Ecole d'été CNRS-ENSTB "apprentissage mobile" 2009

Rôle des composants réseau dans une architecture sensible au contexte

Alain OTTAVI

Orange Labs

sommaire

1 Introduction

Toutes les informations passent par le réseau

- De façon sure et transparente. Pourquoi ne pas en profiter ?
- Il est donc possible de connaître les informations relatives aux utilisateurs du réseau.

Comment les stocker, les analyser pour fournir des services personnalisés aux utilisateurs qu'ils soient en situation de "nomadisme" ou en "mobilité" ?

But de cette présentation : principes, avantages et limites, méthodes et contraintes de la capture des flux et de l'exploitation des traces pour fournir des services personnalisés en cœur de réseau.

/!\ Les architectures présentées sont générales et ne sont pas limitées aux situations de "nomadisme" ou de "mobilité". Cette présentation mettra cependant en avant des cas d'usages plus particulièrement intéressants dans ces deux types de situation.

/!\ Ce n'est pas un cours d'architecture réseau : les architectures présentées sont là pour démontrer l'intérêt de travailler à partir du cœur de réseau pour des applications utilisées en situation de "nomadisme" ou de "mobilité". La présentation se focalisera donc sur les caractéristiques "haut niveau" intéressantes pour la conception d'applications "d'apprentissage mobile".

2 Interception et analyse des flux

Interception : inconvénients et contraintes

- Inconvénients :
  - Impossible de "voir" les flux cryptés entre clients et serveurs en cœur de réseau.
  - Les flux seulement destinés à être exécutés côté client sont difficilement compréhensibles, voire même incompréhensibles (*e.g.* flux Ajax).
  - Possible changement du comportement de l'application. Modification du code applicatif peut-être nécessaire.
  - Dans le cas d'applications sécurisées par certificats, cela peut modifier la configuration cliente et/ou serveur et/ou des machines intermédiaires.

- Contraintes :
  - Intercepter, analyser et rediriger les flux :

    - Reconnaissance applicative ? Cela évolue vite !!!

    - rajout d'une machine équipée (configuration, charge), d'un logiciel spécifique (configuration, mise à jour des bases de signatures et des méthodes de reconnaissance des protocoles et applications).

  - Contraintes de « privacy » importantes (informations des utilisations, droit d'accès et de modification aux informations à caractère personnel, conservation et suppression de ces données ([CNIL], [G29])).

Interception : principes et avantages

- Principe : dispositif en cœur de réseau :
  - en coupure des flux (proxy en mode explicite ou transparent, sonde ou DPI en mode bridge),
  - interception/redirection (WCCP, Policy Based Routing, proxy transparent, switch L4/L7).

- Avantages : accès à tous les flux utilisateurs quels que soient leurs origines/destinations (sens client/serveur ou serveur/client).
  - Appliquer des règles communes à tous les flux : authentification; sécurité, filtrage de toute sorte. Configuration minimale voire nulle côté client.
  - Créer une cartographie des interactions de chaque utilisateur avec tous les autres utilisateurs.
  - Enrichir les flux par des informations de profils.
  - Fournir des services personnalisés en fonction du profil utilisateur et/ou des caractéristiques machines/clients utilisés pour se connecter aux différents services.
  - Composer facilement des applications personnalisées en changeant à volonté l'ordre d'appels des services ou applications *(e.g.* ACLs*)*.
  - Appliquer des mécanismes de « load-balancing » (« Equilibrage de charge »)/ "fail-over » (« protection contre la défaillance ») entre différents serveurs applicatifs permettant une haute-disponibilité de l'application ainsi qu'une plus grande scalabilité.

Les principaux composants pour la capture en cœur de réseau

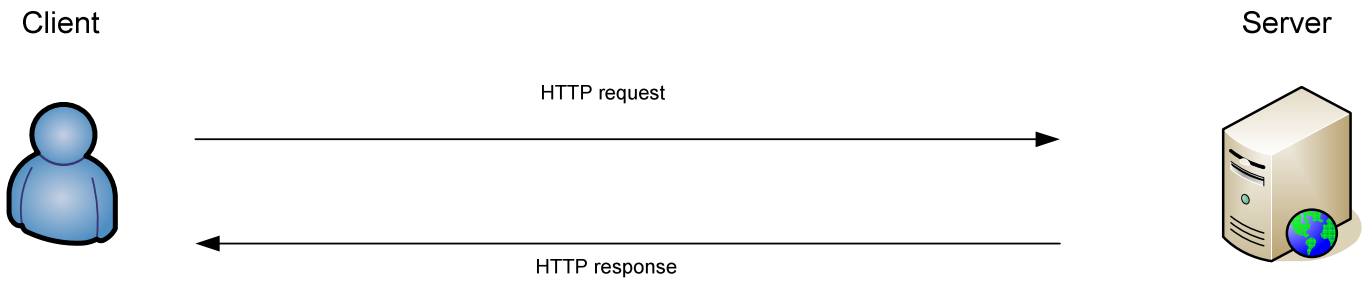
/!\ Firewall, IDS/IPS sont spécifiquement dédiés à la seule protection du réseau et sont donc exclus de cette présentation

Composant	Rôles originels	Rôles actuels supplémentaires
Sonde	Capture traces réseau en mode bridge, réinitialisation connexion.	Reconnaissance applicative ("Deep Packet Inspection"), Shaping (limitation débit par type applicatif/protocole, QoS), redirection, cartographier les flux.
Routeur /(Switch)	Routage des paquets	Applications ACL (limitation débit, filtrage utilisateur et horaire, filtrage et protection du contenu…), redirection conditionnelle (protocole, @IP source et destination, port source et destination), protection contre les DoS.
Proxy	Economie de bande passante HTTP et FTP, sécurité et accélération ("reverse-proxy")	- Optimisation des connexions HTTP, Application ACL sur tout type de protocole (heure, utilisateur, nature protocole), ajout de service (*e.g.* ICAP), enrichissement de flux (rajout de header), redirection vers des services, "TCP protocol tunneling", contrôle connexion SSL, point de terminaison SSL et accélération SSL, compression applicative "à la volée".
WAF	Protection et intégrité des sites Web contre tout type d'attaque.	Détection des comportements utilisateurs (basique), accélération SSL.

Tous ces composants sont maintenant prévus pour :

- faire de l'authentification auprès d'annuaire, de Base de Donnée ou d'un Radius.
- Appliquer des ACL (Access Control List) sur les flux qui les traversent.

