Laboratoire de Cryptologie et Virologie Opérationnelles
38, rue des Docteurs Calmette et Guérin
Parc Universitaire
53000 Laval
02 43 59 24 24

Rapport de stage technique du cycle ingénieur ESIEA

Réarchitecture du pare-feu d'ALCASAR pour un comportement dynamique

Auteur : Jean-Baptiste Couprit 4A

Maître de stage : Franck Bouijoux
Tuteur ESIEA : Richard Rey

Période du 07 avril au 07 août 2014

Soutenu le : 18 juillet 2014

Résumé

Ce rapport traite de la refonte du pare-feu d'ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau). ALCASAR est un contrôleur d'accès au réseau permettant de protéger les connexions de ses utilisateurs, que ce soit à partir d'un ordinateur, d'un téléphone portable, ou encore d'une tablette. Par conséquent, cet équipement doit être placé entre un routeur d'accès Internet (box) et ses utilisateurs. Ce projet libre sous licence GPLv3 est basé sous la distribution française Mageia et intègre plusieurs logiciels libres (Apache, Netfilter, OpenSSL, Dnsmasq, etc). Ce logiciel permet de tracer, imputer et authentifier ses utilisateurs, ce qui lui permet de répondre aux exigences françaises légales et réglementaires.

Mon travail au sein de l'équipe a été de changer la philosophie du pare-feu d'ALCASAR. Un pare-feu est un logiciel dont le but est de faire respecter des politiques de sécurité réseau, à travers un ensemble de règles. De cette manière, les flux de données peuvent être contrôlés. Le pare-feu de l'actuelle version d'ALCASAR (V2.8) définit des règles universelles et figées, ce qui implique que tous les utilisateurs héritent des mêmes droits, préalablement fixés par l'administrateur du système. L'objectif de la refonte du pare-feu était de mettre en place des règles dynamiques, personnalisées à chacun. Grâce à cette nouveauté, chaque utilisateur aura des droits qui lui seront propres, et qui seront gérés indépendamment dans le pare-feu. De cette manière, il sera désormais possible de faire cohabiter sur un même système, une structure d'entreprise où le directeur ne veut pas être filtré, tout en ayant un groupe d'employés avec un premier niveau de filtrage et un second groupe avec un second niveau de filtrage, ainsi qu'une personne avec le plus grand niveau de restrictions. À chaque connexion d'un utilisateur à ALCASAR, le pare-feu s'adaptera automatiquement à celui-ci afin de lui attribuer les règles qui lui sont propres.

Ce document récapitule l'ensemble de mes travaux effectués au cours de quatre mois de stage au laboratoire CVO, de la prise en main d'ALCASAR, à la refonte du pare-feu, en passant par les difficultés rencontrées, et surtout, la manière dont elles ont été surmontées.

Abstract

This report deals with the ALCASAR's (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau) firewall recasting. ALCASAR is a network access controller which is able to protect its users' connections, from a computer, a smartphone, or a tablet. Therefore, this device must be placed between a router and its users. This open-source project under GPLv3 licence is based on the french distribution Mageia and includes severals free softwares like Apache, Netfilter, OpenSSL, Dnsmasq, etc. Its features are to impute, trace and authenticate its users, allowing it to respect legal requirements and regulations.

My work in the ALCASAR team, was to change the philosophy of the firewall. A firewall is a software which aims to enforce network security politics, through a set of rules. By this way, data streams can be handled. The firewall of the current ALCASAR version (V2.8), defines fixed and universal rules, which involves that all users inherit from same rights, beforehand, fixed by the system administrator. For the next version (V2.9) which will be released in a couple of months, the purpose was to implement a dynamic firewall, that is to say, with the adding or not of rules according users. With this innovation, each user will have own rules, which will be handled independently in the firewall. By this way, it will be possible to have, on a same system, without changing any rights, for example, a company structure in which the director doesn't want to be filtered, while having a group of employees with a low level filtering, an other group of employees with a medium level of filtering and a single person with an high filtering level. For each connection to ALCASAR, the firewall will automatically fit, in order to attribute rules to users.

This document summarizes all my work during my four month internship, at the CVO laboratory, from the beginning, to the recasting of the firewall, through the difficulties, and most importantly, how they were overcome.

Table des matières

Table des illustrations

Remerciements

Je remercie toutes les personnes ayant participé de près ou de loin à la réussite de ce stage, l'ensemble du personnel du laboratoire CVO qui a su m'accueillir et m'intégrer à l'équipe et tout particulièrement les personnes suivantes :

Monsieur Richard Rey : pour m'avoir fait l'honneur d'être mon tuteur, pour m'avoir guidé et épaulé durant cette expérience pleine d'intérêts.

Monsieur Franck Bouijoux : qui m'a accompagné tout au long de ce stage, bénéficier de son expérience a été très enrichissant et instructif. Je le remercie pour tout le temps qu'il m'a consacré ainsi que pour les responsabilités qu'il m'a confiées, ce qui m'a permis de beaucoup apprendre.

Introduction

Établissement d'accueil

En tant qu'élève ingénieur en quatrième année à l'ESIEA, j'ai pour but de mettre en pratique l'ensemble de mes connaissances à travers un stage technique en entreprise. L'objectif d'un tel stage est non seulement d'acquérir de l'expérience sur le monde du travail qui m'attend dans quelques mois, mais aussi d'approfondir mes compétences à travers un projet extrascolaire.

L'entreprise m'ayant accueillie est le Laboratoire de Cryptologie et Virologie Opérationnelles (CVO), dirigé par monsieur Éric Filiol, appartenant à l'École Supérieure d'Informatique, Électronique et Automatique, située à Laval (53). Ce laboratoire et l'un des quatre que compte l'ESIEA. Parmi ceux-ci nous trouvons :

- Un laboratoire en Réalité Virtuelle et Systèmes Embarqués (RVSE)
- Un laboratoire d'Art et de Recherche NUMérique (ARNUM)
- Un laboratoire d'Acquisition et Traitement des Images et du Signal (ATIS)

Le laboratoire CVO est implanté à l'ESIEA de Laval depuis 2007. Au départ, il s'agissait d'un laboratoire travaillant en collaboration avec le Laboratoire de Virologie et de Cryptologie de l'École Supérieure et d'Application des Transmissions (ESAT) de Rennes, jusqu'à accueillir définitivement les ressources de l'ESAT en 2008. Sous tutelle du Ministère de la Défense, ce laboratoire est habilité à mener des recherches dites classifiées de défense et est rattaché à l'École doctorale de l'École Polytechnique de Palaiseau. À l'heure actuelle, cette structure comprend quatre enseignants-chercheurs à temps complet, deux ingénieurs de recherche ainsi que six doctorants. Les thèmes des recherches abordés sont les suivants :

- Cryptologie symétrique, cryptologie asymétrique, cryptanalyse et cryptographie
- Systèmes stéganographiques
- Virologie informatique
- Guerre informatique
- Sécurité des environnements embarqués
- Algorithmie et implémentation sécurisée

Présentation d'ALCASAR

ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau) est un contrôleur d'accès au réseau libre, sous licence GPLv3, exploitant Mageia. Il a été créé par *Rexy* et *3abtux* et est toujours en développement. Sa version actuelle est la V2.8. Il s'agit d'un portail permettant de sécuriser l'accès à un réseau, en authentifiant, traçant et imputant les connexions des utilisateurs, quelque soit les supports utilisés (ordinateurs, tablettes ou encore téléphones portables). L'authentification est réalisée à la suite de l'interception d'une station du réseau qui est effectuée par un identifiant et un mot de passe. L'administrateur du système crée les comptes usagers, ou les groupes d'usagers, avec la possibilité d'allouer

Illustration 1: Logo d'ALCASAR

à chacun un grand nombre d'attributs tels que leur période de connexion, la date d'expiration du compte, la taille maximale de transfert de données, et bien d'autres. Si une personne ne possède pas d'identifiants, elle sera bloquée.

