ERA, éditeur de règles pour le module Amon

EOLE 2.5

création : Mai 2015 Version : révision : Avril 2018

EOLE 2.5

Version : révision : Avril 2018

Date : création : Mai 2015

Editeur : Pôle national de compétences Logiciels Libres

Auteur(s) : Équipe EOLE

Pour toute information concernant ce projet vous pouvez nous joindre :

- Par courrier électronique : eole@ac-dijon.fr
- Par FAX : 03-80-44-88-10
- Par courrier : EOLE-DSI - 2G, rue du Général Delaborde - 21000 DIJON
- Le site du Pôle national de compétences Logiciels Libres : http://eole.orion.education.fr

Table des matières

Chapitre 1

Introduction

1. Présentation

Présentation et fonctionnalités

L'outil EOLE de génération de règles de pare-feu[p.35] pour les modules Amon et AmonEcole se nomme ERA[p.34].

Il permet de gérer la description de la politique de sécurité d'un pare-feu[p.35]. Cette politique est sauvegardée intégralement dans un fichier de type XML avec un format spécifique à l'application.

Par un processus de compilation, ERA transforme le fichier XML en un bloc de règles iptables[p.34], de manière à instancier ces règles sur un pare-feu[p.35] cible.

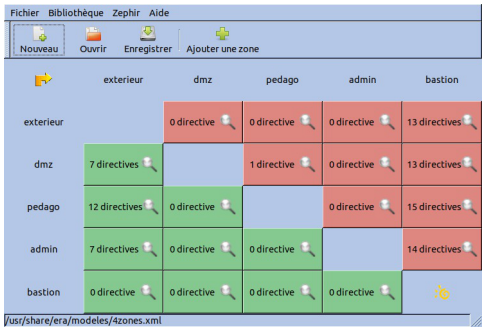
ERA et sa documentation sont sous licence libre.

Un logiciel en deux parties

- L'interface de conception permet d'organiser la politique de filtrage et l'enregistre dans un fichier XML ;
- le compilateur génère le script iptables , par compilation, à partir du fichier XML de description du pare-feu.

Seul le format XML est utilisé par le module Amon. L'exportation au format iptables[p.34] permet d'être utilisable sur un autre pare-feu disposant de Netfilter.

Il n'est bien sûr pas nécessaire de connaître la syntaxe iptables pour manipuler ERA. Le but d'un tel logiciel est justement de s'abstraire de la syntaxe iptables, afin de pouvoir concevoir un pare-feu sans pour autant être un expert. Pour cela, l'interface graphique de ERA est un outil intéressant :



L'interface graphique d'ERA

le fichier lance.firewall

Sur le pare-feu Amon, le fichier `lance.firewall` présent dans `/sbin/` est un fichier de règles iptables qui a été généré par ERA.

> Remarquons que si le serveur sur lequel est lancé le compilateur de règles est en mode conteneurs, ERA va générer autant de fichiers de règles iptables que de conteneurs.

2. Les fichiers XML de modèles

Un modèle[p.34] est un fichier de description du pare-feu. Le format d'enregistrement est un format XML. Divers modèles caractéristiques de description de pare-feu sont disponibles dans le répertoire `/usr/share/era/modeles` et sont des exemples de types de pare-feux (deux, trois, quatre ou cinq cartes réseau).

En général il est préférable, pour commencer un pare-feu, de partir d'un des modèles exemples, et d'y rajouter des directives (ou bien d'en enlever). Partez plutôt du modèle qui correspond au nombre de cartes réseau dont vous disposez sur le serveur.

Lorsque vous modifiez un modèle exemple, il faut impérativement l'enregistrer dans un fichier différent. Sinon, il sera écrasé à la mise à jour suivante.

De plus, il faut que vos nouveaux fichiers XML soient enregistrés dans le répertoire `/usr/share/era/modeles/`.

Charger un modèle à la ligne de commande.

Il est possible d'ouvrir directement un modèle à la ligne de commande. Pour cela, il suffit de spécifier l'option **-f** avec le nom du fichier.

Par exemple :

```
era -f /usr/share/era/modeles/3zones-amonecole.xml
```

Le format XML interne est facilement lisible avec un éditeur de texte (ou un éditeur XML) si l'on est familiarisé avec :

- la notation XML ;
- les différents concepts propres à ERA (tableau des flux, services, directives, ...).

Version des modèles ERA

Les modèles XML ERA sont versionnés.

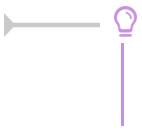
Dans les modèles fournis, la version est indiquée dans l'attribut version de la balise `<firewall>`.

```
1 <firewall name="Concatenated_Do_Not_Edit" netbios="1" qos="0" version="2.4">
```

Le numéro de version des modèles XML ERA est incrémenté lorsque des changements importants sont apportés à la DTD[p.33].

Les numéros de version sont généralement corrélés avec les versions des modules EOLE : 2.0, 2.3, 2.4, 2.42.

Sur EOLE 2.5.1 et supérieur, les commandes `instance`, `reconfigure` et le redémarrage du service bastion affichent un avertissement si le modèle de pare-feu utilisé possède une version inférieure à celle gérée par l'outil ERA du module.



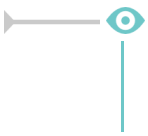
Il est possible de mettre à niveau un modèle XML existant en l'éditant et en le sauvegardant à l'aide du logiciel ERA.

3. Les variables Creole

ERA[p.34] a été conçu dans le cadre du projet EOLE et pour le pare-feu Amon. Il peut très bien être utilisé en dehors de ce cadre, mais c'est sur un module Amon qu'il devient vraiment possible de déployer toutes les possibilités du logiciel.

Il est possible, à plusieurs endroits de l'interface, d'insérer des variables Creole[p.33] (elles commencent par `%%`) plutôt que des valeurs fixes.

Le fichier XML de description de pare-feu devient alors un template[p.35] Creole.



Dans la fenêtre d'édition d'une zone, entrer une valeur du type `%%ip_variable` plutôt qu'une valeur IP fixe.

Ces variables seront instanciées sur un serveur EOLE. Mais elles peuvent aussi être utilisées pour le déploiement d'autres pare-feux tant que Netfilter est présent.

Limitations de l'intégration entre ERA et Creole

Cette intégration des variables Creole dans ERA a des limites dans le cas des variables multivaluées. Une variable multivaluée au sens de Creole est une variable dont les valeurs sont multiples (c'est une liste d'ips, de networks, etc...).

Il est autorisé d'utiliser des variables multivaluées dans ERA, mais il y a une limitation : si dans une directive donnée plusieurs variables multivaluées sont utilisées (par exemple au niveau d'une extrémité source, d'une extrémité de destination ou d'un service, ou d'un port de redirection...), alors il faut que les autres variables multivaluées utilisées soient déclarées dans le dictionnaire Creole comme esclaves de la première variable multi-valuée, sinon le cas d'utilisation ne sera pas géré.

Le support des groupes de variables multivaluées est très partiel dans ERA, si dans une directive une variable multivaluée est utilisée alors elle doit être déclarée comme maître dans le dictionnaire Creole, et il ne faut pas qu'il y ait dans cette directive une deuxième variable multivaluée **indépendante**, donc les autres variables impliquées sont soit des variables multivaluées esclaves, soit simplement des variables Creole non-multivaluées.

Chapitre 2

Utilisation

1. Les zones de sécurité

Présentation

L'éditeur ERA est un outil de conception par zones[p.35]. Une zone[p.35] correspond physiquement à une carte réseau. Cela permet de découper le parc de machines en réseau ou sous-réseau.

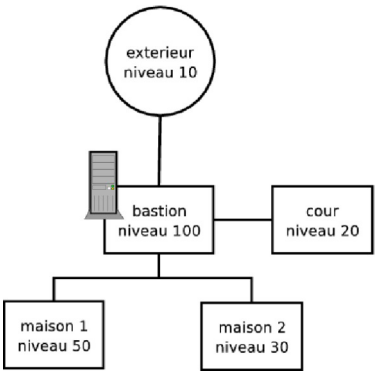
Le pare-feu lui-même étant une zone à part, appelée par convention `bastion`.

Les zones sont ensuite ordonnées par niveau de sécurité[p.35] sous forme d'entiers de 0 à 100.

100 est le niveau de sécurité maximal et correspond à la zone `bastion`. Cela permet de "cartographier" tout le réseau.

Par convention, le niveau de sécurité le plus faible de toutes les zones est affecté à la zone "extérieur".

Les machines de la zone ont un accès complet aux zones de niveau inférieur et aucun accès à celles de niveau supérieur.



Les niveaux de sécurités des différentes zones (vue centrée sur le bastion)

Une zone correspond à un réseau et dans cette zone, on retrouve des sous-réseaux et des machines, correspondant à la notion d'extrémité[p.34] utilisée dans ERA.

Une extrémité est un sous-ensemble d'une zone :

- Elle est définie par un ensemble d'adresses IP ou une adresse réseau.
- Elle hérite du niveau de sécurité de la zone à laquelle elle appartient.

Ajouter une zone

Il est possible à tout moment, même après la conception initiale du modèle, d'ajouter une zone de sécurité. L'ajout d'une zone de sécurité se fait en cliquant sur le bouton `Ajouter Zone` de la barre d'icônes.

Ajouter une zone au tableau des flux

Les cases des noms des zones sont cliquables.

Un clic droit dans une case des noms de zones permet d'afficher les zones ainsi que les extrémités[p.34] qui y sont associées.

Fenêtre d'édition de zone

les trigrammes (préfixes) de zones

Dans le choix des noms de zone :

- les trois premières lettres (trigramme) du nom de la zone sont discriminantes, par exemple : `statistique` et `station` sont des noms de zone incompatibles (c'est la même zone `sta`) ;
- le mot clef `bastion` est réservé (pour la zone du bastion lui-même) ;
- le mot clef `extérieur` est également réservé (pour la zone extérieure, internet).

la gestion des VLAN

Une zone peut aussi représenter un VLAN.

C'est une bonne pratique de créer une nouvelle zone pour gérer un VLAN.

