



Auteur : Raphaël PION

# Rapport de stage technique 4A

Développer, tester et intégrer les évolutions planifiées sur  
ALCASAR

<b>Lieu</b>	ESIEA-Laval
<b>Date</b>	04/04/2016 → 01/07/2016
<b>Directeur</b>	M. Jean Labourdette
<b>Maître de stage</b>	M. Richard Rey
<b>Tuteur</b>	M. Olivier Ferrand
<b>Co-Tuteur</b>	M. Stéphane Weber



## Résumé

Ce rapport traite des améliorations du contrôleur d'accès au réseau ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau). Ce portail captif est un projet libre et gratuit sous licence GNU GPLv3 qui se base sur la distribution française Mageia 5. Il permet aux particuliers ainsi qu'aux professionnels de mettre en place un réseau de consultation Internet répondant aux exigences de la loi française. En effet, ALCASAR protège le responsable du réseau en traçant, en imputant les connexions et en authentifiant tous les usagers du réseau. De plus, ALCASAR dispose d'un système de filtrage de liste noire/blanche pouvant être utilisé dans les établissements scolaires.

Mon travail effectué durant ces 4 mois a été de faire basculer la version 2.9.2 vers la version 3.0 d'ALCASAR. Parmi ces nombreuses améliorations, un nouveau système d'interception pour s'authentifier a été mis en place permettant de traiter les requêtes HTTP et HTTPS des utilisateurs. En plus des améliorations de l'ACC (ALCASAR Control Center), des erreurs présentes dans le PHP causées par le basculement vers la distribution Mageia 5 ont été corrigées. Le système de filtrage a été repensé permettant ainsi de traiter la liste blanche de la même façon que la liste noire. La phase de test a permis de relever des erreurs et de sortir le 20 juillet 2016 la version 3.0 d'ALCASAR.

En parallèle, une solution de filtrage de protocole par utilisateur a été implémentée en ajoutant des règles au pare-feu interne d'ALCASAR. Deux demandes ont été faites par le RSSI du commissariat de la ville de Laval. La première était de simplifier l'extraction des journaux d'imputabilité sous la forme d'un document PDF tout en respectant la vie privée des utilisateurs. La deuxième consistait à créer un rapport PDF hebdomadaire facilement compréhensible résumant l'activité de l'ALCASAR.

Ce document détaille les différentes missions effectuées pendant ce stage qui permettront la sortie de la version 3.0 et de répondre aux demandes extérieures. Chaque mission est expliquée, accompagnée d'une solution décrivant ainsi les difficultés rencontrées et la manière dont elles ont été surmontées.

## Abstract

This report deals with enhancements of ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau). This network access controller is an open source project under GNU GPLv3 which is based on the french distribution Mageia 5. It allows individuals and professionals to set up an Internet consultation network respecting the French law. Indeed, ALCASAR protects the network manager by tracking, imputing connections and by authenticating all users. Moreover, ALCASAR has a filtering system of black (or white) list which can be used in schools.

My work during these four months was to switch the version 2.9.2 to the version 3.00 of ALCASAR. Among the many improvement, a new interception system for authentication has been set up to handle HTTP and HTTPS user's requests. In addition to the improvements of ACC (ALCASAR Control Center), I corrected the errors in the PHP caused by the changeover to the Mageia distribution 5. The filter system has been redesigned to treat in same way the white and black list. The checklist allow me to reveal some bugs and release the version 3.00 of ALCASAR the 20<sup>th</sup> of July 2016.

In parallel, I have implemented a user protocol filtering solution by adding rules to the internal firewall of ALCASAR. Two requests has been made by the RSSI of the police station in Laval. The first was to simplify the extraction of imputability logs into a PDF document by respecting user privacy. The second was to create an understable PDF which summerizes the activity of ALCASAR every week.

This document details every missions whici will allow to realease the version 3 of ALCASAR and answer to external requests. Each mission is explained and accompanied by a solution, a describing about the difficulties encountered and how they were overcome

## Remerciements

Je remercie toutes les personnes ayant contribué au projet ALCASAR et plus particulièrement celles qui ont permis la réussite de ce stage en assurant notamment son financement. Je souhaite remercier le personnel du laboratoire CVO grâce auquel j'ai pu travailler dans de très bonnes conditions.

Je remercie Monsieur REY et son précieux sens de la pédagogie pour m'avoir guidé et aidé durant ce stage riche en expériences et en connaissances.

## Table des matières

1	Introduction.....	6
1.1	Présentation de l' établissement.....	6
1.2	Projet ALCASAR.....	6
1.3	Problématique.....	7
2	Les évolutions d'ALCASAR.....	8
2.1	Configuration du set-up.....	8
2.2	ALCASAR 3.0.....	9
2.2.1	Interception HTTPS.....	9
2.2.2	Modification de la configuration réseau via l'ACC.....	11
2.2.3	Importation de l'autorité de certification d'ALCASAR dans le navigateur client.....	12
2.2.4	Erreurs PHP.....	12
2.2.4.1	Introduction.....	12
2.2.4.2	Corrections du code PHP.....	13
2.2.4.3	Modification sur les utilisateurs.....	13
2.2.5	SafeSearch pour la whitelist.....	14
2.2.6	Modification du traitement de la Liste de Toulouse.....	16
2.3	Améliorations en développement de ALCASAR 3.0b.....	18
2.3.1	Filtrage protocole / utilisateur.....	18
2.3.2	Vérifier l'état de la fenêtre status.php.....	20
2.3.3	Génération du rapport d'imputabilité.....	21
2.3.3.1	Corrélation des informations.....	21
2.3.3.2	Génération du document.....	22
2.3.3.3	Avertir les utilisateurs.....	22
2.3.4	Rapport d'activité hebdomadaire.....	23
2.3.4.1	Consultation via l'ACC.....	25
2.3.5	Création d'un RPM afin d'installer wkhtmltopdf.....	26
2.3.6	Option NTP du DHCP.....	26
2.4	Missions annexes de ALCASAR.....	28
2.5	CheckmyHTTPS.....	28
2.5.1	CheckmyHTTPS SDK.....	28
2.5.2	Présentation/conférence.....	31
2.5.3	Reprise du projet.....	31
3	Conclusion.....	33
4	Annexe.....	34
4.1	Exemple de rapport d'activité de la machine ALCASAR.....	36
5	Liste des illustrations.....	39

# 1 Introduction

## 1.1 Présentation de l'établissement

Mon stage s'est déroulé dans le laboratoire (C+V)<sup>o</sup> (Cryptologie et Virologie Opérationnelles) situé dans les locaux de l'École Supérieure d'Informatique, Électronique et Automatique (ESIEA) à Laval.

Créé en 2007 et dirigé par monsieur FILIOL, ce laboratoire de recherche est spécialisé dans le domaine de la lutte informatique aussi bien défensive qu'offensive, la sécurité informatique, la virologie et la cryptologie.



*Illustration 1: Logo (C+V)<sup>o</sup>*

## 1.2 Projet ALCASAR

ALCASAR (Application Libre pour le Contrôle d'Accès Sécurisé et Authentifié au Réseau) est un projet libre et open source sous licence GNU GPLv3. C'est un contrôleur d'accès au réseau ( ou NAC Network Access Controller) permettant de protéger, de contrôler et d'imputer les accès d'un réseau de consultation Internet dans le respect de la loi française.

Étudiant en 4<sup>ème</sup> année à l'ESIEA, j'ai commencé à appréhender ALCASAR lors de mon projet scientifique et technique (PST). En effet, j'ai travaillé avec Clément SICCARDI, Hugo MEZIANI et Bettyna BOURCIER afin de faire migrer ALCASAR vers la distribution Mageia 5. Avant que mon stage débute, ALCASAR était à la version 2.9.2. Sa

version actuelle est la 3.0, sortie le 20 juillet 2016.

ALCASAR se place entre le réseau de consultation et le point d'accès Internet. Il permet de tracer, d'imputer les connexions entrantes et sortantes du réseau. La création d'utilisateur ou de groupe permet d'authentifier et de filtrer. A l'aide de règles de pare-feu et d'antivirus, ALCASAR permet de protéger le réseau qu'il administre.



*Illustration 2: Logo ALCASAR*

## **1.3 Problématique**

Le sujet de mon stage portera sur les améliorations du projet ALCASAR. Pour commencer, j'ai installé et étudié le fonctionnement de la version de développement d'ALCASAR (3.0b) présent sur le subversion (SVN) officiel du projet.

Ce projet commence à prendre de l'ampleur. Par conséquent de plus en plus de fonctionnalités viennent s'intégrer au contrôleur d'accès. Parmi ces fonctionnalités, il y a le filtrage par utilisateur, l'authentification, la traçabilité, etc.

Afin d'aborder le projet, ma première mission était d'étudier et de tester le fonctionnement de whitelist. Une fois cette première mission terminée, mon stage pouvait commencer.

## 2 Les évolutions d'ALCASAR

### 2.1 Configuration du set-up

J'ai choisi de travailler dans un environnement virtuel, car cela me permettait de continuer à travailler de n'importe où. Voici la configuration de mes machines virtuelles :

-Utilisation de VirtualBox (Version 5.0.16 r105871)

-Machine ALCASAR-3.00b (récupéré depuis le SVN officiel du projet) installée avec l'ISO Mageia-5-dual-DVD.iso

-Machine cliente Mageia : 3.10.60-desktop-1.mga3

→ navigateur web : Firefox ESR 32.2.0

-Configuration réseau :

