Outils de sécurité réseau avec OpenBSD et PF

Matthieu Herrb
CNRS LAAS
7, avenue du Colonel Roche, 31077 Toulouse Cedex 4
Université de Toulouse : UPS, INSA, INP, ISAE, UT1, UTM, LAAS.

Résumé

Cet article présente la mise en place de solutions de sécurité réseau basées sur des logiciels libres au LAAS du CNRS. Dans le domaine de la sécurité réseau, deux composants clé utilisés dans le laboratoire sont le pare-feu principal et le portail captif destiné à accueillir les connexions de visiteurs. Les outils retenus se basent sur le filtre de paquets PF ainsi que sur des extensions du serveur DHCP du système OpenBSD.

Après avoir étudié plusieurs scénarios de déploiement, le pare-feu principal du laboratoire est constitué de deux serveurs redondants contrôlant le trafic à l'aide de PF en mode transparent.

Pour l'accueil de connexions réseaux de visiteurs, le laboratoire souhaite continuer à leur proposer un accès ouvert. Un portail captif « auto déclaratif » sur lequel chaque visiteur s'enregistre pour accéder à l'Internet a été réalisé à l'aide d'outils présents dans OpenBSD.

Mots clefs

Sécurité, pare-feu, OpenBSD, portail captif

1 Introduction

Le LAAS est attaché aux logiciels libres depuis leur apparition dans le paysage informatique au milieu des années 1980. C'est pourquoi les solutions libres permettant de réaliser des fonctions de sécurité ont toujours été suivies de près. Face au besoin de solutions de filtrage réseau plus performantes que celle qui avait été déployée par le passé et au besoin d'une solution pour accueillir les machines de visiteurs selon le cahier des charges du laboratoire, le filtre de paquets PF d'OpenBSD est assez rapidement apparu comme une solution attrayante.

La première partie de cet article va situer rapidement les caractéristiques d'OpenBSD et de son filtre de paquets pf par rapport à d'autres solutions. Ensuite la solution retenue pour le pare-feu principal du laboratoire puis l'application plus spécialisée gérant le portail captif pour l'accueil des visiteurs seront présentées.

2 OpenBSD et PF

OpenBSD est un système d'exploitation dérivé de la distribution BSD4.4 de l'Université de Berkeley. Le projet a démarré en 1995 avec comme objectif de produire un système capable d'assurer la sécurité tout en restant utilisable. Pour ceci, le projet s'est focalisé sur trois axes : l'intégration dans le système d'outils de sécurité de base (les bibliothèques de cryptographie et les applications qui les utilisent : SSH, IPSec, SSL...), le développement d'outils de sécurité tels que le filtre de paquets PF ou des mécanismes d'authentification plus sûrs que le hachage classique des mots de passe Unix et enfin la revue permanente du système à la recherche d'erreurs de programmations (les bugs) qui peuvent avoir ou non des conséquences pour la sécurité, mais qui doivent de toutes façons être éliminées d'un système que l'on souhaite fiable.

C'est pourquoi OpenBSD se compose d'un système de base d'aspect minimaliste, considérant que la complexité inutile crée un milieu favorable au développement d'erreurs qui peuvent se transformer en failles de sécurité.

Le filtre de paquets PF a été introduit en 2001 [1] en remplacement d'un logiciel existant (IPF) qui ne pouvait plus être intégré à OpenBSD en raison d'un changement sa licence d'utilisation. PF a hérité d'IPF son design et ses principes de fonctionnement : filtrage basé sur une inspection des paquets avec un état permettant de traiter implicitement à l'aide d'une seule règle tous les paquets appartenant à un même flux IP entre deux nœuds, ainsi qu'une logique d'évaluation des règles inversée par rapport à la

majorité des autres filtres existants : pf évalue toutes les règles qui peuvent s'appliquer à un paquet et utilise la dernière règle trouvée [2].