Il n'est pas possible de créer une zone pour tous les VLAN.

S'il y en a plusieurs il faut les créer un à un manuellement.

Syntaxe Creole pour la création des VLAN

Il est fréquent que les valeurs des IP des VLAN soient stockées dans une variable Creole, et que cette variable soit multiple (une variable multiple au sens Creole est une variable qui

contient une liste de valeurs). Il faut alors manier correctement la syntaxe Creole pour créer une zone de VLAN.

Exemple de création d'un VLAN de eth1

Dans la widget de création d'une zone, il faut mettre une IP et un netmask variable :

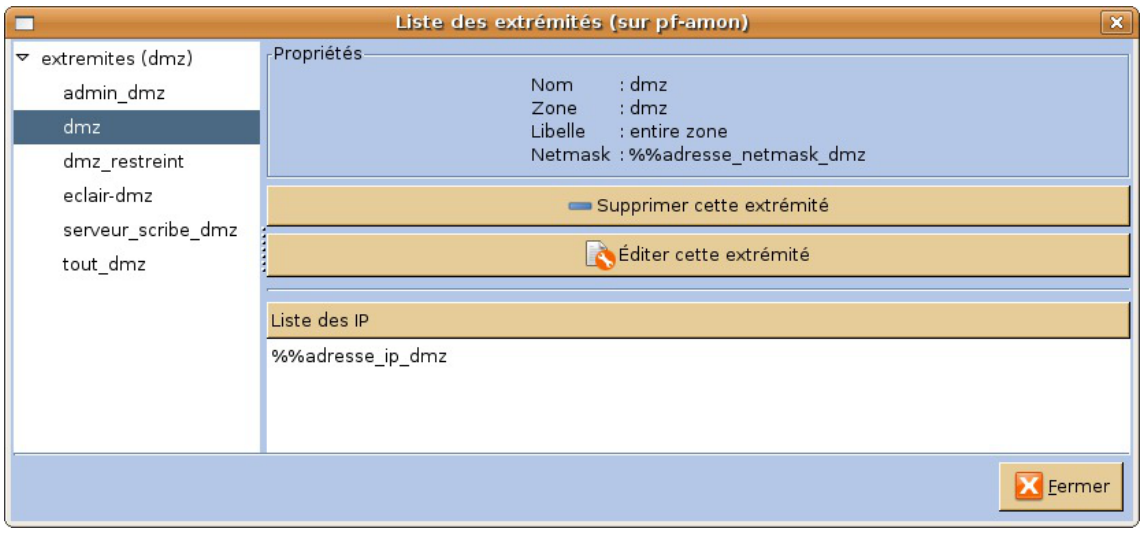
`ip variable : %%id vlan eth1[0].adresse ip vlan eth1`

`netmask variable : %%id vlan eth1[0].adresse netmask vlan eth1`

Ajouter une extrémité

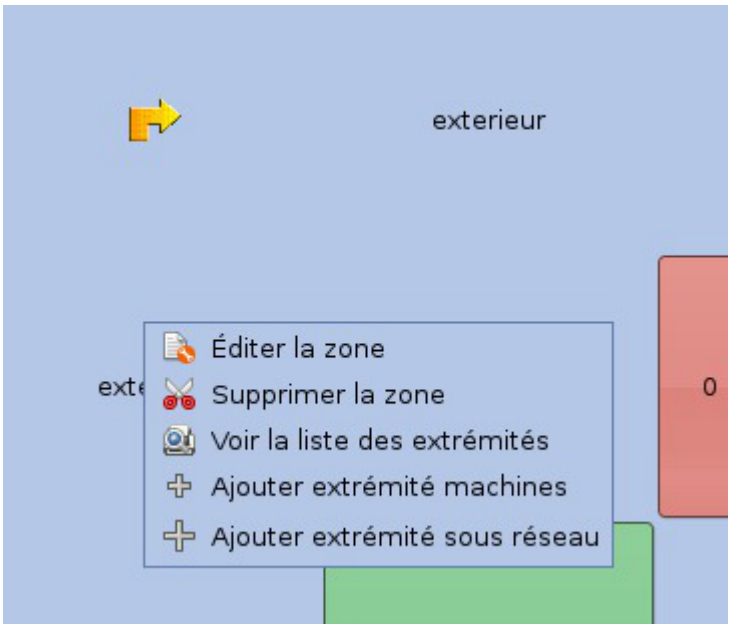
La liste des extrémités est disponible dans le menu `bibliothèque` / `extrémités`.

Il est également possible de lister les extrémités d'une zone, en cliquant droit sur le bouton de la zone et en sélectionnant `voir la liste des extrémités`.

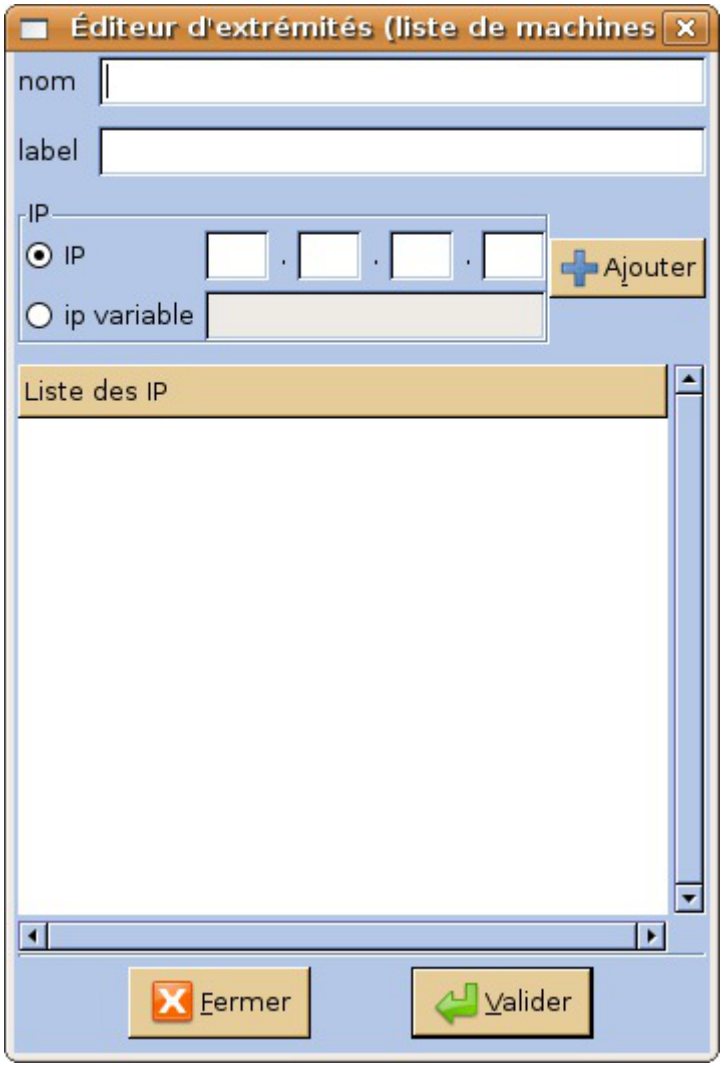


Liste des extrémités

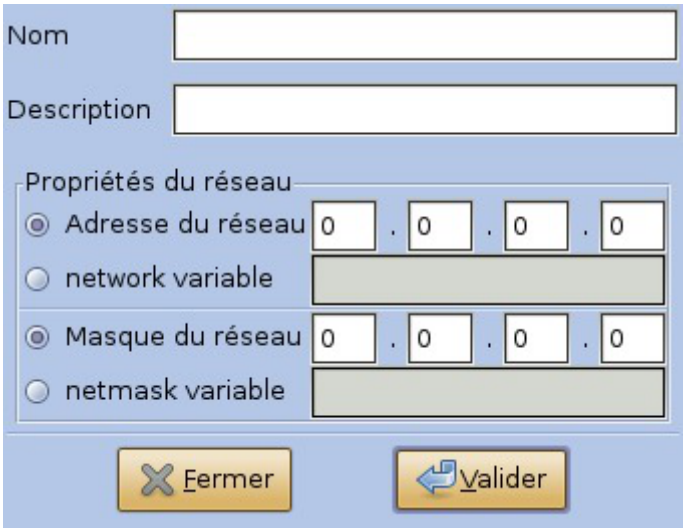
Pour créer une nouvelle extrémité, faire un clic droit dans la zone dans laquelle vous voulez l'inclure. Ensuite, choisir `définir un ensemble de machines` ou `définir un sous-réseau` suivant que vous voulez inclure un groupe d'IP ou un sous-réseau.



Un clic droit sur le nom d'une zone affiche le menu contextuel relatif à cette zone



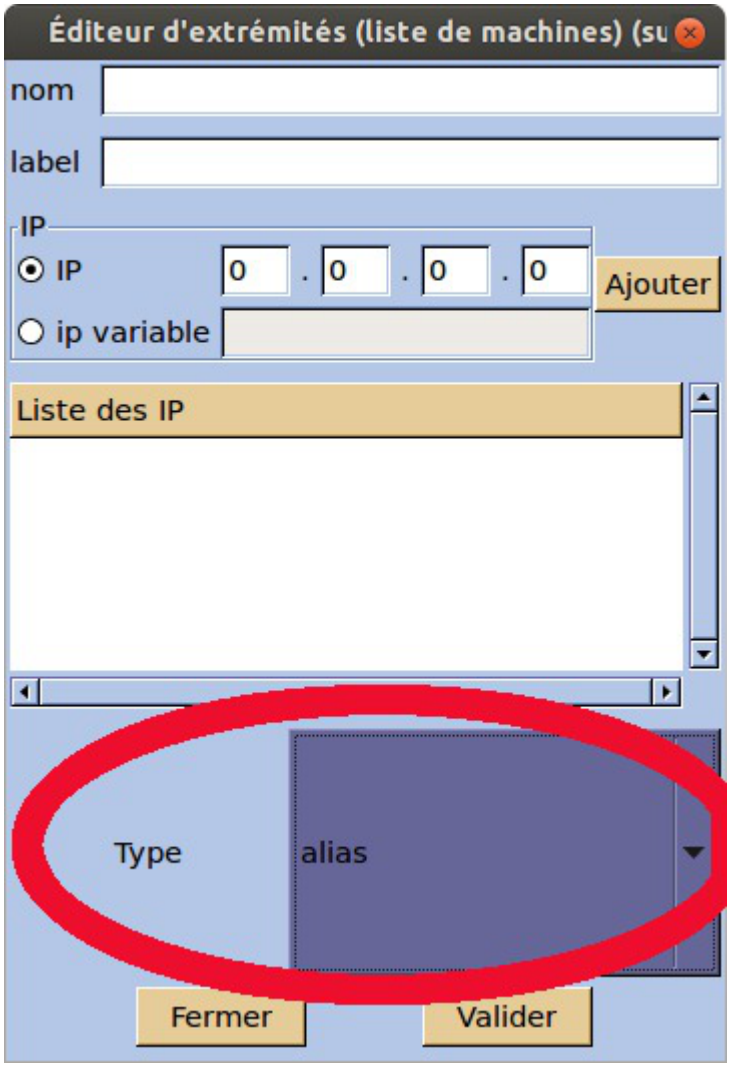
Ajout d'une extrémité dans le cas d'une liste de machines



Ajout d'une extrémité de type sous réseau

Les alias IP doivent être gérés **comme des extrémités** et non comme une zone : **un alias n'est pas une zone**.