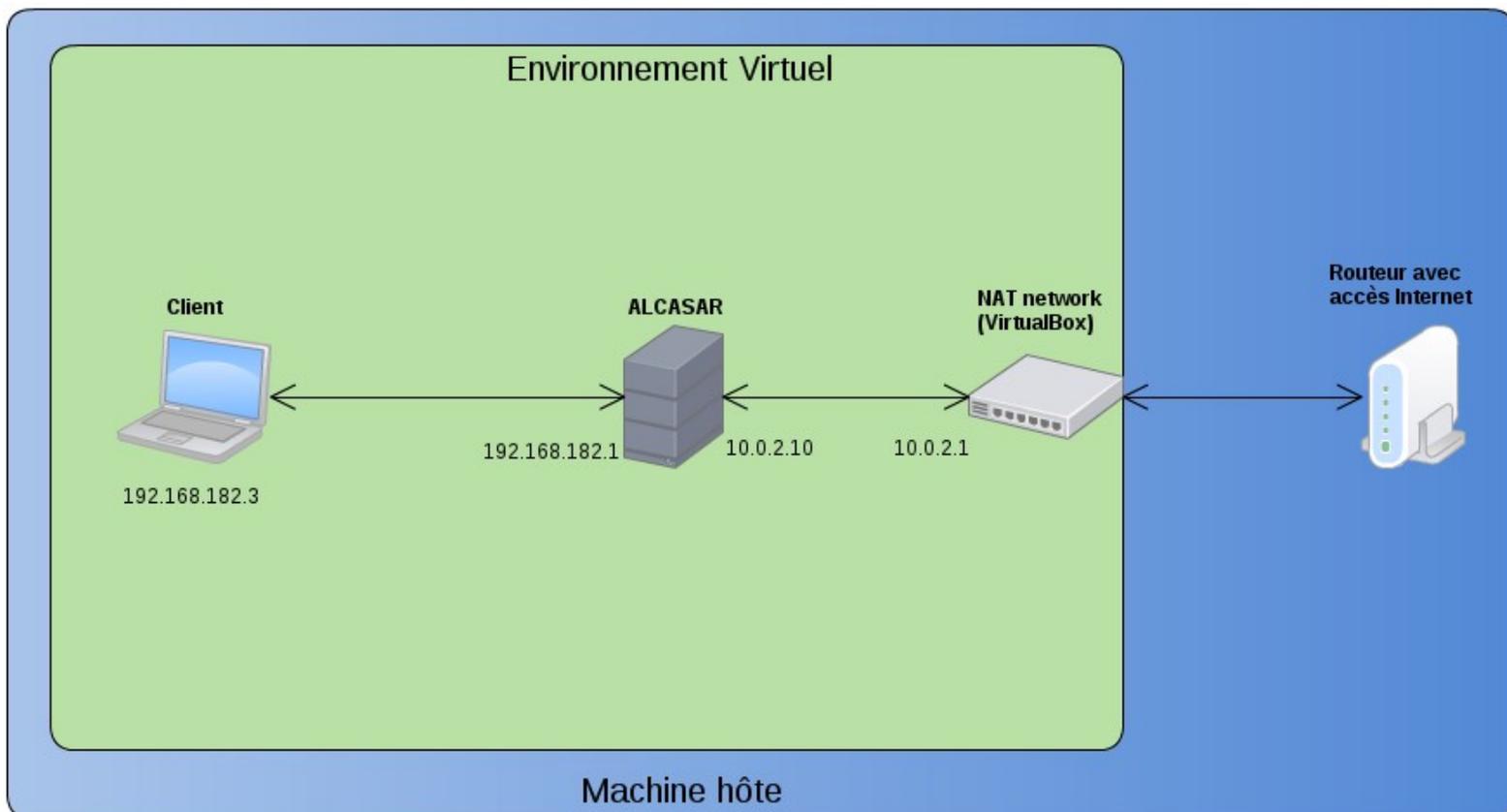


Illustration 3: Schéma réseau du set-up

## 2.2 ALCASAR 3.0

La version 3.0 de ALCASAR est sortie le 20 juillet 2016. Il est intéressant de voir comment cette version a été lancée. Tout d'abord, il y avait une version de développement sur le SVN qui servait à l'implémentation des versions futures. Une fois que cette version contenait des éléments majeurs en terme d'amélioration, on soumettait une version bêta à la communauté afin de la tester et d'avoir un retour sur les erreurs qui se sont produites. Et enfin, lorsque la version de développement est stable, la version officielle peut être publiée.

J'ai appris à développer dans le cadre d'un projet libre. Les contraintes de temps sont plus flexibles afin de sortir une nouvelle version. Cela m'a permis de travailler dans de bonnes conditions tout en enrichissant mes connaissances.

Lors de la sortie officielle de la version 3.0, des problèmes ont été remontés durant le mois d'août concernant l'édition des utilisateurs. Ces problèmes ont été corrigés, il s'agissait de modification mineure en PHP afin de rendre compatibles les anciennes bases de donnée issue d'ALCASAR 2.9.2.

Dans la suite du document, vous aurez toutes les améliorations apportées depuis la version 2.9.2.

### 2.2.1 Interception HTTPS

#### Mission :

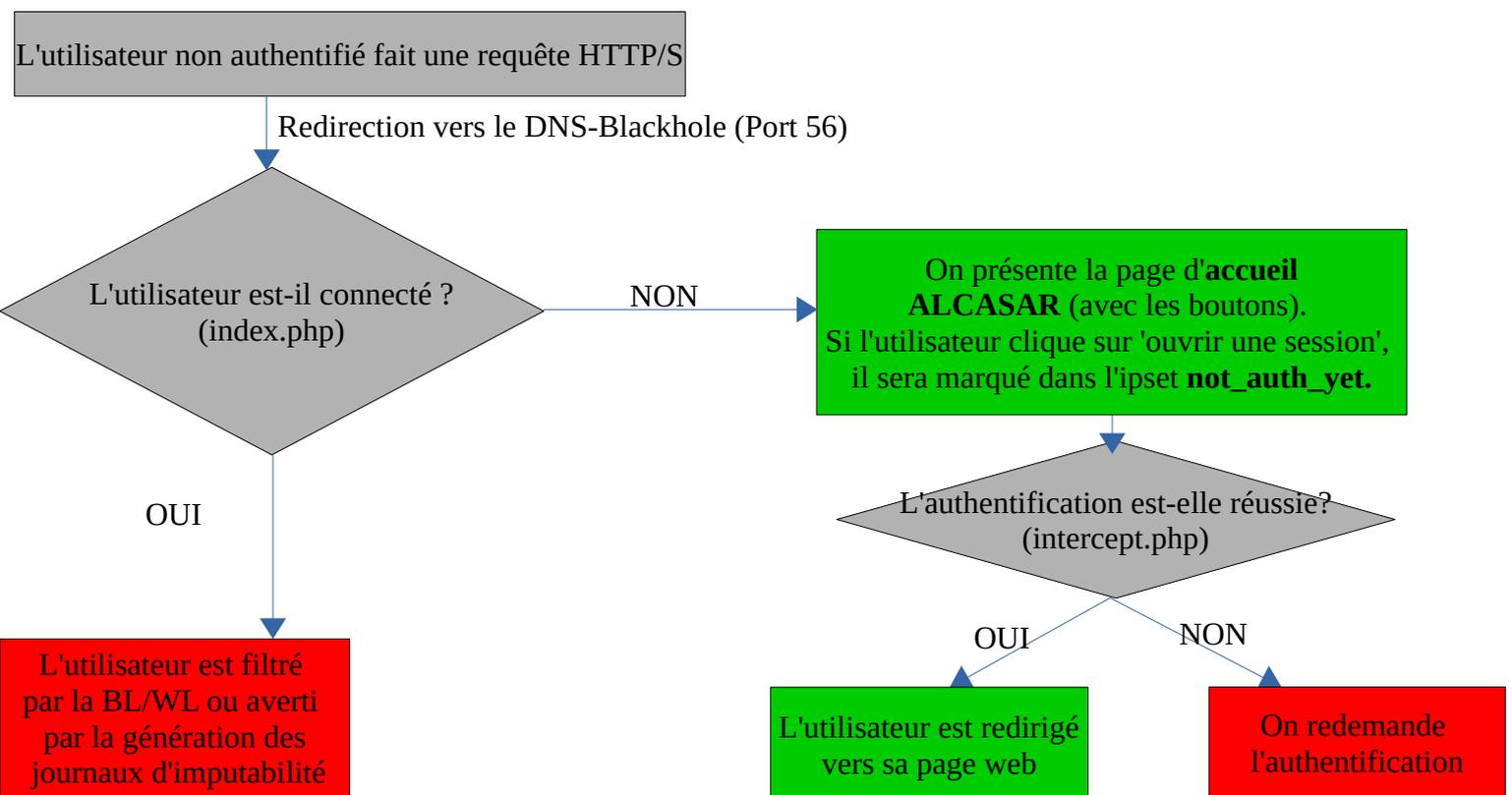
Le service Coova Chilli ne peut intercepter que le trafic HTTP afin de rediriger l'utilisateur vers une page d'authentification (intercept.php).

Si l'utilisateur démarre son navigateur et qu'il souhaite joindre une page HTTPS (exemple : <https://www.google.fr>), il ne pourra pas se faire intercepter par la page d'authentification, car le navigateur est en attente d'une réponse d'ALCASAR et le navigateur n'affiche rien.

#### Solution :

Toutes les requêtes DNS (port 53 vers la machine ALCASAR) d'un utilisateur non authentifié seront redirigées vers le DNS-Blackhole (port 56). ALCASAR présentera alors la page d'accueil (</var/www/html/index.php>).

Pour se faire, un nouvel ipset a été créé, **not\_auth\_yet**, ainsi qu'une liste d'ipset (**users\_list**) utile pour faire correspondre une IP avec plusieurs ipset dans les règles de pare-feu. Voici le fonctionnement de cette interception lorsque l'utilisateur est redirigé dans le DNS-Blackhole :



Pour savoir si un utilisateur est connecté, du point de vue des règles de pare-feu, il suffit de dire que son adresse IP n'est pas présente dans ces ipsets : **havp**, **havp\_wl**, **havp\_bl** et **not\_filtered**, **not\_auth\_yet**. La règle de parefeu PREROUTING a été ajoutée dans le script [/usr/local/bin/alcasar-iptables.sh](#).

Comme vous avez pu le constater, l'IP de l'utilisateur est placé dans l'ipset **not\_auth\_yet**. Cela évite de reboucler le processus de tel sorte que l'utilisateur n'aura pas accès à l'interface de login [/var/www/html/intercept.php](#) vu qu'il sera constamment redirigé sur [/var/www/html/index.php](#).

L'ipset **not\_auth\_yet** se supprime juste après qu'il ait atteint la page 'intercept.php' ou bien lorsque l'utilisateur a réussi son authentification (cf. script [/usr/local/bin/alcasar-conup.sh](#)). Cela permet à l'utilisateur non authentifié de se refaire intercepter. La documentation technique a été mise à jour concernant cette fonctionnalité.

#### Difficultés rencontrées :

- La règle d'interception du pare-feu a été remontée en première position afin d'augmenter sa rapidité. En effet, la correspondance des règles de pare-feu se fait de manière ascendante. Pour voir le pare-feu en direct :

```
watch iptables -nvL (-t nat/INPUT/OUTPUT/FORWARD/[PRE/POST]ROUTING)
```

- La redirection ne se faisait pas proprement. En effet, le cache du navigateur devait

être vidé

- Le navigateur ne renouvelait pas sa requête DNS. Ce qui impliquait que l'utilisateur arrive sur la page d'interception de la Blacklist. Le cache DNS est forcé si dans la page index.php :

-il est toujours dans l'ipset not\_auth\_yet

-il demande à joindre un autre site que ALCASAR (!\$direct\_access)

-il ne demande plus les boutons de la page d'accueil d'ALCASAR

Bien que cela soit en phase de développement, le service dnsmasq-blackhole envoie sa réponse DNS à la machine cliente avec un TTL de '0' afin que le navigateur renouvelle ses requêtes DNS.

