



# BROUILLON



## Documentation technique

Projet : Sécurisation des accès Internet	Auteur : rexy and 3abtux with helps by alcasar team
Objet : Documentation technique du projet	Version : 2.7
Mots clés : Portail captif, captive portal, coova, chilli	Date : Mai 2013

## Table des matières

1 - Rappel de l'architecture.....	3
2 - Choix des constituants.....	3
2.1 - La passerelle d'interception.....	3
2.2 - Les autres constituants.....	3
3 - Schémas de principe :.....	4
4 - Fonction « interception / authentification ».....	7
4.1 - la passerelle « coova-chilli ».....	7
4.1.1 - Fonctionnement de l'interception (capture).....	7
4.1.2 - Exception à l'authentification.....	8
4.2 - Le serveur FreeRadius.....	9
4.3 - Base de données des usagers.....	9
4.3.1 - Accès graphique à la base.....	11
4.3.2 - Accès console.....	11
4.4 - Serveur A.D./LDAP externe.....	12
5 - Fonction « traçabilité et imputabilité ».....	12
5.1 - Journalisation principale.....	12
5.2 - journalisation accessoire.....	13
6 - Fonction « filtrage ».....	13
6.1 - Filtrage Réseau.....	13
6.2 - Filtrage de domaines et d'URLs.....	14
6.3 - Antivirus WEB.....	15
7 - Fonction « Interface de gestion ».....	15
8 - Fonction « modules complémentaires ».....	16
8.1 - Import de comptes – Fichier mots de passe.....	16
8.2 - Watchdog.....	16
8.3 - Statistiques.....	16
8.4 - Contournement (by-pass).....	16
8.5 - Load balancing de connections .....	17
8.6 - Re-Horodatage des fichiers journaux .....	17
8.7 - Module de sauvegarde.....	17
8.7.1 - Sauvegarde du système complet.....	17
8.7.2 - Sauvegarde des journaux d'évènements.....	17
8.7.3 - sauvegarde de la base de données.....	18
9 - Annexes.....	18
9.1 - Coova-chilli.....	18
9.2 - Freeradius.....	18
9.3 - Dnsmasq.....	18
9.4 - Parefeu.....	18
9.5 - Dansguardian.....	18
9.6 - Squid.....	19
9.7 - Ulogd.....	19
9.8 - HAVP + Clamav.....	19
9.9 - Distribution Magia et ses dépôts.....	19




# 1 - Rappel de l'architecture

ALCASAR est positionné en coupure entre l'accès Internet et le réseau de consultation. Il permet d'authentifier les usagers, de contrôler les accès, de tracer les connexions effectuées, de protéger le réseau de consultation. Le coeur d'ALCASAR est constitué des éléments traditionnels d'un portail captif : une passerelle d'interception, un serveur d'authentification et une base de données usagers.

## 2 - Choix des constituants

### 2.1 - La passerelle d'interception

La « passerelle d'interception » constitue le chef d'orchestre d'un portail captif. Pour choisir celle qui serait intégrée dans ALCASAR, les passerelles libres suivantes ont été évaluées :

	NoCat 	Talweg	Wifidog 	Chillispot/Coovachilli 
Site WEB	<a href="http://nocat.net">nocat.net</a>	<a href="http://talweg.univ-metz.fr">talweg.univ-metz.fr</a>	<a href="http://dev.wifidog.org">dev.wifidog.org</a>	<a href="http://www.chillispot.org">www.chillispot.org</a> <a href="http://www.coovachilli.org">www.coovachilli.org</a>
Version	Plusieurs versions pour les différents constituants du produit. Dernière mise à jour : 27/02/2005	0.86-R2 (22/03/2007)	1.0.0_m2 (7/10/2005)	1.1 (24/10/2006)
Langage	C	C# sous mono	- C pour le programme principal - module PHP pour le serveur WEB	- C pour le programme principal - module CGI-BIN pour le serveur WEB (PERL ou C)
Description	NoCat est constitué de plusieurs éléments : « NoCatSplash » est le portail, « NoCatAuth » est utilisé pour l'authentification et « Splash Server » est le service permettant de générer les formulaires de connexion des utilisateurs. Le suivi de ce produit a été arrêté en 2005.	Talweg est un portail dont le contrôle d'accès au réseau est géré protocole par protocole. Tous les protocoles utilisables sur Internet ne sont pas encore intégrés.	WifiDog est composé de 2 modules : « Authentification Server » et « WifiDog Gateway ». Le serveur d'authentification doit être installé sur un serveur fixe alors que la passerelle peut être embarquée dans certains équipements réseau compatibles (routeur, passerelle ADSL, etc.).	Chillispot ne constitue que la partie centrale d'une architecture de type portail captif. Il implémente les 2 méthodes d'authentification (UAM et WPA). Il nécessite la connaissance et l'installation des autres services constitutifs du portail captif.

	NoCat	Talweg	Wifidog	Chillispot
<b>Simplicité d'installation</b>				
<b>Infrastructure nécessaire</b>				
<b>Performances &amp; consommation réseau</b>				
<b>Gestion utilisateurs</b>				
<b>Sécurité authentification</b>				
<b>Sécurité communications</b>				
<b>Protocoles supportés</b>				
<b>Crédit temps</b>				
<b>Interface d'administration / Statistiques</b>				

Légende:  
 : Non Disponible.  
 : Plus ou moins.

Bien que cette liste ne soit pas exhaustive, la passerelle « Chillispot » a été choisie. Depuis, elle a été remplacée par le clone (fork) « coova-chilli » dont le développement est plus actif (<http://coova.org/CoovaChilli>). Avant chaque nouvelle version d'ALCASAR, Coova-chilli est récupéré sous forme d'archive compressée. Il est compilé et empaqueté (RPM) spécifiquement pour être intégré à ALCASAR.

### 2.2 - Les autres constituants

Pour couvrir l'ensemble des besoins d'ALCASAR, les produits libres suivants ont été intégrés. Leur choix est principalement dicté par leur niveau de sécurité et leur reconnaissance.