Pour ajouter une extrémité de type alias, il faut spécifier le type "alias" dans l'éditeur d'extrémité :



Il est fréquent que les valeurs des IP des alias soient stockées dans une variable Creole, et il est fréquent aussi que cette variable soit multiple (une variable multiple au sens Creole est une variable qui contient une liste de valeurs). Il faut alors manier correctement la syntaxe Creole pour créer une extrémité qui est un alias.

Dans la widget de création d'une extrémité, il faut alors mettre une IP et un netmask variable.

Dans la zone correspondant à la carte, créer une extrémité (clic droit sur la case de la zone).

Un alias de eth2 doit être créé de la façon suivante :

`ip variable : %%alias_ip_eth2[0]`

`network variable : %%alias_network_eth2[0]`

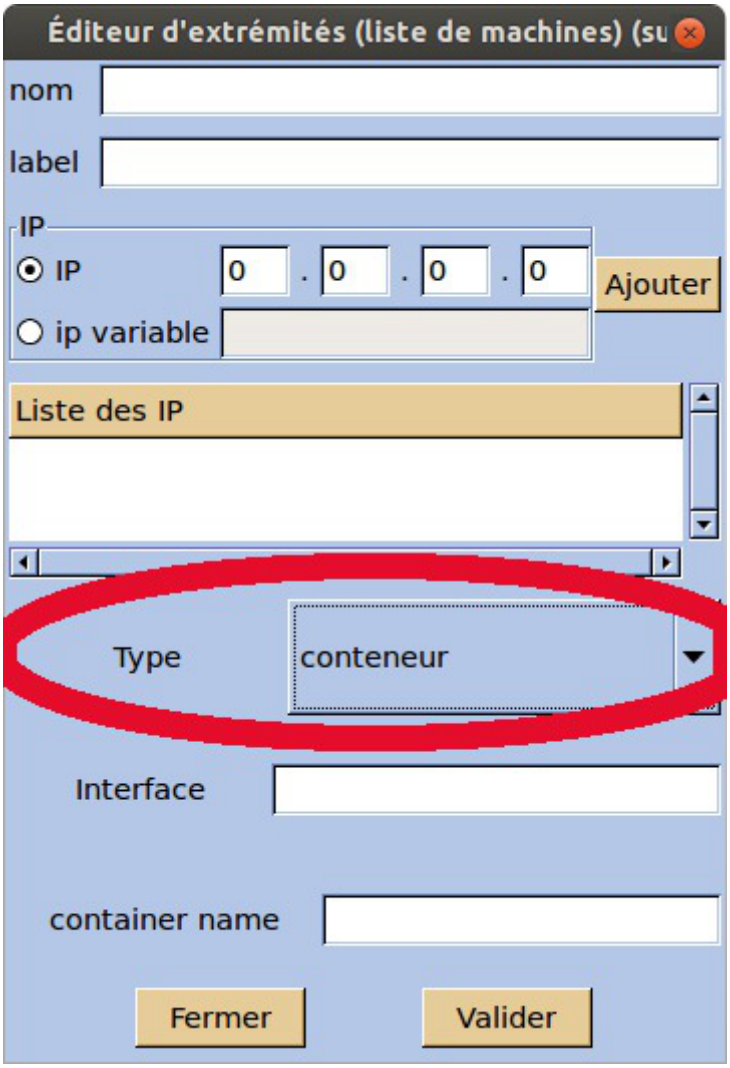
Il sera possible ensuite de créer une directive avec cette extrémité plutôt qu'avec l'extrémité correspondant à l'IP de la zone elle-même.

Les mauvaises fausses bonnes idées pour créer un alias

- créer une zone supplémentaire avec une carte eth0:X ;
- aller de suite dans les inclusions statiques ;
- utiliser la variable eth0 de Creole comme IP multivaluée.

Les extrémités de type conteneur

Il est possible également de créer une extrémité dans la zone **bastion**. Dans la zone bastion il y a depuis la version 2.4 un nouveau type d'extrémité, le type **conteneur**. Ce type d'extrémité permet de créer des directives à destination des conteneurs (directives de type INPUT).



Une extrémité de type conteneur est à destination du conteneur. Elle nécessite deux informations : le nom de l'interface (typiquement : "br0", "eth1", ...), et le nom du conteneur (typiquement : "bdd", "internet"...)

2. Les flux

Présentation

Dans ERA, les règles sont systématiquement classées d'après la zone d'origine et la zone de destination. ERA est donc conçu autour du concept de flux[p.34] plutôt que centré sur la notion de règle. Par voie de conséquence, chaque zone est reliée à une autre zone par des flux.

A l'intérieur d'un flux, on trouve deux flux orientés, le "flux montant[p.34]" et le "flux descendant[p.34]".

Les "flux montants[p.34]" concernent les zones[p.35] d'un niveau de sécurité plus faible vers un niveau de

sécurité plus élevé, et réciproquement pour les "flux descendants[p.34]".

Pour pouvoir ordonner les flux en vue d'une cohérence globale, il convient ensuite de modéliser le " tableau des flux[p.35]".

Ce tableau correspond à l'ensemble des flux du modèle de sécurité à l'intérieur duquel seront rangées les règles (ou directives).

	10	20	100
10		Montant	Montant
20	Descendant		Montant
100	Descendant	Descendant	

Directives montantes et descendantes dans la matrice de flux

Lorsque les flux montants et les flux descendants sont définis, une politique par défaut est automatiquement spécifiée. Ici, la politique de sécurité par défaut qui résulte de la matrice de flux est :

	10	20	100
10		Interdit	Interdit
20	Autorisé		Interdit
100	Autorisé	Autorisé	

Repérage des types de directives (autorisation ou interdiction) dans la matrice de flux

Niveaux de sécurité égaux

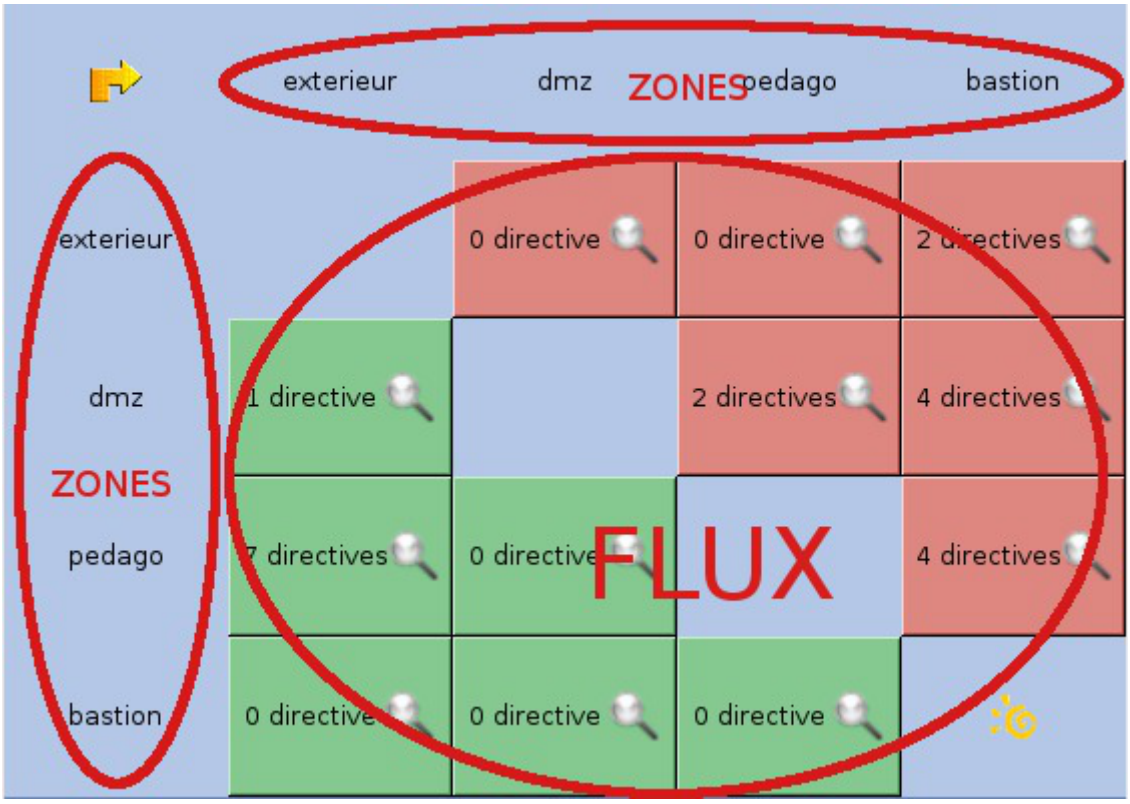
Lorsque deux zones ont deux niveaux de sécurité égaux, alors la politique par défaut est une interdiction des deux côtés (flux montants et descendants).