## 2.2.2 Modification de la configuration réseau via l'ACC

### Mission :

Dans l'ACC → Système → réseau, la page présentée était passive. Elle n'affichait que les informations de la configuration réseau d'ALCASAR. L'objectif est de rendre cette page active en ayant la possibilité de modifier cette configuration.

### Solution :

Via le PHP, il est possible de modifier le fichier de configuration d'ALCASAR (/usr/local/etc/alcasar.conf). Les entrées de l'administrateur sont vérifiées avec des expressions régulières. Les valeurs modifiées dans ce fichier sont : DNS1,DNS2,PUBLIC\_IP,GW,PRIVATE\_IP.

Lorsque l'administrateur valide sa modification, un script applique la configuration à ALCASAR (/usr/local/bin/alcasar-conf.sh --apply).

Network configuration		
<b>INTERNET</b> ✓ Public IP address : 77.148.221.157 DNS1 208.67.222.222 DNS2 208.67.220.220	<b>enp0s3 (Internet connected interface)</b> IP Address 10.0.2.10/24 Gateway 10.0.2.1	<b>enp0s8 (Private network)</b> IP Address 192.168.182.1/24
<input type="button" value="Apply changes"/>		

Illustration 4: Modification de la configuration réseau via l'ACC

Lors de l'application des changements, le service httpd se terminait entraînant ainsi la mort de son fils : PHP. Afin de pallier à ce problème, il fallait modifier le script alcasar-conf.sh afin de modifier le signal envoyé au service httpd.

Ce service est traité de la façon suivante dans le script :

- Httpd s'arrête pendant que les principaux services se terminent.

- Httpd continue et recharge (reload) tous ses fichiers de configuration lorsque les autres services redémarrent.

La documentation d'exploitation a été mise à jour (paragraphe 2.2).

## 2.2.3 Importation de l'autorité de certification d'ALCASAR dans le navigateur client

### Mission :

Lors de la sortie de la 3.0b d'ALCASAR, il a été remonté que l'importation de l'autorité de certification d'ALCASAR ne s'installait plus automatiquement. Cet import de certificat est disponible sur `index.php` et permet d'éviter les alertes de certificat autosigné.

### Solution :

Il suffisait de renommer le fichier `.crt` en `.der` afin de le rendre compréhensible par le navigateur.

### Complément d'information :

DER signifie « *Distinguished Encoding Rules* ». Cet encodage se fait en fonction de 4 critères : l'identifiant, la taille, le contenu, la fin du contenu des octets. Vous pouvez lire le contenu d'un certificat DER avec `openssl` :

```
openssl x509 -in cert.der -inform der -noout -text
```

PEM signifie « *Privacy Enhanced Mail* ». Cet encodage est défini dans la RFC 1421-1424. Les certificats PEM se présentent de la manière suivante :

A pre-boundary line of "-----BEGIN CERTIFICATE-----".

Base64 encoding output of DER encoded certificat.

A post-boundary line of "-----END CERTIFICATE-----".

Vous pouvez lire le contenu d'un certificat PEM avec `openssl` :

```
openssl x509 -in keytool_crt.pem -inform pem -noout -text
```

## 2.2.4 Erreurs PHP

### 2.2.4.1 Introduction

Lors de mon PST 4A, Clément SICCARDI s'est occupé de mettre à jour le code PHP de ALCASAR 2.9 en corrigeant ainsi les warnings PHP. Il restait cependant la partie statistique qui était volontairement laissée.

La 3.0b de ALCASAR a été testée par sa communauté et quelques erreurs ont été remontées. L'une d'entre elles était que la création de tickets de plusieurs utilisateurs ne fonctionnait plus.

### 2.2.4.2 Corrections du code PHP

Clément SICCARDI a mis à jour les fonctions PHP faisant appel au SQL. En effet, nous sommes passés de l'API mysql à mysqli. Et cela n'a pas été pris en compte dans certaines fonctionnalités de l'ACC. En effet, mysql et mysqli retournent deux objets SQL différents. Il fallait donc traiter ce problème.

Ensuite tous les warnings présents dans la partie statistique ont été corrigés rendant ainsi ALCASAR à jour concernant le PHP. Voici une liste des quelques modifications effectuées :

- /etc/freeradius-web/sql.attrs :ajout des attributs 'groupname' et de 'xascendsessionsvrkey'
- 'da\_sql\_num\_fields' a été supprimé dans fonction.php du SQL. Cette fonction ne renvoyait plus la bonne valeur.
- Nettoyage des warnings dans la partie statistique de l'ACC
- /lib/sql/default.php : requête listant les groupes
- suppression de la fonction date2time(). Il faut utiliser strtotime() de l'API PHP.
- Les colonnes download et upload ont été inversées (pour être cohérent) dans 'connections' de l'ACC.
- Dans 'categorie\_help.php', le nom de la catégorie ainsi que la description s'affichent selon la langue.
- Il n'y a plus de warning dans le PHP.

### 2.2.4.3 Modification sur les utilisateurs

Un utilisateur ne peut faire partie que d'un seul groupe. Par exemple, on ne peut pas faire partie d'un groupe blacklist et d'un groupe whitelist en même temps. Le groupe de l'utilisateur peut être modifié via l'éditeur de l'ACC. Il suffit de créer un groupe. Il apparaîtra dans les attributs de l'utilisateur. Pour lui attribuer un groupe, il faut surligner le groupe en question et valider la modification. Si aucun groupe n'existe, l'administrateur en est informé.

Les utilisateurs not\_filtered ont maintenant un Filter-Id (00000000). Auparavant, cet attribut était vide dans la BDD pour les utilisateurs non filtrés.

Lors de l'édition d'un usager, dans l'attribut 'nombre de connexions simultanées', j'ai remarqué le hash avec 'salage' de l'utilisateur. Cette erreur a été corrigée suite à la mise à jour du code PHP.

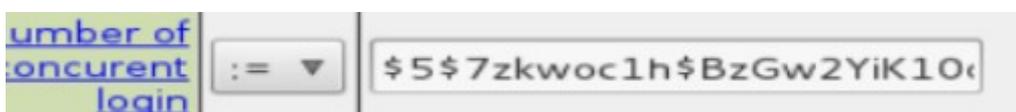


Illustration 5: bug hash utilisateur

## 2.2.5 SafeSearch pour la whitelist

### Mission :

Le safesearch est déjà présent dans la blacklist, il fallait l'adapter pour la whitelist.

### Réflexion :

En étudiant cette fonctionnalité, j'ai décidé de lancer un service dans guardian uniquement pour la whitelist afin de pouvoir modifier l'URL de l'utilisateur en y ajoutant des *flags* particuliers pour activer le safesearch.

Pour se faire, il a fallu ajouter un fichier dans `/usr/lib/systemd/` et le lier vers `/etc/systemd/system/multi-user.target.wants`. Voici le contenu de ce fichier :

```
[Unit]
Description=Dansguardian Whitelist
After=network.target chilli.service

[Service]
Type=forking
PIDFile=/run/dansguardian-wl.pid
ExecStart=/usr/sbin/dansguardian -c /etc/dansguardian/dansguardian-wl.conf

[Install]
WantedBy=multi-user.target
```

Le PID est écrasé par le fichier de configuration `/etc/dansguardian/dansguardian-wl.conf`. Il faut donc spécifier le chemin du fichier PID, le port du service (ici 8100) ainsi que l'emplacement des fichiers de journalisation. 'systemctl daemon-reload' permet de recharger les fichiers des services. Les fichiers de paramètre de dansguardian-wl utilisés sont :

- `urlregexplist-wl` : expression régulière permettant d'ajouter le flag activant le safesearch (HTTP uniquement)
- `bannedsitelist` : permet d'interdire l'accès aux sites par l'adresse IP.

Il faut ensuite modifier les règles de pare-feu afin de diriger l'utilisateur vers le service dansguardian-wl (port 8100) à la place du tinyproxy (port 8090).

Pour tester le fonctionnement de dansguardian-wl, il fallait connecter un utilisateur filtré par la whitelist, autoriser le nom de domaine 'astrolabio.net' (par exemple) via l'ACC et ensuite rejoindre l'URL <http://www.astrolabio.net/casino/> (même si le lien est mort).

Afin de faire les choses proprement, le service dansguardian de base (dédié uniquement à la blacklist) devient maintenant dansguardian-bl (toujours sur le port 8090).

Dans l'ACC, l'interface de configuration de la whitelist a été modifiée sur le même modèle que la blacklist. Deux cases ont été ajoutées : une pour le safesearch et l'autre pour interdire l'accès des sites par l'adresse IP.

Pour la recherche sécurisée de Google, le safesearch fonctionne de la façon suivante, toutes les URLs de ce moteur de recherche sont redirigées vers un DNS spécial permettant de rediriger l'utilisateur vers le safesearch. Pour plus d'information, veuillez consulter le lien suivant :

<https://support.google.com/websearch/answer/186669?hl=fr>

En modifiant la configuration du service dnsmasq-whitelist, on peut rediriger l'utilisateur vers le safesearch de Google. Il faut donc autoriser l'IP du forcessafesearch de Google en l'ajoutant dans le fichier */usr/local/share/ossi-ip-wl*.

Au final, le filtrage des URLs de la whitelist n'est pas pertinent. En effet, l'accès laissé par la WL est très sélectif et la modification de l'URL pour activer le safesearch se fait en HTTP, or, la plupart des moteurs de recherche sont en HTTPS (à l'exception de Bing, pour l'instant). En effet, le HTTPS ne peut pas être traité par dansguardian.

On peut donc se débarrasser du service dansguardian concernant la WL. De plus le safesearch de YouTube via un ID n'est plus possible, car le site n'est accessible qu'en HTTPS. Le lien expliquant la démarche pour obtenir cet ID a été mis à jour :

<https://support.google.com/youtube/answer/174084?hl=fr>

Cependant, après avoir discuté sur le sujet avec mon maître de stage, nous avons décidé de garder la restriction concernant l'accès aux sites par l'IP en modifiant la configuration de tinyproxy. Tinyproxy peut filtrer les URLs en activant l'option « Filter "/etc/tinyproxy/filter" » dans le fichier de configuration */etc/tinyproxy/tinyproxy.conf*.

Il faut donc ajouter une expression régulière bloquant les adresses IP :

```
[012]\?[0-9][0-9]\?.[012]\?[0-9][0-9]\?.[012]\?[0-9][0-9]\?.[012]\?[0-9][0-9]\?
```

Cette règle n'est pas précise, car elle filtre les adresses IP allant de 0.0.0.0 à 300.300.300.300. Ceci est dû au fait que tinyproxy comprend très mal les expressions régulières complexes (désécialisation, syntaxe différente...). Cependant le contournement de cette règle est toujours possible en passant par le HTTPS.