La combinaison de ces deux caractéristiques permet de décrire la politique de sécurité en termes de règles de filtrage de façon descendante, en commençant par les règles générales et en affinant les cas particuliers par la suite. Cela conduit en général à un ensemble de règles plus compact et plus lisible pour une politique de filtrage donnée.

Le recours systématique à une table d'états (activée par défaut sur toutes les règles), couplée à un optimiseur qui factorise automatiquement les règles fournies par l'utilisateur de manière à obtenir un ensemble équivalent de longueur minimale permet de minimiser les recherches dans les règles. OpenBSD obtient ainsi des performances suffisantes pour traiter des flux de l'ordre du giga-bit par seconde sur des matériels de type PC relativement standard.

PF a été conçu initialement pour fonctionner sur les interfaces d'un routeur traitant les paquets au niveau 3. Cependant toujours en traitant le niveau 3, il peut également fonctionner en mode « transparent » sur les interfaces d'un équipement fonctionnant en pont de niveau 2.

Enfin PF est bien intégré avec plusieurs autres sous-systèmes d'OpenBSD comme le gestionnaire de bande passante (ALTQ) ou le serveur DHCP et, depuis OpenBSD, 4.9 pf supporte pleinement IPv6 pour l'ensemble de ses fonctionnalités.

3 Pare-feu redondant

Le LAAS souhaitait améliorer l'efficacité du filtrage réseau réalisé jusqu'au début de 2010 sur un routeur Cisco à l'aide de listes de contrôle d'accès (ACL) « étendues ».

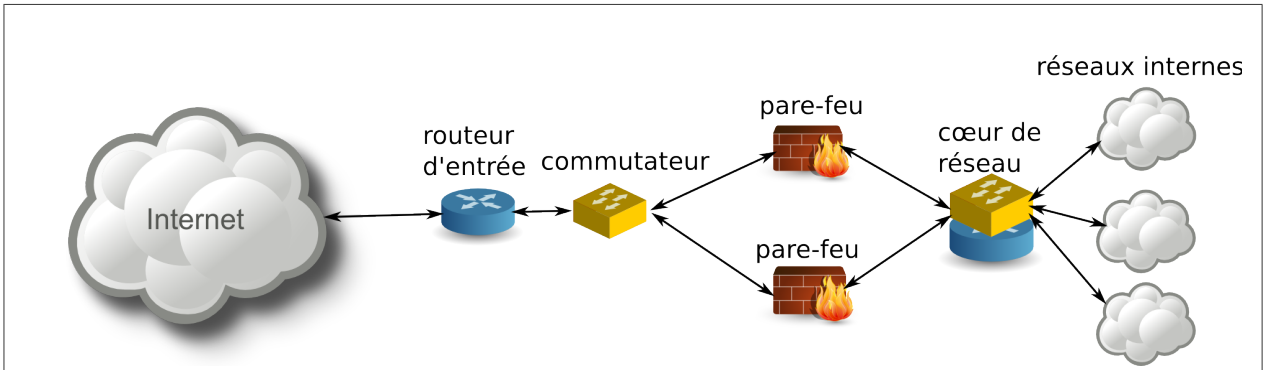

Figure 1: Architecture niveau 2

Les ACLs disponibles dans la version d'IOS installée sur ce routeur ne permettent de traiter l'état des connexions que via le mot clé « *established* » qui définit de manière très large et floue les connexions TCP déjà ouvertes.

Ce type de contrôle d'accès oblige à faire des concessions par rapport à une politique de sécurité définie afin de permettre le bon fonctionnement de certains protocoles.

Par ailleurs, les nombreuses exceptions demandées par les équipes de recherche du laboratoire dans le cadre des projets autour des réseaux et de la sécurité rendaient la base des règles du pare-feu peu lisible donc difficile à maintenir et à modifier.

Sur la base d'expériences précédentes avec PF, ainsi qu'en raison du coût souvent élevé de solutions basées sur des routeurs filtrants ou des pare-feux commerciaux, l'équipe d'administration système et réseau s'est assez rapidement ralliée à la proposition de construire un système basé sur OpenBSD et PF. Il a cependant été décidé de construire une architecture qui permette, au moins pendant les premiers mois de l'exploitation réelle, de re-basculer sur la solution précédente en cas de problème.