Changement de la politique par défaut

Il est possible d'inverser le comportement de la politique par défaut. On peut choisir d'interdire les flux d'une zone vers une autre par défaut.

L'interface de conception

La fenêtre principale représente un tableau composé de cases de zones et de cases de flux.



Interface d'ERA avec un modèle chargé

Les cases des flux sont colorés. Les cases de couleur verte sont en "autorisation" par défaut et les cases de couleur rouge sont en "interdiction" par défaut.

La couleur rouge indique que le flux orienté est interdit, tandis que la couleur verte montre que le flux orienté est autorisé.

3. Les directives

3.1. Présentation

Une directive[p.34] est une règle concernant un service ou un groupe de services entre deux extrémités.

Cette règle peut être de type "interdiction", "redirection", "source NET" ou "destination NAT".

3.2. Les services et les groupes de services

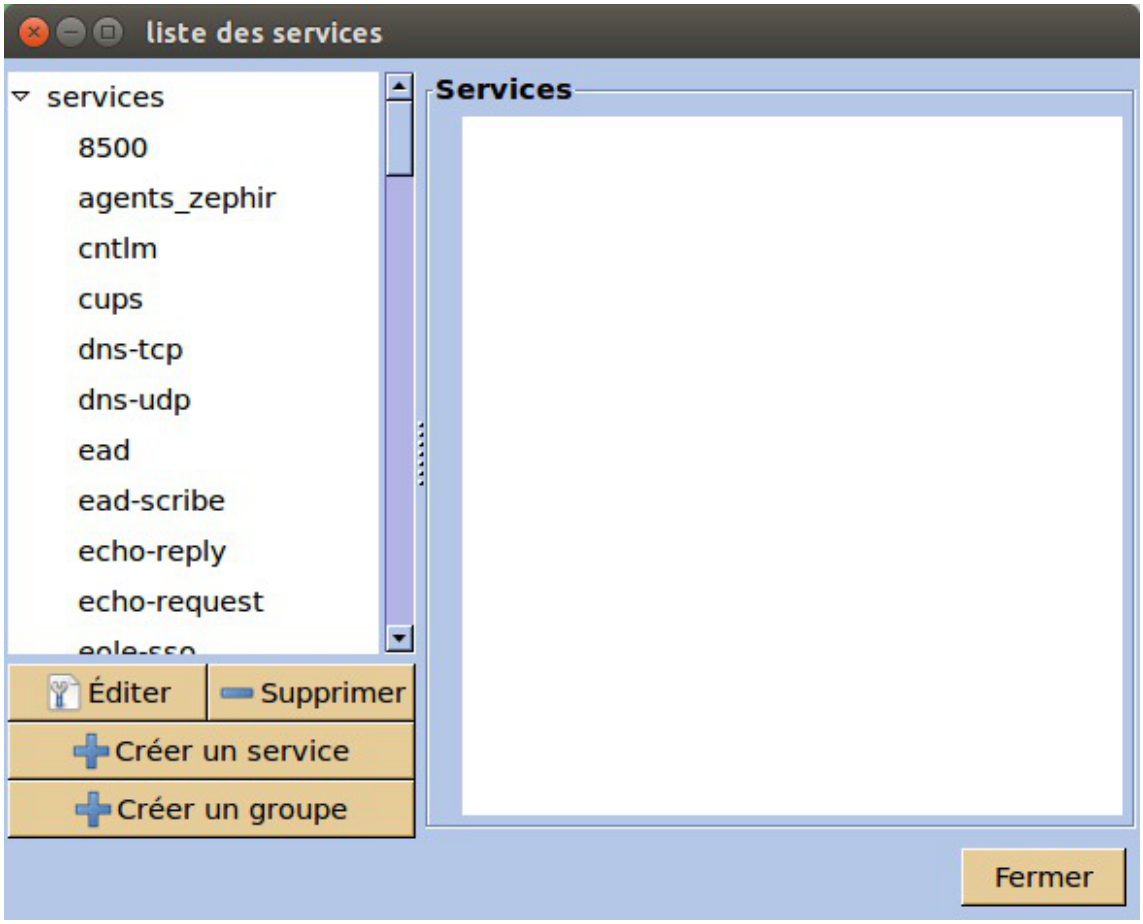
Avant de pouvoir créer une directive, il faut d'abord créer un service[p.35].

Par exemple, le service "serveur web" est défini par le protocole HTTP sur le port 80.

Il y a déjà une bibliothèque de services prédéfinis dans ERA.

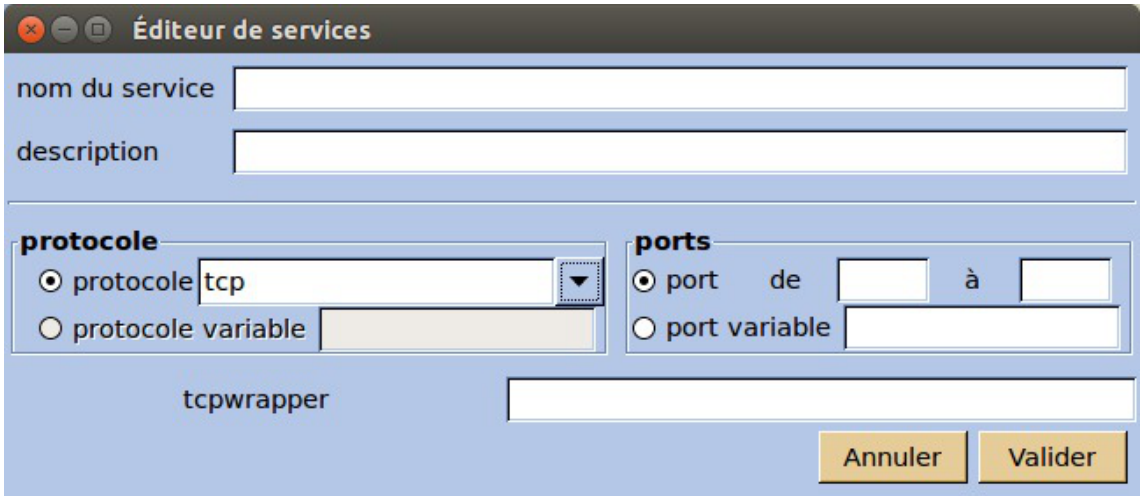
> Pour lister ces services, aller dans le menu : `Bibliothèque > services`.

Pour créer un service, aller dans le menu : `Bibliothèque > services`.



Liste et édition des services

Créer ou modifier un service signifie renseigner les noms, descriptions, protocoles et ports.



Ajout d'un service

Remarquons que si on choisit un port égal à 0, cela équivaut à saisir de 0 à 65535.

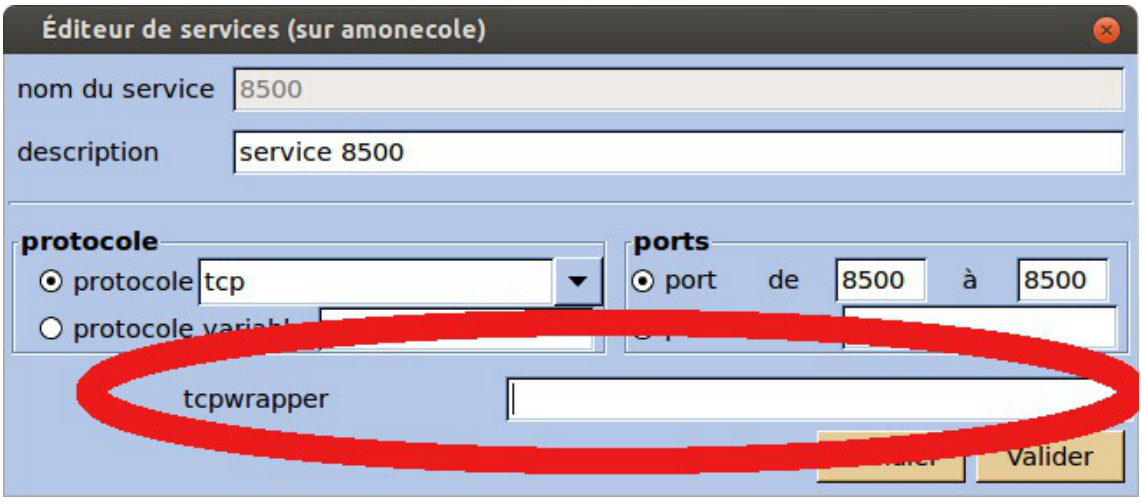
> Si on veut que le service ne concerne qu'un seul port, il faut mettre deux fois le même numéro de port.

Depuis EOLE 2.4, l'utilisation d'une variable Creole pour définir le type de protocole à utiliser n'est plus fonctionnelle.

Cette fonctionnalité n'est plus disponible dans l'interface à partir d'EOLE 2.6.

implémenter un service avec tcpwrapper

Pour prendre en compte le tcpwrapper avec ERA, ça se passe au niveau des services. Il suffit de renseigner le nom tcpwrapper du service (le nom tel qu'il doit apparaître dans le fichier **hosts.allow**) et le tcpwrapper sera pris en compte dès qu'une directive utilisant ce service sera créée.



le tcpwrapper en mode conteneur

Remarquons que ERA va générer un fichier tcpwrapper, classiquement le fichier /**etc**/**hosts.allow**, mais que en mode conteneur autant de fichiers seront générés que de conteneurs.

3.3. L'éditeur de directives

Un clic droit ou un double clic dans une case de flux du tableau permet de visualiser la liste des directives de façon synthétique.

Les directives sont triées par ordre croissant. C'est l'ordre dans lequel seront appliquées les règles sur le pare-feu cible.

Les directives de `nat` et `redirection` sont appliquées forcément avant les autres. Ceci est le comportement de Netfilter[p.35].

Depuis cette fenêtre, il est possible d'éditer une nouvelle directive (en double-cliquant dessus) ou d'en ajouter une si nécessaire.