L'imputabilité et la traçabilité sont réalisés grâce à un système de journalisation par fichiers, ce qui permet d'avoir une trace de chaque activité utilisateurs (téléchargements, consultation de courriers électroniques, navigation, etc.). De plus, ALCASAR garantit le respect de la vie privée de chaque utilisateur en chiffrant les flux liés à l'authentification et l'impossibilité de modifier les fichiers de traces des connexions. De ce fait, il y a non-répudiation des traces. Tout ceci permet par conséquent de répondre aux exigences des politiques d'accès.

La sécurité, quant à elle, est réalisée grâce un pare-feu couplé à un antivirus afin de protéger chaque utilisateur des menaces extérieures d'une part, ainsi que de modules spécifiques afin de protéger chaque utilisateur de tentatives de piratage interne d'autre part.

ALCASAR fonctionne sur un ordinateur dédié placé entre le routeur d'accès à Internet (box) et un réseau de consultation. Il est totalement indépendant des technologies utilisées.

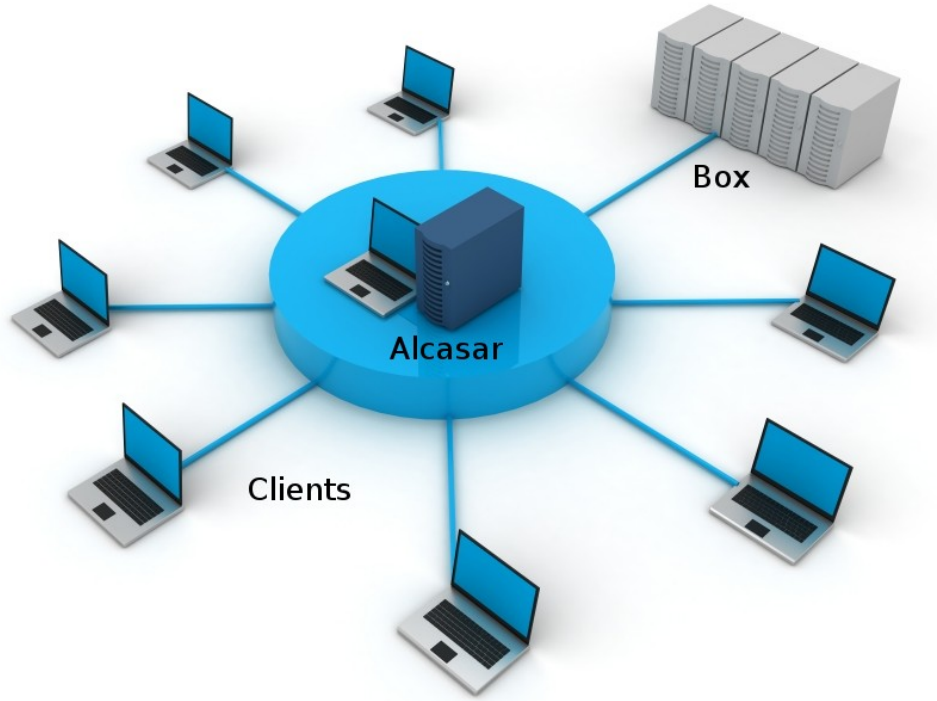


Illustration 2: Exemple de réseau avec ALCASAR

ALCASAR est destiné aux grandes structures telles que les hôtels, les écoles, les aéroports, etc. Il est aussi utilisé comme outils d'apprentissage pour les filières de la sécurité réseau.

Illustration 3: Interception client

Du point de vue du client, ALCASAR intercepte les utilisateurs lors de la demande d'une page web suite au lancement d'un navigateur, et demande des identifiants pour pouvoir bénéficier d'un accès à Internet.

Nous pouvons trouver sur cette page les principales informations de l'entreprise et du portail. L'usager, s'il le souhaite, a la possibilité de changer son mot de passe. Tant qu'une personne n'est pas authentifiée, l'accès à Internet est totalement bloqué. Une fois authentifié, un utilisateur se retrouve dirigé sur la page qu'il demandait lors de son interception. Dès lors, une session sécurisée lui est attribuée, et sa vie privée respectée. En fonction de son niveau de filtrage,

déterminé par un administrateur, un usager aura ou n'aura pas le droit de naviguer sur certains sites.

Du point de vue de l'administrateur, ALCASAR offre un centre de gestion complet permettant de visualiser l'état du réseau et des services lancés, de gérer les utilisateurs en leur allouant par exemple une durée de connexion maximale, une limitation de la bande passante, une date d'expiration de compte, etc. Un système de liste noire / liste blanche est intégré afin de filtrer la navigation des usagers en sélectionnant des catégories de sites par exemple le shopping, chat, audio-vidéo, etc. Un système de visualisation de statistiques d'exploitation du réseau est aussi intégré, permettant de voir en temps réel le taux de données transféré. Cette interface offre également la possibilité de récupérer les archives des fichiers journaux, pouvant être utilisées dans la cadre d'enquêtes judiciaires. Cette interface est accessible à distance, au moyen d'un tunnel sécurisé et chiffré (SSH).