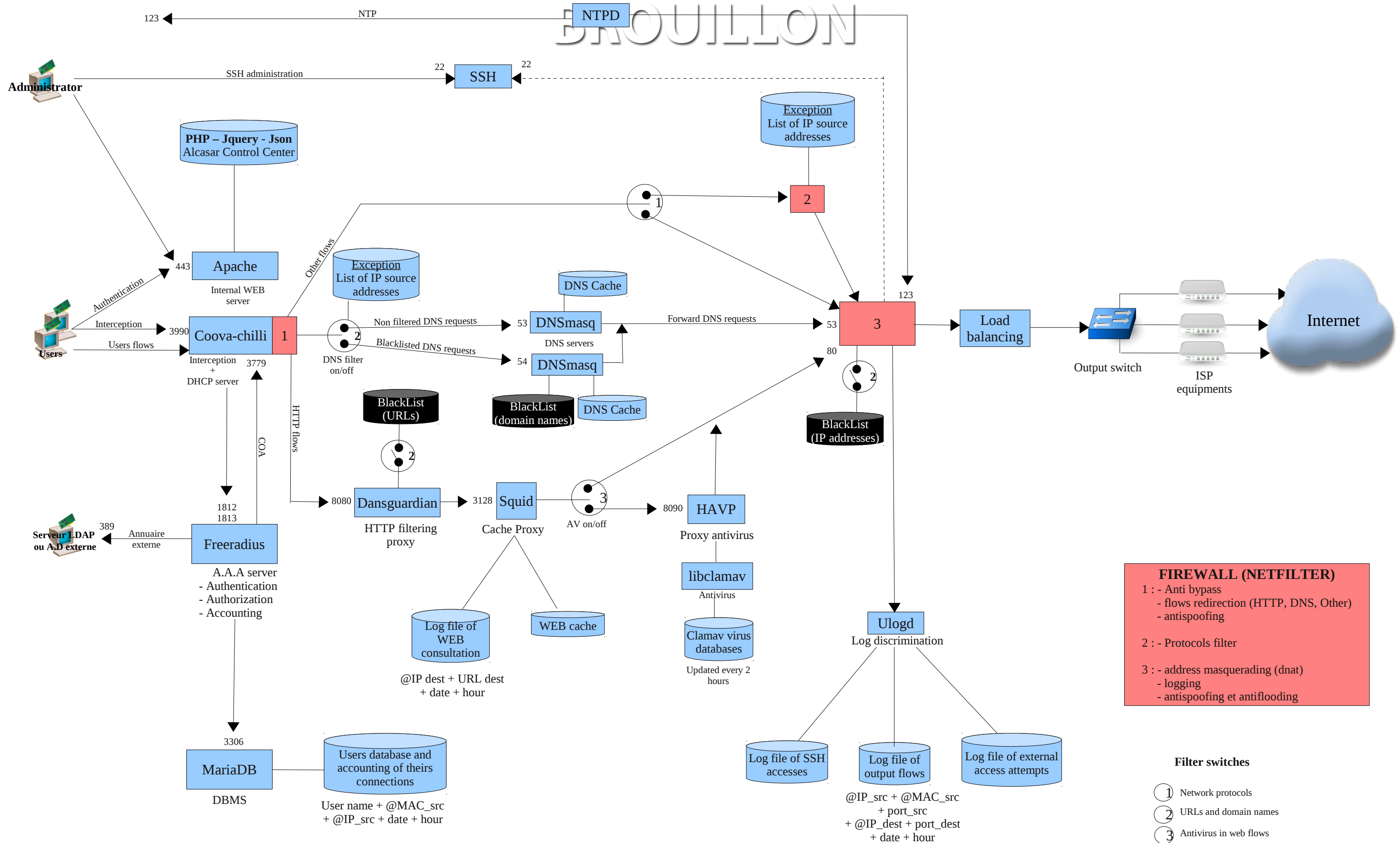
Version d'ALCASAR		<=1.6	1.7	1.8	1.9	2.0	2.1 2.2	2.3 à 2.5	2.6	2.7
Système d'exploitation	Linux Mandriva	2007.0	2009.0	2010.0	2010.0	2010.1	2010.2	2010.2	2010.2	
	Mageia									2
noyau Linux		2.6.17	2.6.27	2.6.31	2.6.31	2.6.33		2.6.33.7		3.4.32
Passerelle d'interception	Coova-chilli	1.0	1.0.12			1.2.2	1.2.5	1.2.8	1.2.9	
DNS	Bind				9.6.1					
	Dnsmasq							2.52-1		2.63
Serveur DHCP (mode bypass)	dhcpd server	3.0.4	3.0.7	4.1.0						
Serveur Web	Apache	2.2.3	2.2.9	2.2.14	2.2.14	2.2.15	2.2.15	2.2.15	2.2.22	2.2.24
PKI locale (chiffrement des flux)	OpenSSL							1.0.0.a		1.0.0.k
Middleware	PHP	5.1.6	5.2.6	5.3.1	5.3.1	5.3.4	5.3.6	5.3.6	5.3.14	5.3.19
Serveur d'authentification	FreeRadius	1.1.2	2.1.0	2.1.7	2.1.7	2.1.8	2.1.8	2.1.8.6	2.1.8.6	2.1.12
Serveur de base de données usagers	Mysql	5.0.24	5.0.67	5.1.40	5.1.42	5.1.46	5.1.55	5.1.58	5.1.63	
	MariaDB									5.5.25
Cache WEB (proxy)	Squid	2.6	3.0.8	3.0.22	3.0.22			3.1-14		3.1.19
Serveur de temps	ntpd	4.2				4.2.4				4.2.6
Journalisation	Ulogd	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24	1.24
Filtrage d'URL WEB	SquidGuard	1.2.0								
	Dansguardian		2.9.9					2.10.1		
Statistiques de consultation	Awstat	2.5	2.5.3	6.9	6.9	6.95	6.95	6.95	6.95	7.0
Lecture des journaux du parefeu	FirewallEyes							1.0		
Info système	Phpsysinfo							2.5.3		
Archivage à chaud du système	Mondo	2.09	2.24	2.28	2.28		2.29			
	Mindi	1.09	1.24			2.0.7				
Chiffrement des fichiers journaux	Gnupp	1.4.5	1.4.9				1.4.10			
Connexion distante sécurisée	openssh- server	4.3-P2	5.1-P1	5.3-P1				5.5p1		5.9p1
Passerelle antivirus WEB	HAVP				0.91			0.92a	0.92a-1	
Antivirus	LibClamav				0.96-0	0.96-1	0.97-0	0.97-3	0.97-5	0.97-7
Serveur de courrier	Postfix									2.8.8

### 3 - Schémas de principe :

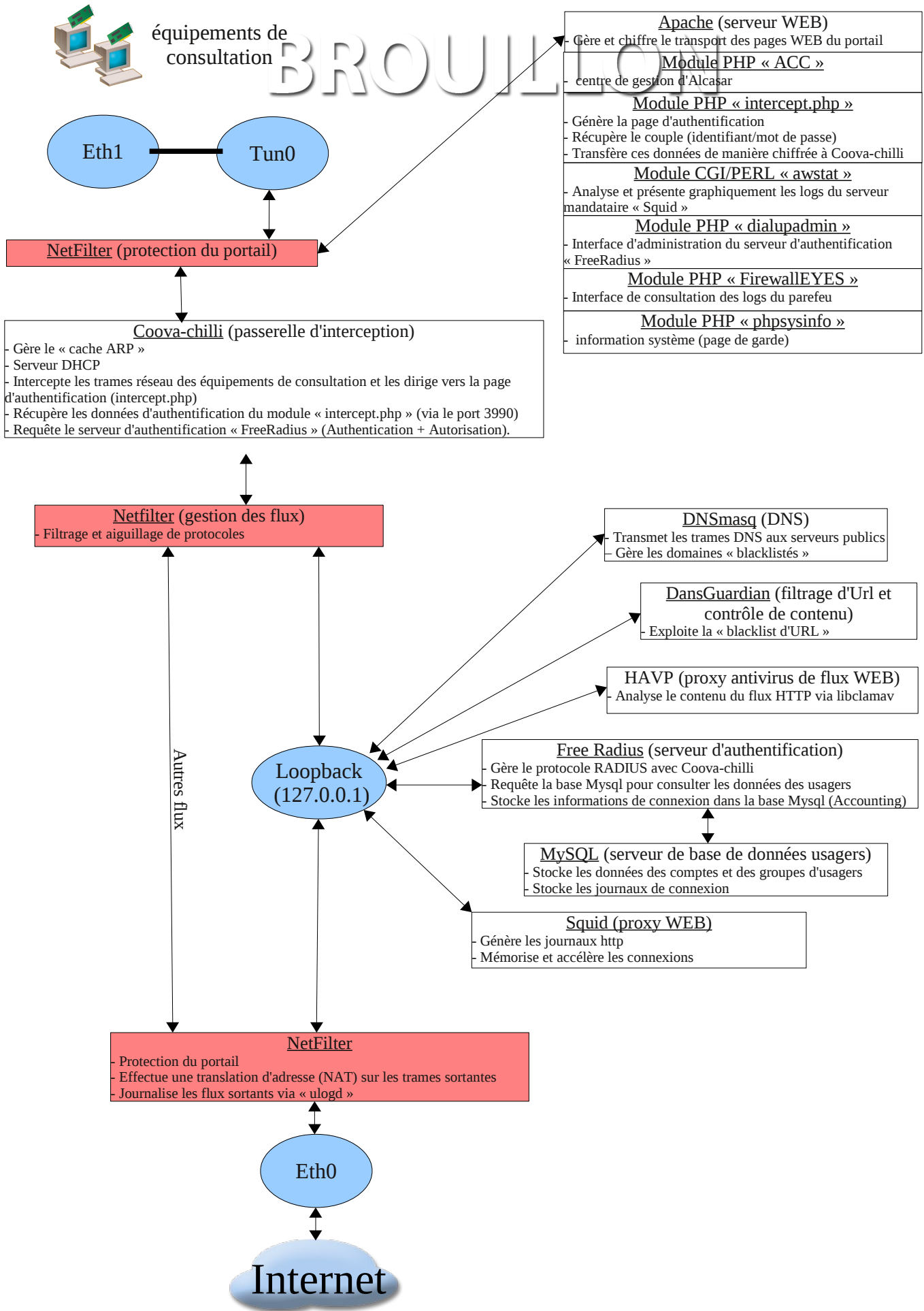
ALCASAR peut être décomposé en cinq fonctions qui sont détaillées dans la suite du document :

- fonction « interception / authentification » réalisée par Coova-chilli, DNSMasq, Apache et le couple Freeradius / Mysql voire LDAP pour l'authentification ;
- fonction « traçabilité / imputabilité des connexions » constituée des journaux du parefeu et du couple Freeradius / Mysql) Le journal du proxy Squid peut aussi être exploité ;
- fonction « filtrage » de domaine, d'URL, antivirus WEB et réseau. Cette fonction est réalisée par le parefeu (Netfilter), le couple (HAVP, LibClamav), DNSMasq et Dansguardian ;
- fonction « interface de gestion » réalisée en PHP et PERL et servie par Apache ;
- fonction « modules complémentaires ». Ces modules ont pour objectif d'améliorer la sécurité globale du portail (anti-contournement, anti-usurpation MAC/IP, chiffrement des fichiers journaux, gestion des certificats, etc.) ou d'enrichir les possibilités du portail (installation, mise à jour, by-pass, archivage, chiffrement des journaux, accélération de la consultation, cron, etc.)

# ALCASAR – ARCHITECTURE



# ALCASAR – Schéma des flux réseau internes



## 4 - Fonction « interception / authentification »

Un des objectifs d'ALCASAR est d'être le plus universel possible. Ainsi, la méthode d'interception et d'authentification choisie s'appuie sur l'« UAM » (Universal Access Method). Cette méthode n'utilise que des protocoles standards ne nécessitant qu'un navigateur WEB pour authentifier un usager situé sur un équipement de consultation. Parmi les autres méthodes, on peut citer celle exploitant des modules clients à installer sur les équipements de consultation (méthode exploitée par le parefeu authentifiant « NuFW » par exemple) ou celle reposant sur des protocoles réseau dédiés (802.1X par exemple).