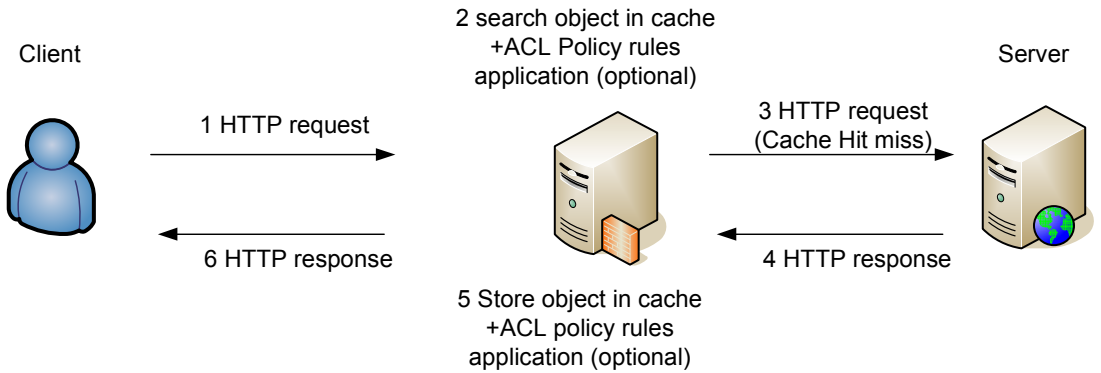
HTTP : connexion sans interception



HTTP Level	GET http://server:port/url Host : server
TCP Level	(Client @IP, Client source port) >> (Server @IP, Server Port)

HTTP : interception avec proxy en mode "explicite"

Déploiement au niveau "client"



HTTP Level	GET http://server:port/url HTTP/1.x Host : server	GET /url HTTP/1.x Host : server
TCP Level	(Client @IP, Client source port) >> (Proxy @IP, Proxy Port)	(Proxy @IP, Proxy source port) >> (Server @IP, Server Port)

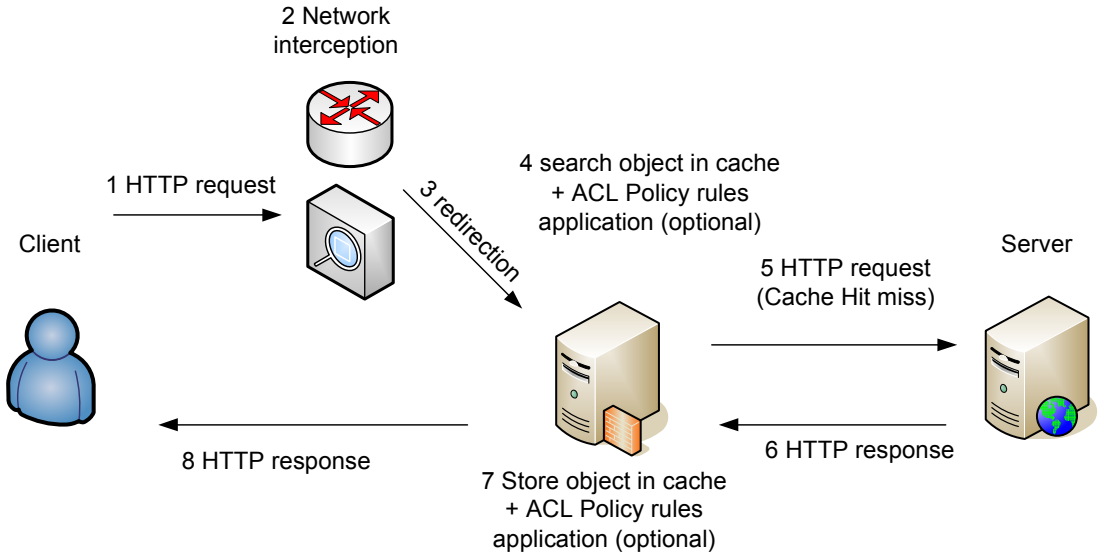
Le logiciel client connaît l'existence du proxy et s'adresse explicitement à lui pour chaque requête.

La configuration côté client peut se faire de deux façons :

- à la main, en rentrant l'URL du proxy, du script .pac (Proxy Auto-Config).
- chargé par un script au démarrage *(e.g.* lors de l'authentification de l'utilisateur*)*.

HTTP : interception avec proxy en mode "transparent"

Déploiement au niveau "réseau"



HTTP Level	GET http://server:port/url HTTP/1.x Host : server	GET /url HTTP/1.x Host : server
TCP Level	(Client @IP, Client source port) >> (Server @IP, Server Port)	(Proxy @IP, Proxy source port) >> (Server @IP, Server Port)

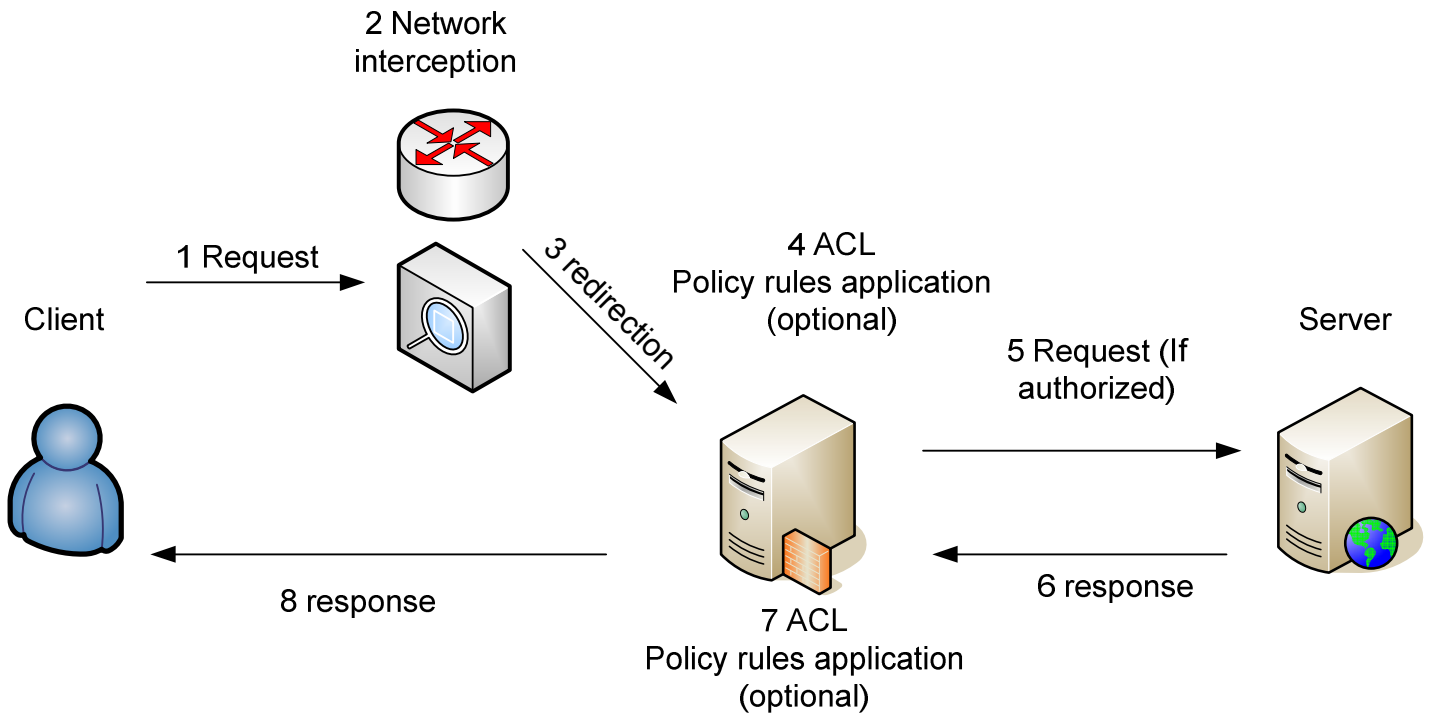
Le logiciel client ignore l'existence du proxy et pense s'adresser au serveur.

