l'administrateur de la sécurité de spécifier des valeurs initiales alternatives aux valeurs par défaut lorsqu'un objet ou une information est créé (choix du rôle par exemple).

La fonction de sécurité F.GESTION_CLES_ACCES (changement du rôle par exemple) met en œuvre cette exigence.

**FMT_MTD.1 Administration des données de la TSF**

Seuls l'administrateur de la sécurité a la possibilité de gérer les stratégies de sécurité (ou « policies).

Cette exigence est implémentée par la fonction de sécurité F.CONFIGURATION_TOE qui vérifie la signature des politiques à appliquer.

**FMT_SMF.1 Spécification des fonctions d'administration**

La TOE permet de réaliser :

- Les fonctions de gestion des accès des conteneurs
- La fonction de recouvrement
- La fonction de collecte d'information pour le support

Cette exigence fonctionnelle est implémentée par les fonctions de sécurité F.GESTION_CLES_ACCES (gestion des accès et du recouvrement), F.OPERATIONS_CRYPTO (chiffrement des informations récoltées dans l'environnement) et F.CONFIGURATION_TOE (policies).

**FMT_SMR.1 Rôles de sécurité**

La TOE supporte les rôles utilisateur, administrateur des accès et administrateur de la sécurité.

Cette exigence est implémentée par F.GESTION_CLES_ACCES qui identifie les droits administrateur et utilisateur par l'intermédiaire de leur clé s'accès.

**FPT_TST.1 Auto tests de la TSF**

Les tests cryptographiques réalisés au démarrage de la TOE ou périodiquement pour vérifier le bon fonctionnement du générateur aléatoire et la conformité des algorithmes est assuré par F.OPERATIONS_CRYPTO.

## 8.4. Argumentaire pour les annonces de conformité à un PP

Cette cible de sécurité ne déclare aucune conformité à un Profil de Protection. Aucun argumentaire n'est donc requis.

# 9. ANNEXE A : EXIGENCES FONCTIONNELLES DE SECURITE DE LA TOE

Cette annexe contient les textes officiels de la partie 2 des Critères Communs en version 3.1 d'avril 2017 avec l'ensemble des opérations réalisées pour la TOE.

Les composants fonctionnels CC sélectionnés pour répondre aux objectifs de sécurité de la TOE sont les suivants :

| Composants CC retenus | |
|---|---|
| FCS_CKM.1 | Cryptographic key generation |
| FCS_CKM.3 | Cryptographic key access |
| FCS_CKM.4 | Cryptographic key destruction |
| FCS_COP.1 | Cryptographic operation |
| FDP_ACC.1 | Subset access control |
| FDP_ACF.1 | Security attribute based access control |
| FDP_ITC.1 | Import of user data without security attributes |
| FDP_RIP.1 | Subset residual information protection |
| FDP_SDI.2 | Stored data integrity monitoring and action |
| FIA_AFL.1 | Authentication failure handling |
| FIA_UAU.2 | User authentication before any action |
| FIA_UID.2 | User identification before any action |
| FMT_MOF.1 | Management of security functions behaviour |
| FMT_MSA.1 | Management of security attributes |
| FMT_MSA.2 | Secure security attributes |
| FMT_MSA.3 | Static attribute initialisation |
| FMT_MTD.1 | Management of TSF data |
| FMT_SMF.1 | Specification of Management Functions |
| FMT_SMR.1 | Security roles |
| FPT_TST.1 | TSF testing |