Pour construire une directive, il faudra au moins deux extrémités (entre deux zones) et un service (ou groupe de services), qui doit être renseigné par glisser-déplacer.

Si vous ne renseignez pas les extrémités, c'est la zone entière qui est prise (plus précisément l'extrémité désignant la zone entière).

Différence entre zone entière et zone restreinte

La zone entière est le réseau correspondant à une carte réseau du pare-feu. Cela correspond au réseau local ainsi que d'éventuels sous-réseaux derrière une passerelle. Elle est nommée *<nom de la zone>*.

La zone restreinte ne correspond qu'au sous-réseau. Elle est nommée *<nom de la zone>_restreint.*

A chaque directive est associé un service ou un groupe de services qu'il est nécessaire de renseigner par glisser-déposer.

3.3.1. Les types de directives

Les types de directives

Il y a plusieurs types de directives :

- autorisation
- interdiction
- redirection
- SNAT
- DNAT
- FORWARD

Les directives d'autorisation et d'interdiction

Une directive est dans une case de flux et elle s'oppose à la politique par défaut du flux. Si le flux est en autorisation, la directive propose une interdiction et inversement.

En plus du filtrage simple, d'autres fonctionnalités sont proposées.

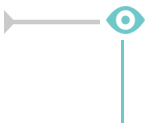
Les directives de redirection

Une directive de type redirection permet de rediriger une requête d'un port déterminé vers un port de la machine elle-même (bastion).

Cette fonctionnalité est particulièrement intéressante dans le cas du proxy transparent. Toutes les requêtes destinées à des serveurs web seront redirigées automatiquement vers le service proxy installé sur le serveur.

Les directives de DNAT/SNAT

Le NAT permet de modifier l'adresse source (SNAT) ou destination (DNAT) d'une requête.



Le SNAT est utilisé pour toutes les requêtes provenant de la zone pédago vers l'extérieur. Cela permet de transformer les adresses source locales en adresses source extérieures.

Le DNAT et le SNAT ne sont pas autorisés si la directive est authentifiée.

Les directives FORWARD

Le FORWARD permet d'autoriser la translation un réseau vers un autre

3.3.2. Les plages horaires

Création d'une plage horaire

Les plages horaires sont définies depuis le menu `Bibliothèque > plage horaire`.

Il y a trois manières de définir une plage horaire :

- les heures de début et de fin ;
- les dates de l'année de début et de fin ;
- les jours de la semaine.

Les informations indispensables sont : le nom et une ou plusieurs de ces trois manières.

Affectation d'une plage horaire à une directive

Il est possible de définir une plage horaire à l'intérieur de laquelle la directive sera activée.

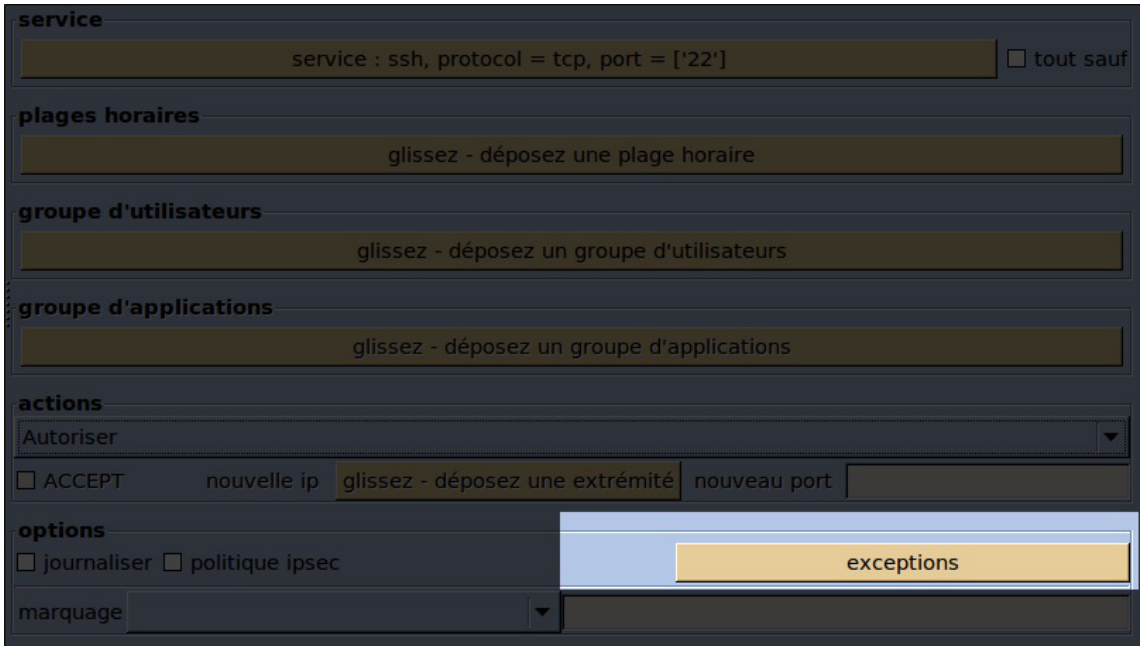
Depuis l'éditeur de directives, glisser-déposer une plage horaire. Affecter une plage horaire à une directive.

3.3.3. La journalisation

La case "journaliser" permet de tenir un journal des évènements (logs) de la directive (grâce à ULOG).

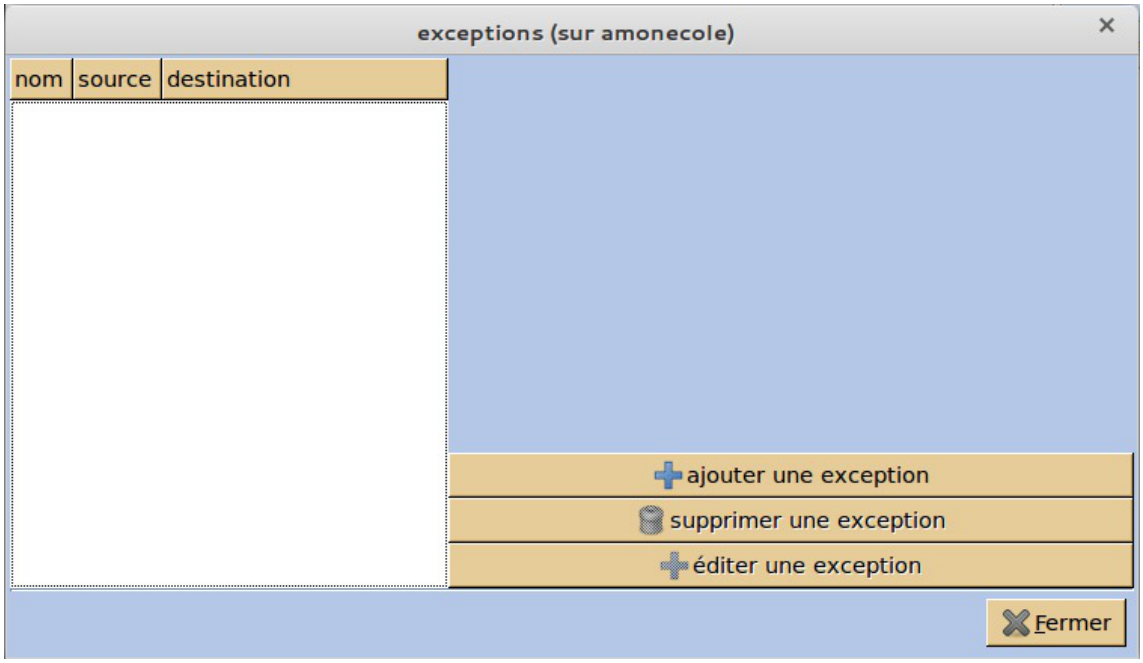
3.3.4. Gérer des exceptions

Dans l'éditeur de directives il est possible d'ajouter des exceptions.



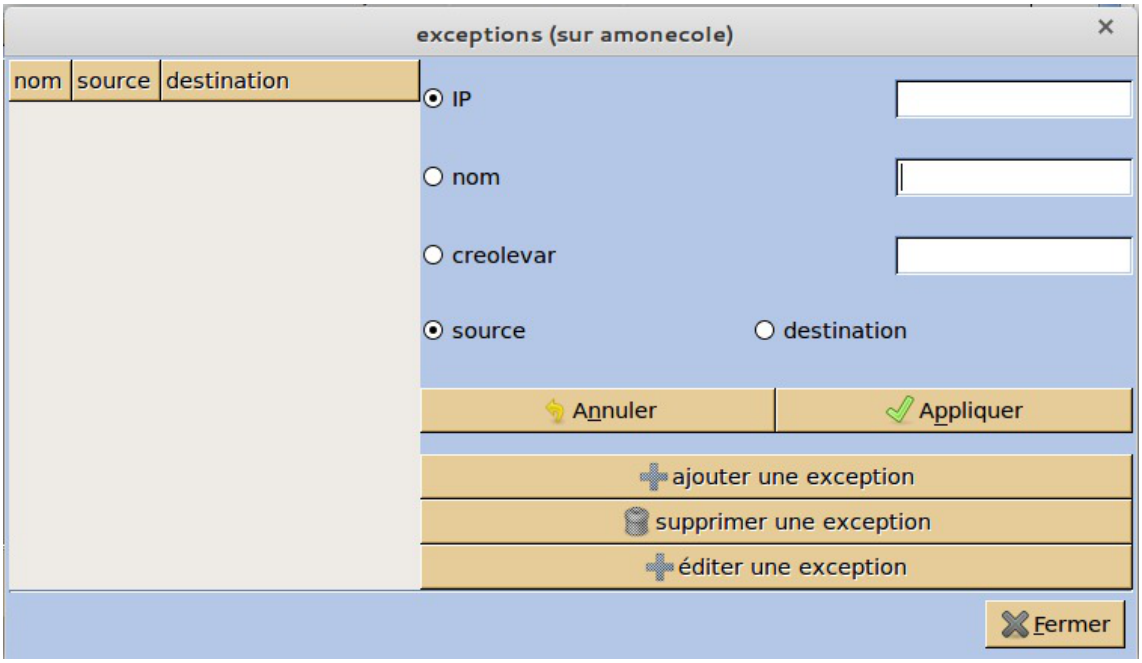
L'éditeur d'exceptions permet :

- d'ajouter une exception ;
- d'éditer une exception ;
- de supprimer une exception.