Enfin, le routeur Cisco existant assure aussi le routage des paquets multicast avec le protocole PIM *sparse mode.* L'absence d'une solution de remplacement éprouvée et intégrée au système de base d'OpenBSD a conduit à choisir de déployer le pare-feu en mode pont derrière le routeur existant, plutôt qu'en remplacement complet du routeur Cisco.

3.1 Redondance

Pour faire face à d'éventuelles défaillances matérielles ou logicielles et pour faciliter les interventions de mise à jour des logiciels ou du matériel, une solution redondante de type actif-passif était souhaitée.

OpenBSD permet de réaliser des pare-feux redondants grâce au protocole *pfsync* qui assure la synchronisation de l'état de plusieurs pare-feux, afin qu'un paquet soit traité avec le même résultat quel que soit le pare-feu réel qui va le traiter.

La solution habituellement utilisée avec OpenBSD pour assurer le contrôle du trafic vers un ensemble de pare-feux redondants est le protocole CARP, similaire au protocole VRRP de Cisco mais non encombré de brevets logiciels. Cependant CARP (ou VRRP) suppose que le pare-feu intervient en tant que routeur [3].

Dans la solution retenue (pare-feu au niveau 2), il existe d'autres possibilités pour assurer l'aiguillage du trafic entre les pare-feux Les deux principales solutions étudiées sont l'agrégation de liens de niveau 2 (telle que définie par la norme IEEE 802.3ad) et les mécanismes d'arbres de recouvrement définis par IEEE 802.1D.

Dans les deux cas, l'architecture physique correspond à celle qui est présentée sur la figure 1.

L'interface interne du routeur de sortie est connectée à un commutateur qui va aiguiller le trafic vers l'un ou l'autre des pare-feux. Ces derniers sont ensuite connectés directement sur deux interfaces du commutateur-routeur central.

3.1.1 Agrégation de liens

L'agrégation de liens semblait a priori la solution la plus élégante pour assurer le contrôle de la commutation des paquets. Dans cette configuration les pare-feux sont en effet complètement transparents, et la seule contrainte est que les commutateurs situés de part et d'autre supportent un mode d'agrégation de liens de type actif-passif mutuellement compatible, ce qui semblait possible en utilisant le protocole LACP (IEEE 802.8ad) [4].

Les premiers tests de basculement entre les pare-feux ont été réalisés en déconnectant physiquement le câble sur la branche du pare-feu actif. Les commutateurs Cisco réagissent alors de manière conforme aux attentes et basculent sur le second lien avec un délai imperceptible pour l'utilisateur ayant des connexions ouvertes traversant le pare-feu.

Mais bien que LACP soit configuré en mode actif, l'implémentation de Cisco semble reposer entièrement sur l'observation de l'état des liens au niveau physique. Dans le cas d'une interruption du lien causée par un arrêt logiciel du pare-feu (soit une défaillance soit pendant une mise à jour) plutôt que par un problème matériel, la coupure du lien n'est pas détectée et par conséquent le trafic ne bascule pas sur le lien de secours. Il n'a pas été possible de trouver une explication satisfaisante à ce comportement.

3.1.2 Arbre de recouvrement

La technologie d'arbre de recouvrement (*Spanning Tree*) est utilisée traditionnellement pour protéger un réseau en cas de modification de sa topologie de niveau 2, en particulier en cas d'apparition de boucles. Elle permet également de provisionner de la redondance [5].

Avec le protocole de *spanning tree* RTSP configuré sur les quatre éléments qui constituent le domaine de niveau 2 du réseau d'inter-connexion (les 2 commutateurs et les 2 pare-feux en mode pont), on obtient un comportement cohérent en cas de coupure physique ou logique d'un lien. Dans le cas d'une coupure physique, le délai de convergence de RTSP est plus long (jusqu'à 20 secondes sur la plate-forme de test) qu'avec un mécanisme d'agrégation de liens mais ce délai reste compatible avec les réglages du protocole TCP. Une interruption de la communication est perceptible sur les protocoles de téléphonie ou de vidéo, mais la durée moyenne observée a été jugée acceptable.