La fonction « interception / authentification » s'appuie sur la passerelle d'interception « Coova-chilli » (processus « chilli »), le serveur WEB « apache » (processus « httpd »), le serveur d'authentification « Freeradius » (processus « radiusd ») et le système de gestion de bases de données « Mysql » (processus « mysqlmanager » et « mysqld »).

### 4.1 - la passerelle « coova-chilli »

Elle est lancée via son script de démarrage (`/etc/rc.d/init.d/chilli start`) qui a été légèrement adapté par le script d'installation (« `alcasar.sh` »). Ce script utilise le fichier de configuration (« `/etc/chilli.conf` »). Le processus « chilli » est alors lancé en mode « daemon ». Ce dernier crée l'interface virtuelle « tun0 »<sup>1</sup> liée en point à point à l'interface physique connectée au réseau de consultation (eth1). Cela lui permet de gérer sa propre table de résolution ARP en espace utilisateur. Une particularité dans cette gestion consiste à verrouiller les couples (@MAC , @IP) rencontrés sur le réseau de consultation. Un empoisonnement du cache ARP par le réseau est alors impossible (« cache poisoning »). Dans certains cas, ce comportement peut être bloquant (équipement reparamétré après avoir déjà généré des trames IP vers ALCASAR). La commande « `chilli-query list` » permet d'afficher et de contrôler le cache ARP de « chilli ». Cette commande est utilisée par l'interface de gestion (menu « ACTIVITÉ ») pour supprimer les mauvaises associations @IP/@MAC. Complémentaire à cette fonction d'anti-« cache poisoning » intégrée à « chilli », ALCASAR utilise un module spécifique de sécurité (`alcasar-watchdog.sh`) permettant d'éviter l'usurpation d'adresses MAC et d'adresses IP des stations de consultation connectées sur le réseau (cf. fonctions de sécurité).

#### 4.1.1 - Fonctionnement de l'interception (capture)

Lorsqu'un équipement de consultation tente de se connecter sur une URL Internet ([www.free.fr](http://www.free.fr) dans l'exemple qui suit) :

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.182.129	192.168.182.1	DNS	Standard query A www.free.fr
2	0.007977	192.168.182.1	192.168.182.129	DNS	Standard query response A 212.27.48.10
3	0.013100	192.168.182.129	212.27.48.10	TCP	iclprv-nlc > http [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
4	0.018826	212.27.48.10	192.168.182.129	TCP	http > iclprv-nlc [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=
5	0.022528	192.168.182.129	212.27.48.10	TCP	iclprv-nlc > http [ACK] Seq=1 Ack=1 Win=96768 Len=0
6	0.095262	192.168.182.129	212.27.48.10	HTTP	GET / HTTP/1.1
7	0.112659	212.27.48.10	192.168.182.129	TCP	http > iclprv-nlc [ACK] Seq=1 Ack=384 Win=6912 Len=0
8	0.119483	212.27.48.10	192.168.182.129	TCP	[TCP segment of a reassembled PDU]
9	0.122374	192.168.182.129	192.168.182.129	HTTP	HTTP/1.1 302 Moved Temporarily (text/html)
10	0.126880	212.27.48.10	192.168.182.129	TCP	http > iclprv-nlc [FIN, ACK] Seq=1511 Ack=384 Win=6912 Len=0
11	0.131796	192.168.182.129	212.27.48.10	TCP	iclprv-nlc > http [ACK] Seq=384 Ack=1512 Win=96768 Len=0
12	0.135296	192.168.182.129	212.27.48.10	TCP	iclprv-nlc > http [FIN, ACK] Seq=384 Ack=1512 Win=96768 Len=0
13	0.142579	212.27.48.10	192.168.182.129	TCP	http > iclprv-nlc [ACK] Seq=1512 Ack=385 Win=6912 Len=0
14	0.173829	192.168.182.129	192.168.182.1	TCP	iclprv-wsm > https [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=1
15	0.182402	192.168.182.1	192.168.182.129	TCP	https > iclprv-wsm [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=
16	0.182724	192.168.182.129	192.168.182.1	TCP	iclprv-wsm > https [ACK] Seq=1 Ack=1 Win=96768 Len=0
17	0.259992	192.168.182.129	192.168.182.1	SSL	Client Hello
18	0.266048	192.168.182.1	192.168.182.129	TCP	https > iclprv-wsm [ACK] Seq=1 Ack=367 Win=6912 Len=0
19	0.268301	192.168.182.1	192.168.182.129	TLSv1	server Hello, change Cipher Spec, Encrypted Handshake Mess

```
Transmission Control Protocol, Src Port: http (80), Dst Port: iclprv-nlc (3394), Seq: 1461, Ack: 384, Len: 0
[Reassembled TCP segments (1510 bytes): #8(1460), #9(50)]
Hypertext Transfer Protocol
  HTTP/1.1 302 Moved Temporarily\r\n
  Connection: close\r\n
  Cache-Control: no-cache, must-revalidate\r\n
  P3P: CP="IDC DSP COR ADM DEVI TAIi PSA PSD IVAi IVDi CONi HIS OUR IND CNT"\r\n
  [truncated] Location: https://192.168.182.1/intercept.php?res=notyet&uamip=192.168.182.1&uamport=3990&challenge=6595f6
  Content-Type: text/html; charset=UTF-8\r\n
```

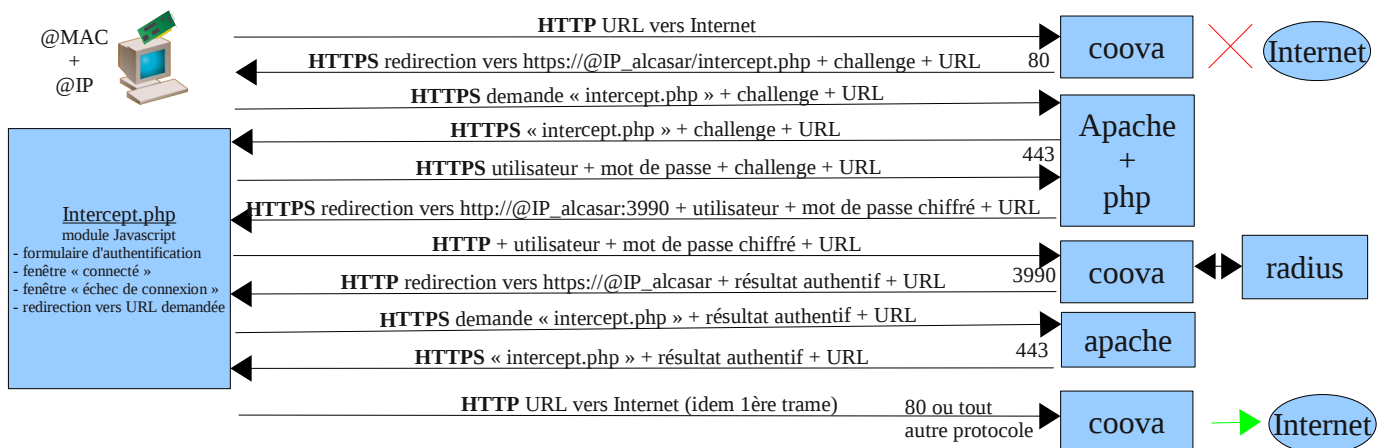
- [trame 1] La requête DNS de l'équipement est récupérée par le serveur DNS d'ALCASAR (dnsmasq). Les tentatives de connexion vers d'autres serveurs DNS sont bloquées par le parefeu interne. Cela permet de prévenir le contournement du DNS d'ALCASAR ainsi que les tunnels DNS.
- DNSMasq résout le domaine localement s'il est dans sa base (cf. fonctions de filtrage), sinon il transfère

1 - Les périphériques « Tap » et « Tun » des noyaux Linux sont des interfaces réseau virtuelles de niveau 2 (c.-à-d. Ethernet) pour « Tap » ou 3 (c.-à-d. IP) pour « Tun » permettant à des processus exécutés en espace utilisateurs (les interfaces physiques fonctionnent en espace noyau) d'envoyer ou de recevoir des trames sur ces interfaces via les fichiers spéciaux (/dev/tapX ou /dev/tunX). Ces interfaces virtuelles peuvent être exploitées comme des interfaces physiques (configuration, émission/réception, routage). Ces interfaces autorisent un traitement sur les trames à la réception ou avant l'émission de celles-ci. L'interface Tap est souvent utilisée dans la création de tunnels RVP/VPN afin d'encapsuler un flux dans un autre (cf. projet « OpenVPN »).

la requête vers les serveurs DNS Internet définis lors de l'installation d'ALCASAR. Les réponses sont retournées à l'équipement de consultation [trame 2].