Au final, la modification du tinyproxy n'était pas nécessaire, car la whitelist possédait déjà cette fonctionnalité en utilisant l'ipset *whitelist\_ip\_allowed*.

#### Solution finale:

Les modifications sont :

- *alcasar-url\_filter\_wl.sh* : modification de la configuration *dnsmasq-whitelist.conf* et correction de l'importation des fichiers à la main (*ossi-ip-wl*).
- *alcasar-url\_filter.sh* a été renommé en *alcasar-url\_filter\_bl.sh* (anciennement)

- bl\_filter.php : mise à jour de la démarche pour acquérir un YouTube ID
- wl\_filter.php : ajout du safesearch uniquement pour Google, car ce sont les seuls pour le moment à disposer d'un DNS spécial.

La fonctionnalité de safesearch pour la WL a été ajoutée dans l'ACC et modifie uniquement la configuration du service dnsmasq-whitelist (/etc/dnsmasq-whitelist.conf).

#### Problèmes rencontrés:

- Redirection pour les autres moteurs de recherche
  - duckduckgo : safe.duckduckgo.com est un alias pour duckduckgo.com.
  - Qwant : En ajoutant 'address=/.qwant.com/194.187.168.102' dans les fichiers de configuration de dnsmasq. La redirection n'affiche pas les résultats provenant de qwant.com affichant ainsi une page presque vide.



*Illustration 6: problème avec le safesearch de Qwant*

## **2.2.6 Modification du traitement de la Liste de Toulouse**

### Mission :

Après avoir étudié le fonctionnement de la whitelist (documentation mise à jour) et celui de la blacklist, M. Rey a décidé de réorganiser son traitement.

### Solution

Dorénavant, on traite la blacklist et la whitelist de la même façon.

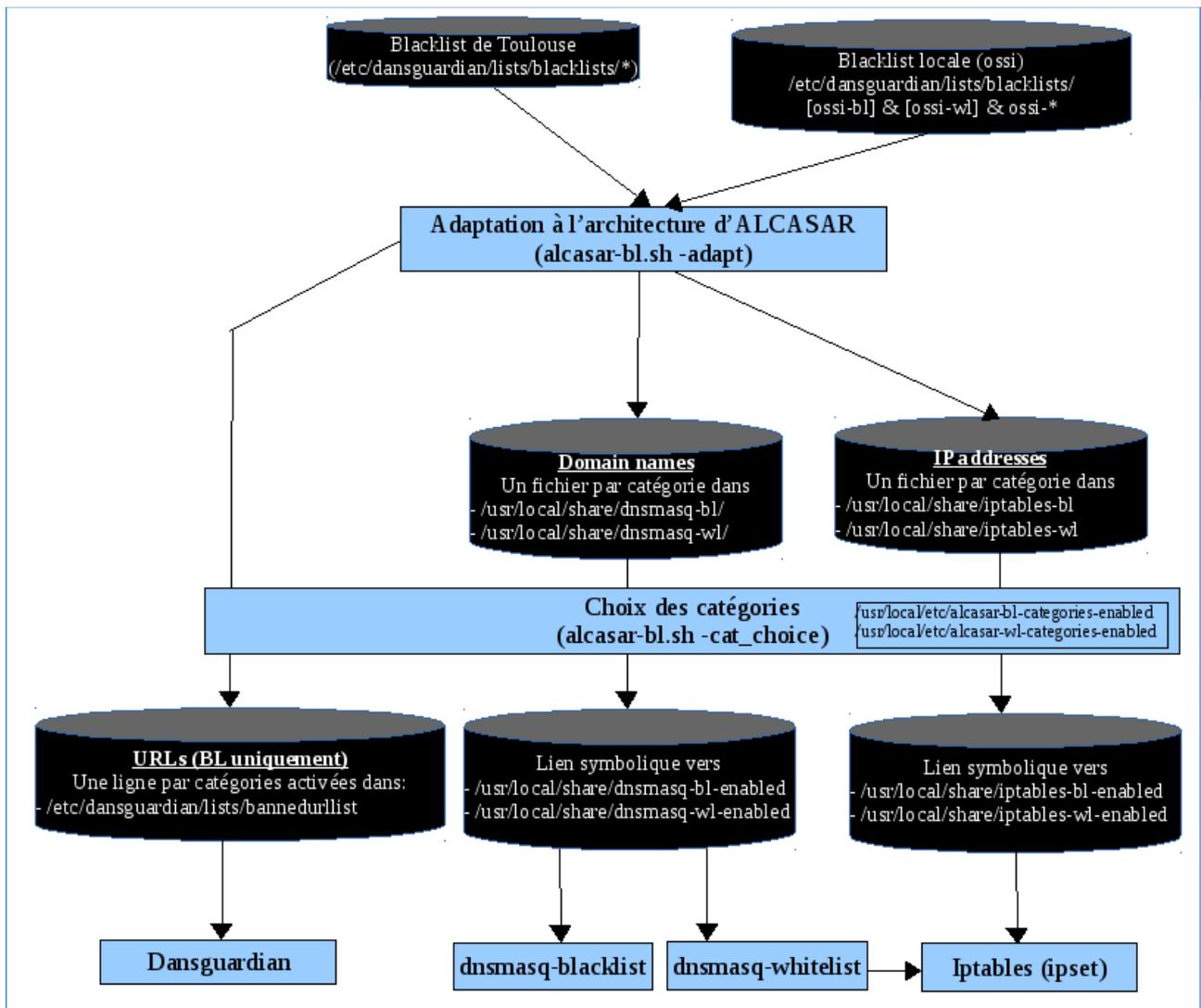


Illustration 7: Traitement de la liste de Toulouse

Il est maintenant possible d'importer manuellement un fichier contenant des IP/nom de domaine avec la whitelist.

Les fichiers à importer depuis l'ACC se trouvent dans le dossier </etc/dansguardian/lists/blacklist/>.

Via l'ACC, il est possible de désactiver/supprimer les fichiers ajoutés à la main (ossi). De plus lorsqu'on clique sur ces fichiers, on peut voir les 10 premiers noms de domaine ou IP afin de permettre une meilleure ergonomie.

On peut aussi mettre à jour la Blacklist avec le protocole rsync (remote synchronization, fonctionnant sur le port 873). Cette fonctionnalité est désactivée par défaut, car elle reste en bêta. En l'activant depuis l'ACC, le crontab suivant va se lancer :

```
0 */12 * * * /usr/local/bin/alcasar-bl.sh --update-cat
```

Le script se lancera toutes les 12 heures afin de lire le fichier si la case est cochée, le fichier renseigne le nom de la catégorie à mettre à jour ainsi que le lien des catégories rsync :

`/usr/local/etc/update-cat.conf` qui se présente sous la forme : “\$categorie \$url”

Ainsi, nous pouvons rajouter nos propres URLs afin de mettre à jour des catégories existantes.

Une nouvelle catégorie a été rajoutée : celle des noeuds de Tor (disponible ici <https://www.dan.me.uk/torlist/>). Cette catégorie est désactivée par défaut et permet d'être en accord avec la PSSI des établissements scolaires. Dans une amélioration future, on pourra autoriser uniquement la connexion vers ses nœuds afin de forcer l'anonymisation des usagers.

## 2.3 Améliorations en développement de ALCASAR 3.0b

En parallèle avec le développement de la version 3.0 officielle, j'ai dû implémenter plusieurs améliorations qui verront le jour dans une version future d'ALCASAR.

Dans cette version, un attribut de la BDD radius dans la table radcheck sera utilisé pour le filtrage. Voici la comparaison de cet attribut entre la version 2.9.2 et la version 3.0b.

N° bit	1	2	3	4	5	6	7	8
valeur	<i>vide</i>	<i>vide</i>	<i>vide</i>	<i>vide</i>	<i>vide</i>	WL	BL	HAVP

*Filter-Id dans la version 2.9.2*

N° bit	1	2	3	4	5	6	7	8
valeur	Profile 1	Profile 2	Profile 3	Avertir ?	<i>vide</i>	WL	BL	HAVP

*Filter-Id dans la version 3.0b*

Comme vous pouvez le constater, j'ai pris 4 octets de l'attribut Filter-Id. Dans la suite du document, nous allons voir à quoi correspond la valeur de ces nouveaux octets. Nous verrons aussi les futures améliorations d'ALCASAR.

### 2.3.1 Filtrage protocole / utilisateur

Mission:

La fonctionnalité de filtrage des protocoles de l'ACC (présente dans Filtrage → Protocoles) s'applique sur tous les utilisateurs. Il faut maintenant pouvoir l'appliquer sur chaque utilisateur du réseau de consultation.

À faire :

Sur ALCASAR 2.9, lorsque cette fonctionnalité est activée, seul le protocole HTTP est autorisé par défaut. Tous les autres protocoles sont bloqués. Un fichier par défaut est

chargé contenant une liste de protocole non exhaustive. Ce fichier se situe dans [/usr/local/etc/alcasar-services](#). Les fichiers PHP permettant sa modification sont `protocols_filter.php` et `protocols_filter2.php` présent dans le dossier [/var/www/html/acc/admin/](#).

Les règles de pare-feu [/usr/local/bin/alcasar-iptables.sh](#) peuvent être traitées en fonction de chaque ipset. J'ai créé quatre ipset afin de dissocier les quatre profils différents :

Les profils correspondent à :

- profil 0 : pas de filtrage
- profil 1 : http/s
- profil 2 : http/s, pop3/s, imap/s,ftp, ssh/sftp
- profil 3 : personnalisable

Le profil 3 est personnalisable via l'ACC dans la section "Network protocols filter". L'administrateur sera libre d'ajouter/supprimer ses propres protocoles.

L'implémentation de cette solution se déroulera de la façon suivante :

- Création et traitement d'un ipset pour chacun des profils (avec les règles iptables)
- Sélection des profils dans la création et l'édition de l'utilisateur.
- Le profile est codé sur les 3 premiers bits du `filter_id` de l'utilisateur (cf introduction 2.3).