C'est cette solution basée sur les arbres de recouvrement qui a été retenue pour la version opérationnelle.

3.2 Autres tests

Des tests simples de performance ont été effectués sur les machines retenues pour assurer le rôle de pare-feu. Ceux-ci n'ont pas réussi à montrer de dégradation de performances du pont avec un jeu de règles représentatif de celui qui allait être utilisé par rapport au cas où les pare-feux étaient déconnectés. La principale difficulté de ces tests a été de générer un trafic représentatif d'une charge réelle capable de provoquer une charge mesurable sur les pare-feux. Avec le logiciel *iperf* plusieurs flux, totalisant 750 Mb/s ont pu traverser le pare-feu, avec un jeu de règles proche de celui utilisé actuellement en production. Ces performances

étant largement supérieures au trafic observé sur le routeur du laboratoire, il a été décidé de passer rapidement à la phase de production, sachant que le retour en arrière en cas de problème restait facile.

3.3 Déploiement

La solution retenue a été installée physiquement début septembre 2009 avec une version préliminaire d'OpenBSD 4.6. Dans un premier temps pf n'a pas été activé, le routeur Cisco continuant à assurer le filtrage. Dans cette configuration, on n'a pas pu observer le moindre changement de performance des connexions réseau vers l'extérieur et la charge sur le pare-feu actif restait en permanence inférieure à 5 % d'utilisation du processeur (essentiellement passé en mode noyau dans le traitement des interruptions des cartes réseau).

Ensuite, les règles ont été progressivement migrées du routeur Cisco vers pf en surveillant l'éventuelle apparition de problèmes. Au bout de 2 semaines, l'ensemble des règles de filtrage avait été migré vers pf, et le routeur Cisco n'assure depuis que le routage du trafic.

La version d'OpenBSD sur les pare-feux a été mise à jour vers 4.7 puis vers 4.9 en profitant de la redondance pour réaliser ces mises à jour sans interruption du trafic. Ces deux opérations ont été les seules occasions de voir le mécanisme de basculement à l'œuvre, les 2 pare-feux n'ayant connu aucune défaillance depuis leur mise en service.

En moyenne sur le mois de septembre 2011, le pare-feu du LAAS a traité un trafic de 11,5 Mb/s avec des pointes, mesurées sur un intervalle de 5 min à 110 Mb/s. Durant les périodes d'activité, cela représente entre 1200 et 2000 paquets par seconde en moyenne. La charge CPU du pare-feu actif reste en dessous de 10% d'utilisation. Il faut remarquer que ce trafic est loin de saturer les possibilités du pare-feu.

Au courant du mois de juillet 2011, une adresse IP non utilisée du LAAS a été la cible d'une petite attaque de type déni de service distribué. Hormis des pics sur le graphe des paquets rejetés par seconde (jusqu'à 6000 par seconde), cette attaque n'a eu aucun effet perceptible par les utilisateurs.

Pour être complet, il faut signaler que jusqu'à la version d'OpenBSD 4.9 incluse, le ré-assemblage des fragments IPv6 n'était pas assuré par pf. Cette fonctionnalité a été ajoutée peu après la sortie de la version 4.9. La version actuelle d'OpenBSD fonctionnant sur les pare-feux intègre cette fonctionnalité et permet maintenant d'assurer la même qualité de filtrage en IPv6 qu'en IPv4.

3.4 Administration

L'administration du pare-feu comporte deux types de tâches : l'archivage et le suivi des traces d'une part et les modifications à apporter aux règles de filtrage en fonction des besoins (évolutions du parc des machines, modification de la politique de filtrage) d'autre part.