- Une requête de connexion sur le port 80 (http) du serveur WEB est alors envoyée [trame 3] par la station de consultation. Cette requête est interceptée par « chilli » qui vérifie si un usager n'est pas déjà « autorisé » sur cet équipement :
  - Si tel est le cas, Chilli « ouvre la barrière » et laisse transiter toutes les trames de l'équipement quelque soit le protocole. Le parefeu prend alors le relais. Il oriente les flux WEB vers la chaîne de filtrage WEB. Il filtre ou transfère les autres flux vers Internet (cf. fonction de filtrage).
  - Si tel n'est pas le cas, Chilli simule une connexion WEB standard [trames 4 à 6] et répond à la requête de l'équipement [trames 7 à 9] par une trame HTTP de redirection de service (« HTTP/1.0 302 Moved Temporarily ») contenant l'URL d'une « splash-page » avertissant de la redirection (directive « uamhomepage » du fichier `/etc/chilli.conf`). Dans ALCASAR cette « splash-page » a été supprimée afin de récupérer directement la page d'authentification définie par la primitive « uamserver » (URL de redirection : « https://alcasar/intercept.php ») [cf. détail de la trame 9]. Cette session se termine [trames 10 à 13] et le navigateur initie une session chiffrée avec le serveur WEB intégré dans ALCASAR (Apache) afin de récupérer cette page [trame 14 et suivantes]. L'usager renseigne les champs d'authentification (identifiant + mot de passe) qui sont envoyés de manière chiffrée à Apache pour être traités (chiffrement du mot de passe avec une clé secrète partagée entre apache et chilli). Apache retourne le résultat au navigateur afin que ce dernier redirige une nouvelle fois ces informations au processus « chilli » (port 3990<sup>2</sup>). Chilli les récupère afin de pouvoir requêter le serveur radius. Le résultat de cette requête est retourné au navigateur afin d'être traité par les scripts javascript de la page « intercept.php » (échec ou réussite de la connexion).
  - La communication entre chilli et Freeradius exploite le protocole « radius ». Les paramètres de cette communication sont définis à la fois dans le fichier « `/etc/raddb/client.conf` » et via les directives « `hs_radius` », « `hs_radius2` » et « `hs_radsecret` » du fichier « `/etc/chilli.conf` ».
- Pour la déconnexion, les navigateurs Web génèrent une requête adéquate sur le port d'écoute de Chilli (3990).

Cette phase d'interception peut être schématisée comme suit pour un usager non authentifié sur une station de consultation identifiée par son @MAC et son @IP :



#### 4.1.2 - Exception à l'authentification

Coova-chilli a la possibilité de laisser transiter des trames spécifiques vers Internet sans authentification préalable. Cette possibilité est exploitée dans ALCASAR pour permettre la mise à jour automatique des antivirus et des patches systèmes. Les paramètres « uamallowed » et « uamdomain » pointent vers deux fichiers contenant la liste des adresses IP (ou adresse réseau) ou des noms de domaine joignables sans authentification (`/usr/local/etc/alcasar-uamallow` et `/usr/local/etc/alcasar-uamdomain`).

2 - « chilli » écoute sur un port défini par la primitive « `hs_uamport` » du fichier `/etc/chilli/config` (3990 par défaut). Le format des requêtes envoyées sur ce port détermine l'action demandée (ex. « `@IP:3990/prelogin` » pour une demande de connexion, « `@IP:3990/logout` » pour une demande de déconnexion. La requête contient bien entendu l'ensemble des paramètres nécessaires au traitement de la demande (@MAC, challenge, identifiant, etc.).



## 4.2 - Le serveur FreeRadius

Le service radiusd est utilisé dans le portail comme unité d'identification, d'authentification et d'accounting (mesure d'usage des comptes).

L'identification utilise uniquement le SGBD (par défaut) mais freeradius pourrait être adapté pour utiliser par exemple un annuaire externe.

L'authentification utilise par défaut le SGBD local mais dispose d'un module LDAP pour comparer le couple login/MDP à un annuaire (AD, etc.)

L'accounting utilise uniquement le SGBD pour stocker les traces d'usages des comptes.

Le fichier principal est *radiusd.conf*. Il s'appuie sur le fichier *client.conf*, *sql.conf* pour les paramètres de connexions SQL et sur *ldap.attrmap* pour la mappage des attributs LDAP.

Un fichier alcasar situé sous */etc/raddb/sites-available* définit les paramètres spécifiques à ALCASAR. Un lien symbolique relie */etc/raddb/sites-enable/alcasar* vers ce fichier pour rendre actif ce fichier. Remarque, pour limiter les effets de bords des migrations de freeradius qui rajoute systématiquement 3 liens symboliques vers « inner-tunnel, control-socket et default », ces 3 fichiers sont fixés à 0 volontairement. Ne pas les supprimer !!!

Commande de test de radius : `radtest <userLogin> <userPassword> 127.0.0.1 0 <radsecret>`

## 4.3 - Base de données des usagers

La base de données des usagers est gérée par le SGBD « MariaDB ». Le schéma de cette base est entièrement compatible avec le service d'authentification Radius. La structure de cette base est mise en place lors de l'installation d'ALCASAR en exploitant un script SQL (cf. fonction « *init\_db* » du script *alcasar.sh*) :

```
# Ajout d'une base vierge
mysql -u$DB_USER -p$radiuspwd $DB_RADIUS < $DIR_CONF/radiusd-db-vierge.sql
```

Le Modèle Conceptuel de Données (MCD) de cette base est le suivant :

BASE DE DONNEE RADIUS V2.x (ALCASAR > V1.7\*)

**Légende**  
 En rouge : Index – Clé Primaire  
 En Italique \* : Clé secondaire  
 [Blue Box] Table créée par FreeRadiusWEB  
 [Red Box] Table créée par FreeRadiusSQL  
 [Hatched Box] Table inexploitée par ALCASAR

BROUILL

**1. badusers**  
*id* int(10)  
*UserName* \* varchar(30)  
*Date* \* datetime  
 Reason varchar(200)  
 Admin varchar(30)

**2. mtotacct**  
*MtotAcctld* bigint(21)  
*UserName* \* varchar(64)  
*AcctDate* \* date  
 ConnNum bigint(12)  
 ConnTotDuration bigint(12)  
 ConnMaxDuration bigint(12)  
 ConnMinDuration bigint(12)  
 InputOctets bigint(12)  
 OutputOctets bigint(12)  
*NASIPAddress* \* varchar(15)

**12. userinfo**  
*id* int(10)  
*UserName* \* varchar(64)  
 Name varchar(200)  
 Mail varchar(200)  
*Department* \* varchar(200)  
 WorkPhone varchar(200)  
 HomPhone varchar(200)  
 Mobile varchar(200)

**10. totacct**  
*TotAcctld* bigint(21)  
*UserName* \* varchar(64)  
*AcctDate* \* date  
 ConnNum bigint(12)  
 ConnTotDuration bigint(12)  
 ConnMaxDuration bigint(12)  
 ConnMinDuration bigint(12)  
 InputOctets bigint(12)  
 OutputOctets bigint(12)  
*NASIPAddress* \* varchar(15)

**4. radacct**  
*radacctld* bigint(21)  
*acctsessionid* \* varchar(32)  
*acctuniqueid* \* varchar(32)  
*username* \* varchar(64)  
 groupname varchar(64)  
 realm varchar(64)  
*nasipaddress* \* varchar(15)  
 nasportid varchar(15)  
 nasporttype varchar(32)  
*acctstoptime* \* datetime  
 acctstarttime int(12)  
 acctauthentic varchar(32)  
 connectinfo\_start varchar(50)  
 connectinfo\_stop varchar(50)  
 acctinputoctets bigint(12)  
 acctoutputoctets bigint(12)  
 calledstationid varchar(50)  
 acctterminatecause varchar(32)  
 servicetype varchar(32)  
 framedprotocol varchar(32)  
*framedipaddress* \* varchar(15)  
 acctstartdelay int(12)  
 acctstspdelay int(12)  
 xascendsessionsvrkey varxchar(12)

**11. radusergroup**  
*username* \* varchar(64)  
*groupname* varchar(64)  
 priority int(11)

**5. radcheck**  
*id* int(11)  
*username* \* varchar(64)  
*attribute* varchar(64)  
*op* char(2)  
 value varchar(253)

**9. radreply**  
*id* int(11)  
*username* \* varchar(64)  
*attribute* varchar(32)  
*op* char(2)  
 value varchar(253)

**8. radpostauth**  
*id* int(11)  
 user varchar(64)  
 pass varchar(64)  
 reply varchar(32)  
 date timestamp(14)

**6. radgroupcheck**  
*id* int(11)  
*groupname* \* varchar(64)  
*attribute* varchar(32)  
*op* char(2)  
 value varchar(253)

**7. radgroupreply**  
*id* int(11)  
*groupname* \* varchar(64)  
*attribute* varchar(32)  
*op* char(2)  
 value varchar(253)

**3. nas**  
*id* int(10)  
*nasname* \* varchar(128)  
 shortname varchar(32)  
 type varchar(30)  
 ports int(5)  
 secret varchar(60)  
 community varchar(50)  
 description varchar(200)

\* dans les version < 2.0 : la table « radusergroup » s'appelait « usergroup » et le champs « groupname » de la table « radacct » n'existait pas  
 \* à partir de la version 2.6, le champs 'username' de la table 'userinfo' change de type pour être compatible avec les autres tables (bascule de varchar(30) en varchar(64))

### 4.3.1 - Accès graphique à la base

Afin de pouvoir afficher de manière conviviale et pédagogique le contenu de la base usager, vous pouvez utiliser l'interface WEB « phpmyadmin ».