Aucune configuration côté client.

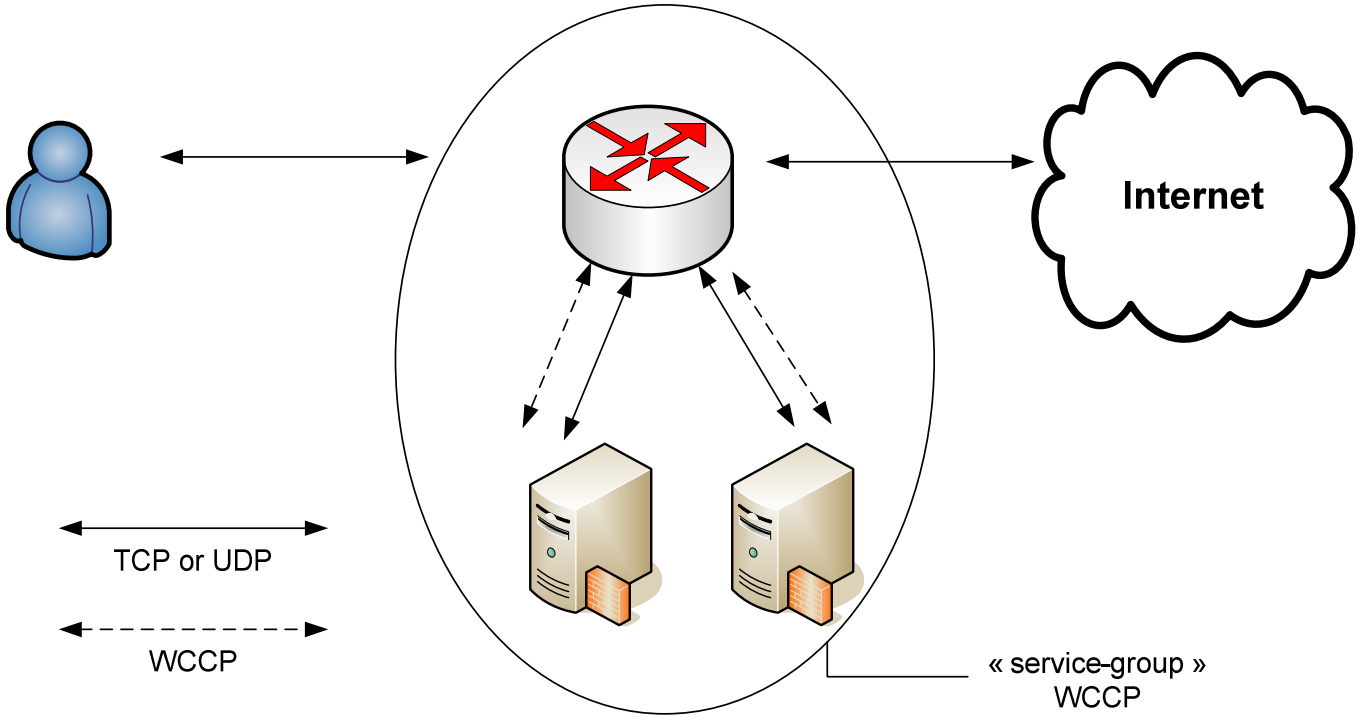
- C'est un composant réseau qui intercepte les flux et les re-route vers un autre élément du réseau (*e.g.* proxy)

- L'élément intercepteur peut-être une sonde ou un proxy en mode bridge, un routeur utilisant PBR (Policy Based Routing) ou WCCP, un switch L4/L7.

Cas général : interception avec proxy en mode "transparent"



Cas général : WCCP avec routeur et proxy compatibles



Principe : redirection vers un ensemble de proxy d'un flux TCP ou UDP qui arrive sur une interface du routeur.

Avantages notables :

- Marche pour n'importe quel flux TCP ou UDP (**dont on connaît le port**).
- Load balancing entre les proxy, mécanisme de "heart-beat",

Inconvénients notables :

-- Ne marche que pour un petit nombre de ports simultanés à prédéfinir à la mise en place du "service-group".
-- Surcharge du routeur.
-- Nécessite un routeur et des proxy compatibles

3 Services à Valeur Ajoutée en cœur de réseau

Service à Valeur Ajoutée en cœur de réseau

Une fois le flux intercepté, il est possible de le rediriger sur d'autres équipements (fonction des capacités du composant intercepteur). Nous allons présenter succinctement les quelques cas suivants :

- DPI →Equipement spécifique
- (DPI/Routeur/[Switch L4/L7]/WAF/Proxy) →Proxy
- Proxy →Equipement ou service spécifique
- Proxy →Serveur Web avec réécriture d'URL pour protocole HTTP.
- Proxy avec traitement des flux par application sur mesure pour le protocole HTTP.
- Proxy →Serveur iCAP pour les protocoles HTTP/FTP

Service à Valeur Ajoutée en cœur de réseau

- DPI →Equipement spécifique

-Ex : Partenariat Allot (DPI) et Aladdin (société spécialisée dans la sécurité) pour faire du content protection / content filtering, sur les contenus HTTP/FTP/SMTP/POP. La DPI et la machine supportant le logiciel tiers communiquent directement sans intermédiaire par des câbles réseaux en utilisant la couche MAC.