L'exception peut se faire :

- sur une adresse IP ;
- sur un nom de domain ;
- sur une variable Creole.



3.3.5. Le marquage

Le marquage est une fonctionnalité avancée de iptables[p.34] permettant d'identifier un paquet grâce à une marque spécifiée dans l'interface.

3.3.6. Les directives optionnelles

Présentation

Une directive optionnelle[p.33] est une directive qui va être activable ou désactivable depuis l'interface EAD[p.33].

Pour cela, il est indispensable d'affecter un libellé optionnel à cette directive. Il est aussi possible de choisir un libellé optionnel préexistant dans la liste des libellés affectés aux directives, ce qui crée une notion de groupe de directives optionnelles.

La directive est étiquetée comme optionnelle

Un libellé optionnel sert de tag (d'identifiant). Il peut être composé de caractères alphanumériques [1-9] [a-z] [A-Z] et éventuellement de "_" ou d'espaces. Il est impératif de ne utiliser d'autres caractères accentués.

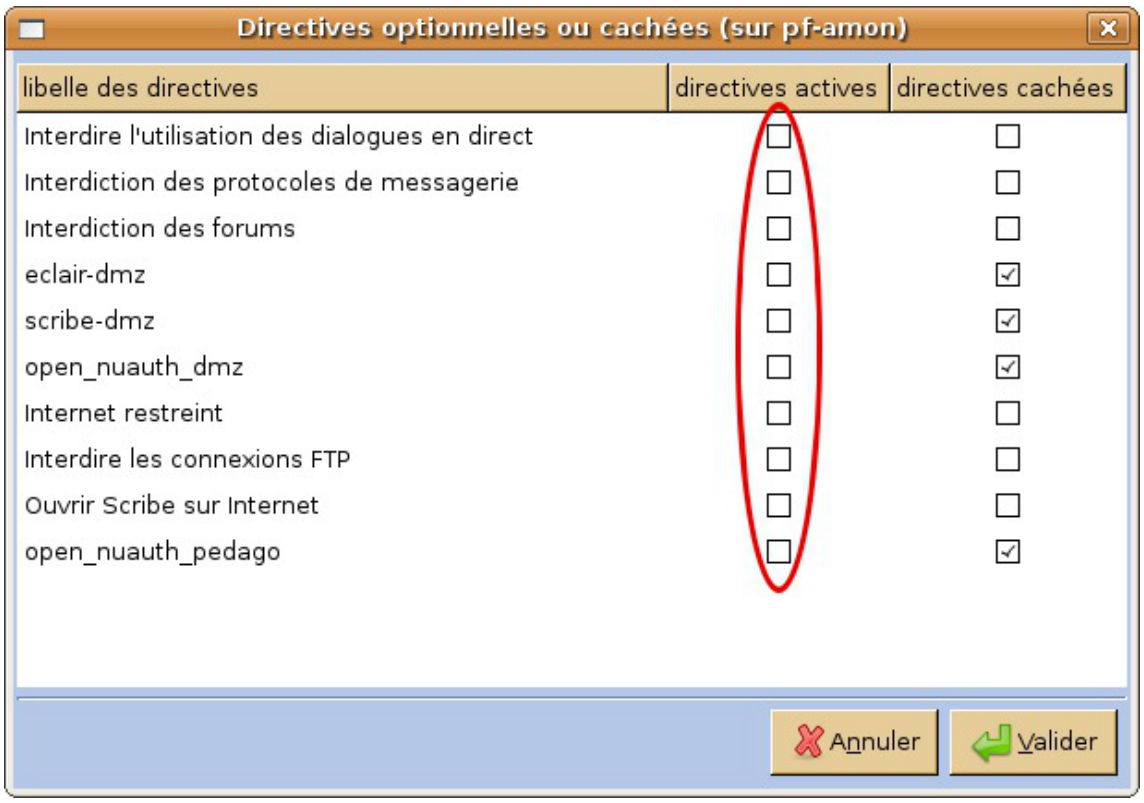
Directive optionnelle active

Une directive optionnelle n'est pas active par défaut dans ERA, c'est-à-dire que la directive ne sera pas appliquée sur le serveur cible. Pour l'appliquer, il faut aller la cocher comme active dans l'interface EAD.

Mais il est possible de rendre une directive active par défaut dans ERA. Dans ce cas, il faudra aller dans l'interface EAD pour la désactiver.

L'état actif et la possibilité de marquer une directive comme optionnelle sont deux notions différentes.

Pour activer une directive, aller dans `Bibliothèque` / `Directives optionnelles`.



Fenêtre de la bibliothèque permettant d'étiqueter une directive comme optionnelle

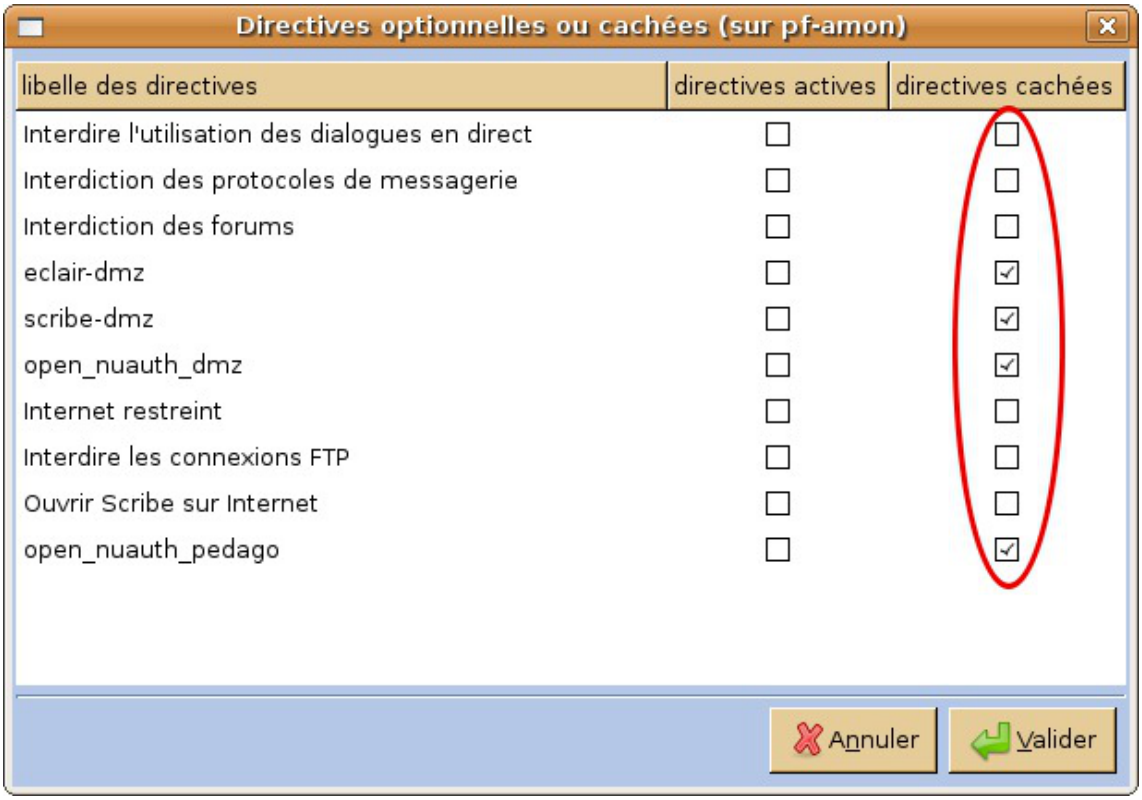
Directive optionnelle active et inactive

Dans le cas de l'activation/désactivation d'une directive optionnelle, il faut bien comprendre que c'est l'EAD qui prime par rapport à ERA. À la première instanciation du serveur, ERA détermine si la directive optionnelle est active ou inactive, mais une fois le serveur est instancié c'est depuis l'interface EAD qu'il faut renseigner le statut actif/inactif de la directive optionnelle en question.

Les directives optionnelles cachées

Une directive optionnelle cachée est une directive optionnelle qui n'apparaîtra pas dans l'EAD. Elle est activable uniquement par une procédure particulière.

Pour créer une directive optionnelle cachée, aller dans `Bibliothèque` / `Directives Optionnelles` et cocher `directives cachées`.



Les directives cachées

Une directive cachée est désactivée par défaut. Pour l'activer, il faut patcher le template `active_tags` afin d'y ajouter le libellé optionnel de la directive (un libellé par ligne).

Il est préférable d'utiliser un libellé optionnel court (par exemple "`ActiverProxy`" plutôt que "activer le proxy").

Dans le template `active_tags`, ne pas mettre de commentaire.

Voir aussi...

Directives optionnelles ERA depuis l'EAD [p.27]

4. La qualité de service

La qualité de service ne concerne que les flux des zones internes vers l'extérieur. C'est une QOS[p.35] *externe*.

Le qualité de service est un système de **minimum garanti**.

Elle n'entraîne pas de sous-utilisation de la bande passante, car si une zone n'atteint pas son minimum d'utilisation, ce qui reste est réparti dynamiquement entre les autres zones.

Il est possible d'accéder à la fenêtre de gestion de la QOS[p.35] de deux manières :

- depuis le menu `Bibliothèque` / `Qualité de service (QOS)` ;
- en cliquant sur la zone *Extérieur* depuis le tableau des flux.

Dans cette boîte de dialogue, il faut :

- fixer des valeurs de bande passante en *upload* et en *download* (c'est-à-dire les flux globaux disponibles entre l'intérieur et l'extérieur), en kilo bits par seconde (soit un débit de 1000 bits par seconde) ;
- à l'aide des poignées de manipulation des différentes boîtes représentant chaque zone, affecter un pourcentage de flux relatif à chaque zone.