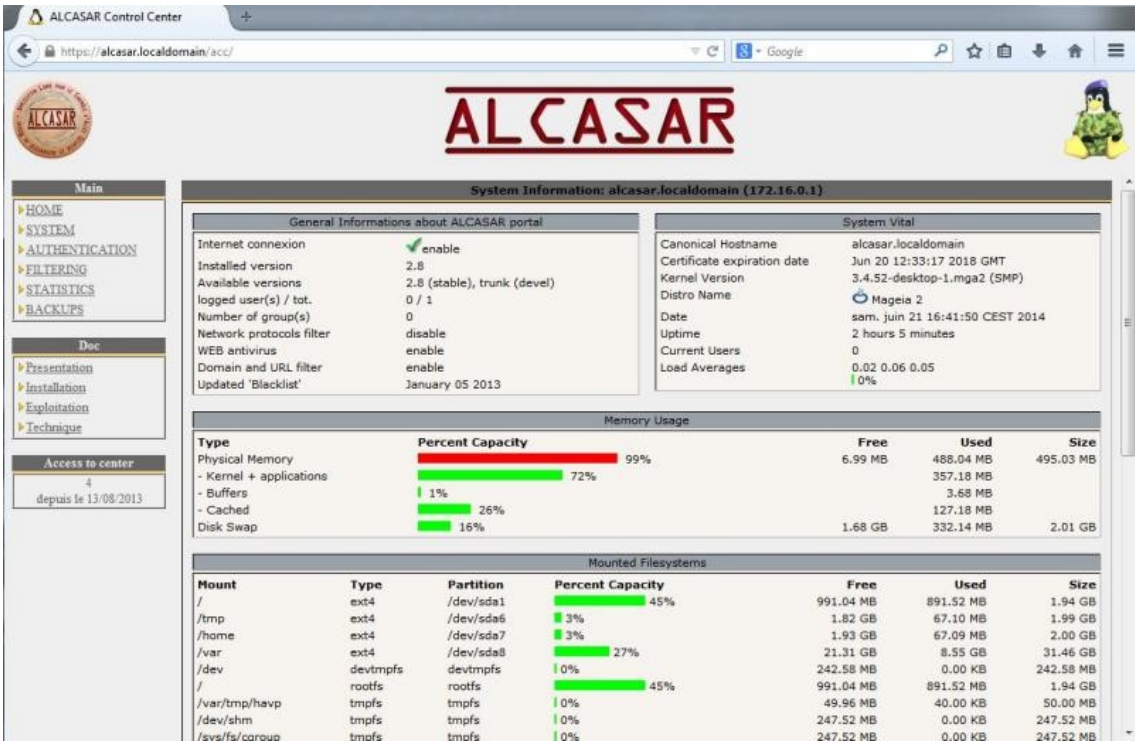


Illustration 4: Panel d'administration d'ALCASAR

Architecture de ce document

Ce document décrit tout d'abord la problématique liée à ma période de stage, en détaillant l'utilité d'un pare-feu dynamique dans le projet ALCASAR. Il se poursuit par l'amélioration de certains points de l'interface d'administration, comme l'amélioration de la gestion de la liste noire, ou la simplification du menu d'administration. Dans la partie suivante, nous détaillerons l'amélioration du filtrage par liste noire grâce à l'ajout du filtrage par adresse IP. Cette précédente étape nous permet d'arriver au coeur de mon stage : la modification du pare-feu. Cette partie récapitule les étapes de mes études et recherches sur le pare-feu en vue de l'amélioration de son comportement. Enfin, j'évoquerai la bascule du support de Mageia 2 vers Mageia 4.

Problématique

Sujet

Afin de débuter mon stage dans les meilleures conditions possible, j'ai étudié l'installation et le fonctionnement d'ALCASAR avant le début de celui-ci afin de me familiariser avec ses principales fonctionnalités et son architecture. ALCASAR étant beaucoup plus qu'un simple contrôleur d'accès, il intègre un grand nombre de fonctionnalités (authentification, filtrage, traçage, etc.), et l'étudier dans sa globalité demande beaucoup de temps. À la suite de cette étude, ayant eu la chance de choisir mon sujet de stage, je me suis dirigé vers la partie de filtrage des usagers réalisée par le pare-feu Netfilter. Le but étant de rendre un pare-feu non plus statique, mais dynamique, c'est à dire s'adaptant à chaque utilisateur, d'où le sujet suivant : ***Réorganisation des flux du pare-feu d'ALCASAR.***

Un pare-feu, pouvant être matériel ou logiciel, permet de faire respecter des politiques de sécurité réseau, à travers un ensemble de règles. De cette manière, les flux de données entre Internet et un réseau de consultation peuvent être contrôlés.

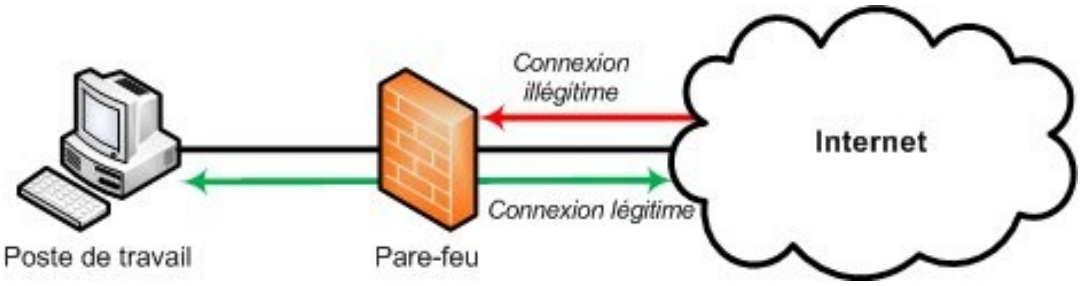


Illustration 5: Pare-feu

La mise en place d'un pare-feu joue un rôle crucial pour la sécurité d'un réseau local. En effet, un pare-feu permet d'interdire les connexions définies comme illégitimes en provenance d'Internet, tout en laissant circuler librement les autres connexions. Pour cela, un pare-feu analyse les adresses IP, les protocoles, les ports source et de destination, et, suivant la politique qui lui aura été dictée, autorisera, ignorera ou rejettera les paquets qui transitent. Un pare-feu n'offre cependant pas une sécurité absolue. En effet, plusieurs autres éléments sont nécessaire pour couvrir l'ensemble du domaine (antivirus, serveur mandataire, détecteurs d'intrusion, bon comportement des utilisateurs, etc.). ALCASAR étant un projet libre et gratuit, il se devait d'intégrer un pare-feu libre, et surtout réputé. C'est la raison pour laquelle Netfilter a été choisi.

Pourquoi un pare-feu dynamique ?

Le pare-feu de l'actuelle version d'ALCASAR (V2.8) étant statique, cela signifie que ses règles s'appliquent pour l'ensemble des utilisateurs du réseau de consultation. La future version d'ALCASAR V2.9 ayant pour but d'attribuer des niveaux de filtrage différents par utilisateur, cette approche a donc dû être repensée afin d'attribuer des règles différentes en fonction des usagers. Cela signifie qu'un utilisateur ne sera plus dirigé vers un unique chemin, mais sera aiguillé lors de sa connexion en fonction de son niveau de filtrage.

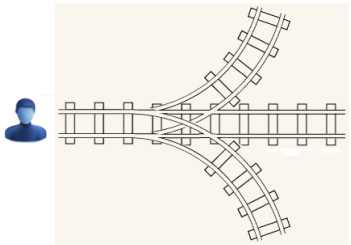


Illustration 6: Aiguillage d'un usager

1 Ajout de fonctionnalités

Les premières tâches qui m'ont été confiées avaient pour but de me faire découvrir le fonctionnement d'ALCASAR pour pouvoir descendre plus en profondeur par la suite : la partie pare-feu, se situant au coeur du système. J'ai donc eu pour mission d'améliorer certains points de l'interface d'administration, et plus particulièrement le filtrage des sites web (liste noire). ALCASAR dispose d'un système de filtrage de sites web qui s'appui sur la liste noire de Toulouse (http://dsi.ut-capitole.fr/blacklists/), nous offrant différentes catégories à interdire allant des sites de jeux aux sites pédophiles. Dans la version 2.8 d'ALCASAR, la description des catégories était écrite à la main par un membre de l'équipe d'ALCASAR. Cela peut poser problème, car cette liste est en constante évolution et par conséquent, des catégories peuvent être ajoutées ou enlevées, ce qui génère des erreurs. Mon rôle a donc été d'automatiser cette tâche. La liste noire intégre une description des catégories, j'ai donc basculé le système afin qu'il s'appuie pour chaque catégorie, sur la description fournie. Ainsi, lors du téléchargement, la liste est traitée afin d'être à différents endroits et c'est à ce moment que le script récupère les descriptions et les affecte aux catégories. Ensuite, lorsque l'on clique sur une catégorie dans le panel d'administration, la page PHP va chercher l'information et l'affiche.