- (DPI/Routeur/[Switch L4/L7]/WAF/Proxy) →Proxy

- DPI → Proxy : mis en place par Optenet [OPTENET] avec son composant CCOTTA. Aussi à l'étude chez d'autres éditeurs de DPI.

- Routeur →Proxy : WCCP (routeur et proxys compatibles) ou PBR (Policy Based Routing).

- Switch L4/L7 → Proxy : PBR.

- WAF → Proxy ou Proxy → Proxy évidents car les deux composants réseau ont été créés pour comprendre HTTP et pour être chainés entre eux.

- Proxy →Equipement ou service spécifique

- Applications de protection du contenu / filtrage du contenu. Ces applications étant souvent le résultat d'un partenariat précis entre un constructeur d'équipement et un éditeur de logiciel, le tout fonctionnant de façon propriétaire. Ex : BlueCoat (Leader sur le marché des Proxy) et les accords avec éditeurs d'AntiVirus ou de solutions de contrôle parental.

Service à Valeur Ajoutée en cœur de réseau

- Proxy →Serveur Web avec réécriture d'URL pour protocole HTTP

- Utilisation du proxy en mode "reverse-proxy" (le client pense s'adresser au serveur Web alors qu'il s'adresse au proxy, ce dernier s'occupant de toutes les translations d'URL).

- Utilisation d'un serveur Apache avec le module mod_rewrite pour la réécriture d'URL. D'autres serveurs Web ont des capacités comparables.

- Proxy avec traitement des flux en local par une application sur mesure pour le protocole HTTP.

- Utilisation des "squid redirector" : SQUID envoie sur sa sortie standard le quadruplet (Uri, Client/FQDN, identifiant, méthode) qui peut être alors traité par n'importe quel programme sur la machine qui renvoie sur la sortie standard de la machine l'URI résultante.

- Proxy →Serveur iCAP pour les protocoles HTTP/FTP

iCAP : "Internet Content Adaptation Protocol" : Protocole "léger" pour exécuter un RPC ("Remote Procedure Call") sur les messages HTTP".

- Clients iCAP : BlueCoat, NetApp (BlueCoat), Squid (GPL),

- Serveurs iCAP : Webwasher (McAfee), BlueCoat AV, IBM Datapower, Netasq, Orange et différents services iCAP (voir ci-dessous).

Service à Valeur Ajoutée en cœur de réseau : le protocole iCAP

-Proxy →Serveur iCAP pour le protocole HTTP/FTP

-"idée derrière iCAP" : la démarche générale pour rajouter des services en cœur de réseau est la suivante :

1 - Intercepter le flux (*e.g*…. vers un proxy) et rediriger le vers l'équipement particulier supportant le service adéquat après authentification de l'utilisateur (auprès d'un annuaire, une base Radius, une Base de Donnée, une authentification HTTP basique…) pour faire de la personnalisation de services.

2 - Rajouter un load-balancer en frontal de ces équipements pour assurer la scalabilité du service.

3 - Rajouter les autres équipements supportant les autres services dont vous avez besoin.

4 – Définissez des règles (ACLs et ordre d'appels) pour l'accès aux services de chaque utilisateur.

-Vous obtenez une architecture à base de services iCAP (RFC 3507) !!

Cette architecture est basée :

- Sur un client iCAP (en général un proxy) qui gère l'authentification utilisateur, l'ordre d'appels des différents services iCAP en fonction d'ACL (composition de services), le stockage des réponses des serveurs iCAP et la communication avec le client internet.

- Sur un serveur iCAP qui contient l'application.

Il existe deux modes de fonctionnement du service iCAP :

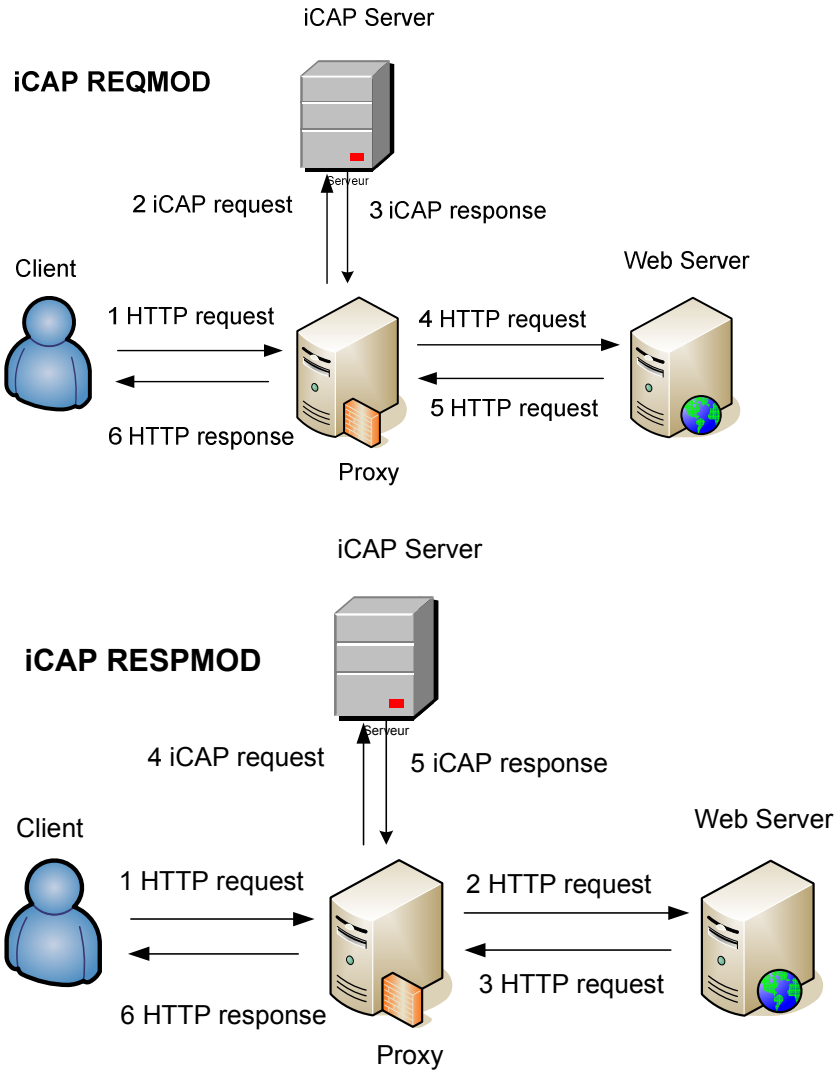
1 - modification de la requête par le serveur iCAP : "reqmod",

2 - modification de la réponse par le serveur iCAP : "respmod".