Remarquons que il est tout-à-fait possible de mettre des variables Creole comme valeurs possibles de QOS en upload et en download

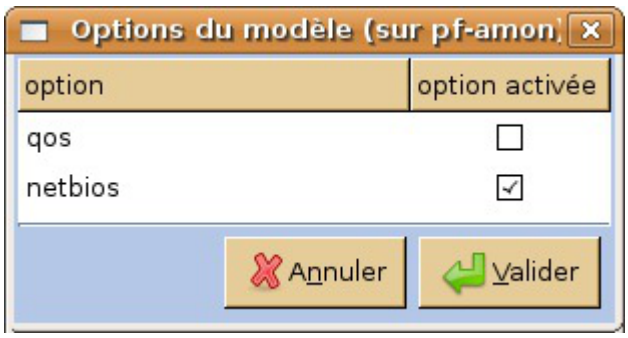
La QOS peut être définie dans un modèle sans être activée !

Pensez à l'activer dans les options du modèle (`Bibliothèque->Options du modèle`).

La désactivation de la QOS n'est effective que si le fichier `/etc/qoseole.conf` est supprimé.

5. Les options du modèle

Il est possible d'ajouter des règles spécifiques à netbios et à la QOS depuis le menu `Bibliothèque` / `Options du modèle` .



Fenêtre des options du modèle

Activer netbios permet d'ajouter des règles permettant de bloquer les requêtes du réseau Microsoft vers l'extérieur. Cette règle est activée par défaut.

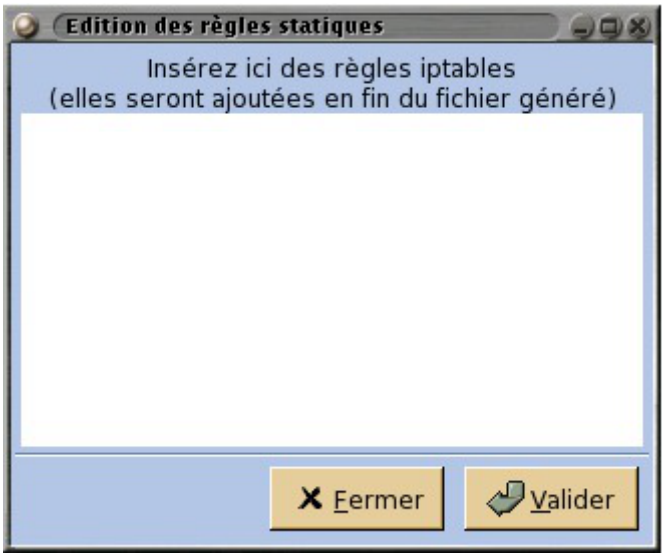
Si vous voulez utiliser les règles de Qualité de Service (QOS), il est indispensable de l'activer dans cette fenêtre. Par défaut, les règles de QOS ne sont pas actives.

6. L'inclusion statique

Il est possible d'insérer des règles iptables personnalisées. Ces règles vont venir s'insérer à la fin du fichier généré.

On accède à la fenêtre des inclusions statiques par le menu `Bibliothèque` / `Inclusion Statique`.

Il s'agit d'une zone de saisie de texte.



Fenêtre d'insertion des inclusions statiques

Aucune validation n'est faite par ERA sur ces règles insérées directement par l'utilisateur. Précisons que cette possibilité est réservée à un utilisateur avancé, qui maîtrise parfaitement la syntaxe iptables.

7. Imbriquer des modèles :l'héritage

Il est possible d'imbriquer des modèles, c'est-à-dire de faire dépendre des modèles les uns des autres. Un modèle devient un modèle père, les autres modèles héritent de toutes ses caractéristiques (directives, bibliothèques, flux, zones, ...).

Pour imbriquer des modèles, créez d'abord un modèle de manière habituelle. Ce modèle deviendra le modèle père. Ensuite, créez un nouveau fichier dans l'éditeur, et choisissez dans le menu `Fichier` / `importer un modèle`.

Le modèle est chargé comme d'habitude mais les directives importées ne sont plus éditables (elles sont grisées).

Ne seront éditables que les directives que vous allez rajouter. En plus de l'existant, vous pouvez faire toute modification utile (ajout de zone, création de directives, etc...).

Vous ne pouvez plus changer le fichier père de place ni le renommer, le chemin du fichier père est enregistré comme attribut.

L'héritage multiple entre modèles

L'héritage d'un modèle XML est donc la possibilité de d'utiliser plusieurs fichiers XML liés entre eux par référence. Le fichier référencé dans un autre fichier est appelé le fichier père. On peut voir si on édite le fichier XML avec un éditeur de texte, que le chemin du fichier XML père est renseigné dans l'attribut **model** à la racine de la balise **firewall**.

> Il est possible, mais ce n'est pas une action qui est accessible depuis l'interface gtk, d'hériter de plusieurs fichiers. Il suffit dans ce cas de mentionner dans l'attribut **model** une liste de noms longs de fichiers, séparés par des virgules. Pour des exemples de ces fonctionnalités, regarder dans le dossier **template** dans les sources du projet ERA, car les modèles XML eux-mêmes sont générés à partir de templates qui sont imbriqués entre eux avec cette fonctionnalités de l'héritage multiple.

8. Communication avec Zéphir

La connexion au Zéphir est possible depuis le menu `Zephir`.

Connection à Zéphir

Importer un modèle

ERA intègre la possibilité d'échanger des modèles avec un serveur Zéphir.

Lors du première utilisation des fonctions d'importation Zéphir, des informations de connexion vous seront demandées.

Vous devez spécifier ici l'adresse du serveur Zéphir, et le nom et le mot de passe d'un utilisateur ayant les droits nécessaires (lecture pour l'import et écriture pour l'export). Une fois connecté, vous pourrez saisir l'identifiant du serveur.

Le modèle est alors téléchargé et ouvert dans ERA. Cette procédure n'est valable que si vous avez déjà remonté le modèle de pare-feu dans Zéphir.

Importation d'un modèle XML depuis le Zéphir

A l'enregistrement, il vous sera demandé si vous voulez remonter le modèle sur Zéphir.

Exporter un modèle Zéphir

Lorsque vous avez construit un modèle de pare-feu, vous pouvez l'envoyer directement sur un serveur Zéphir avec le menu `Envoi à zephir`. Si vous ne les avez pas encore renseignées, ERA vous demandera les informations nécessaires à la connexion.

Les options suivantes vous sont proposées pour la sauvegarde sur Zéphir :

- pour un serveur : sauvegarde le modèle sur le serveur et change le modèle actif dans la configuration du serveur ;
- pour une variante : le fichier est ajouté à la liste des fichiers de la variante ;
- pour un groupe : idem que pour un seul serveur, mais sur tous les Amon présents dans le groupe choisi.

Exportation vers Zéphir

Chapitre 3
Directives optionnelles ERA depuis l'EAD

Les modèles de pare-feu ERA peuvent contenir des directives optionnelles[p.33].

Une règle peut être :

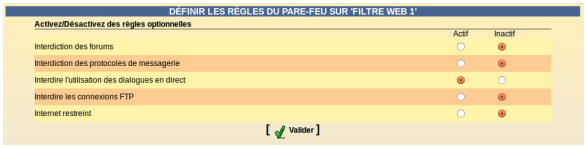
- générale, si elle concerne l'interface externe ;
- spécifique à une zone de configuration, si elle concerne une interface interne de la zone.

La configuration générale est accessible par le menu EAD : `Configuration générale` / `Règles du pare-feu`.

La configuration spécifique est accessible par le menu EAD : `Filtre web X` / `Règles du pare-feu` :

Pour valider une directive optionnelle :

- choisir Actif ;
- valider.



Activation des directives optionnelles dans l'EAD

Lien entre ERA et les directives optionnelles de l'EAD

Pour les règles optionnelles, l'EAD prime sur l'ERA : elles sont pilotées par l'EAD. Une directive peut être marquée comme étant active par défaut dans ERA et ne pas être active car désactivée dans l'interface EAD.

Voir aussi...

Les directives optionnelles [p.20]

Chapitre 4

Compléments techniques

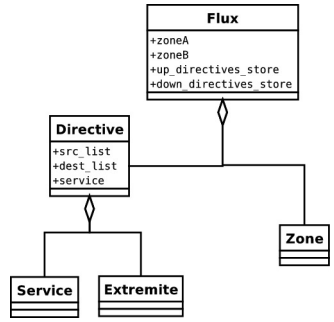
1. Le format XML interne

Les composantes du tableau des flux sont :

- les flux ;
- les zones ;
- les directives montantes et descendantes.

Le format XML interne suit une DTD qui correspond à la modélisation par flux. Les noms des balises correspondent aux noms des objets ERA. Il y a la liste des zones, puis les extrémités et les services, les groupes de services, et enfin les flux contenant les directives.

La représentation interne en objets est la suivante :



La représentation interne (objets)

- Directive(FwObject) : directive ;
- Service(FwObject), ServiceGroupe(FwObject) : service et liste de services ;
- Zone(FwObject), Extremite(FwObject) ;
- Flux(FwObject).

Les directives optionnelles

Dans le fichier `era.noyau.constants.py` il y a deux constantes intéressantes ici

- `DIRECTIVE_OPTIONAL = 1`
- `DIRECTIVE_ACTIVE = 2`

Ces filtres permettent de savoir si une directive est optionnelle ou non. Pour cela, il faut regarder l'attribut `attrs` de la directive.

Si `directive.attrs = 0`, alors la directive n'est ni optionnelle, ni active.

- `attrs=0` : pas optionnelle
- `attrs=1` : optionnelle mais pas active

- `attrs=3` : optionnelle et active
- la valeur 2 correspond à non optionnelle mais active, ce qui n'a pas de sens. Les valeurs autorisées sont donc `[0,1,3]`
- `ACTION_DENY` = 1 : barrage
- `ACTION_ALLOW` = 2 : pont
- `ACTION_FORWARD` = 4 : redirect
- `ACTION_DNAT` = 8 : dnat
- `ACTION_MASK` = 16 : masque

Exemple d'une directive de type masque :

```
action="16" attrs="0" nat_extr="exterieur_bastion" nat_port="0"
```

Exemple d'une directive de type dnat :

```
action="8" attrs="0" nat_extr="serveur_web" nat_port="80"
```

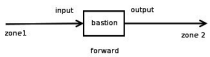
2. Comportement du Backend

Règles implicites : le REDIRECT

Un redirect doit inclure aussi une chaine input chaine xxx-bas. A une règle de forward vient donc se greffer une règle de type input.

règles implicites : le redirect

Il y a une règle de forward (une redirection) :



La règle de forward

la chaine input qui vient se greffer sur le redirect (sur le forward) est implicite.un forward z1->z2 doublé d'une redirection, ajoute une règle de type input vers le bastion.