- installez phpmyadmin : « `urpmi phpmyadmin` »
- modifiez le fichier « `/etc/httpd/conf/webapps.d/phpmyadmin.conf` » afin d'autoriser votre station de consultation à y accéder (allow from votre\_@IP) ;
- connectez-vous à la base à partir de votre station de consultation à l'URL : « `https://@ip_alcasar/phpmyadmin` »
- récupérez le nom et le mot de passe du compte d'administration de la base dans le fichier « `/root/ALCASAR-passwords.txt` »
- identifiez-vous sur le SGBD et choisissez la base « radius »
- Vous pouvez maintenant accéder aux contenus des tables.

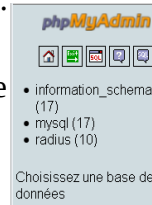
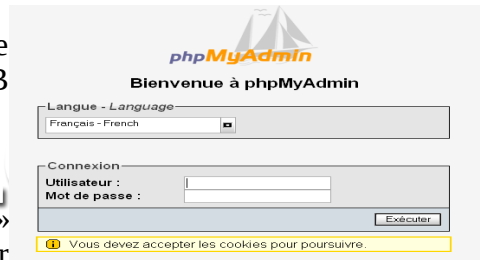


Table	Action	Enregistrements <sup>1</sup>	Type	Interclassement	Taille	Perte
mtotacct		0	MyISAM	utf8_unicode_ci	1,0 Kio	-
radacct		47	MyISAM	utf8_unicode_ci	21,0 Kio	-
radcheck		3	MyISAM	utf8_unicode_ci	3,2 Kio	-
radgroupcheck		0	MyISAM	utf8_unicode_ci	1,0 Kio	-
radgroupreply		0	MyISAM	utf8_unicode_ci	1,0 Kio	-
radpostauth		47	MyISAM	utf8_unicode_ci	3,5 Kio	-
radreply		0	MyISAM	utf8_unicode_ci	3,0 Kio	36 o
radusergroup		0	MyISAM	utf8_unicode_ci	1,0 Kio	-
totacct		0	MyISAM	utf8_unicode_ci	1,0 Kio	-
userinfo		2	MyISAM	utf8_unicode_ci	6,0 Kio	-
10 table(s)	Somme	99	MyISAM	utf8_unicode_ci	41,7 Kio	36 o

### 4.3.2 - Accès console

Avec le login/mot de passe issu du fichier « `/root/ALCASAR-password.txt` », taper la ligne ci-dessous :

```
mysql -uradius -p radius
```

Entrez le mot de passe associé à l'utilisateur radius.

Voir les tables : `SHOW TABLES ;`

Voir le contenu : `SELECT * FROM <tables_name> ;`

Voir les tutoriels concernant le SQL et notamment MySQL.

## 4.4 - Serveur A.D./LDAP externe

Freeradius peut interroger une base externe via le protocole LDAP quand la primitive « ldap » est décommentée dans le fichier « /etc/raddb/sites-enable/alcasar ». Tous les paramètres de connexion sont regroupés dans le fichier « /etc/raddb/modules/ldap ». Ces paramètres sont modifiables via l'interface de gestion ou « à la main ». Les modifications sont prises en compte après avoir relancé le service radiusd : « **service radiusd restart** ».

Paramètres	Définition	Remarques
server	Nom du serveur LDAP (server = "ldap.example.com" ou server = "@IP")	Le port de connexion par défaut est 389. Pour le changer : @serveur:port
basedn	Base de recherche des usagers à authentifier	Voir l'exemple ci-dessous dans le cas d'Active Directory.
filter	Recherche de l'identifiant ou attribut pour l'authentification	<u>Pour un ldap standard</u> : filter = "(uid=%{Stripped-User-Name}:%{User-Name})" <u>Pour Active Directory</u> : filter = "(samAccountName=%{Stripped-User-Name}:%{User-Name})"
base_filter	Filtre de recherche ldap complémentaire	Exemples : <ul style="list-style-type: none"><li>- par défaut, vide</li><li>- base_filter="(objectclass=radiusprofile)"</li><li>- base_filter="(memberof=groupe_alcasar)"</li></ul>
identity	Compte possédant des droits en lecture sur l'annuaire.	Vide = connexion anonyme (LDAP) <u>Obligatoire pour Active Directory</u> (sur le serveur AD, créer un compte standard qui sera utilisé par ALCASAR pour l'interroger l'annuaire à distance).
password	Mot de passe associé au compte avec des droits de lecture sur l'annuaire ldap.	Vide = connexion anonyme (LDAP). <u>Obligatoire pour Active Directory.</u>

Par rapport à l'exemple d'annuaire présenté dans le document d'exploitation, les paramètres de ce fichiers seraient les suivants :

```
basedn = "ou=User,ou=Utilisateur,ou=SITE_I2SC,dc=i2sc,dc=local"
filter = "(samAccountName=%{Stripped-User-Name}:%{User-Name})"
identity= "cn=rldap,ou=Admin,ou=Utilisateur,ou=SITE_ISC,dc=i2sc,dc=local"
password = "*****"
```

Il est possible de tester la liaison A.D./LDAP vers le serveur d'annuaire à partir du poste ALCASAR après avoir installé le paquetage « openldap-clients ». La commande « ldapsearch -vWx -h @ip\_A.D -b "ou=User,ou=Utilisateur,ou=SITE\_i2sc,dc=i2sc,dc=local" -D "rldap@i2sc.local" » permet de lister l'ensemble des usagers contenu dans l'O.U. « User » (-v : verbeux, -b : la base recherchée, -D : le dn de l'utilisateur autorisé à requêter la base, -W : demande le mot de passe de manière interactive, -x : exploite l'authentification simple plutôt que SASL).

Les usagers authentifiés de cette manière sont affectés dans le groupe d'utilisateurs ALCASAR nommé « ldap ». Il est ainsi possible de leur affecter des attributs particuliers. Pour modifier ce nom de groupe, il suffit d'éditer le fichier « /etc/raddb/sql/mysql/dialup.conf » et modifier la valeur « **default\_user\_profile = "<nouveau groupe>"** ». Relancer le service « radiusd » pour la prise en compte de la modification.

Il est aussi possible d'affecter des attributs ALCASAR à un seul usager authentifié A.D./LDAP. Pour cela il suffit de créer un usager ALCASAR ayant le même nom que sur l'annuaire externe. Exemple d'utilisation : tous les élèves d'une école sont gérés dans un annuaire. Il est possible de limiter la bande passante ou les créneaux de connexion à l'élève ayant abusé de téléchargements.

## 5 - Fonction « traçabilité et imputabilité »

### 5.1 - Journalisation principale

La traçabilité des connexions est assurée par le parefeu d'ALCASAR associé au processus « Ulog ». Ce couple permet de dissocier les logs exclusivement liés à la traçabilité des connexions par rapport aux autres. Ainsi, Le parefeu distribue les flux de journalisation sur 3 canaux distincts. À partir du premier canal, « ulog » génère les fichiers « /var/log/firewall/tracability.log » liés aux traces de connexions des usagers (un fichier par semaine). Le deuxième canal génère les fichiers « /var/log/firewall/ssh.log » liés aux flux d'administration à distance via le

protocole ssh. Le troisième canal génère les fichiers « /var/log/firewall/ext-access.log » liés aux tentatives de connexions depuis Internet (fonction « bastion »). Pour ce canal, une protection est mise en place afin de ne pas charger trop le système en cas d'attaque par saturation (flooding).

Trois processus « Ulog » sont donc activés. Les fichiers de configuration respectifs sont « /etc/ulogd-xxxx.conf ».

Tous les dimanche le processus « logrotate » effectue une rotation sur les fichiers de « ulog ». Tous les lundi à 5h35, le gestionnaire de tâche « cron » lance le script « alcasar-archive.sh ». Ce script copie le fichier créé lors de la dernière rotation dans le répertoire « /var/Save/log/ ». Le nom de ces fichiers comprend la date du jour de rotation (ex : tracability.log-20130310.gz).