Dans ces 2 modes, le serveur iCAP peut être appelé avant de regarder dans le cache du proxy ou après, ce qui donne 4 possibilités appelées "vector-point". Deux d'entre eux permettent la personnalisation du service en fonction de l'utilisateur.

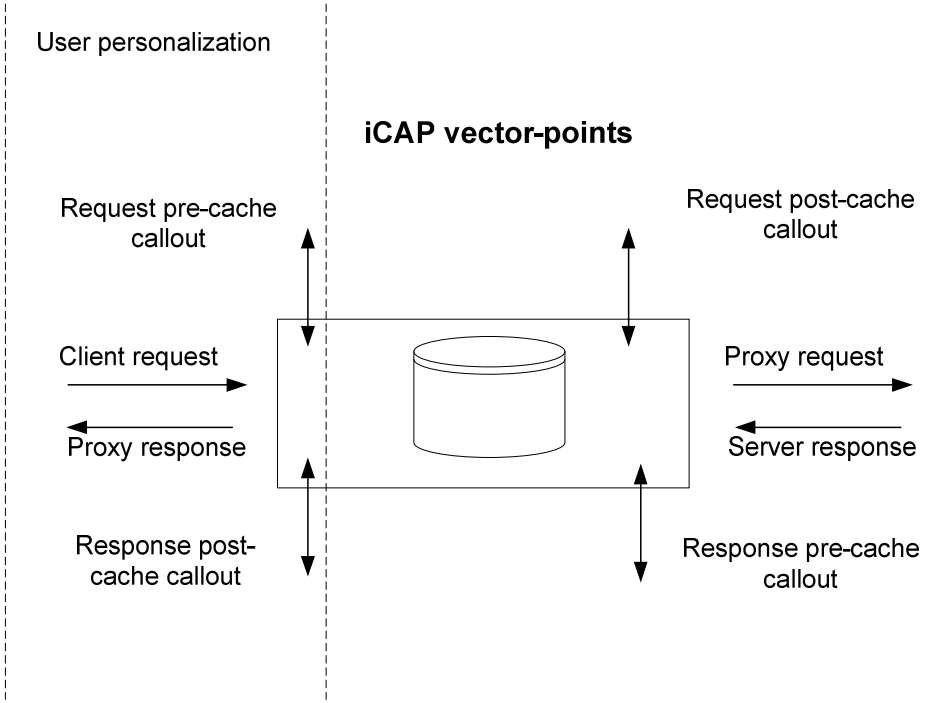
Service à Valeur Ajoutée en cœur de réseau : le protocole iCAP

Les deux modes de fonctionnement :



Service à Valeur Ajoutée en cœur de réseau : le protocole iCAP

Les quatre "vector-points" :



Service à Valeur Ajoutée en cœur de réseau : le protocole iCAP

- Quelques services types en reqmod :

  – Contrôle parental/patronal par liste URLs autorisées ("liste blanche") et/ou interdites ("liste noire").

  – Liste des services utilisables par un utilisateur en fonction de son profil.

  – LEP SSO (Single Sign On)

- Quelques services types en respmod :

  - Contrôle parental par filtrage de contenus de page.

  – Traduction de pages à la volée.

  – Compression de la réponse du serveur Web pour économiser de la bande passante (important dans le monde mobile).

  –Suppression des composants dangereux, interdits ou non certifiés par l'éditeur dans un page Web.

  – Application de feuilles de style (CSS) pour respecter une charte graphique prédéfinie.

  – Rajout d'un bandeau comportant différents items en haut de chaque page. Ce bandeau peut être personnalisé en fonction du profil utilisateur.

  – LEP SSO (Single Sign On)

Idées de services pour "l'apprentissage mobile"

  – Sélection du support de cours adapté en fonction du profil utilisateur (langue, cours déjà vu) : REQMOD.

  – Adaptation de la page aux caractéristiques et capacités réseau du terminal : RESPMOD.

  – Obtention de la géolocalisation mobile ou WiFi d'un utilisateur : REQMOD.

  – Liste des apprenants connectés aux différentes applications, ainsi que leur "état" : RESPMOD.

Service à Valeur Ajoutée en cœur de réseau : conclusion

- Les protocoles "orienté binaire" donnent moins de possibilités d'enrichissement / modification que les protocoles "orienté texte". Les enrichissements / modification des protocoles "orienté binaire" se limitent en général à :

- Aux ACLs disponibles pour tous les protocoles (limitation du débit et de la taille de l'objet, filtrage @IP source et @IP destination, port source, port destination, … en fonction du profil utilisateur)
- Aux applications de protection du contenu / filtrage du contenu (AntiVirus, composant signé). Ces applications étant souvent le résultat d'un partenariat précis entre un constructeur d'équipement et un éditeur de logiciel, le tout fonctionnant de façon propriétaire.

- HTTP est le protocole permettant le plus de latitude par modification / enrichissement :

- des entêtes (requête et réponse),
- de la requête,
- de l'objet renvoyé en réponse

en plus des possibilités offertes par les ACL disponibles pour les autres types de protocole.

- Les ACL permettant de faire la composition de services en coordonnant l'ordre de leurs appels.

4 Création de logs et centralisation automatique

Création de fichiers de log

- Tous les serveurs d'applications et beaucoup de composants réseaux (proxy, WAF, annuaire) créent des fichiers de log au format "Common Log" et "extended" du W3C.

- Des outils de génération, de centralisation et de gestion de logs ont été crées pour faciliter la gestion de logs de toute sorte :

  –Syslogd ([SYSLOG]) pour les applications en C/C++ principalement.

  – Log4J ([LOG4J]) pour les applications sous Java (package org.apache.log4j).