Illustration 7: Description d'une catégorie

À ce moment précis, nous ne disposions que d'une description et l'administrateur ne pouvait pas savoir ce qui se trouvait réellement dans les catégories (étant composées de noms de domaine, d'URL et d'adresses IP). Dans la continuité de cette démarche, j'ai implémenté un code permettant d'afficher le nombre de noms de domaine, d'URL et d'adresses IP que contient chaque catégorie et c'est à ce moment que j'ai rencontré ma première difficulté. Certaines catégories contiennent plus d'un million de noms de domaine, d'URL ou d'adresses IP et calculer leur nombre prenait environ neuf secondes en PHP sur disque dur, ce temps d'attente étant beaucoup trop long pour l'administrateur, j'ai décidé de faire appel à un code plus rapide, le BASH, qui lui offre des fonctions en langage C optimisé. Après la mise en place de celui-ci, nous sommes passés de neuf secondes à une seconde pour les très grosses catégories, ce qui est plus acceptable. Enfin, dans le but de permettre à l'administrateur de visualiser ce que contiennent réellement les catégories, des liens sur les nombres de noms de domaine, d'URL et d'adresses IP ont été créés. Lors d'un clic sur ces liens, les premières lignes de chaque fichier sont affichées, ce qui donne un aperçu de ce que contient une catégorie. À ce point, si un utilisateur cliquait par exemple, sur le nombre d'adresses IP, puis sur le nombre d'URL, le calcul de ces nombres était refait à chaque fois, ce qui n'était pas une solution adaptée. J'ai donc choisi de faire les calculs une seule fois lors d'un clic sur les catégories et de passer les nombres obtenus en paramètres lors de l'appel de la page web (méthode GET). De cette façon, le temps d'attente n'intervient qu'une seule fois.

Cette première partie s'est achevée avec la mise en place du nombre total d'adresses IP, d'URL et de noms de domaines que compte la liste noire dans le panel d'administration, afin de donner un aperçu de ce qu'elle contient, globalement.

2 Fusion des pages ticket rapide et ticket usager

Ma mission s'est poursuivie par la simplification de l'interface d'administration en fusionnant la page de création d'un usager et celle de création d'un ticket rapide, car celles-ci convergent vers le même but. De cette façon, l'administration du système est simplifiée. La page de création d'un ticket rapide offre la possibilité de créer un usager avec des attributs généraux tels que l'e-mail, la date d'expiration du compte, le nombre de connexions simultanées, etc., contrairement à la page de création d'un usager qui elle, offre une plus large gamme d'attributs. Parmi ceux-ci, nous pouvons trouver le temps maximal d'une connexion, le nombre maximal de données transférées ou encore la limitation de la bande passante. Le but a été de créer une unique page (création d'un usager), pouvant s'étendre si l'administrateur le souhaite. La page de création d'un ticket rapide a été employée, sur laquelle j'ai rajouté un bouton permettant d'étendre le choix des attributs, ce qui permet de rendre l'utilisation de l'interface plus intuitive.

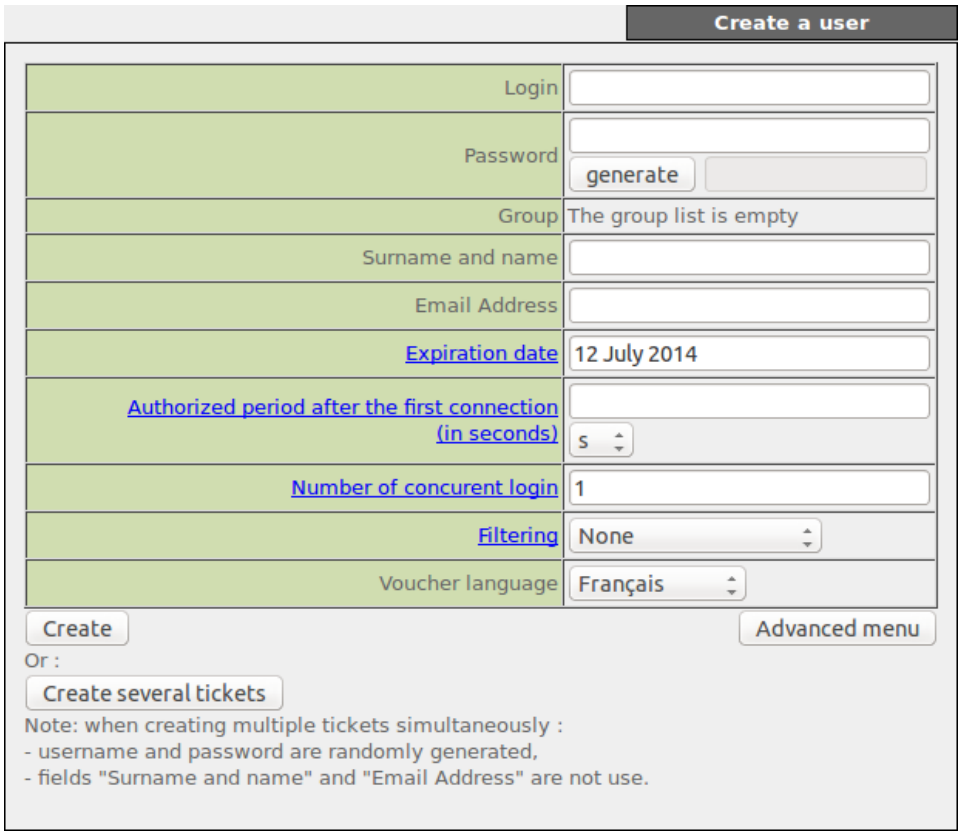


Illustration 8: Création d'un usager

C'est aussi grâce à cette page qu'ALCASAR offre la possibilité aux usagers de s'authentifier automatiquement lorsque leur adresse MAC est reconnue, en renseignant leur adresse MAC à la place du nom d'utilisateur et le mot “password” pour le mot de passe, lors de la création de leur compte. Dans la version 2.8 d'ALCASAR, l'administrateur, pour la création d'un tel compte, doit entrer une adresse MAC uniquement de la forme 08-00-27-F3-DF-68, ce qui peut poser problème si l'administrateur se trompe. C'est la raison pour laquelle j'ai mis en place un système d'expressions régulières afin que l'administrateur puisse entrer une adresse MAC du type 08-00-27-f3-df-68, 08:00:27:F3:DF:68 ou encore 08:00:27:f3:dF:68. Une fois l'adresse MAC reconnue, celle-ci est transformée en un format unique compatible avec les champs de la base de données. Ce mécanisme permet de limiter le risque d'erreurs de la part de l'administrateur.

Une fois un compte créé, un ticket est généré par ALCASAR, récapitulant les informations d'un compte, pour être donné à l'utilisateur.