#### Problèmes rencontrés :

Vous trouverez ci-après l'ancienne solution qui a été abandonnée, car elle ne correspondait pas aux attentes de la mission. En effet le filtrage devait se faire par utilisateur, en sélectionnant un profile de filtrage protocole particulier. Or ici, on filtre les protocoles selon les caractéristiques ci-dessous:

- Tout le monde (ipset : **users\_list**)
- aucun filtrage (ipset : **not\_filtered**)
- filtré par l'antivirus (ipset : **havp**)
- antivirus+whitelist (ipset : **havp\_wl**)
- antivirus+blacklist (ipset : **havp\_bl**)

Une fois cette fonctionnalité implémentée, l'ACC ressemblait à cela :

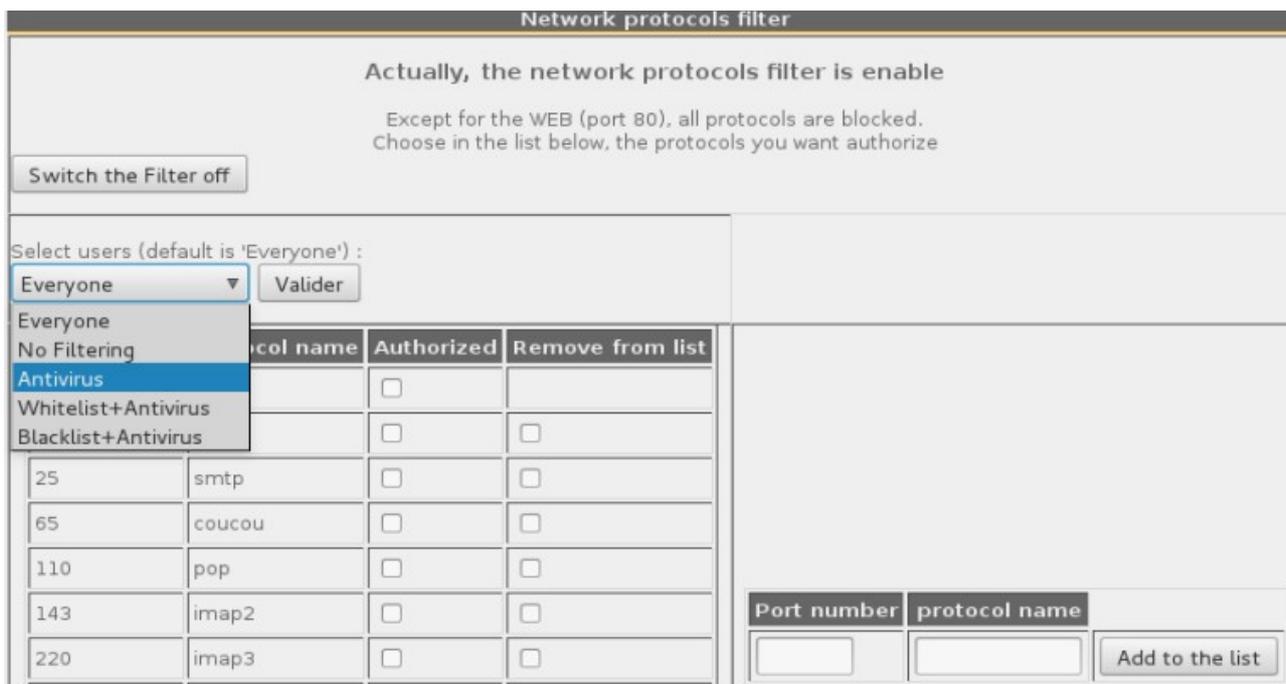


Illustration 8: Filtrage protocole par type de filtre

## 2.3.2 Vérifier l'état de la fenêtre status.php

### Mission :

La fenêtre status.php utilise des fonctions non supportées par certains navigateurs (Safari par exemple). Il s'agit de la fonction *loadbeforeunload*. Cette fonction permet de détecter la fermeture du navigateur et d'exécuter ainsi des instructions.

### Solution:

Afin de trouver une solution générale pour tous les navigateurs, j'ai décidé de modifier le fichier status.php. En effet, cette fenêtre va pointer sur un iframe (still\_connected.php) qui s'actualisera toutes les 3 minutes.

Cette nouvelle page va écrire dans le fichier /tmp/current\_users.txt l'IP de l'utilisateur connecté. De ce fait, ce fichier nous informera des utilisateurs ayant la fenêtre status.php ouverte.

Le script alcasar-watchdog.sh va vérifier deux fois si l'utilisateur est connecté en vérifiant si son IP est présent dans ce fichier.

- 1er passage : Prendre les adresses IP connectées qui ne sont pas présentes dans le fichier '/tmp/current\_users.txt'. Mettre le résultat dans '/tmp/watchdog.txt'
- 2ème passage : Pour chaque IP de '/tmp/watchdog.txt', déconnecter les utilisateurs.

En ce qui concerne les autorisations par @MAC, nous avons décidé de ne pas les prendre en compte et ainsi ne pas les déconnecter du réseau de consultation.

### 2.3.3 Génération du rapport d'imputabilité

Suite à mon entretien avec Jean-François BELLANGER, le RSSI du commissariat de Police de Laval, nous nous sommes mis d'accord sur un cahier des charges concernant la génération des journaux d'imputabilité. Cette mission doit remplir les objectifs suivants:

- La corrélation entre les utilisateurs et leurs connexions (journaux d'imputabilité).
- Pour générer ce document, nous avons besoin du nom du demandeur, de la raison ainsi que d'un mot de passe afin de protéger le document.
- Les utilisateurs du réseau de consultation doivent être avertis de la création de ce document lors de leur prochaine connexion.

#### 2.3.3.1 *Corrélation des informations*

Pour corréler les informations entre les utilisateurs et leurs connexions, nous devons interroger la table 'radacct' de la base de données radius afin d'y extraire la valeur des attributs :

- username : Nom de l'utilisateur du réseau de consultation.
- Callingstationid : Son adresse MAC
- framedipaddress : Son adresse IP
- acctstarttime : la date d'ouverture de sa session
- acctstoptime : la date de fermeture de sa session
- acctinputoctets : La quantité de données envoyées
- acctoutputoctets : La quantité de données téléchargées
- acctterminatecause : La cause de la fermeture de la session (soit l'utilisateur s'est déconnecté volontairement, soit il s'est déconnecté à cause de l'administrateur ou du temps de session qui est arrivé à échéance (par exemple)).

Nous pouvons corréler ces informations avec les journaux générés par la sonde NetFlow. En effet, nous disposons de la date de fermeture et d'ouverture de la session ainsi que de l'adresse IP attribuée lors de cette connexion. Pour se faire, nous allons interroger la sonde NetFlow à l'aide de la commande suivante :

```
nfdump -O tstart -R DOSSIER_LOG -t YYYY/MM/DD.HH-YYYY/MM/DD.HH (intervalle de temps) -o "fmt:FORMAT_DE_LA_REPONSE"
```

Dans notre format de réponse, nous allons extraire l'adresse IP et le port source, de même

pour la destination ainsi que la date de cette connexion. Ainsi nous allons lister toutes les connexions de chaque utilisateur pour un intervalle donné. Chaque ligne de connexion est numérotée afin de faciliter la lecture de ce rapport.

### 2.3.3.2 Génération du document

Le document PDF généré est protégé par un mot de passe dans une archive au format 'zip'. Pour créer ce fichier PDF, nous allons dans un premier temps générer une page HTML avec du CSS et du Javascript. Cette page sera ensuite convertie en fichier PDF à l'aide de l'outil *open source* suivant : wkhtmltopdf. Cet outil utilise le moteur de rendu Qt WebKit, vous pouvez consulter le site du projet pour plus d'information : <http://wkhtmltopdf.org/>.

Ce document peut être généré depuis l'ACC en remplissant les champs 'mot de passe', 'le nom du demandeur' et 'la raison de cette demande'. Il est possible de sélectionner un intervalle de date ou encore de générer ce rapport depuis une date spécifique. Enfin, à chaque génération du document, un historique est présent récapitulant toutes les demandes effectuées précédemment.

Date	User	Reason	IP address
2016-06-27 09:35:14	azdazd	azdazd	192.168.182.5
2016-06-27 10:00:23	coucou	diazjdilazj	192.168.182.5
2016-06-27 10:00:34	aaaaa	dzdazdzdzzzz	192.168.182.5

Illustration 9: Interface génération des journaux d'imputabilité

Bien sûr, cette pratique doit respecter la loi française, c'est pour cela qu'il faut prévenir les utilisateurs de la génération de ce document lors de leur prochaine connexion.

### 2.3.3.3 Avertir les utilisateurs

Il faut donc marquer ces utilisateurs à l'aide de l'attribut Filter-Id de la table 'radacct' de la BDD radius. En effet, cet attribut, stocké sur 8 caractères, nous informe quel est le type de filtrage à utiliser pour un utilisateur donné. Lorsque le 4ème caractère est à '1',

l'utilisateur se fera intercepter par une page informant ainsi que son historique de connexion a été consulté.