  Ces deux outils, par leur commodité :

  - ont été souvent portés sous d'autres langages,

  - sont souvent complétés par de nombreux logiciels compagnons pour étendre encore leurs possibilités.

Création et gestion de logs : Syslogd ([SYSLOG])

Syslog utilise le daemon syslogd sous UNIX pour définir pour chaque programme des fichiers de log sous la forme :

<date> <Nom DNS de la machine> <Service.Niveau de criticité> <Message>

En paramétrant dans le fichier syslogd (syslog.conf) :

- les services et niveaux de criticité à logger.

- Les noms des fichiers de logs

- Les machines (noms DNS ou adresses IP) sur lesquelles envoyer les fichiers de log.

-Les paramètres de rotation des fichiers de log : par taille, périodicité notamment, compression, nombre de fichiers historiés avant suppression.

Syslogd permet de centraliser de gros fichiers de logs sur une machine dédiée.

Syslogd autorise une grande souplesse de l'organisation des fichiers de logs et permet une administration système "propre".

Création et gestion de log : Log4J ([LOG4J])

Génération de traces de façon hiérarchique et modulaire à l'aide de "loggers".

- Six niveaux de traces : trace < debug < info < warn < error < fatal (par ordre de criticité/priorité croissante).

- Organisation de façon hiérarchique des noms des "loggers" : existence d'un "logger" "root" parent de tous les loggers de l'application. Tous les autres loggers seront définis sous la forme d'une arborescence de syntaxe de type DNS.

- Définition d'un niveau de priorité par défaut p : ne logger que les messages de priorité q>=p.

- Définition de la destination des traces ("Appenders") : de type console (sortie ou erreur standard), fichiers, composants graphiques, socket serveur distant, message JMS ou démon syslog d'un système UNIX distant.

- Définition des formats de présentation= ("Layers") : définition des champs composant la ligne de log.

- Définition des paramètres dans un fichier de configuration au format texte de la forme paramètre=valeur ou au format XML.

- Héritage par défaut entre un logger parent/ancêtre et un des descendants à la fois des niveaux de priorité par défaut si le niveau de priorité du logger courant n'est pas défini et des "appenders" de façon "additive" (les "appenders" des enfants sont ajoutés à ceux des parents/ancêtres). Ces deux héritages peuvent être désactivés pour chaque descendant.

5 Exemple de plate-forme de centralisation et traitement des traces, d'appels de services externes

Plate-forme : Expression des besoins

- Fonctionnalités souhaitées de la plate-forme centralisée de gestion des logs

  – Ouverture : protocoles ouverts, normalisés et standard industriel. Code source de la plate-forme librement modifiable par Orange.

  – Géolocalisation d'un mobile par le réseau Orange : 1er service externe à intégrer*.

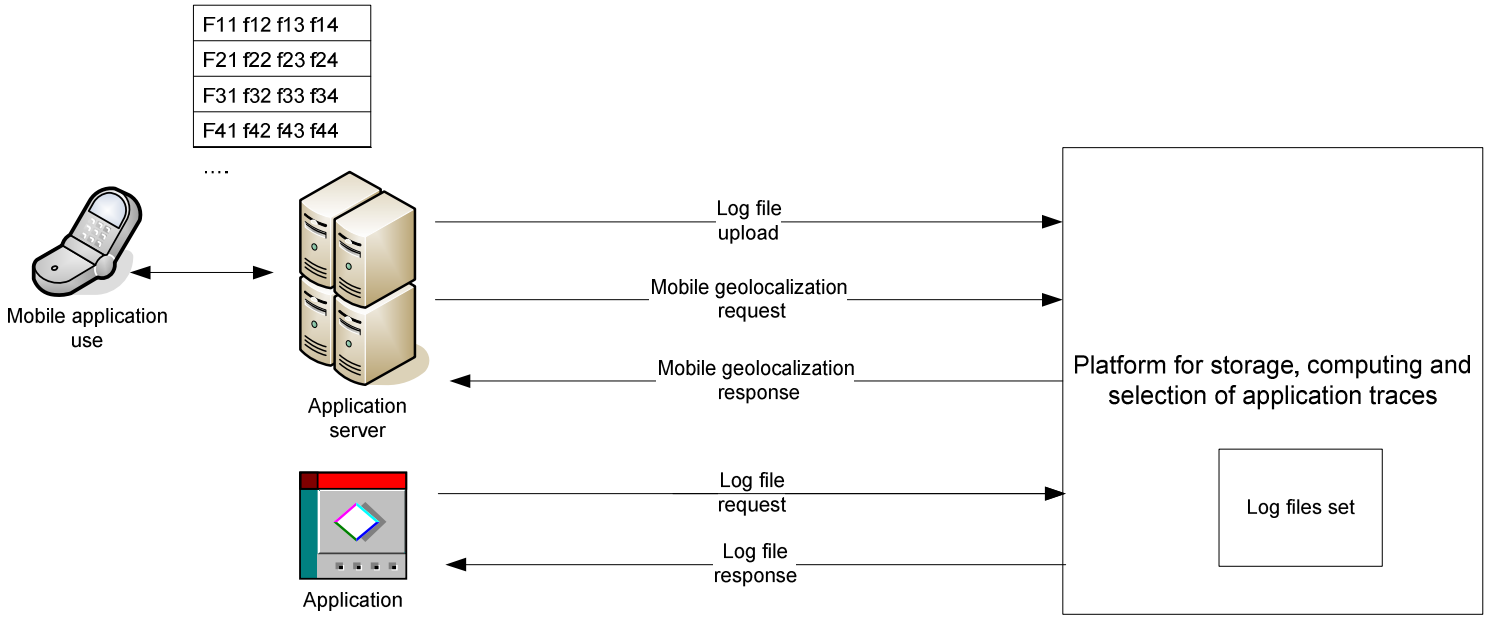
  – Respect de la loi "Informatique et Libertés" : les traces applicatives contiendront souvent des "données à caractère personnel". ([CNIL], [G29])

  – La plate-forme est accessible depuis Internet : connexion à travers des APIs publiques. Cela facilite la mise en place d'une politique d'accès cohérente sur la plate-forme.

  *Dans cette expérimentation, la géolocalisation est faite par une plate-forme réservée aux expérimentations. Cette plate-forme supporte beaucoup de formes de géolocalisation (IP, réseau mobile, A-GPS, WiFi Orange…) et peut-être interrogée en utilisant des requêtes XML sur HTTPS.