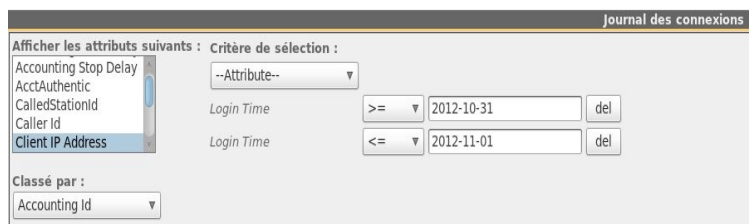
Dans le fichier « tracability.log » chaque ligne correspond à une connexion d'un poste usager. Cette ligne est composée des champs principaux suivants :

Oct 28 03:14:06 alcasar RULE\_F\_http --ACCEPT IN=tun0 OUT= MAC= SRC=192.168.182.2 DST=77.67.27.11 LEN=52 TOS=00 PREC=0x00 TTL=128 ID=19347 DF PROTO=TCP SPT=62934 DPT=80  
 SEQ=16441080 WCK=0 WINDOW=8192 SYN URGP=0

Annotations :

- date
- Nom du parefeu
- Nom de la règle (rule) du parefeu ayant générée cette ligne
  - accept : la trame est sortie sur Internet
  - drop : la trame a restée boquée
- Carte rso par laquelle est entrée la trame
- @IP de la station de consultation
- @IP du serveur destinaire
- Port source
- Port destination

Pour imputer chaque trame, il suffit d'extraire à partir du fichier de la base de données « radius.sql » (de la même semaine) le nom de l'utilisateur connecté sur la station de consultation possédant l'adresse IP source. Cette dernière information peut être récupérée directement à partir de l'interface graphique d'ALCASAR (menu « statistique » + « connexions »). Exemple pour chercher les usagers connectés dans la journée du 31/10/2012 :



## 5.2 - journalisation accessoire

Le proxy cache « Squid » génère des logs dans le répertoire /var/log/squid sous le nom : « access.log ». Ces log ne sont présents que pour permettre le calcul de statistiques de consultation. Le résultat de ce calcul est présenté dans l'interface de gestion par le produit « awstat ».

Le proxy de filtrage d'URL « DansGuardian » génère des logs dans le répertoire /var/log/dansguardian sous le nom : « access.log ». Ces log ne sont pas exploités par ALCASAR. Ils présentent les URL ayant été bloquées.

Le proxy antivirus HAVP génère des logs dans /var/log/havp/ sous le nom « access.log ». Ces logs ne sont pas exploités par ALCASAR. Ils présentent les virus détectés et bloqués par le couple (HAVP + libclamav).

## 6 - Fonction « filtrage »

### 6.1 - Filtrage Réseau

Cette couche est gérée à l'aide du parefeu intégré (NetFilter).

Le portail est configuré en mode 'Bastion' vis-à-vis du monde Internet et canalise les flux en provenance du réseau de consultation. Lors de l'installation, ce fichier a été exécuté et permet de générer le fichier « /etc/sysconfig/iptables » ; à chaque démarrage du serveur, ces règles sauvegardées sont appliquées.

Le fichier de configuration principal qui conditionne le fonctionnement de cova-chilli et des proxy web est « /usr/local/bin/alcasar-iptables.sh ». Il est déconseillé de le modifier afin d'éviter des effets de bords sur le fonctionnement global du portail.

Toutefois, certaines règles du parefeu peuvent être surchargées pour permettre d'accéder à certaines fonctionnalités (accès SSH depuis l'extérieur pour l'administration par exemple).

Pour permettre ces paramétrages 'locaux', le fichier « `/usr/local/etc/alcasar-iptables-local.sh` » est appelé par le fichier principal du parefeu. Les lignes pour l'administration externe par SSH sont commentées dans ce fichier pour exemple.

Par défaut, le portail autorise tous les protocoles lorsqu'une session utilisateur est ouverte. Cette fonction 'libertine' peut-être restreinte par une liste blanche de services autorisés. C'est le rôle du fichier « `/usr/local/etc/alcasar-filter-exceptions` » qui est appelé par le script principal du parefeu si la variable `FILTERING` est positionnée à « `yes` ». Cette dernière est modifiable par le biais de l'interface de gestion. Dans ce cas-là, les services listés dans le fichier `alcasar-filter-exception` sont les seuls à être joignables depuis le réseau de consultation. Cette liste n'est pas exhaustive ; elle est modifiable par le biais de l'interface de gestion.

Le fichier journal des traces du parefeu est « `/var/log/firewall/tracability.log` ». Il est 'rotaté' toutes les semaines dans le répertoire `/var/log/firewall` sous le nom `tracability.log-<date>.gz`.

Exemples de logs :

```
Dec 23 00:25:22 alcasar-cirisi-lyon RULE direct-DNS -- REDIRECT IN=tun0 OUT= MAC= SRC=192.168.182.20 DST=192.168.182.1 LEN=65 TOS=00 PREC=0x00 TTL=64 ID=49813
CE DF PROTO=UDP SPT=37550 DPT=53 LEN=45
Dec 23 00:25:22 alcasar-cirisi-lyon RULE Transfert2 -- ACCEPT IN=tun0 OUT= MAC= SRC=192.168.182.20 DST=150.214.142.197 LEN=60 TOS=00 PREC=0x00 TTL=64 ID=52139
CE DF PROTO=TCP SPT=39359 DPT=80 SEQ=3976661343 ACK=0 WINDOW=5840 SYN URGP=0
```

Pour forcer les usagers à passer par le service DNS du portail, le parefeu effectue une redirection de port 53 vers [l'@IP](#) locale. Cet artifice permet de couper court aux éventuels tunnels DNS (sur le port 53 uniquement). Remarque : les seuls serveurs DNS interrogés par ALCASAR restent ceux qui ont été renseignés lors de l'installation et qui sont définis dans le fichier `/etc/dnsmasq.conf` (primitive 'server').

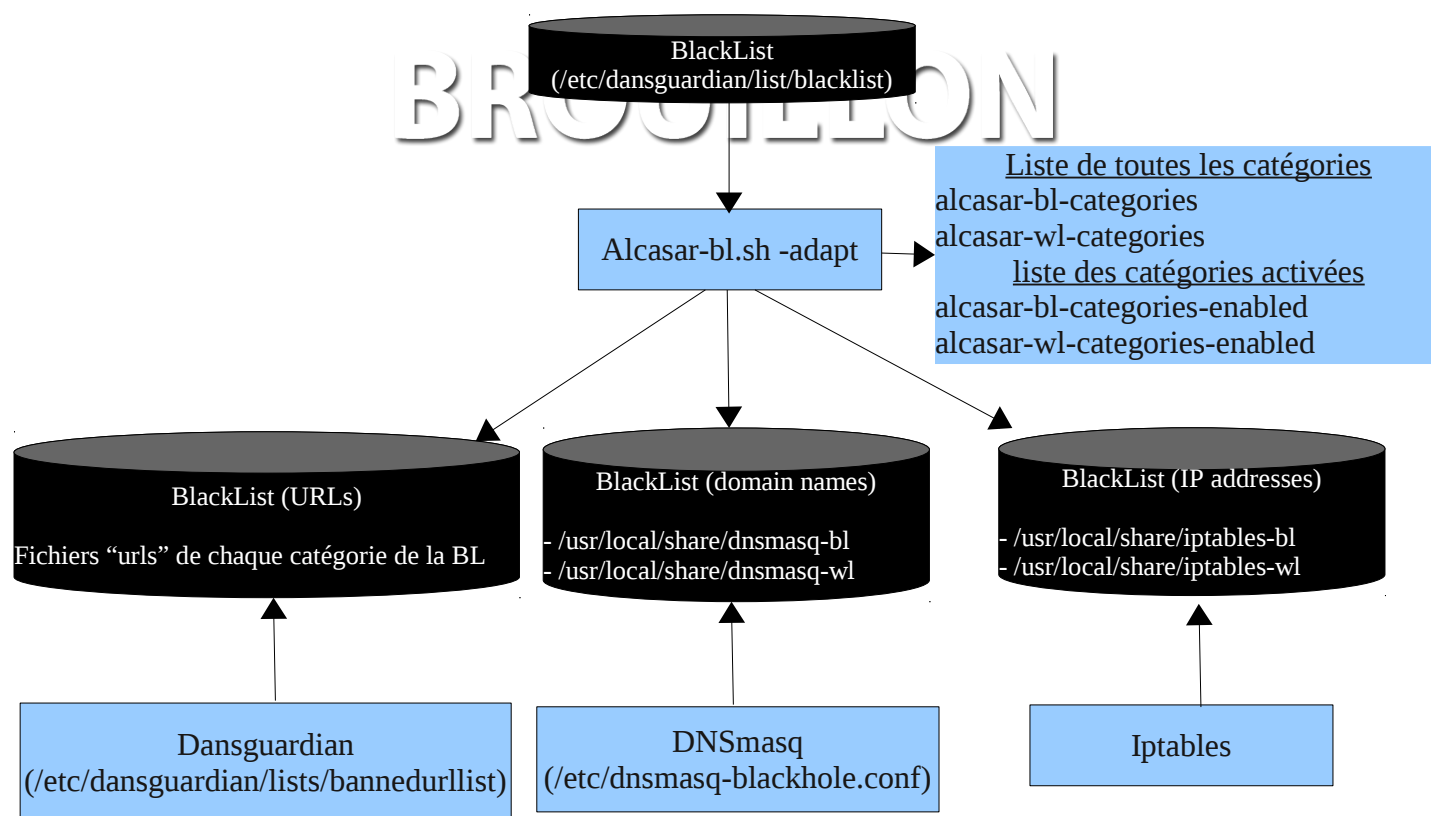
## 6.2 - Filtrage de domaines et d'URLs

Ce filtrage est paramétrable à travers l'interface de gestion. Il s'appuie sur la liste noire (BL) de l'Université de Toulouse qui intègre des « noms de domaine », des « URLs » et des « @IP ». ALCASAR traite cette BL afin de pouvoir l'exploiter selon 3 axes techniques différents :

1. filtrage de noms de domaine  
Pour cela, il s'appuie sur son serveur DNS interne (« `dnsmasq` »). Un domaine interdit dans la BL renvoie [l'@IP](#) du portail (page de filtrage). Ce filtrage offre l'avantage de pouvoir interdire un nom de domaine quelque-soit le protocole demandé (HTTP, HTTPS, FTP, etc.). Ce type filtrage est souvent appelé « DNS blackhole ».
2. filtrage d'URLs  
ALCASAR s'appuie sur le proxy HTTP « `dansguardian` » pour exploiter les URLs de la BL. Les catégories d'URLs blacklistées sont stockées dans le fichier « `/etc/dansguardian/lists/bannedurllist` ».
3. filtrage d'adresses IP  
Le parefeu de sortie d'ALCASAR intègre les @IP de la BL afin de les bloqués.