**Tableau 13 : Exigences fonctionnelles de sécurité pour la TOE**

## 9.1. Class FCS : Cryptographic support

| | |
|---|---|
| **FCS_CKM** | **Cryptographic key management** |
| FCS_CKM.1 | Cryptographic key generation |
| FCS_CKM.1.1 | The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [<br>- **génération de nombres pseudo-aléatoires utilisés pour la génération des clés de chiffrement et des clés RSA de listes d'accès en utilisant les générateurs Hash_DRBG, HMAC_DRBG ou CTR_DRBG décrit dans la publication « Recommendation for Random Number Generation Using Deterministic Random Bit Generators » (référence SP 800-90A révision 1) du NIST ;**<br>- **diversification de clés PKCS#5 à partir des mots de passe**<br>- **Génération de mot de passe par le carnet de mot de passe pour en dériver une clé]**<br>and specified cryptographic key sizes [**de 128, 192 et 256 bits pour les clés symétriques et de 2048 bits à 4096 bits pour les clés asymétriques**] that meet the following: [**exigences cryptographique de l'ANSSI définies dans [CRYPTO_STD] et [CLES_STD]].** |
| FCS_CKM.3 | Cryptographic key access |
| FCS_CKM.3.1 | The TSF shall perform [**l'utilisation de clés**] in accordance with a specified cryptographic key access method [**utilisation du driver clavier et déchiffrement (déwrapping) des clés par la clé d'accès**] that meets the following: [**Aucun**]. |
| FCS_CKM.4 | Cryptographic key destruction |
| FCS_CKM.4.1 | The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **[réécriture de motifs composés de zéros suivi par une opération de lecture-vérification. Si la lecture-vérification de la réécriture échoue, le processus doit être répété**] that meets the following: [**Aucun**]. |
| **FCS_COP** | **Cryptographic operation** |
| FCS_COP.1 | Cryptographic operation |
| FCS_COP.1.1 | The TSF shall perform [**le hachage, le calcul et la vérification d'intégrité, le chiffrement, le déchiffrement, la vérification de la signature des politiques de sécurité, la génération de clés, le wrapping et dewrapping de clés et la dérivation de clés**] in accordance with a specified cryptographic algorithm [**HMAC, SHA-256 et SHA-512, RSA PKCS#1 v2.2, AES mode CBC**] and cryptographic key sizes [**de 128, 192 et 256 bits pour les clés symétriques et de 2048 bits pour les clés asymétriques**] that meet the following: [**exigences cryptographique de l'ANSSI définies dans [CRYPTO_STD] et [CLES_STD]**]. |

## 9.2. Class FDP : User data protection

| | |
|---|---|
| **FDP_ACC** | **Access control policy** |
| FDP_ACC.1 | Subset access control |
| FDP_ACC.1.1 | The TSF shall enforce the [**SFP.ACCESS_OBJ**] on [<br>**Sujets: Administrateur et utilisateurs de la TOE**<br>**Objets: Conteneurs contenant les fichiers utilisateur et le fichier de contrôle.**<br>**Opérations : Gestion des conteneurs et utilisation**]. |
| **FDP_ACF** | **Access control functions** |

| | |
|---|---|
| FDP_ACF.1 | Security attribute based access control |
| FDP_ACF.1.1 | The TSF shall enforce the [**SFP.ACCESS_OBJ**] to objects based on the following: [<br>**Sujets: Administrateur et utilisateurs de la TOE**<br>**Attributs de sécurité : Clés d'accès utilisateur permettant ou non d'ouvrir le conteneur et rôle]** |
| FDP_ACF.1.2 | The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [<br>**Objet : Conteneur**<br>**Opération: Gestion des conteneurs et utilisation**<br>**Règle : authentification réussie après mise à disposition de la clé d'accès associée au conteneur concerné avec accès à la gestion du conteneur uniquement pour le rôle administrateur des accès (gestion des accès) et administrateur de la sécurité (toute opération de gestion)].** |
| FDP_ACF.1.3 | The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [**Aucune**]. |
| FDP_ACF.1.4 | The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [**Aucune**]. |
| **FDP_ITC** | **Import from outside TSF control** |
| FDP_ITC.1 | Import of user data without security attributes |
| FDP_ITC.1.1 | The TSF shall enforce the [**SFP.ACCESS_OBJ**] when importing user data, controlled under the SFP, from outside of the TOE. |
| FDP_ITC.1.2 | The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE. |
| FDP_ITC.1.3 | The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: [**Aucune**]. |
| **FDP_RIP** | **Residual information protection** |
| FDP_RIP.1 | Subset residual information protection |
| FDP_RIP.1.1 | The TSF shall ensure that any previous information content of a resource is made unavailable upon the [**désallocation de la ressource de**] the following objects: [**clés de chiffrement des conteneurs et clés d'accès**]. |
| **FDP_SDI** | **Stored data** |
| FDP_SDI.2 | Stored data integrity monitoring and action |
| FDP_SDI.2.1 | The TSF shall monitor user data stored in containers controlled by the TSF for [**erreurs d'intégrité**] on all objects, based on the following attributes: [**HMAC global ou par partie du conteneur**]. |

| | |
|---|---|
| FDP_SDI.2.2 | Upon detection of a data integrity error, the TSF shall [**retourner un message d'erreur et interdire l'ouverture du conteneur**].<br>**Raffinement non éditorial :**<br>**La version Q.2021.1 effectue le contrôle d'intégrité sur la globalité du conteneur, les anciennes versions de Zed ! effectuent le contrôle d'intégrité sur les différentes parties du conteneur.**<br>**Le contrôle du HMAC global s'effectue au moment de l'ouverture du conteneur. Si le conteneur contient une protection en intégrité par partie, les fichiers techniques et le watermark sont contrôlés au moment de l'ouverture du conteneur, les fichiers utilisateurs sont contrôlés lorsqu'ils sont ouverts.** |