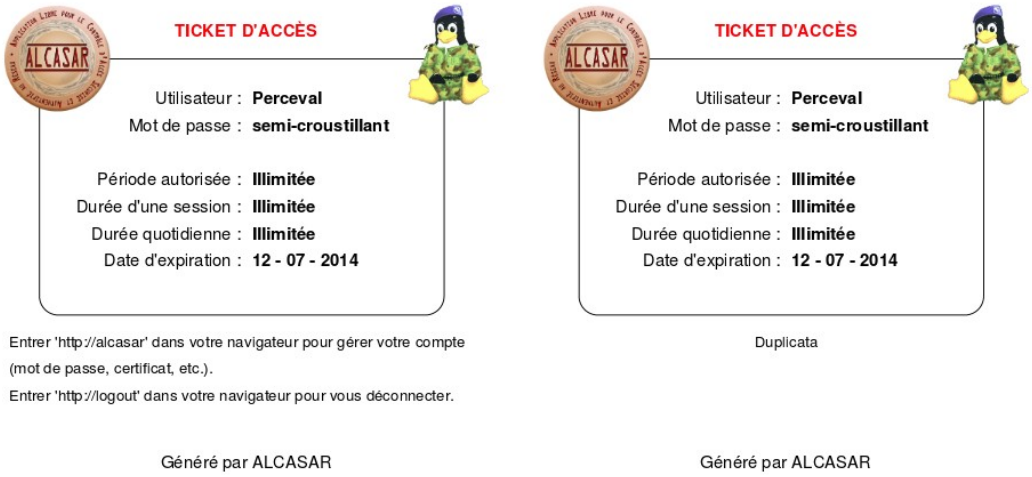


Illustration 9: Ticket généré par ALCASAR

Dans la version 2.8, lorsqu'un administrateur veut créer un compte avec une adresse MAC, un ticket est généré, ce qui peut s'avérer inutile, parce qu'un utilisateur n'a besoin d'aucune information (mot de passe, identifiant) pour se connecter, son adresse MAC étant reconnue, il se fait automatiquement connecter. J'ai donc supprimé l'appel à la génération automatique d'un ticket lors de la création d'un tel compte.

3 Améliorations de la partie « Filtrage »

Liste noire

Ma mission s'est ensuite poursuivie en descendant un peu plus en profondeur dans le système. Pour la partie filtrage des usagers, ALCASAR s'appuie sur la liste noire de Toulouse (http://dsi.ut-capitole.fr/blacklists/). Il s'agit d'une archive diffusée par l'Université de Toulouse 1, permettant un meilleur contrôle de l'utilisation d'Internet. En effet, cette archive contient d'une part une liste noire composée d'adresses IP, d'URL et de noms de domaine frauduleux, pornographiques, dangereux, etc., regroupés en catégories (Radio, Adult, Bank, etc), et d'autre part, une liste blanche regroupant des sites jugés fiables et inoffensifs. Cette archive est activement maintenue non seulement, par des dizaines de contributeurs, mais aussi par des robots explorant Internet.

Dans son menu d'administration, ALCASAR propose un filtrage des usagers basé sur la liste noire. Avec ce filtrage, un administrateur a la possibilité de mettre à jour la liste noire, et de sélectionner la ou les catégories qu'il souhaite interdire pour son réseau de consultation.

arjel ☐	astrology ☐	audio-video ☐	blog ☐	celebrity ☐	chat ☐	cooking ☐	filehosting ☐	financial ☐	forums ☐
games ☐	lingerie ☐	manga ☐	mobile-phone ☐	publicite ☐	radio ☐	reaffected ☐	shopping ☐	social_networks ☐	sports ☐
webmail ☐	adult ☑	agressif ☑	dangerous_material ☑	dating ☑	drogue ☑	gambling ☑	hacking ☑	malware ☑	marketingware ☑
mixed_adult ☑	phishing ☑	redirector ☑	remote-control ☑	sect ☑	strict_redirector ☑	strong_redirector ☑	tricheur ☑	warez ☑	

Illustration 10: Catégories de la liste noire

S'il le souhaite, l'administrateur peut réhabiliter ou filtrer des noms de domaine précis, ou encore des URL. Dans la version 2.8, le filtrage par adresse IP n'est pas présent. Mon travail a donc été d'intégrer au système un filtrage par adresse IP venant compléter celui des URL et des noms de domaine, jusqu'à maintenant non utilisé du fait d'un temps d'attente trop important au niveau de l'ajout de ces adresses dans le pare-feu, mais désormais surmontable au moyen de l'intégration de SET par le noyau Linux. Un SET peut être assimilé à une sorte de tableau dans lequel nous pouvons ajouter ce que l'on désire dans le but de le traiter au niveau du pare-feu, ce qui évite de créer des centaines de règles (coûteux en temps).

Ma première approche a consisté à faire une boucle alimentant un SET dans le pare-feu nommé “Blacklist_ip_blocked” avec les adresses IP des catégories sélectionnées par l'administrateur, puis d'ajouter des règles bloquant ces adresses IP. La taille du SET correspondant au nombre d'adresses IP des catégories sélectionnées. Les limites de cette approche sont vite apparues. En effet, cette méthode fonctionne parfaitement sur un petit nombre d'adresses IP, mais lorsqu'une catégorie d'un million d'adresses IP est ajoutée, le temps d'attente devient alors inacceptable. Pour résoudre ce problème, après des recherches plus approfondies, j'ai remarqué que les SET, gérés avec la commande BASH “ipset” pouvaient être sauvegardés dans des fichiers textes, afin de repeupler un SET dans le futur. J'ai donc décidé de tester cette méthode afin de comparer le temps d'attente avec le précédent. Les tests se sont avérés fructueux, puisque la remise en mémoire d'un SET de plus d'un million d'entrées, grâce un fichier de sauvegarde, prend quelques secondes contre quelques minutes pour la première méthode. Suite à cette étude, afin d'optimiser le code, j'ai donc pris l'initiative, avec l'accord de mon maître de stage, de créer un fichier de sauvegarde par catégorie, respectant le format attendu par la commande “ipset”, à la suite du téléchargement de la liste noire. Cette méthode me permettant donc de directement, et rapidement, peupler le SET “Blacklist_ip_blocked” avec les fichiers de sauvegarde de chaque catégorie sélectionnée. Par la suite, si un utilisateur connecté souhaite aller sur une des adresses IP appartenant à ce SET, il est bloqué par le pare-feu, dans le cas

Illustration 11: Blocage par le pare-feu

inverse, il navigue librement.

Une fois cette solution intégrée, il fallait offrir la possibilité à l'administrateur, de pouvoir réhabiliter ou filtrer les adresses IP qu'il souhaite, sans pour autant désactiver ou activer une catégorie complète. De plus, la réhabilitation et le filtrage des URL étant inappropriés (le plus souvent, un site complet est interdit d'accès ou autorisé), ils ont donc été supprimés afin de mettre en place le système d'adresse IP. L'administrateur peut alors, librement ajouté ou réhabiliter, des adresses IP. Au niveau du pare-feu, le filtrage personnel d'adresses IP se traduit par l'ajout de ces adresses dans le SET “Blacklist_ip_blocked”. Si des adresses doivent être réhabilitées, elles sont supprimées du SET. La taille du SET est égale au nombre d'adresses IP des fichiers de sauvegarde sélectionnés par l'administrateur, plus les adresses IP filtrées, moins les adresses IP réhabilitées.