  /!\ Cette plate-forme est un "PoC" (Proof of Concept) pour mesurer l'intérêt et les limites de bâtir une plate-forme centralisée de gestion de log.

Plate-forme : flux d'entrées/sorties



La plate-forme recevra :

- des fichiers de logs au format texte,

- des demandes de géolocalisation de mobiles avec stockage de la requête et des réponses,

- des requêtes sur les différents fichiers de traces ou de géolocalisation stockés sur la plate-forme.

La plate-forme enverra des réponses au format texte, CSV ou XML suite à une requête sur les différents fichiers de traces stockés sur la plate-forme correspondant :

- aux fichiers de log envoyés par les applications,

- aux coordonnées de géolocalisation des mobiles suite à des requêtes applicatives.

Plate-forme : choix composants logiciels et protocoles réseau

- Premier choix : utilisation d'un SGBD/R pour stocker les fichiers de log, les requêtes et les réponses de géolocalisation mobile. Cela permet de bénéficier des performances et des richesses fonctionnelles de ce type d'outil (héritage, procédure événementiel, procédures stockées…).
- Deuxième choix : HTTP devient LE protocole universel. Pour envoyer des fichiers et sélectionner un ensemble de paramètres l'association **formulaire HTTP + méthode POST basique** est très efficace. Ce choix supprime les limitations de la méthode GET (manque de confidentialité, taille limitée de la requête…) et évite les défauts des WebServices pour ce genre de service (lourdeur protocolaire, nécessité de définir précisément les interfaces des services …).
- Troisième choix : nous ne prenons que des logiciels libres réputés pour leur maturité et leur ouverture (en termes d'opérabilité protocolaires et interfaçage avec d'autres applicatifs). D'où la sélection de **Linux**, **Tomcat** et **PostgreSQL** pour constituer notre plate-forme.

Plate-forme : choix des fichiers de logs et politique de privacy

- Un fichier de log = (qui, quand, quoi) en général dans un fichier texte. Pour simplifier, le format "Common Log" du W3C a été retenu ([LOGW3C]).
  - **remotehost, rfc931, authuser, [date], "request", status, bytes.**

Bien que simple, il offre beaucoup de possibilités et est quasi-universel.

- Politique de privacy choisie :

"**Des applications** génèrent des fichiers de log ou des demandes de géolocalisation de mobiles si elles ont les droits adéquats. D'autres applications (éventuellement les mêmes) peuvent lire les informations relatives à ces fichiers de log ou positions de géolocalisation de mobiles si elles ont les droits adéquats".

Droits données par l'administrateur du SGBD de la plate-forme. Ils sont de deux types :

READ : **l'application** pourra accéder aux enregistrements du fichier de log considéré contenus dans les tables des Bases de Données en lecture seule.

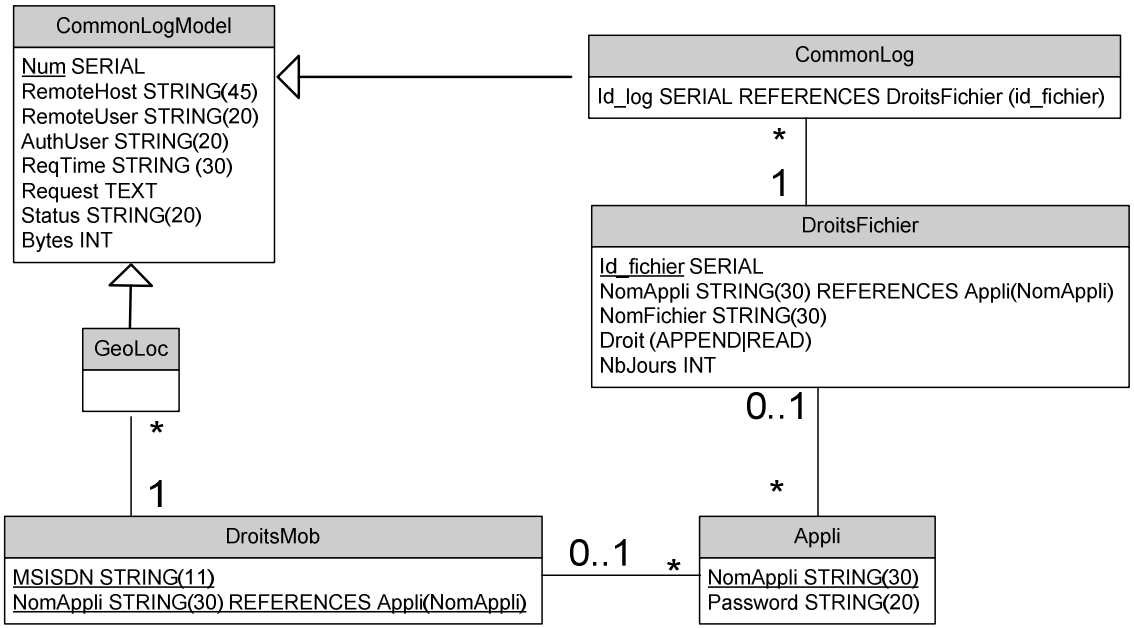
APPEND : **l'application** pourra accéder aux enregistrements du fichier de log considéré contenus dans les tables des Bases de Données en lecture ou en mode ajout.

Pas de droits **modification** ou **suppression** pour les applications. Seul l'administrateur peut le faire.

Sur la géolocalisation mobile le droit READ n'existe pas. Seules les **applications** habilitées à demander la géolocalisation d'un mobile (*i.e.* le droit "APPEND") peuvent accéder à ces informations.

Pas de droits, pas d'accès

Plate-forme : Schéma de la Base de Données



- Remarques :

  – Les tables CommonLog et GeoLoc héritent de CommonLogModel : interrogation des trois tables à volonté.

  – Usage de la fonction TEXT de PostgreSQL pour stocker des requêtes de taille quelconque.

  –Presque tous les champs sont des chaînes de caractère : possibilité de stocker n'importe quel type d'informations. La signification de chaque champ est laissé sous la responsabilité des applications qui les utilisent (ce sont **leurs traces et elles n'ont de sens que pour elles**)

Plate-forme : quelques conclusions

Cette plate-forme centralisée a montré beaucoup d'avantages pour gérer la Privacy. Cependant les obligations légales liées à la privacy nécessitent une réponse au niveau organisationnel de l'entreprise en réunissant les différents acteurs concernés - CIL, juriste, DSI.