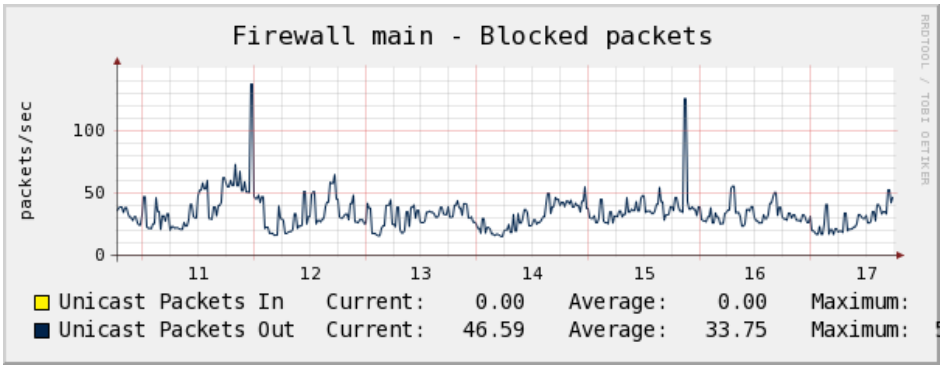

Figure 2: Graphe des paquets rejetés

Les traces de pf sont collectées dans des fichiers au format *pcap*, lisibles par l'outil *tcpdump*. Les fichiers produits sont archivés quotidiennement et transférés vers le serveur d'archivage des traces.

Une analyse en temps réel du trafic traversant le pare-feu et du nombre de paquets bloqués est collecté par pf et mis en forme graphique à l'aide de l'outil *cacti*. PF propose des outils plus sophistiqués pour une analyse plus fine du trafic, notamment un collecteur de données au format *netflow*, qui n'a pas été mis en œuvre pour l'instant.

Les modifications du jeu de règles se font en éditant le fichier `/etc/pf.conf` sur le pare-feu maître. Un script assure ensuite la validation des nouvelles règles et leur chargement dans la configuration active des deux pare-feux. Par ailleurs, le fichier des règles ainsi que les autres fichiers principaux de configuration du système sont maintenus dans un logiciel de gestion de version (git) afin d'avoir un suivi des modifications et une possibilité de retour en arrière en cas de problème.

3.5 Extensions et évolutions possibles

L'utilisation d'un logiciel libre à la base de la solution permet une infinité de possibilités d'extensions. Parmi celles qui sont envisagées pour le futur au LAAS, il y a en particulier les possibilités d'accroître la dynamique de la politique de filtrage, en permettant par exemple à certaines catégories d'utilisateurs, authentifiés et autorisés, d'ouvrir à la demande certains ports de leur poste de travail sans intervention d'un administrateur réseau.

Un développement récent dans la pile réseau d'OpenBSD permet de détourner certains paquets traités par PF vers une application en mode utilisateur, pour réaliser une inspection en profondeur du trafic. Cela permettra, si le besoin s'en fait sentir de connecter un module de détection de contenus malicieux afin de les bloquer au plus tôt. L'impact de cette technologie sur les performances globales du système n'a cependant pas été évaluée.

Enfin la possibilité de collecte de données de type *netflow* permettra d'avoir une analyse plus fine de la nature des flux.

4 Portail captif

Le projet de portail captif a démarré avant celui du pare-feu redondant. L'objectif était de maintenir un accès facile à une connexion Internet en wifi ou en filaire aux visiteurs du laboratoire.

Le choix exprimé par le conseil de laboratoire a été d'utiliser un système d'auto-enregistrement sur lequel les utilisateurs indiquent eux-même leur identité et une adresse de messagerie (utile pour les joindre au cas où un incident de sécurité susceptible de les affecter soit détecté sur le réseau visiteur).