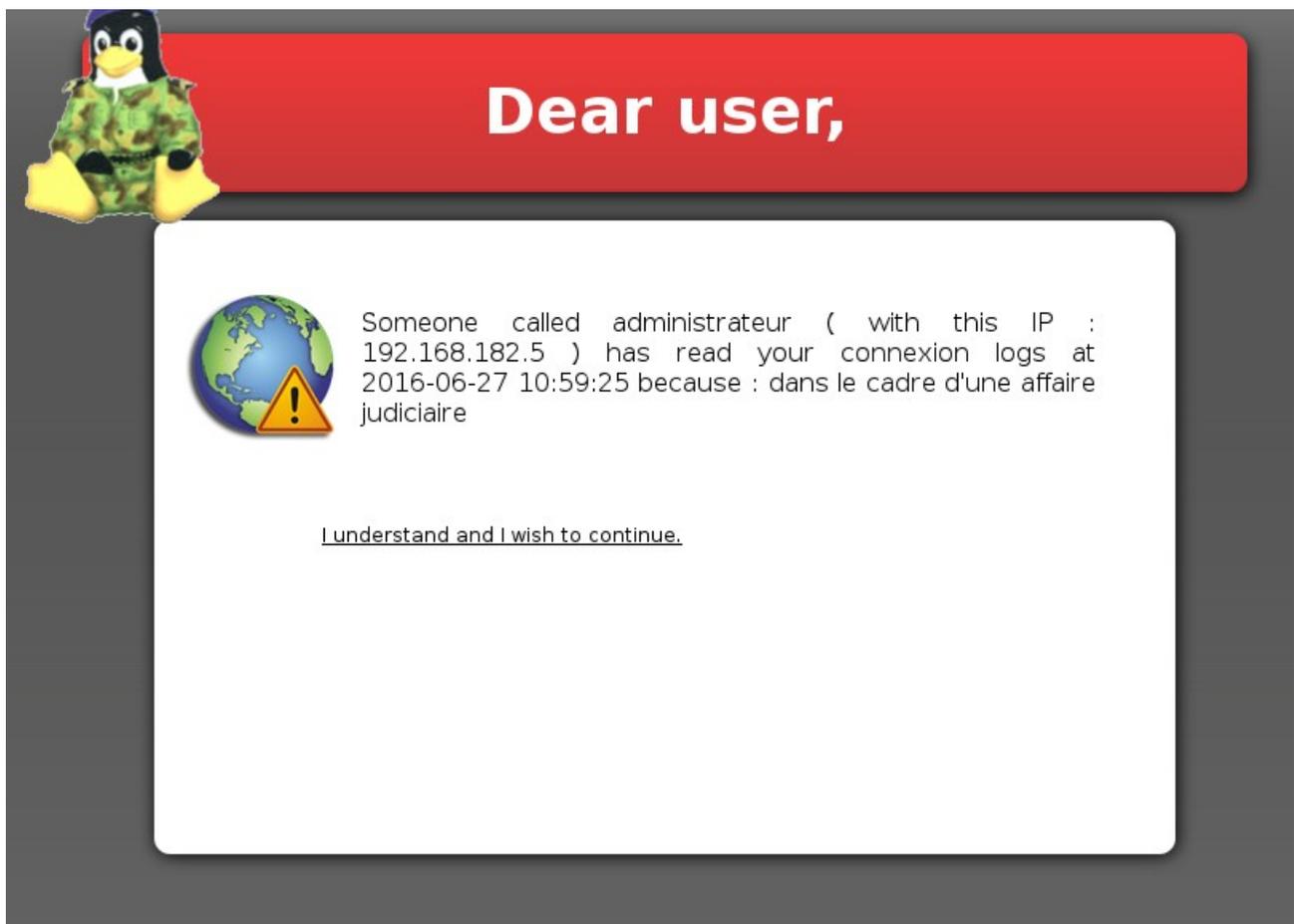


Illustration 10: Avertissement destiné aux utilisateurs

## 2.3.4 Rapport d'activité hebdomadaire

### Mission :

Le but étant ici de générer un rapport d'activité hebdomadaire compréhensible. Chaque dimanche à 5h35 du matin, un script s'exécutera afin de réunir les informations nécessaires pour la génération de ce document.

### Solution :

Le rapport se génère à l'aide d'un crontab que voici :

```
35 5 * * 0 root $DIR_DEST_BIN/alcasar-activity_report.sh
```

Le rapport contient les éléments suivants :

- Un tableau de bord indiquant les informations du système et de l'ALCASAR. Il renseigne aussi les logiciels mis à jour pendant la semaine :

## Logiciels mis à jour (cette semaine)

NOM	DATE	VERSION
lib64php5_com mon5	2016-07-08 10:38:22	5.6.23
php-curl	2016-07-08 10:38:22	5.6.23
php-dom	2016-07-08 10:38:22	5.6.23
php-ftp	2016-07-08 10:38:22	5.6.23
php-json	2016-07-08 10:38:22	5.6.23
php-mysql	2016-07-08 10:38:22	5.6.23
php-mysqli	2016-07-08 10:38:22	5.6.23
php-mysqldb	2016-07-08 10:38:22	5.6.23
php-openssl	2016-07-08 10:38:22	5.6.23
php-pdo	2016-07-08 10:38:22	5.6.23
php-pdo_sqlite	2016-07-08 10:38:22	5.6.23
php-xmlreader	2016-07-08 10:38:22	5.6.23

*Illustration 11: Logiciel mis à jour de la semaine*

- Un camembert informe les sites bloqués par la blacklist depuis l'installation et de la semaine :
  - création des logs dnsmasq-blacklist :
    - renseigner dans le fichier `/etc/dnsmasq-blacklist` :  
`log-queries`  
`log-facility=/var/log/dnsmasq/dnsmasq-blacklist.log`
  - application du logrotate
    - renseigner dans le fichier `/conf/logrotate.d/dnsmasq-blacklist` :  
`/var/log/dnsmasq/dnsmasq-blacklist.log {`  
`missingok`  
`notifempty`  
`rotate 52`  
`weekly`  
`dateext`  
`sharedscripts`  
`postrotate`  
`/usr/bin/systemctl restart dnsmasq-blacklist`

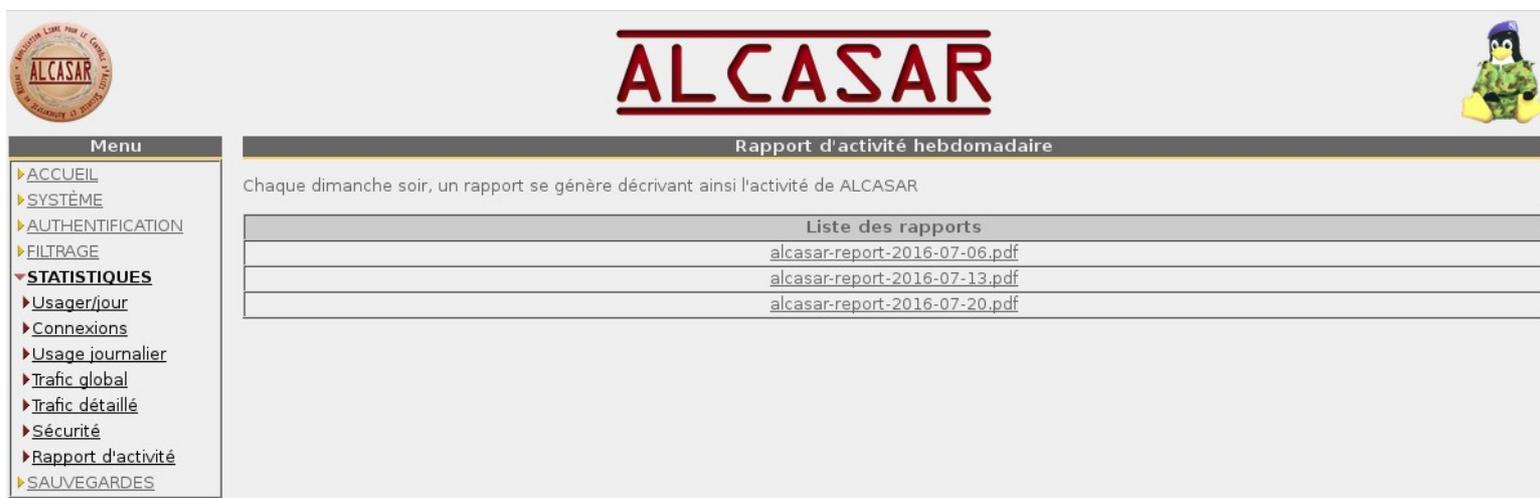
*endscript*

}

- Il est possible de voir les catégories bloquées par ALCASAR y compris celle ajoutée à la main par l'administrateur réseau (OSSI).
- Menaces virales : nous avons un graphique indiquant le nombre de menaces bloquées par l'antivirus HAVP par semaine.
  - Les menaces bloquées par l'antivirus de proxy sont présentes dans le fichier : `/var/log/havp/access.log`
- Statistiques de volumétrie des connexions : insertion du tableau de l'ACC, disponible dans le menu 'STATISTIQUES' → 'usage journalier'.
  - Pour se faire, il faut accéder à l'ACC en créant un utilisateur temporaire avec `htdigest`. Ensuite, nous téléchargeons et nettoyons la page de statistique afin d'obtenir le tableau.
- TENTATIVE DE CONNEXION : Indique les connexions autorisées et interdites par ALCASAR. On renseigne aussi le `fail2ban`.
  - Pour connaître les connexions autorisées et interdites, il faut requêter l'attribut 'acctterminatecause' de la table `radacct` de la base de données `radius`.
  - Pour connaître le nombre de `fail2ban` de ALCASAR, il faut lire le fichier de journal suivant : `/var/log/fail2ban.log`

### 2.3.4.1 Consultation via l'ACC

Une nouvelle catégorie a été ajoutée dans le menu "STATISTIQUES" de l'ACC permettant de consulter tous les PDF générés toutes les semaines depuis l'installation de la machine.



The screenshot shows the ALCASAR web interface. At the top center is the ALCASAR logo. On the left is a navigation menu with the following items: ACCUEIL, SYSTÈME, AUTHENTIFICATION, FILTRAGE, STATISTIQUES (highlighted), Usager/jour, Connexions, Usage journalier, Trafic global, Trafic détaillé, Sécurité, Rapport d'activité, and SAUVEGARDES. The main content area is titled 'Rapport d'activité hebdomadaire' and contains the text: 'Chaque dimanche soir, un rapport se génère décrivant ainsi l'activité de ALCASAR'. Below this text is a table titled 'Liste des rapports' with three rows of report links:

Liste des rapports
<a href="#">alcasar-report-2016-07-06.pdf</a>
<a href="#">alcasar-report-2016-07-13.pdf</a>
<a href="#">alcasar-report-2016-07-20.pdf</a>

Illustration 12: Affichage dans l'ACC

Un exemple de rapport d'activité a été joint en annexe 1.

### 2.3.5 Création d'un RPM afin d'installer wkhtmltopdf

#### Mission:

Afin de convertir le HTML en PDF, nous utilisons wkhtmltopdf. Il s'agit d'un outil *open source*. Cet outil utilise le moteur de rendu Qt WebKit, vous pouvez consulter le site du projet pour plus d'information : <http://wkhtmltopdf.org/>. Cependant il n'existe pas de RPM compatible avec mageia. Le but est de *packager* wkhtmltopdf afin d'indiquer aux fichiers sources le bon emplacement. Ici, il n'y a pas besoin de compilation.

#### Solution :

Il faut lire le fichier d'instructions présent sur le SVN : `svn/alcasar/trunk/rpms/rpm-build-howto`. Le fichier SPEC concernant wkhtmltopdf est présent dans le même répertoire.

Dans ce fichier SPEC, on y indique la description de notre paquet (résumé, nom, version ...) ainsi que les instructions à effectuer pour s'installer dans l'arborescence de la machine ciblée (création de dossiers, copie des fichiers).

### 2.3.6 Option NTP du DHCP

#### Mission :

Le serveur NTP doit être spécifié dans la requête DHCP afin d'indiquer à la machine cliente son adresse IP.

#### Solution :

En lisant la RFC des options du DHCP (RFC 1533), il est possible de renseigner dans la requête du serveur DHCP l'adresse IP du serveur NTP. Le serveur DHCP est Coova-Chilli, il est donc nécessaire de le compiler avec le flag '--enable-dhcpopt' afin de prendre en compte les options du DHCP.