Le BL est organisée en catégories. Une catégorie peut être en « liste noire » ou en « liste blanche ». ALCASAR permet de sélectionner ces catégories via l'interface de gestion. La liste des noms de catégorie et la liste des catégories activées sont situées dans le répertoire de configuration d'ALCASAR (`/usr/local/etc`).

Afin de permettre cette triple exploitation de la BL, ALCASAR effectue le traitement suivant lors du processus d'import de la BL :



*L'architecture d'ALCASAR rend le contournement du filtrage très compliqué. Celui-ci est toujours possible par l'ouverture d'un tunnel (VPN) à destination d'un équipement maîtrisé situé sur Internet. Pour fonctionner, ce tunnel doit faire transiter l'ensemble des protocoles de la station de consultation (dont le DNS).*

*Ce type de tunnel ne permet pas de contourner l'authentification. Ainsi, ALCASAR trace et impute les trames de ce tunnel. En cas de problème, et si l'enquête détermine que la sortie du tunnel est impliquée, le portail pourra être sollicité pour finaliser la traçabilité.*

Afin de disposer d'une solution de filtrage pour certaines stations et aucune pour d'autres, il est utilisé un second démon dnsmasq (blackhole). Le parefeu redirige (à partir du fichier d'exception de filtrage) les requêtes DNS vers la bonne instance.

### 6.3 - Antivirus WEB

Le proxy HTTP HAVP couplé à la bibliothèque de l'antivirus « Clamav » est utilisé pour analyser le contenu des pages web.

Le fichier de paramétrage de HAVP est `/usr/havp/havp.config` ; il regroupe les ports d'écoute et de transfert au proxy 'parent'.

Le script `alcasar-havp.sh` est appelé par l'interface de gestion pour activer/désactiver l'antivirus.

D'autres antivirus peuvent être associés au moteur HAVP. Des configurations sont disponibles dans le fichier principal `/etc/havp/havp.conf`. Un répertoire monté en tmpfs sur `/var/tmp/havp` permet d'augmenter la fluidité du scanner antivirus. Ce répertoire est nettoyé à chaque démarrage du démon havp ( fonction modifiée directement dans le démon havp).

La base de donnée antivirale qui est située dans `/var/lib/clamav` est mise à jour à toutes les deux heures par le processus « freshclam » (fichier de configuration : `/etc/freshclam.conf`). HAVP recharge cette base toutes les heures.

## 7 - Fonction « Interface de gestion »

Cette fonction est réalisée en PHP. Les possibilités de cette interface sont décrites dans la documentation d'exploitation.

L'interface de gestion (réécrite depuis la version 2.0) se trouve dans `/var/www/html/acc`.

Elle est protégée en accès par le module d'authentification d'Apache.

Dans le fichier de configuration `/etc/httpd/conf/webapps.d/alcasar.conf`, le répertoire `/usr/local/etc/digest/` contient les fichiers des mots des identifiants/mots de passe :

- `key_all`
- `key_admin`
- `key_manager`
- `key_backup`

## 8 - Fonction « modules complémentaires »

### 8.1 - Import de comptes – Fichier mots de passe

Dans le cadre de la gestion des comptes d'authentifications, il est possible d'importer une liste de comptes attachés à un groupe prédéfini. Cette fonctionnalité accessible depuis l'interface de gestion génère un fichier `<import-user>.pwd` pour chaque importation et ajoute les usagers dans le groupe (optionnel) de la base de données. Pour l'instant, seul le groupe peut-être attaché aux identifiants ; c'est-à-dire qu'aucun renseignement supplémentaire n'est importable pour le moment.

Le script `import_user.php` du répertoire « `/var/www/html/acc/manager/htdocs` » permet d'importer le fichier au format csv ou txt et le script `import_file.php` permet de ...

L'importation d'un fichier génère un fichier associé comportant les mots de passe en clair des utilisateurs importés. Ce dernier est téléchargeable pour être distribué aux usagers. Afin de les supprimer périodiquement, une tâche, planifiée toutes les 30mn, cherche et supprime les fichiers datant de plus de 24h00.

Le script lancé est `alcasar-import-clean.sh`.

### 8.2 - Watchdog

Ce script (« `alcasar-watchdog.sh` ») est lancé toutes les 3 minutes par le Daemon « cron ». Il permet de couvrir les fonctions suivantes :

- éviter les « oublis » de déconnexion liés aux pannes (réseau ou équipement de consultation) ;
- limiter le risque lié à l'usurpation d'adresse IP et d'adresse MAC sur le réseau de consultation (pirate interne) ;
- modifier la page WEB présentée aux navigateurs en cas de problèmes de connectivité détectés sur le réseau local (lien Ethernet désactivé sur « eth0 » ou routeur de site injoignable).

Il peut arriver que dans certaines architectures, le watchdog mette en évidence des bizarreries réseau ; en effet, le script `alcasar-watchdog.sh` effectue des requêtes de type `arping` sur l'ensemble des machines censées être 'vivantes'. Hors dans certains cas, elles fonctionnent bien ( elles accèdent à l'internet par le biais d'ALCASAR, elles voient les autres stations, etc. mais elles ne répondent pas à une requête `arping` ???). Cela a été mis plusieurs fois en évidence sans réponse connue aujourd'hui.

Pour pallier ce phénomène qui a l'incidence de tuer toutes les sessions Internet ouvertes depuis ces stations 'muettes', il peut être utile le temps de trouver le problème de commenter le cron qui tourne chaque 3minutes.

→ commenter la ligne comme ci-dessous le fichier `/etc/cron.d/alcasar-watchdog`

```
# activation du "chien de garde" (watchdog) toutes les 3'  
##*/3 **** root /usr/local/bin/alcasar-watchdog.sh > /dev/null 2>&1
```

### 8.3 - Statistiques

Les statistiques d'usages et de navigation ne comportent pas d'éléments permettant de lier les contenus aux usagers. Cela permet de protéger la vie privée des usagers conformément aux préconisations de la CNIL.

### 8.4 - Contournement (by-pass)

En cas de problème technique concernant une des briques logicielles du portail (principalement « coova-chilli »), il est possible de court-circuiter le module d'authentification tout en maintenant le traçage des logs réseau (parefeu).

Un script lancé localement en root `alcasar-bypass.sh -on` | `--off` permet au choix de mettre :



- en mode « On » le bypass → le portail désactive les services coova-chilli, squid, dansguardian
- en mode « Off » : le portail est en mode normal. Tous les services nécessaires sont activés.

## 8.5 - Load balancing de connections

Le script `alcasar-load_balancing.sh` permet de disposer de plusieurs passerelles d'accès à l'Internet. Le script (actif ou non) est lancé au démarrage du serveur par `/etc/rc.local`.

Les interfaces virtuelles qui s'appuient sur eth0 sont créées au démarrage du script et sont définies dans le fichier `/usr/local/etc/alcasar.conf`. Le formatage de ces interfaces est de la forme : `WANx="active[1|0],@IPx/mask,GWx,Weight,MTUx` avec

- WANx avec x, un indice de 1 à ..., pour définir le nom de l'interface virtuelle,
- le premier paramètre (non traité pour le moment) qui prend en compte le côté actif de la carte ou non,
- `l@IP` / masque de l'interface virtuelle,
- `@IP` de la passerelle correspondant à cette interface virtuelle,
- le poids (weight) : un poids identique indique une répartition "égale" sur les interfaces de mêmes poids : la valeur par défaut est "1", identique à la passerelle par défaut,
- la valeur du MTU correspond à cette interface.

Pour être actif, le `load_balancing` nécessite de positionner le paramètre "MULTIWAN=on" (ou 'On').

Le test de connectivité qui s'effectue sur google et sur [www.example.com](http://www.example.com) (voir le paramètre TESTIPS dans le script `alcasar-load_balancing.sh` pour modifier les `@ip` de tests) est actif lorsque le paramètre FAILOVER comporte une valeur différente de "0". Il est effectué tous les x secondes. La valeur doit être positive et entière.

Le script est dorénavant lancé par défaut au démarrage du serveur. Il peut être manipulé par la commande `alcasar-load_balancing.sh` avec les options suivantes :

- `start` : lancement du script qui crée les interfaces virtuelles, les monte et renseigne la table de routage,
- `stop` : arrêt du script et démontage des interfaces virtuelles
- `status` : état du script (uniquement visible en mode failover) ; en effet, en dehors de ce mode, le script n'est pas lancé en mode 'démon'.