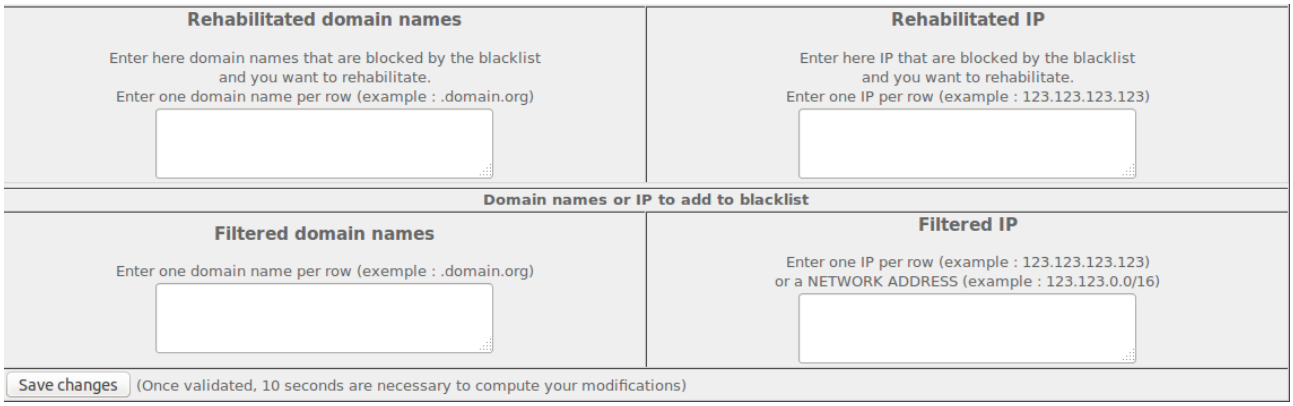


Illustration 12: Réhabilitation et filtrage d'adresses IP

De plus, afin de satisfaire à une demande des utilisateurs d'ALCASAR, j'ai mis au point un système d'ajout de fichiers d'adresses IP destinés à être filtrés. Si un administrateur veut ajouter sa liste noire d'adresses IP à bloquer, il pourra désormais le faire dans la version V2.9 d'ALCASAR.

Illustration 13: Fichiers d'adresses IP

Une fois envoyés à ALCASAR, ces fichiers d'adresses IP ne sont pas simplement ajoutés au SET “Blacklist_ip_blocked”. Ils sont d'abord « reconstruits » afin d'éviter les erreurs au niveau du pare-feu. C'est-à-dire que chaque fichier est parcouru dans son intégralité afin d'en extraire uniquement les adresses IP (via des expressions régulières). De cette façon, même si le fichier contient du texte comme des commentaires, il ne sera pas pris en compte. Ensuite, le fichier est modifié afin de ne mettre qu'une seule adresse IP par ligne et les doublons sont éliminés. Une fois cette procédure terminée, la syntaxe des fichiers de sauvegarde pour les SET est mise en place, afin que l'ajout des adresses IP dans le SET “Blacklist_ip_blocked” soit le plus rapide possible. Bien entendu, la taille du SET est recalculée en fonction de ces fichiers (la taille du SET est égale au nombre d'adresses IP des fichiers de sauvegarde sélectionnés par l'administrateur, plus les adresses IP filtrées, moins les adresses IP réhabilitées, plus le nombre d'adresses IP que contient chaque fichier ajouté par l'administrateur). Si le fichier d'adresses IP ajouté par l'administrateur contient des adresses IP déjà présentes dans la liste noire de Toulouse, celles-ci sont ignorées. Ce problème permet d'être résolu avec le paramètre « -! » à passer à la commande « ipset ».

Liste blanche

Le système de liste blanche n'étant pas présent dans la version V2.8 d'ALCASAR, j'ai eu pour tâche de la rendre opérationnelle pour la future version. Contrairement à une liste noire qui filtre certains sites, une liste blanche n'autorise que certains sites et bloque les autres. Concernant les noms de domaine, il a fallu mettre en place un nouveau serveur léger DNS : Dnsmasq, n'autorisant que les noms de domaine des catégories sélectionnées par l'administrateur, et bloquant les autres. Ce problème étant résolu, un deuxième a fait son apparition : une personne connaissant à l'avance une adresse IP peut joindre le site sans être filtrée. Dans le but de résoudre ce problème, j'ai dans un premier temps contacté monsieur Fabrice Prigent de la liste noire de Toulouse afin qu'il fournisse les adresses IP des noms de domaine dans l'archive de la liste noire. La réponse s'est avérée négative, car un site peut en cacher d'autres, et par conséquent leurs accès pourraient être bloqués par le premier. Dans un second temps, j'ai décidé de récupérer les adresses IP de tous les noms de domaine, avec des résolutions DNS, afin de faire le lien au niveau du pare-feu entre les noms de domaine et les adresses IP (si l'adresse IP d'un site n'est pas répertoriée, son accès est bloqué). Pour ce faire, lors du téléchargement de la liste noire, je faisais une résolution DNS par nom de domaine. En les lançant toutes en parallèle, un temps d'attente d'environ deux minutes était nécessaire. Cette seconde solution fonctionnait, jusqu'à ce que Google (utilisé comme serveur DNS) assimile ce procédé à une attaque DOS (déni de service) et par conséquent bloque l'accès à ses services. Enfin, en effectuant plus de recherches, une solution a émergé. Il se trouve que Dnsmasq (logiciel utilisé dans ALCASAR) possède une option permettant de peupler un SET du pare-feu, avec les adresses IP qu'il a résolues. Grâce à ce procédé, soit un site est libre d'accès et nous avons son adresse IP, soit il est interdit par Dnsmasq qui ne renseigne pas son adresse IP.

La solution a été la suivante : j'ai créé un SET “whitelist_ip_allowed” accueillant les adresses IP proposées par Dnsmasq. Ensuite, j'ai ajouté une règle dans le pare-feu permettant de bloquer l'accès à un site si son adresse IP ne figure pas dans le SET. De ce fait, soit un utilisateur demande un site autorisé, Dnsmasq le laisse passé et l'adresse IP est ajoutée au SET “whitelist_ip_allowed”, soit il entre directement une adresse IP dans la barre de recherche de son navigateur, si celle-ci est présente dans le SET, il accède au site, sinon il se trouve bloqué. Concernant la taille à allouer à ce SET, je me suis basé sur une moyenne globale des adresses IP des sites de la liste blanche : sur l'ensemble des sites, certains ont une seule adresse IP, d'autres en ont deux ou trois, et des sites comme Google peuvent avoir jusqu'à six adresses IP. Après cette étude j'ai donc choisi de prendre la moyenne (6+2+1 / 3 = 3), ce qui donne trois fois plus d'espace pour stocker les adresses IP, qu'il y a de noms de domaine, en sachant que les utilisateurs n'iront jamais sur tous les sites de la liste blanche. Dans le cas, où les usagers consulteraient tous les sites et que la taille du SET viendrait à ne plus suffire (ce qui semble impossible), la liste blanche fonctionnera encore, car si un SET est plein, Dnsmasq n'ajoute plus d'adresse IP, et donc permet le maintien du filtrage.

Enfin, la liste blanche intègre également un système d'adresse IP et de noms de domaine autorisés, au cas où l'administrateur voudrait autoriser certains sites bloqués.

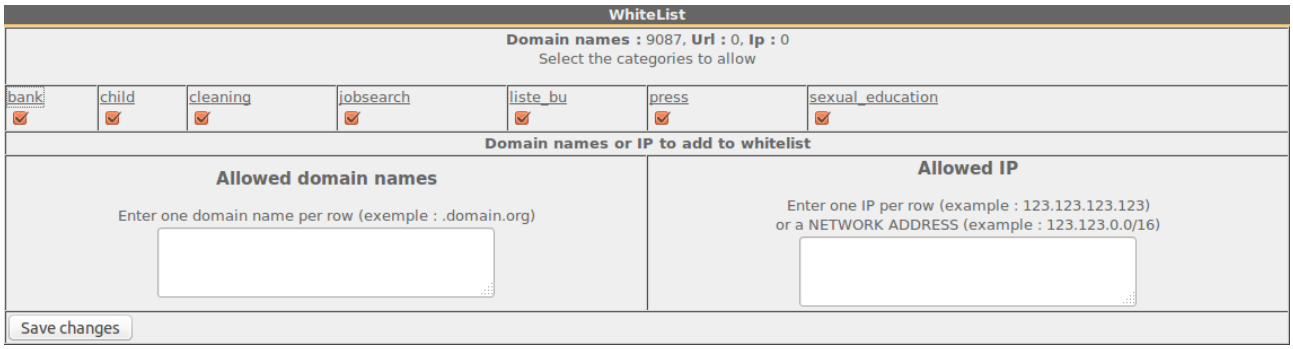


Illustration 14: Catégories de la liste blanche

4 Pare-feu dynamique

Mon travail s'est poursuivi en descendant au coeur du système, le but étant non plus d'avoir un pare-feu global, mais d'avoir un pare-feu dynamique, c'est à dire s'adaptant à chaque utilisateur en fonction de son niveau de filtrage. Pour cela, la première étape a été de définir les niveaux de filtrage des utilisateurs. Afin de satisfaire un maximum de clients d'ALCASAR et de répartir au mieux les filtres, quatre grands niveaux de filtrages ont été retenus :