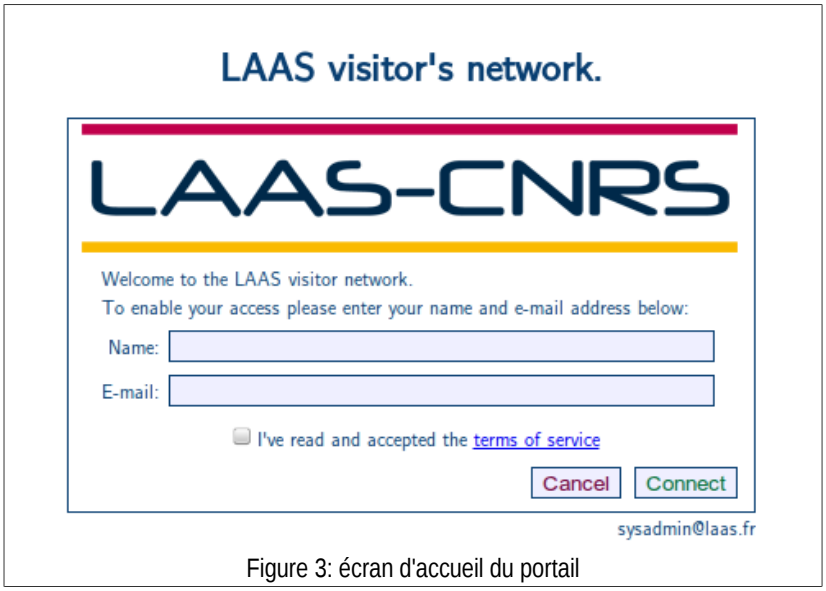

Figure 3: écran d'accueil du portail

L'analyse de plusieurs solutions existantes (blue socket, Alcatel Omni-Access, chillispot) n'a pas été satisfaisante [6]. En parallèle à cette étude, un prototype de portail captif avec authentification LDAP s'appuyant sur PF et son intégration avec le serveur DHCP avaient été développés pour démontrer la capacité des outils d'OpenBSD à produire rapidement des solutions de sécurité intéressantes.

Il a été alors décidé d'adapter ce prototype pour développer en interne, dans l'esprit d'OpenBSD (KISS - K*eep It Simple and Stupid*) la solution de portail pour les visiteurs.

4.1 Principe de fonctionnement

Lorsqu'une machine dont l'adresse MAC n'est pas connue se connecte sur un port d'un commutateur ou sur un point d'accès sans fil du laboratoire, elle est affectée au VLAN dédié aux visiteurs.

Ce VLAN est connecté à la zone semi-ouverte du laboratoire par un routeur sous OpenBSD qui assure à la fois le rôle de serveur DHCP, de routeur, de portail web et de pare-feu pour ce réseau.

Par défaut toute communication du réseau visiteur vers l'extérieur est bloquée par le pare-feu. Seules les requêtes DHCP sont autorisées.

Toute nouvelle machine qui émet une requête DHCP sur le réseau se voit attribuer une adresse IP par le serveur qui place alors cette adresse dans la table ACTIVE de PF. Cette table est utilisée par une règle de pf qui redirige tout le trafic HTTP de ces adresses vers la page d'accueil du portail.

Le serveur DHCP distribue l'adresse du serveur DNS à utiliser. Le serveur DNS est assuré par le logiciel *unbound* en mode cache, qui apporte une validation du contenu des requêtes.

Sur cette page d'accueil, un formulaire permet aux visiteurs de s'enregistrer et de prendre connaissance des conditions d'utilisation associées à ce service (y compris la charte d'utilisation des moyens informatiques du CNRS). Après avoir accepté ces conditions en cochant la case adéquate, l'utilisateur peut sélectionner le bouton « *connect* » pour demander sa connexion.