## 9.3. Class FIA : Identification and authentication

| | |
|---|---|
| **FIA_AFL** | **Authentication failures** |
| FIA_AFL.1 | Authentication failure handling |
| FIA_AFL.1.1 | The TSF shall detect when [**cinq**] unsuccessful authentication attempts occur related to [**l'ouverture d'un conteneur**]. |
| FIA_AFL.1.2 | When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [**temporiser l'accès à ce conteneur**]. |

| | |
|---|---|
| **FIA_UAU** | **User authentication** |
| FIA_UAU.2 | User authentication before any action |
| FIA_UAU.2.1 | The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user. |
| **FIA_UID** | **User identification** |
| FIA_UID.2 | User identification before any action |
| FIA_UID.2.1 | The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user. |

## 9.4. Class FMT : Security management

| | |
|---|---|
| **FMT_MOF** | **Management of functions in TSF** |
| FMT_MOF.1 | Management of security functions behaviour |
| FMT_MOF.1.1 | The TSF shall restrict the ability to [**activer ou désactiver**] the functions [**collecte d'information et recouvrement]** to [**administrateur de la sécurité**]. |
| **FMT_MSA** | **Management of security attributes** |
| FMT_MSA.1 | Management of security attributes |
| FMT_MSA.1.1 | The TSF shall enforce the [**SFP.ACCESS_OBJ]** to restrict the ability to [**changer la valeur par défaut, modifier ou supprimer**] the security attributes [**clés d'accès et rôles**] to [**administrateur des accès et administrateur de la sécurité**]. |
| FMT_MSA.2 | Secure security attributes |
| FMT_MSA.2.1 | The TSF shall ensure that only secure values are accepted for [**clés d'accès**]. |
| FMT_MSA.3 | Static attribute initialisation |

| | |
|---|---|
| FMT_MSA.3.1 | The TSF shall enforce the [**SFP.ACCESS_OBJ**] to provide [**restrictive**] default values for security attributes that are used to enforce the SFP. |
| FMT_MSA.3.2 | The TSF shall allow the [**administrateur des accès et administrateur de la sécurité**] to specify alternative initial values to override the default values when an object or information is created. |
| **FMT_MTD** | **Management of TSF data** |
| FMT_MTD.1 | Management of TSF data |
| FMT_MTD.1.1 | The TSF shall restrict the ability to [**changer la valeur par défaut, modifier ou supprimer**] the [**stratégies de sécurité**] to [**administrateur de la sécurité**].<br>**Raffinement non éditorial :**<br>**Ce composant est implémenté par la signature des politiques de sécurité.** |
| **FMT_SMF** | **Specification of Management Functions** |
| FMT_SMF.1 | Specification of Management Functions |
| FMT_SMF.1.1 | The TSF shall be capable of performing the following management functions: [<br>- **Les fonctions de gestion des accès**<br>- **La fonction de recouvrement**<br>- **La fonction de collecte d'information pour le support]** |
| **FMT_SMR** | **Security management roles** |
| FMT_SMR.1. | Security roles |
| FMT_SMR.1.1 | The TSF shall maintain the roles [**administrateur de la sécurité, administrateur des accès et utilisateur de la TOE**]. |
| FMT_SMR.1.2 | The TSF shall be able to associate users with roles. |

## 9.5. Class FPT: Protection of the TSF

| | |
|---|---|
| **FPT_TST.1** | **TSF self test** |
| FPT_TST.1 | TSF testing |
| FPT_TST.1.1 | The TSF shall run a suite of self tests [**during initial start-up, periodically during normal operation**] to demonstrate the correct operation of [the TSF]. |
| FPT_TST.1.2 | The TSF shall provide authorised users with the capability to verify the integrity of [**no parts of TSF data**]. |
| FPT_TST.1.3 | The TSF shall provide authorised users with the capability to verify the integrity of [**no parts of TSF**]. |