1. pas de filtrage
2. filtrage par antivirus
3. filtrage par antivirus + liste noire
4. filtrage par antivirus + liste blanche

J'ai ensuite intégré ces différents niveaux de filtrage au système de création de comptes afin que l'administrateur ait la possibilité d'attribuer à chaque utilisateur, un niveau de filtrage.

Illustration 15: Filtrage usager

En base de données, ce paramètre est codé de la manière suivante :

1. **aucun paramètre** : pas de filtrage
2. **00000001** : l'utilisateur est filtré uniquement par l'antivirus
3. **00000011** : l'utilisateur est filtré par l'antivirus et la liste noire
4. **00000101** : l'utilisateur est filtré par l'antivirus et la liste blanche

Le premier bit correspond à l'antivirus, le second à la liste noire et le troisième à la liste blanche. De cette façon, ce paramètre pourra facilement s'adapter à d'autres niveaux de filtrage, si l'équipe d'ALCASAR décide d'en mettre d'autres un jour. L'ajout d'aucun paramètre en base de données si l'utilisateur n'a pas de filtrage est très important, car c'est grâce à cela que s'il fait partie d'un groupe, les attributs du groupe prendront le dessus sur ceux de l'utilisateur. En revanche, si un niveau de filtrage différent a été attribué à l'utilisateur et que celui-ci fait partie d'un groupe, ce seront les attributs utilisateur qui seront prioritaires.

Au niveau du pare-feu, ces niveaux de filtrages se sont traduits par quatre SET :

1. **no_filtering_set** pour les usagers sans filtrage
2. **havp_set** pour les usagers filtrés avec l'antivirus (HAVP+LibClamav)
3. **havp_bl_set** pour les usagers filtrés avec l'antivirus et par la liste noire
4. **havp_wl_set** pour les usagers filtrés avec l'antivirus et la liste blanche

Aucune taille n'a été renseignée pour ces SET car de base, un SET peut contenir 65536 entrées, ce qui est jugé comme plus que satisfaisant en vue du nombre potentiel d'utilisateurs d'ALCASAR connectés simultanément.

Lorsqu'un utilisateur se connecte, la passerelle d'interception Coova-Chilli génère un Hook (déclenche un évènement) nous permettant de récupérer en direct les attributs propres à l'utilisateur. Grâce à cette fonctionnalité, en fonction de l'état de la variable correspondant au filtrage, on ajoute l'adresse IP de l'utilisateur dans le SET correspondant. De cette façon, les utilisateurs se trouvent aiguillés vers leur niveau

de filtrage. Ensuite, dans le pare-feu, je dirige les SET dans le chemin qui leur est destiné (voir le schéma de l'architecture d'ALCASAR en annexe), en ayant pris soin de camoufler les ports utilisés afin d'éviter les scans de port, et de filtrer les SET **havp_bl_set** et **havp_wl_set**.

Concernant le filtrage par l'antivirus, deux instances de l'antivirus doivent être présentes. Un antivirus étant en mode non-transparent afin d'accueillir les paquets provenant de Dansguardian (servant au filtrage des URL) dans le cas du filtrage par la liste noire et un en mode transparent afin d'y envoyer directement les paquets dans le cas du filtrage par l'antivirus uniquement (via le SET **havp_set**). J'ai donc intégré ce procédé au système.

Aussi, afin de rendre le système de filtrage par noms de domaine opérationnel, deux nouvelles instances de Dnsmasq ont été mises en place. Une nommée « Dnsmasq-blacklist », afin de rediriger les utilisateurs vers une page de blocage s'ils tentent d'accéder à un site bloqué. Une seconde nommée « Dnsmasq-whitelist », afin d'interdire tous les noms de domaine sauf ceux autorisés par la liste blanche.

5 Mise à jour de Mageia 2 vers Mageia 4

Basculement

La future version d'ALCASAR V2.9, devant sortir sous Mageia 4, j'ai donc dû adapter mon travail à cette nouvelle distribution afin que tout reste fonctionnel. Les scripts d'installation et de désinstallation ont été modifiés afin qu'ALCASAR intègre mes modifications lors de son installation, et les supprime lors de sa désinstallation. De plus, Mageia 4 exploitant le nouveau lanceur « SystemD », différents paquets ont dû être modifiés et recompilés afin qu'ils puissent s'adapter (Proxy antivirus : HAVP, Dnsmasq). Enfin, j'ai réeffectué tous les tests afin d'être complètement sûr que tout fonctionne convenablement sous ce nouveau système.

Compilation d'un paquet

ALCASAR V2.9 fonctionnant sous Mageia 4, se servait encore du logiciel HAVP (Proxy antivirus) packagé pour Mageia 2. Désirant acquérir un maximum d'expérience durant ces quatre mois de stage, j'ai voulu le packager pour Mageia 4. J'ai donc effectué des recherches sur la compilation de paquets, et bien entendu modifié le code de HAVP pour qu'il s'adapte à l'architecture de Mageia 4. Une fois ce paquet terminé, et s'installant correctement, j'ai eu pour objectif de le proposer à Mageia. Mageia fonctionne avec un système de Mentoring pour les nouveaux contributeurs. Dans le but de devenir contributeur, j'ai pris contact avec l'équipe de développement sur le forum IRC de Mageia et j'ai demandé la procédure à suivre. Un site m'a alors été indiqué, sur lequel je me suis inscrit en indiquant mes connaissances, afin qu'un Mentor m'accepte. Je suis à l'heure actuelle en attente d'une réponse.

Résolution d'un « bogue »

Suite à la mise à jour du logiciel antivirus LibClamAV (0.98.4), HAVP ne pouvait plus fonctionner, indiquant que la base de données ne pouvait être chargée à cause d'un problème d'allocation de mémoire. Après quelques recherches, il s'est trouvé que Clamav avait intégré OpenSSL et que certaines fonctions d'initialisation devaient être appelées par HAVP. J'ai donc intégré les fonctions **int cl_initialize_crypto(void)** et **void cl_cleanup_crypto(void)** au fichier d'en-tête correspondant (/usr/include/clamav.h), puis modifié le code source d'HAVP afin qu'il prenne en compte l'appel à ces fonctions. Après recompilation, tout fonctionnait parfaitement. J'ai donc proposé un patch à l'équipe de HAVP qui m'a répondu que ma solution était correcte, mais qu'ils attendent un retour de l'équipe de Clamav pour le moment.