Les informations entrées sont alors validées (pour l'instant seule l'existence d'un relais de messagerie valide pour la partie droite de l'adresse est effectivement vérifiée, mais d'autres tests peuvent être imaginés) et le script associé à ce formulaire fait alors passer l'adresse IP utilisée dans la table pf CLIENTS qui identifie les adresses des utilisateurs autorisés à utiliser le réseau visiteur. Les

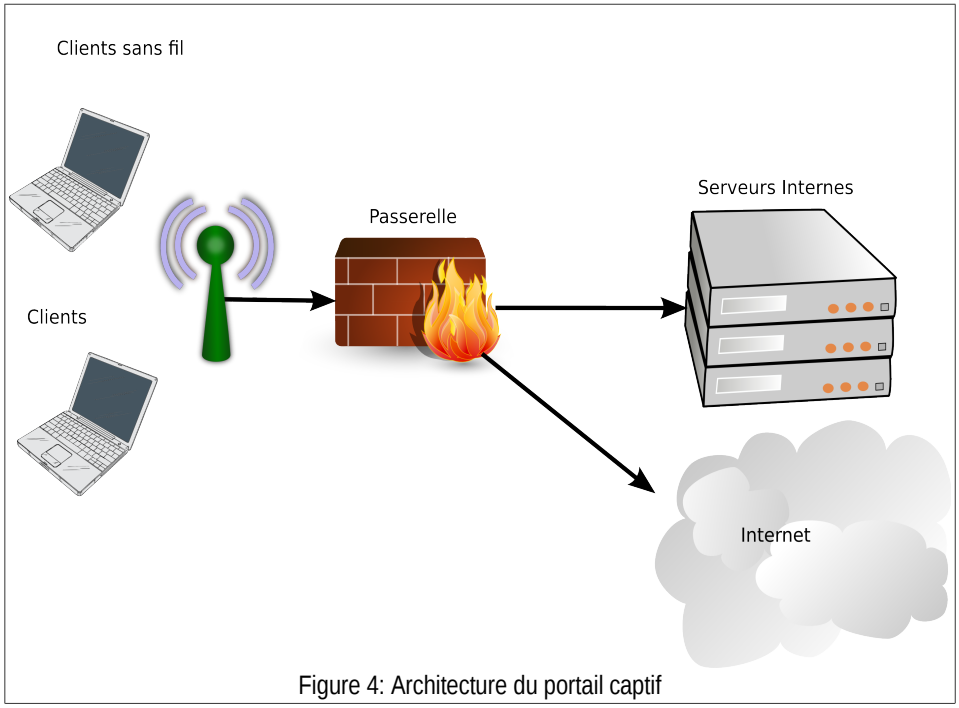

Figure 4: Architecture du portail captif

règles de pf pour les adresses dans cette table permettent l'accès aux services réseaux les plus courants (messagerie, http[s], SSH, clients VPN) pour la durée du bail DHCP. Lorsque le bail DHCP expire ou si le serveur DHCP reçoit un paquet RELEASE du client avant la fin du bail, celui-ci retire l'adresse IP des tables de PF, ce qui a pour effet d'isoler à nouveau complètement l'adresse IP qui vient d'être libérée.

Le serveur enregistre pour chaque bail DHCP l'adresse MAC, l'adresse IP associée et les informations d'identification fournies par l'utilisateur et conserve ces traces pendant un an, conformément au décret d'application du 25 février 2011 de la loi pour la confiance dans l'économie numérique (LCEN).

4.2 Sécurité

Plusieurs attaques contre ce portail ont été prises en compte pour assurer un niveau de sécurité optimal. L'utilisation d'une adresse IP libre prise au hasard n'est pas possible puisque seules les adresses des postes ayant passé la phase d'identification par le portail peuvent accéder à l'extérieur. L'utilisation d'une adresse IP déjà identifiée provoque une alerte qui est utilisée pour supprimer l'identification. On peut donc provoquer un déni de service, mais il reste impossible d'accéder à l'internet.

Parmi les attaques classiques sur les portails captifs, l'utilisation de tunnels IP dans DNS peut être contournée de plusieurs manières. L'utilisation d'un serveur cache tel que unbound oblige le tunnel à utiliser des requêtes DNS bien formées (un simple tunnel UDP sur le port 53 sera rejeté). En limitant le nombre de requêtes par seconde vers le serveur, pour les adresses non identifiées (qui n'ont a priori besoin que de quelques requêtes avant de voir la page d'accueil du portail) un tel tunnel devient pratiquement inutilisable.