Une directive de redirection génère donc deux règles :

- une règle input vers le bastion
- une règle forward z1->z2

La règle dite "implicite" est la règle de type INPUT. Une règle implicite se place en fin de pile pour chaque flux (elle n'est pas placée directement à côté de sa règle de FORWARD dans le fichier de règles générées).

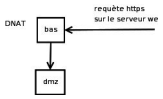
Règles implicites : Le DNAT et le SNAT

Lors d'un DNAT, une règle de type input est doublée d'un forward (elle s'ajoute à un FORWARD).

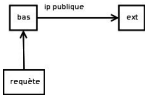
Même chose pour le masque de SNAT.

Exemple : un serveur de la DMZ répond à une requête sur le port 80 du bastion.

Un INPUT est transformé en FORWARD.



requête de DNAT



Une règle de type SNAT

Un poste de travail peut surfer sur le web avec l'IP publique du bastion. Cela permet de surfer masqué.

3. Intégration avec Creole

Creole propose un concept de variables multivaluées qui peuvent être utilisées dans ERA. ERA utilise bien-sûr les variables de dictionnaire Creole "simples", mais la fonctionnalité d'utilisation des variables de dictionnaires dans ERA peut-être étendue aux fonctionnalité Creole.

Si une variable `%%variable` est multi-valuée (au sens de Creole, c'est-à-dire que ça peut-être une liste), et que cette variable est présente dans une règle iptables, alors la règle iptables sera répétée autant de fois que de valeurs dans `%%variable` cette fonctionnalité génère du code avec une boucle for :

`%for %%v in %%variable` `/sbin/iptable bla bla %%v bla bla` `%end for`

4. Le compilateur

La génération des règles iptables

A la compilation du fichier XML, un certain nombre d'actions sont effectuées. Ce sont des règles iptables :

- définition d'une sous-chaîne pour chaque flux (liaison entre zone/extrémité) ;
- création de la politique par défaut (en fonction du niveau des zones) ;
- ajout des règles correspondant aux directives ;
- ajout de règles implicites liées au directives ;
- insertion des inclusions statiques (règle iptables de bas niveau).

Sur Amon, le compilateur gère aussi l'affichage des règles optionnelles dans l'EAD et récupère leur configuration en cas de mise à jour, et contrôle l'activation des directives cachées.

Le script iptables peut-être généré depuis l'interface ERA ou bien depuis un utilitaire ligne de commande plus complet.

Le bouton `générer`, bouton de génération des règles iptables n'est utile que si l'on n'est pas sur un Amon.

Il est donc possible depuis l'interface de transcrire directement en règles iptables ce qui est enregistré dans le fichier XML.

Bouton de génération de la sortie au format iptables

C'est aussi au moment de la compilation que sont gérées les directives cachées. Elles sont activées ou désactivées selon ce qui a été spécifié.

Utilisation en ligne de commande

Aller dans le répertoire era `/usr/share/era` et lancer le compilateur avec le fichier de modèles adapté

`[era]` `$ ./backend/compiler --help`

`compiler [options] era_model_file.xml`

par exemple :

`[era]` `$ ./backend/compiler modeles/3zones.xml`

différentes options sont possibles, taper `--help` pour les détails ou regarder le fichier `/usr/share/era/bastion.sh` qui correspond à ce qui est lancé par le service **bastion**

`service bastion restart` est lancé

Chapitre 5

Quelques références

- Site officiel du logiciel (présentation, téléchargement) : http://eole.orion.education.fr
- Code source du logiciel (versions, branches, tags) : https://dev-eole.ac-dijon.fr/projects/era/repository

Glossaire

Creole *= Création EOLE*	Creole gère la personnalisation des options de configuration des modules, le redémarrage des services, l'installation de paquets additionnels, la mise à jour du système. Il a été conçu pour être facilement personnalisable pour l'utilisateur final. Un ensemble d'outils est proposé pour modifier ou étendre les fonctionnalités offerte par EOLE.
Directive optionnelle	Directive paramétrée dans ERA et qui peut être activée ou désactivée depuis une autre interface. Les directives optionnelles le sont depuis l'EAD et les directives optionnelles cachées le sont par l'intermédiaire du template Creole `active_tags` des modules Amon et AmonEcole.
DTD *= Document Type Definition*	La Définition de Type de Document, est un document permettant de décrire un modèle de document SGML ou XML. Le modèle est décrit comme une grammaire de classe de documents : grammaire parce qu'il décrit la position des termes les uns par rapport aux autres, classe parce qu'il forme une généralisation d'un domaine particulier, et document parce qu'on peut former avec un texte complet. Une DTD décrit les documents à deux niveaux : • la structure logique, que l'on peut assimiler à la syntaxe abstraite ; • la structure physique, que l'on peut assimiler à la syntaxe concrète. Source : http://fr.wikipedia.org/wiki/Document_Type_Definition

EAD *= EOLE ADmin*	L'EAD est l'interface d'administration des modules EOLE. Il s'agit d'une interface web, accessible uniquement en HTTPS avec un navigateur web à l'adresse `https://<adresse module>:4200`. L'authentification peut être locale et/ou au travers d'EoleSSO (authentification unique). L'EAD est composé de deux parties : • un serveur de commandes (service ead-server), présent et actif sur tous les modules ; • une interface web (service ead-web), présent et actif sur tous les modules. Chaque module dispose d'une interface utilisateur EAD. Certains modules (Zéphir, Sphynx, ...) ne disposent que de la version de base qui permet d'effectuer les tâches de maintenance (mise à jour du serveur, diagnostic, arrêt du serveur, ...). Une version plus complète existe pour les autres modules (Horus, Scribe, Amon, ...) incluant des fonctionnalités supplémentaires.
ERA *= Éditeur de Règles pour le module Amon*	ERA est une application graphique de génération et de gestion de règles de sécurité adaptée au module pare-feu Amon. À partir du fichier XML de description du pare-feu, un script de règles iptables pour Netfilter est généré de manière à implémenter ces règles sur le module pare-feu Amon. La génération directe de règles iptables est également possible, permettant d'utiliser ERA pour d'autres types de serveurs sous GNU/Linux.
Extrémité	Une extrémité est un sous ensemble d'une zone. Elle est définie par une ou plusieurs adresses IP ou bien un sous-réseau. Elle hérite du niveau de sécurité de la zone à laquelle elle appartient.
Flux	Lien entre deux zones.
Flux descendant	Interactions d'un niveau de sécurité plus fort vers un niveau de sécurité plus faible avec une politique par défaut "autorisé".
Flux montant	Interactions d'un niveau de sécurité plus faible vers un niveau de sécurité plus fort avec une politique par défaut "interdit".
iptables	iptables est un logiciel libre grâce auquel l'administrateur système peut configurer les chaînes et règles dans le pare-feu dans l'espace noyau composé par des modules Netfilter. Netfilter est un framework implémentant un pare-feu au sein du noyau Linux à partir de la version 2.4 de ce dernier. Il prévoit des accroches (hooks) dans le noyau pour l'interception et la manipulation des paquets réseau lors des appels des routines de réception ou d'émission des paquets des interfaces réseau.

Modèle	ERA enregistre la description d'un pare-feu dans un fichier XML situé par défaut dans un répertoire nommé `/usr/share/era/modeles/`. Ce fichier est souvent dérivé d'un modèle livré de base, fichiers de référence présent dans le dossier `/usr/share/era/modeles` sur lequel se base l'utilisateur. Par extension, un modèle est n'importe quel fichier de description de pare-feu dans ERA.
Netfilter	Netfilter est un outil de filtrage de paquets sous linux. Le logiciel qui lui est associé est iptables.
Niveau de sécurité	Nombre entier (entre 0 et 100) permettant d'ordonner les zones par ordre croissant.
Pare-feu *= firewall*	Un pare-feu est un logiciel et/ou un matériel, permettant de faire respecter la politique de sécurité du réseau, celle-ci définissant quels sont les types de communication autorisés sur ce réseau informatique. Il a pour principale tâche de contrôler le trafic entre différentes zones de confiance, en filtrant les flux de données qui y transitent. Généralement, les zones de confiance incluent Internet (une zone dont la confiance est nulle) et au moins un réseau interne (une zone dont la confiance est plus importante). Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège. Le filtrage se fait selon divers critères. Les plus courants sont : • l'origine ou la destination des paquets (adresse IP, ports TCP ou UDP, interface réseau, etc.) ; • les options contenues dans les données (fragmentation, validité, etc.) ; • les données elles-mêmes (taille, correspondance à un motif, etc.) ; • les utilisateurs pour les plus récents. Source Wikipédia : http://fr.wikipedia.org/wiki/Pare-feu_(informatique)
Qualité de service *= QOS*	Régulation des flux du trafic sur un réseau, définition de Wikpedia [http://fr.wikipedia.org/wiki/QoS]
Service	Couple protocole et/ou port (ou plage de ports).
Tableaux de flux	Ensemble de lien entre les zones permettant de définir une politique par défaut et de classer un ensemble de règles (directives).
Template *= Modèle Creole*	Un template est un fichier contenant des variables Creole, qui sera instancié pour générer un fichier cible (typiquement un fichier de configuration serveur).

Zone	Découpage d'un réseau en restant centré sur le pare-feu, le pare-feu lui-même étant une zone nommée par convention **bastion**, c'est la zone la plus sécurisée (niveau 100). Chaque zone est définie par un nom, une adresse réseau, et un niveau de sécurité.