Ensuite dans le fichier de configuration, il suffit de rajouter le champ 'dhcpopt' suivi de sa valeur en hexadécimal. L'option 42 (serveur NTP) du DHCP s'écrit de la façon suivante avec l'IP d'ALCASAR 192.168.182.1:

Code de l'option	Taille en octet	Adresse IP 1	Adresse IP 2	Adresse IP 3	Adresse IP 4
42 → 2a(h)	4 → 04(h) (car une seule @IP)	192 → c0(h)	168 → a8(h)	182 → b6(h)	1 → 01(h)



## 2.4 Missions annexes de ALCASAR

- Problème de mise à jour de la blacklist : La taille de certains Top-level domain (ex : .fr, .com, .net ...) était trop grande. Tous les TLDs dépassant les 3 caractères étaient tronqués. Ce qui impliquait des problèmes d'importation de la blacklist. Maintenant, la taille maximale (actuellement fixée à 18 caractères) du TDL est mise à jour lors de l'importation de la blacklist via le site de l'IANA.
- Le problème suivant a été corrigé : si l'utilisateur décoche toutes les catégories de la WL/BL, la whitelist ne se charge plus et impossible de recocher d'autres catégories.
- Modification de l'expression régulière de la création d'utilisateurs pour mettre des accents (exemple : Rémi).
- Tous les binaires du répertoire `/usr/local/sbin` ont été déplacés vers `/usr/local/bin`.
- Les dossiers `dbx`, `oracle`, `pg`, `sqlreplay` dans `/web/pass/sql/drivers` ont été supprimés. ALCASAR utilise le SGBD MariaDB et n'utilise que mysql.
- Une confirmation en JavaScript a été ajoutée concernant la purge de la base de données de ALCASAR.
- Lors de la checklist d'ALCASAR du protocole LDAP, nous nous sommes rendu compte que les mots de passe soumis à Coova-Chilli étaient tronqués à partir du 15ème caractère. En se renseignant sur internet, nous avons appliqué un patch permettant de corriger ce problème.
- Certains *drivers* réseau sont passés en *non-free*. Le média doit être ajouté afin d'installer le RPM 'kernel-firmware-nonfree'.

## 2.5 CheckmyHTTPS

CheckmyHTTPS est un projet qui a été développé en parallèle avec le PST ALCASAR. Les créateurs de ce projet sont Monsieur REY, Hugo MEZIANI et moi même. L'idée est venue du fait qu'il fallait un nouveau système d'interception pouvant intercepter les requêtes HTTP et HTTPS sans compromettre le HTTPS sur ALCASAR. De ce fait, nous voulions étudier le SSL afin de comprendre comment fonctionne une attaque man-in-the-middle SSL et de créer un outil permettant de mettre en évidence ce type d'attaque.

### 2.5.1 CheckmyHTTPS SDK

#### Mission:

CheckmyHTTPS est une extension Firefox permettant de vérifier si une connexion HTTPS est sécurisée en comparant le certificat serveur reçu par le client avec celui reçu par l'équipement réseau contrôlé se situant sur Internet. Si les certificats ne correspondent pas, la connexion peut être compromise.

J'ai reçu un mail récemment me disant de mettre à jour le fonctionnement de l'extension. En effet, l'extension étant basé sur XUL, il fallait le migrer sur le SDK de Firefox afin qu'il soit compatible pour les versions futures du navigateur.

### Solution

L'extension fonctionne sur le SDK. Il diffère légèrement de l'ancienne version, il est impossible de faire un clic droit pour afficher les détails de l'*addon*. J'ai donc décidé d'alerter l'utilisateur si un MITM SSL était présent via une notification de Firefox. Si l'utilisateur clique sur cette notification, un onglet s'ouvre affichant ainsi les détails.

Je me suis connecté sur l'IRC de Mozilla afin de me renseigner sur comment mettre à jour l'extension XUL avec le SDK.

Server :irc.mozilla.org  
port:6667  
channel :#amo-editors

Il suffit de rajouter le champ « id » dans le fichier package.json :

```
"id" : "info@checkmyhttps.net",
```

Les avantages du SDK sont les suivants :

- L'*addon* se positionne directement dans la barre d'outils de Firefox.
- L'installation ne nécessite pas de redémarrage.
- L'*addon* est compatible avec toutes les versions à jour de Firefox (*Firefox Extended Support Release*)

Le nom "checkmyhttps" est disponible dans l'URL, contrairement à l'ancien qui était "check-my-https". Cela posait problème lorsque quelqu'un souhaitait retrouver notre *addon* sur le *market place* de Mozilla. En effet, il devait taper « check-my-https » pour le retrouver.

Une description française/anglaise ainsi que de nouvelles images sont disponibles sur le *market place* de Firefox. L'utilisateur est mieux informé des informations qui sont envoyées au serveur. Un schéma simple est disponible afin d'expliquer aux utilisateurs le fonctionnement de l'extension :

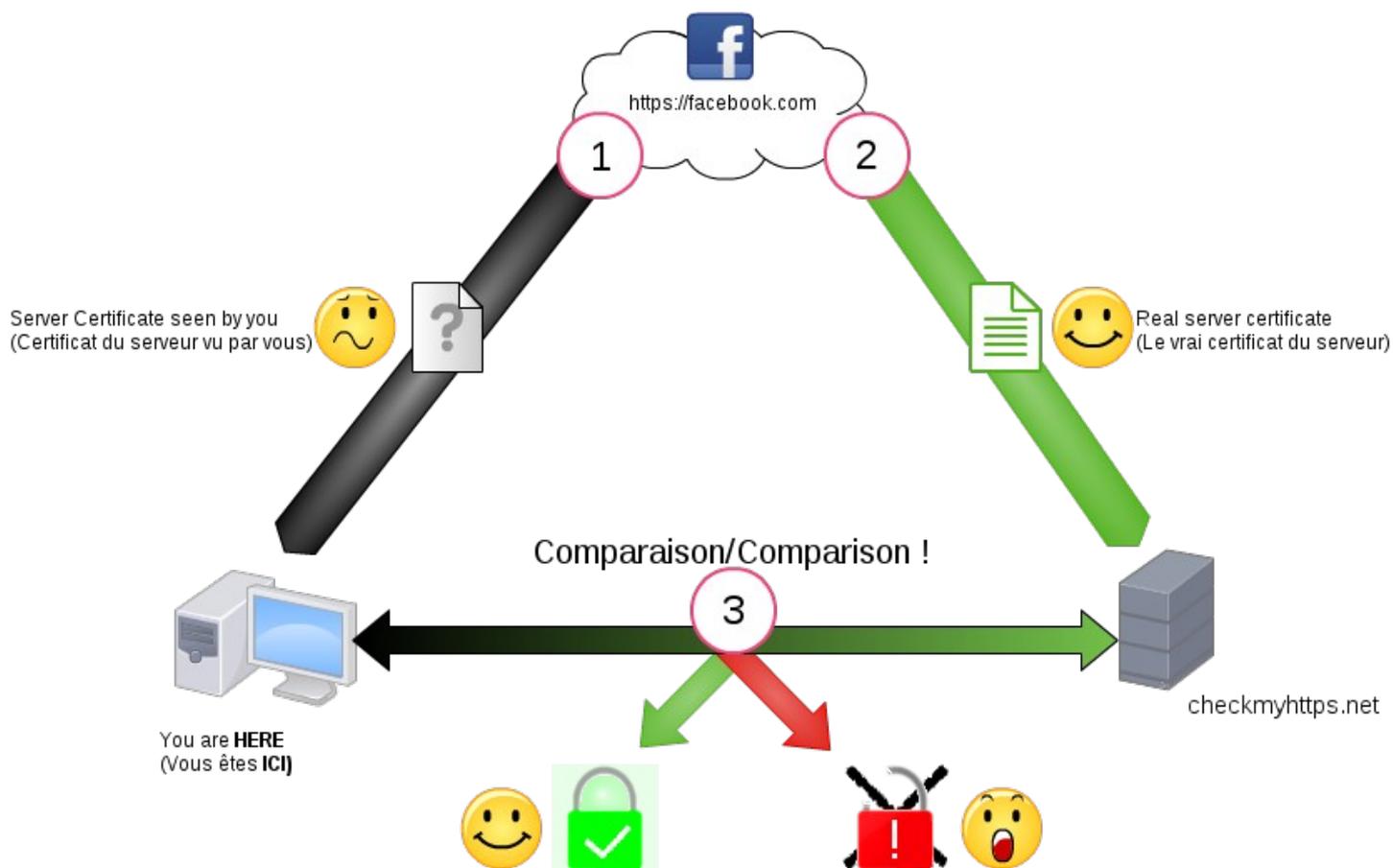


Illustration 14: Explication grand public de checkmyHTTPS

Le site checkmyhttps a été mis à jour. Suite à la demande d'un daltonien par mail, la forme des cadenas a été modifié.

-  La connexion HTTPS n'est pas interceptée
-  La connexion HTTPS est interceptée !
-  Le site CheckMyHTTPS est injoignable ...
-  Le test est en cours

Illustration 15: Mise à jour des icônes de checkmyhttps

Il n'est pas évident de concevoir les difficultés que peut rencontrer ce public-là pour ce qui est de l'interprétation des icônes lors du développement du logiciel.

## 2.5.2 Présentation/conférence

La présentation de checkmyhttps a été modifiée afin d'y inclure les images ci-dessus. De plus, un démonstrateur a été mis en place afin de simuler un MITM SSL. Cette attaque permet de visualiser en clair le trafic HTTPS en imposant à la victime un faux certificat SSL.

J'ai présenté le projet avec Hugo Meziani à l'ESE 2016 (ESIEA Secure Edition) . L'ESE est un événement concernant la sécurité informatique organisé par l'ESIEA. Les entreprises, les forces de l'ordre ainsi que les étudiants peuvent tenir des conférences.

J'ai aussi présenté le projet devant des PDG sarthois ainsi que des DSI. J'ai eu beaucoup de retours positifs concernant ces présentations.

Cet exercice m'a appris à adapter mon discours pour un public n'ayant pas forcément de notions techniques.