Conclusion

Au cours de cette expérience de quatre mois, j'ai eu la chance de participer au développement d'un projet libre, au travers duquel j'ai pu mettre en pratique et améliorer mes connaissances, que cela soit du point de vue de la sécurité des systèmes, ou de celui de la programmation. En travaillant au sein de ce projet, j'ai eu la chance de pouvoir élargir mon expérience en m'intéressant à la compilation de paquets et à la résolution de bogues, ce qui m'a permis d'entrer en contact avec les équipes d'HAVP et de Mageia, pour peut-être une contribution future. ALCASAR étant un projet collaboratif, j'ai pu découvrir la façon de fonctionner d'un tel projet et donc me familiariser avec des logiciels de travail collaboratif.

Travailler sur ce projet c'est aussi prendre en considération les problèmes que peuvent rencontrer ses utilisateurs. C'est la raison pour laquelle, j'ai pu entrer en contact avec quelques membres du forum afin de leur venir en aide sur des problèmes que j'étais en mesure de régler, et d'avoir quelques retours de personnes externes à l'équipe d'ALCASAR à propos de la prochaine version.

Pour conclure, j'aurai souhaité avoir un peu plus de temps afin de poursuivre mon travail sur ALCASAR, pour étudier d'autres branches d'évolution.

Glossaire

BASH : Bourne-Again Shell

DNS : Domain Name System

GET : tableau des valeurs passées au script courant via les paramètres d'URL en PHP

GPL : General Public Licence

IP : Internet Protocol

MAC : Media Access Control

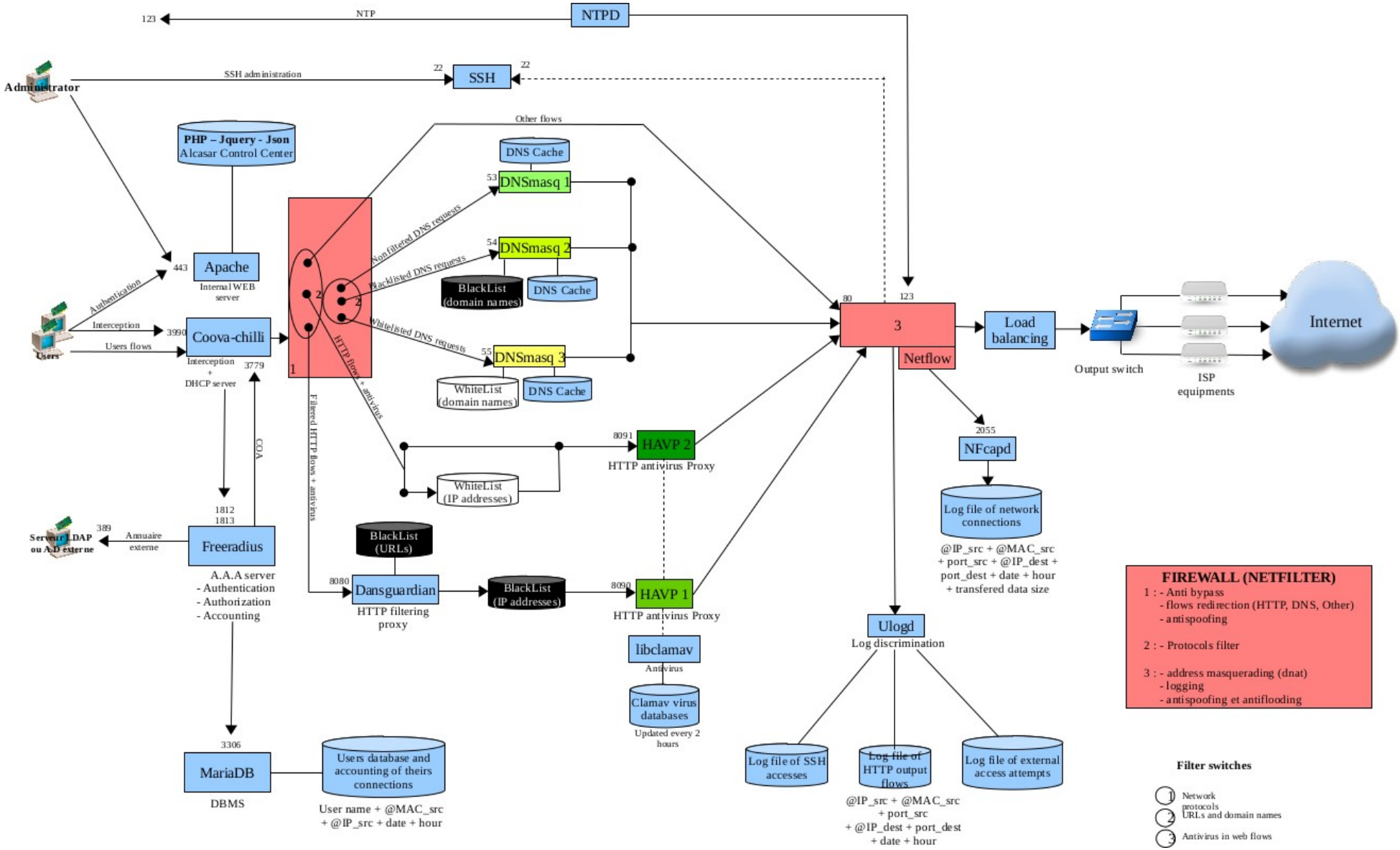
PHP : Language de programmation libre pour produire des sites web

SSH : Secure Shell

URL : Uniform Resource Locator

Annexes

Annexe 1 : Architecture d'ALCASAR


NTP
123
NTPD
SSH administration
22
SSH
22
Administrator
PHP – Jquery - Json
Alcasar Control Center
Other flows
DNS Cache
53
DNSmasq 1
Non filtered DNS requests
54
DNSmasq 2
Blacklisted DNS requests
BlackList (domain names)
DNS Cache
Authentication
443
Apache
Internal WEB server
Interception
3990
Coova-chilli
Users flows
Users
Interception + DHCP server
3779
1
2
3
HTTP flows + antivirus
Whitelisted DNS requests
55
DNSmasq 3
WhiteList (domain names)
DNS Cache
80
123
3
Netflow
Load balancing
Output switch
Internet
ISP equipments
8091
HAVP 2
HTTP antivirus Proxy
WhiteList (IP addresses)
2055
NFcapd
Log file of network connections
@IP_src + @MAC_src + port_src + @IP_dest + port_dest + date + hour + transfered data size
Filtered HTTP flows + antivirus
COA
1812
1813
Freeradius
A.A.A server
- Authentication
- Authorization
- Accounting
Serveur LDAP ou A.D externe
389
Annuaire externe
BlackList (URLs)
8080
Dansguardian
HTTP filtering proxy
BlackList (IP addresses)
8090
HAVP 1
HTTP antivirus Proxy
libclamav
Antivirus
Clamav virus databases
Updated every 2 hours
Ulogd
Log discrimination
Log file of SSH accesses
Log file of HTTP output flows
Log file of external access attempts
@IP_src + @MAC_src + port_src + @IP_dest + port_dest + date + hour
3306
MariaDB
DBMS
Users database and accounting of theirs connections
User name + @MAC_src + @IP_src + date + hour
FIREWALL (NETFILTER)
1 : - Anti bypass
- flows redirection (HTTP, DNS, Other)
- antispoofing
2 : - Protocols filter
3 : - address masquerading (dnat)
- logging
- antispoofing et antiflooding
Filter switches
1 Network protocols
2 URLs and domain names
3 Antivirus in web flows