## 8.6 - Re-Horodatage des fichiers journaux

Lors de la réinstallation d'un serveur, il peut être utile de réinstaller les fichiers journaux d'origines (avant le crash). Afin que les fichiers disposent d'une date cohérente et que l'effacement des logs s'effectue régulièrement (au bout d'1 an), les journaux doivent disposer de la date en relation avec leur rotation originale. C'est tout l'objet du script `alcasar-dateLog.sh` qui plaque les bons attributs 'date:heure' à partir du nom de fichier (qui comprend un suffixe <date>).

## 8.7 - Module de sauvegarde

Les sauvegardes d'ALCASAR sont disponibles sous 3 formes : le système complet, les archives regroupant les journaux d'évènements et l'export de la base de données et la base de données seule.

### 8.7.1 - Sauvegarde du système complet

Depuis la version 2.5, le système de sauvegarde à chaud est supprimé au bénéfice d'une archive reprenant la configuration spécifique d'un site.

Le répertoire de sauvegarde se trouve dans `/var/Save/archive`, il est automatiquement synchronisé chaque semaine ou à la demande avec la commande `alcasar-archive.sh -now`

### 8.7.2 - Sauvegarde des journaux d'évènements

Les journaux d'évènements du système ainsi que ceux des services utiles à ALCASAR sont situés sous `/var/log/`.

Les journaux du firewall, de l'interface Web et de squid sont rotatés régulièrement (chaque semaine). Pour rendre ces archives consultables et téléchargeables par le biais de l'interface de gestion, ces logs sont copiés dans le répertoire grâce à une tâche planifiée qui appelle le script `alcasar-log-export.sh` chaque semaine. Ils sont visibles par le userSystème apache pour permettre aux gestionnaires de les récupérer par le biais de l'interface.

Afin de limiter la conservation des traces à 1 an, le script `alcasar-log-clean.sh` est lancé chaque semaine et efface tous les fichiers dont la date système est supérieures à 365 jours. Tous les lundis matin, la tâche de purge des

logs est planifiée à 4h30 grâce au fichier `/etc/crond.d/alcasar-clean_log` et à 5h00 pour l'export au moyen du au fichier `/etc/crond.d/alcasar-export_log`.

### 8.7.3 - sauvegarde de la base de données

Chaque semaine, la base de données est exportée et sauvegardée dans le répertoire `/var/Save/base` sous la forme : `<db_radius>-<date>.sql`. Cette tâche, planifiée chaque semaine, appelle le script `alcasar-mysql.sh -dump`.

Une tâche journalière est lancée chaque lundi ( 04h45) `/etc/crond.d/alcasar-mysql` pour réaliser l'export de la base dans le répertoire `/var/Save/base`.

Ces sauvegardes sont téléchargeables par le biais de l'interface web.

Chaque nuit ( `/etc/crond.d/alcasar-mysql` ), les utilisateurs dont leur date d'expiration a dépassé 7 jours sont supprimés.

## 9 - Annexes

Ce chapitre reprend les fichiers de configuration spécifiques à ALCASAR.

### 9.1 - Coova-chilli

Les fichiers se situent sous « `/etc/`, `/etc/chilli` et `/usr/local/etc` ».

- Fichier principal : `chilli.conf` (sous `/etc`)
- Exceptions Domaines : `alcasar-uamdomain` (sous `/usr/local/etc`)
- Exceptions URLs : `alcasar-uamallowed` (sous `/usr/local/etc`)
- Exceptions d'authentification par MAC Adresses : `alcasar-macallowed` (sous `/usr/local/etc`)
- L'association dynamique `d'@IP` statiques s'effectue par le biais du fichier : `alcasar-ethers` (sous `/usr/local/etc`)

### 9.2 - Freeradius

Les fichiers du démon radius se situent sous « `/etc/raddb` ».

- Fichier principal : `radiusd.conf`
- Fichier de connexion BDD : `sql.conf`
- Fichier clients autorisés à requêter le service radiusd : `clients.conf`
- Fichier dédié : `alcasar` (sous `/etc/raddb/sites-available` avec un lien symbolique qui lie les « `sites-enable` »)

### 9.3 - Dnsmasq

En fonctionnement normal, dnsmasq ne fournit que les services liés au DNS (filtrage de domaine « DNS-blackhole, transfert de requête « FORWARD DNS » et cache « CACHE-DNS »). En mode secours (bypass), il fournit aussi le service DHCP (coova-chilli s'occupe de ce service en mode normal).

- Fichier principal : `/etc/dnsmasq.conf`
- Filtrage de domaines : `alcasar-dnsfilter-enabled` (sous « `/usr/local/etc` »). Il active le filtrage des classes de domaines.
- Pour intégrer un Active Directory dans l'architecture, il est nécessaire de modifier le fichier de dnsmasq `/etc/sysconfig/dnsmasq` ; en effet un bug empêche la prise en compte d'un redirecteur pour un domaine donné dans le fichier de configuration principal ( cf. Doc Exploitation § 8.5)

### 9.4 - Parefeu

- Fichier principal du parefeu d'Acasar : `alcasar-iptables.sh` (sous `/usr/local/bin`)
- Règles personnalisées du parefeu : `alcasar-iptables-local.sh` (sous `/usr/local/etc`)
- Fichier de filtrage Réseau (associé à `alcasar-nf.sh`) : `alcasar-iptables-exception`
- Activer/désactiver le filtrage web : `alcasar-bl.sh` ( sous `/usr/local/bin`)
- Fichier listant les classes de filtrage (associé à `alcasar-nf.sh`) : `alcasar-bl-categories-enabled` ; utilisée par le fichier `alcasar-bl.sh` pour le filtrage dnsmasq et dansGuardian.
- Fichier contenant la liste complète des domaines par classe issue de la liste noire de Toulouse : `alcasar-dnsfilter-available` (sous `/usr/local/etc/`)
- Fichier du parefeu d'Acasar utilisé en mode ByPass : `alcasar-iptables-bypass.sh` (sous `/usr/local/bin`)

### 9.5 - Dansguardian

Les fichiers de DansGuardian se situent sous « `/etc/dansguardian` ».

- Fichier principal de configuration : `dansguardian.conf`
- Fichier concernant le groupe 1 utilisé par ALCASAR : `dansguardianf1.conf`
- Le répertoire « `lists` » contient les fichiers de filtrage proprement dits :

- « *bannedsitelist* » : non exploité (cf. §6.2)
- « *exceptionsitelist* » : non exploité
- « *bannediplist* » : non exploité
- « *exceptioniplist* » : exploité pour la liste des adresses IP en exception de filtrage
- « *exceptionurllist* » exploité pour la liste des URLs réhabilitées
- « *bannedurllist* » contient la liste des catégories d'URL à filtrer
- « *blacklists* » : contient la BL de Toulouse. Les répertoires « urls » de chaque catégorie sont exploités pour le filtrage d'URLs.

## 9.6 - Squid

Les fichiers de Squid se situent sous « */etc/squid* ». Hormis le fichier principal, tous les autres sont utilisés par défaut.

- Fichier principal : *squid.conf* ; squid est paramétré en mode proxy transparent.

## 9.7 - Ulogd

Le démon ulogd centralise les logs du parefeu (dissociés des logs 'messages') ; tous les journaux d'évènements sont gérés en mode texte.

- Fichier de configuration : *ulogd.conf*
- Fichier concernant les flux Ssh extérieurs en provenance de eth0 : *ulogd-ssh.conf*
- Fichier concernant les flux bloqués en provenance du réseau extérieur : *ulogd-ext-access.conf*

La rotation des logs s'effectue hebdomadairement pour httpd, squid et **tracability**

## 9.8 - HAVP + Clamav

Le moteur HAVP est paramétré avec la bibliothèque libClamav

- HAVP :
  - Fichier de configuration du moteur antivirus : *havp.config*
    - le format des logs a été modifié pour faciliter la remontée des stations ayant demandé un fichier 'infecté'. En croisant les fichiers de havp.log, tracability et une requête sur la BDD, il est ainsi possible de trouver *l'@IP* et *@MAC*.
  - Un répertoire au format tmpfs ( */var/tmp/havp* ) est utilisé pour mettre en mémoire le répertoire du scan ; il est monté au démarrage du démon havp et nettoyé et démonté à son arrêt.
- libClamav
  - la périodicité de mise à jour des signatures est paramétrée par défaut à 12 fois /jour).

## 9.9 - Distribution Mageia et ses dépôts

La distribution Mageia est utilisée comme système d'exploitation du portail. Les mises à jours et l'installation des paquets s'effectuent à l'aide des outils natifs : « urpmi ».

Les fichiers de configurations se trouvent sous */etc/urpmi* :

- source des miroirs : *urpmi.cfg* ;
- exceptions des mises à jour de paquets : *skip.list* ; permet d'exclure des mises à jours certains paquets pouvant éventuellement troubler le fonctionnement du portail.
- Pour effectuer une mise à jour automatique ( sans répondre aux questions ) : *urpmi --auto-update --auto*
- Pour effectuer du ménage : *urpme --auto-orphans --auto*