## 2.5.3 Reprise du projet

Dans l'optique d'une éventuelle reprise du projet, j'ai donc décidé de mettre à jour le code de l'*addon* en le développant sous le SDK. Le serveur hébergeant le site a été mis à jour, nettoyé :

- Nettoyage du serveur (*virtual hosting*, droits sur les fichiers, réorganisation des fichiers)
- Les identifiants pour le GitHub, le *market place* et le serveur ont été transmis à M. REY
- Une sauvegarde du site checkmyhttps.net a été effectuée et remise à M. Rey contenant :
  - le dossier `/etc/httpd` (toute la conf de httpd)
  - les certificats et la clé privée
  - le dossier `/var/www/html/checkmyhttps`
- Ajout d'un fichier CHANGELOG et TODO au GitHub de checkmyhttps
- Parution des articles parlant de Checkmyhttps :
  - L'e-bdo de l'ESIEA.
  - Écrit par M. Rey : Global Security, Informatique News.
  - Rédaction de l'article « Programmez » écrit par Hugo MEZIANI et moi même.
- Amélioration/Modification de l'*addon*
  - Le nom CNS a été changé en « CNS-CVO »
  - Utilisation de l'*addon* dans la navigation privée

"permissions": {"private-browsing": true},

- utilisateur averti lors d'un test sur une IP privée

regexp : « /(^https:\W127\.)(^https:\W10\.)(^https:\W172\.1[6-9]\.)(^https:\W172\.2[0-9]\.)(^https:\W172\.3[0-1]\.)(^https:\W192\.168\.)/ »

- Remarques

- Les éditeurs d'antivirus peuvent être des autorités de certification (ex:Symantec sur le site de paypal.com). On peut imaginer que si Norton Antivirus active l'option de scan HTTPS, même en regardant l'émission du certificat imposé par l'antivirus, il est impossible de savoir s'il y a un MITM SSL local. Ce qui renforce l'utilité du projet !

### 3 Conclusion

Après avoir effectué mes quatre mois de stage dans le laboratoire CVO à Laval, j'ai fini par remplir les objectifs fixés avec la sortie de la version 3 d'ALCASAR. Ce stage qui est dans la continuité de mon PST 4A m'a permis d'utiliser les connaissances que j'ai vues en cours afin de les appliquer dans un projet conséquent.

Le fait de travailler pour ce projet libre m'a permis de mieux comprendre le fonctionnement de son développement. En effet, le projet respecte une hiérarchie ainsi qu'une méthode de réalisation de projet qui est la méthode agile.

La communauté de ALCASAR est active et dispose d'un forum pour venir en aide aux autres ou corriger des problèmes qui contribuent constamment à l'évolution de ce projet. J'ai beaucoup apprécié la philosophie du projet ALCASAR. En effet celui-ci respecte la loi française et surtout la vie privée des usagers.

En plus du côté humain que ce projet peut avoir, j'ai pu enrichir mes connaissances en réseau. Par exemple, la manipulation des règles de pare-feu ou encore l'étude de nouveaux protocoles réseau comme le LDAP, HTTP/S, DHCP ou NTPD. Par ailleurs, j'ai renforcé mes compétences en programmation en développant en Bash et en PHP.

Les diverses missions proposées pendant ce stage, allant de l'intervention dans la gendarmerie à la demande du RSSI du commissariat de la ville de Laval, m'ont permis de découvrir de nouvelles expériences.

Enfin, c'était un grand plaisir de participer à ce projet et bien entendu je resterai disponible pour la correction de problèmes qui pourraient survenir sur la version 3 d'ALCASAR.

## 4 Annexe



## 4.1 Exemple de rapport d'activité de la machine ALCASAR

### Rapport d'activité de l'ALCASAR-esiea

Date de création 2016-07-20

#### Configuration de l'ALCASAR

Organisme	Installation	Version	@IP publique	@IP privée	Passerelle	DNS1	DNS2
esiea	05 mai 2016 - 19h27	3.0b2	10.0.2.5/24	192.168.182.1/24	10.0.2.1	172.16.0.1	172.16.0.1

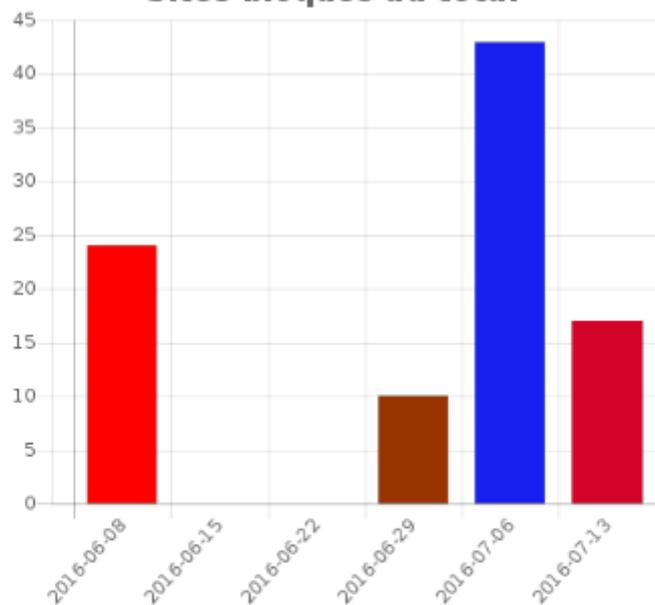
#### Configuration du Système

Hostname	Version	Last reboot	MAJ Antivirus	MAJ Blacklist
alcasar.test	4.1.15-server-2.mga5 [ x86_64 ]	2016-07-20 15:33	2016-05-05 19:30:16	14 January 2016

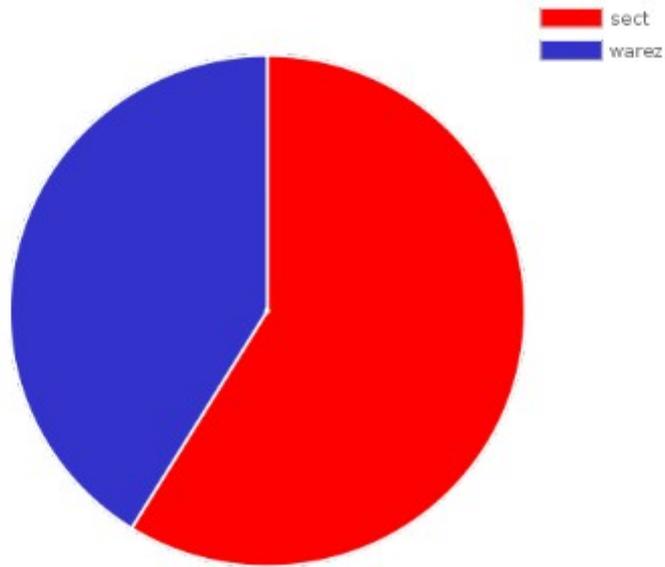
#### Logiciel mis à jour (cette semaine)

NOM	DATE	VERSION
Pas de RPM mis à jour cette semaine		

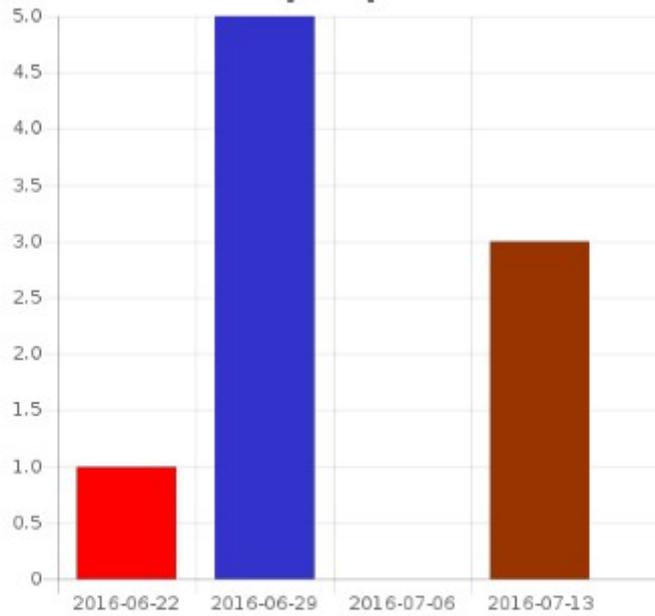
#### Sites bloqués au total



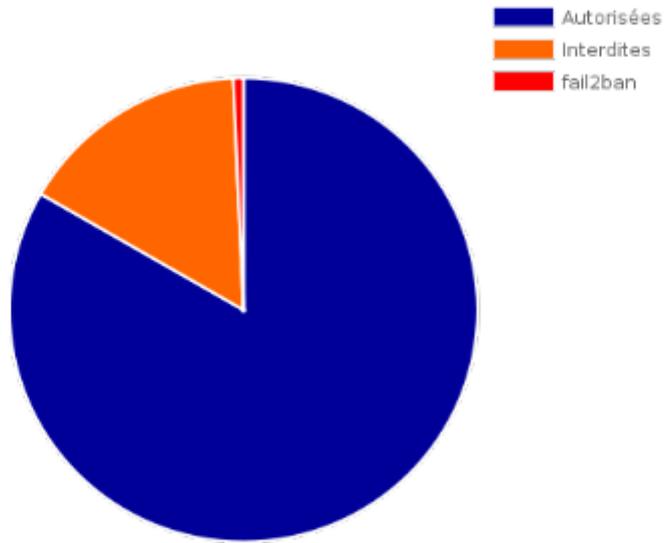
### Sites bloqués cette semaine



### Menaces bloqués par l'antivirus



## Connexions des utilisateurs



## Statistiques volumétrie connexions

### Analyse journalière

date	sessions	temps d'utilisation total		downloads	
2016-07-13	0%	00:00:00	0%	0.00 KBs	0%
2016-07-14	0%	00:00:00	0%	0.00 KBs	0%
2016-07-15	0%	00:00:00	0%	0.00 KBs	0%
2016-07-16	0%	00:00:00	0%	0.00 KBs	0%
2016-07-17	0%	00:00:00	0%	0.00 KBs	0%
2016-07-18	0%	00:00:00	0%	0.00 KBs	0%
2016-07-19	0%	00:00:00	0%	0.00 KBs	0%
2016-07-20	0%	00:00:00	0%	0.00 KBs	0%
2016-07-21	0%	00:00:00	0%	0.00 KBs	0%

### Récapitulatif journalier

	sessions	temps d'utilisation total		downloads	
maximum		00:00:00		0.00 KBs	
moyenne	0	00:00:00		0.00 KBs	
récapitulatif	0	00:00:00		0.00 KBs	

## 5 Liste des illustrations

### Index des illustrations

Illustration 1: Logo (CVO) <sup>2</sup> .....	6
Illustration 2: Logo ALCASAR.....	7
Illustration 3: Schéma réseau du set-up.....	8
Illustration 4: Modification de la configuration réseau via l'ACC.....	11
Illustration 5: bug hash utilisateur.....	13
Illustration 6: problème avec le safesearch de Qwant.....	16
Illustration 7: Traitement de la liste de Toulouse.....	17
Illustration 8: Filtrage protocole par type de filtre.....	20
Illustration 9: Interface génération des journaux d'imputabilité.....	22
Illustration 10: Avertissement destiné aux utilisateurs.....	23
Illustration 11: Logiciel mis à jour de la semaine.....	24
Illustration 12: Affichage dans l'ACC.....	25
Illustration 13: Mise en évidence de l'option 42 du DHCP.....	27
Illustration 14: Explication grand public de checkmyHTTPS.....	30
Illustration 15: Mise à jour des icônes de checkmyhttps.....	30