Ce type de plate-forme permet d'éviter les problèmes typiques d'une plate-forme mal conçue :

- mal sécurisée = confidentialité compromise.

- Mal dimensionnée = goulet d'étranglement sous forte charge.

- Mal supervisée/administrée = défaillance des services se basant sur elle.

En effet les serveurs Apache, Tomcat, SGBD/R PostgreSQL supportent le protocole SSL/TLS et sont tous capables de faire de l'équilibrage de charge applicatif ce qui permet une réelle scalabilité de la plate-forme..

De plus :

- protocole HTTP/HTTPS = capacité d'intégration dans un SI déjà existant+ sécurisation des accès.

- Enrichissement des flux entrants et sortants de la plate-forme en utilisant des services externes (*e.g.* géolocalisation mobile, géolocalisation des utilisateurs par leur adresse IP…) pour fournir des fonctionnalités à toutes les applications clientes de la plate-forme tout en respectant une politique d'accès forte à ces fonctionnalités.

- Politique d'accès centrée sur les applications = Les applications "ayant-droit" accèdent à toutes les informations pertinentes pour elles.

-Base de données "objet" = capacités d'héritage des tables entre elles pour enrichir facilement les capacités de la plateforme et la politique de Privacy mise en place.

6 Conclusions

Conclusions

- Par construction le réseau est :

  - une place privilégiée pour la connaissance d'informations "contextuelles" sur un utilisateur ou un groupe d'utilisateurs.
  - Une place privilégiée pour construire des "enablers" pour toutes les applications.
  - Un élément important du contexte lui-même par sa typologie physique même : géolocalisation par le réseau mobile de l'opérateur (*e.g.* "cell-id"), @IP (*e.g.* librairie geoIP), ou le calcul de la distance physique entre l'utilisateur et le serveur.

- L'interception/redirection des informations en cœur de réseau ne peut pas tout faire mais … elle permet beaucoup !!

  – Appliquer des politiques de sécurité uniforme à tous les flux sans exception et facilitant l'administration (*e.g.* pas de configuration sur les postes clients).
  – Composer des applications personnalisées facilement en ordonnançant à volonté les différents services et applications *(e.g*. ACLs).
  – Enrichir/modifier les flux en rajoutant des entêtes notamment pour tous les flux applicatifs basés sur HTTP.

Conclusions

- Evolution des composants réseaux : les rôles de chacun ne sont plus figés :
  - Les routeurs font aussi de l'analyse de port, de la redirection vers des proxy.
  - Les sondes deviennent des DPI et font aussi de la redirection.
  - Les proxy deviennent des passerelles de sécurité applicative et enrichissent les flux.
  - Avec l'apparition progressive de composant réseau jouant sur tous les tableaux.

  Tout cela permet de construire des applications facilement par composition de services réutilisables à partir d'éléments situés en cœur de réseau

  - /!\ Privacy !! Pouvoir ➔ responsabilité envers les utilisateurs.

Merci

Glossaire

ACL : Access Control List : règle de mise en œuvre d'une politique d'accès à une ressource.

DoS : Denial of Service : attaque d'un serveur par génération d'un grand nombre de requêtes simultanées pour l'empêcher de fournir le service prévu lors de sa conception.

DPI : "Deep Packet Inspection" : sonde applicative (niveau 7 ISO) qui est capable d'identifier les applications contenues dans les trames TCP ou UDP qu'elle reçoit par analyse de ces dernières.

Failover : Redondance de machines permettant de suppléer à la défaillance de l'une d'entre elles de façon automatique.

ICAP : Internet Content Adaptation Protocol : protocole permettant la modification de la requête ou de la réponse HTTP. Protocole défini par le RFC3507 IETF.

Load-balancing : "équilibrage de charge" : permet de répartir les requêtes sur un ensemble de serveurs supportant la même application pour éviter de faire crouler sous la charge un des serveurs.

POC : Proof Of Concept : Preuve de faisabilité d'un service. En général concrétisé par une maquette.

Privacy : protection des données à caractère personnel : protection de la vie privée.

Proxy : "serveur mandataire" : passerelle applicative.

Switch L4/L7 : switch capable d'identifier et de prendre des décisions sur le numéro de port TCP contenu dans les trames TCP ou UDP.

WAF : Web Application Firewall : logiciel ou/et matériel qui protège les applications Web HTTP(S) des attaques logicielles visant les sites Web.

WCCP : Web Cache Communication Protocol : protocole de communication entre des routeurs et des proxy.

Références

Problématique générale

[EIAH07] Alain Ottavi, Sylvain Baron, Luigi Lancieri. Capture et exploitation de traces dans un contexte de mobilité. EIAH 2007, Lausanne, Juin 2007.

[OTT08] Alain Ottavi. "Plate-forme d'interrogation de services externes, de collecte, traitement et mise a disposition de traces applicatives", Mémoire Ingénieur CNAM, 2008.

Fichiers de log

[LOG4J] Log 4 Java, http://logging.apache.org/log4j, Juin 2009.

[LOGW3C] Logging Control In W3C HTTPD, http://www.w3.org/Daemon/User/Config/Logging.html, Juin 2009.

[SYSLOG] Système de génération et de collecte de logs : http://www.syslog.org, Juin 2009.

HTTP, Proxy et WAF

[BC] Blue Coat : http://www.bluecoat.com, Juin 2009.

[OPTENET] Optenet : http://www.optenet.com, Juin 2009.

[RFC2616] norme IETF n° 2616 définissant le protocole HTTP/1.1.

[RFC3507] norme IETF n° 3507 définissant le protocole iCAP.

[SQUID] proxy-cache logiciel sous licence GPL : http://www.squid-cache.org, Juin 2009.

[WAF] Web Application Security Consortium : http://www.webappsec.org, Juin 2009.

Références

Logiciels de la plate-forme

[PGPOOLII] PGPOOL-II : permet de faire de la réplication, du fail-over et du load-balancing entre plusieurs serveurs PostgreSQL, http://pgpool.projects.postgresql.org/, Juin 2009.

[PGSQL] PostgreSQL : SGDB relationnel, http://www.postgresql.org, Juin 2009.

[TOMCAT] http://tomcat.apache.org, Juin 2009.

Privacy

[CNIL] Commission Nationale Informatiques et Libertés, http://www.cnil.fr , Juin 2009.

[G29] Groupe de travail Article 29 sur la protection des données, http://ec.europa.eu/justice_home/fsj/privacy/index_fr.htm, Juin 2009.