Enfin, le choix explicite d'un portail web très simple (un formulaire avec deux champs) limite considérablement la surface d'attaque sur le serveur HTTP du portail. Les valeurs des champs d'identification sont assainies et uniquement stockées pour la journalisation.

4.3 Extensions possibles

En raison de sa conception volontairement simple, ce portail captif présente des limitations qu'il peut être intéressant de supprimer, mais il permet également d'envisager des extensions qui ne sont pas disponibles même sur des produits beaucoup plus complets.

En termes de confort d'utilisation, une extension souvent demandée est l'utilisation d'un moyen (*cookie* par exemple) permettant de mémoriser une identification, afin de permettre à un utilisateur de passer l'étape d'identification pendant un certain temps après une première identification. Cette extension, à l'étude, est réalisable sans compliquer outre mesure le système existant.

Le support d'IPv6 serait une extension assez inédite dans le panorama des portails captifs. Il s'agit cependant d'un projet plus ambitieux puisque le serveur DHCP d'OpenBSD qui fournit les extensions utilisées pour communiquer avec PF ne supporte pas DHCPv6. Il a donc été envisagé d'ajouter à un serveur DHCPv6 disponible pour OpenBSD ces fonctions de communication avec PF.

Enfin pour améliorer la fiabilité de la collecte de l'identité de l'utilisateur, un mécanisme de confirmation par courrier électronique de l'adresse saisie pourrait être implémenté, par exemple sur le modèle proposé par le réseau « Île sans fil » à Montréal (ouverture initiale pour 20 à 30 minutes, prolongation après confirmation d'un code envoyé à l'adresse indiquée, sans possibilité d'utiliser la même adresse MAC avec une autre identité).

5 Conclusion

En utilisant les outils du système libre OpenBSD, le LAAS a pu se doter de solutions de sécurité qui correspondent à ses besoins, tout en ayant un coût tout à fait raisonnable, à la fois pour le développement initial et pour l'exploitation quotidienne. La solution de pare-feu redondant utilisant PF et les arbres de recouvrement assure le traitement du trafic réseau depuis plus d'un an et permet d'exprimer la politique de sécurité en règles de filtrage de manière à la fois simple et complète grâce au formalisme de PF.

La réalisation du portail captif pour l'accueil des machines des visiteurs en s'appuyant sur des outils existants dans le système a permis de créer un mécanisme d'auto-enregistrement qui contraint au minimum les utilisateurs.

Dans les deux cas, en plus de la satisfaction de voir des besoins couverts au meilleur coût, le fait d'avoir une solution maîtrisée debout en bout présente plusieurs avantages : les possibilités de faire évoluer les solutions en fonctions des besoins ou des contraintes nouvelles, un aspect pédagogique qui permet aux administrateurs de s'approprier davantage la technologie tout en étant capable de comprendre le fonctionnement fin des différents composants via des échanges avec les communautés existantes d'utilisateurs et de développeurs des outils libres utilisés.

Bibliographie

[1] H. Brauer et R. McBride, Ten years of PF, dans EuroBSDCon 2011, Maarsen, Pays-Bas, Octobre 2011. http://bulabula.org/papers/2011/pf10yrs/

[2] P. N. M. Hansteen, *The book of PF, A no-nonsense guide to the OpenBSD firewall*, 2nd edition, NoStarch Press, 2010, ISBN-13: 978-1-59327-274-6.

[3] B. Palmer, J. Nazario, *Secure Architectures with OpenBSD,* Addison-Wesley, 2004, ISBN 03-21193-66-0

[4] Agrégation de liens, article Wikipedia : http://fr.wikipedia.org/wiki/Agr%C3%A9gation_de_liens

[5] *Spanning tree protocols,* article Wikipedia : http://fr.wikipedia.org/wiki/Spanning_tree_protocol

[6] N. Rakotomampionona, Étude des solutions d'accès sécurisées au réseau sans fil. *Rapport de stage LAAS-CNRS*, Octobre 2007