LES SERVICES DE L'INTERNET

Noureddine AWANE

noureddine.awane@sorbonne-paris-nord.fr

Département R&T, IUT de Villetaneuse, Université Sorbonne Paris Nord

Plan

- **Le modèle OSI**
- **L'Internet**
- **Le modèle TCP/IP**
- **Les services Internet :**
  - **Le système de résolution de noms DNS**
  - **Les protocoles de messagerie : SMTP, POP3, IMAP**
  - **Le service de transfert de fichiers :**
    - **FTP**
    - **FTPS**
  - **Les protocoles du web :**
    - **HTTP**
    - **HTTPS**
- **Annexe: Linux**

Le modèle OSI

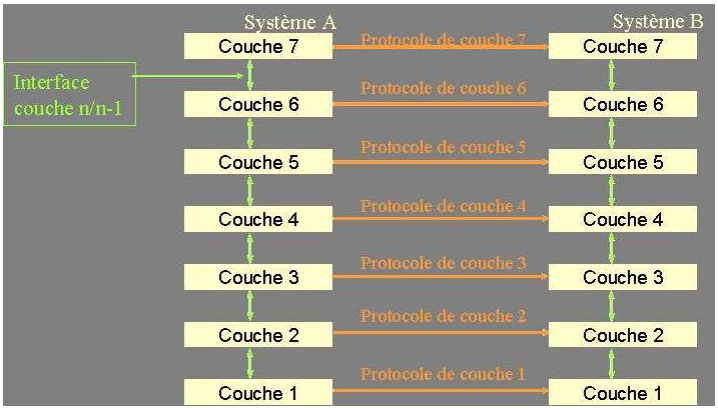
3

Le besoin

- **Les premiers réseaux informatique sont basés sur des technologies propre à chaque constructeur => impossibilité d'interconnecter ces différents réseaux entre eux.**
- **Le modèle OSI (Open Systems Interconnection) publié en 1978 par l' ISO (International Standard Organization) décrit un ensemble de spécifications théoriques décrivant une architecture réseau permettant la connexion d'équipements hétérogènes => assurer la compatibilité**

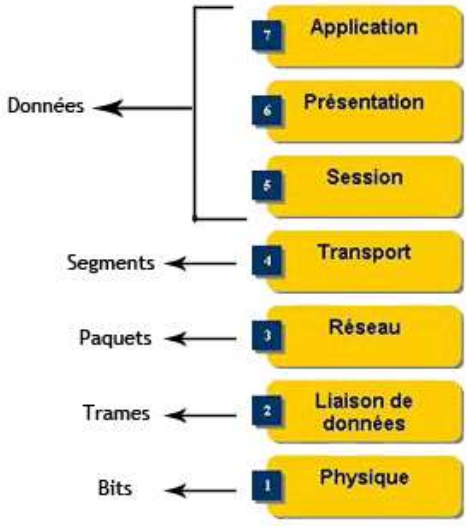
4

Systéme en couche



5

Le modèle OSI



6

Rôle des couches

Les 7 couches du modèle OSI sont les suivantes :

- La couche 7 (application) => interface avec les applications.
- La couche 6 (présentation) => représentation des données compression/chiffrement.
- La couche 5 (session) => ouverture/fermeture des sessions de communication.
- La couche 4 (transport) => transport des données, découpage/réassemblage/gestion des éventuelles erreurs de transmission.
- La couche 3 (réseau) => adressage et routage des données.
- La couche 2 (liaison) => interface avec la carte réseau et le partage du média de transmission.
- La couche 1 (physique) => conversion des données en signaux physiques sur le média de communication.

L'Internet

Historique d'internet

- Née du besoin du DoD (Departement of Defense) de pouvoir interconnecter des réseaux isolées avec des protocoles incompatibles entre eux .
- En 1962, mise en place du projet ARPA (Advanced Research Projects Agency) qui consiste en la mise en place d'un réseau interconnectant des ordinateurs hétérogène (OS et constructeurs différents)
- En 1969, ARPANET est le premier réseau à commutation de paquet
- RFC (request for Comments) a pour but de présenter les travaux des différents chercheurs
- En 1972 : mise au point de TCP/IP (détection de pertes et de retransmission) qui sera adopté pour l'utilisation d' ARPANET dix ans plus tard
- En 1984, les premiers serveurs DNS sont utilisés
- En 1990, HTTP permettant l'utilisation de lien hypertextes

Les organisations de normalisation

- ISOC (Internet society) : travaille sur le développement de l'Internet en supervisant l'IESG et l'IAB
- IESG (Internet Engineering Steering Group): établit les spécifications et réalise les implantations des nouveaux protocoles définit par l'IETF
- IAB (Internet Architecture Board) : comité qui supervise l'IETF et l'IRTF
- IRTF (Internet Research Task Force) : prépare l' évolution des protocoles, des architectures et des technologies - sur le long terme
- IETF : fournir les RFC (normes) – sur le court terme

Le modèle TCP/IP

11

Le modèle TCP/IP

Couche	Nom	Description
4	Application	Couches 7 à 5 du modèle OSI
3	Transport	Qualité de transmission
2	Internet	Sélection du chemin
1	Accès au réseau	Reprend les couches 1 et 2 du modèle OSI

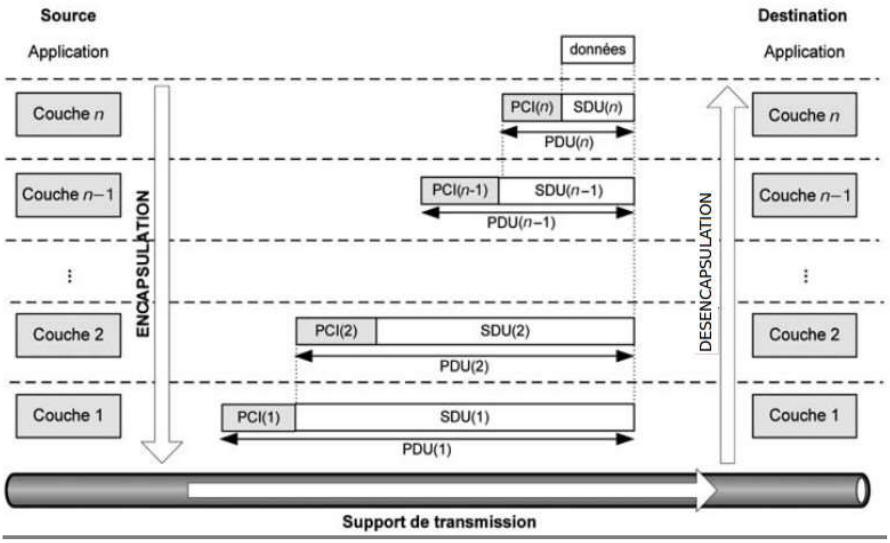
12

L'Encapsulation

- Chaque protocole ajoute un en-tête contenant toutes les informations nécessaires au traitement
- A l'émission, la couche n encapsule les données de la couche n+1
- A la réception, la couche n desencapsule les données de la couche n-1.

- Vocabulaire
  - PDU (Protocol Data Unit, unité de données de protocole) : paquet provenant de la couche supérieure + en-tête
  - SDU (Service Data Unit, unité de données de service) : données issues de la couche supérieure
  - PCI (Protocol Control Information, information de contrôle du protocole) : entête

La traversée des couches



Les caractéristiques du modèle TCP/IP

- Protocoles ouverts
- Standardisation des protocoles (couches réseau, transport, application)
- Protocoles indépendants du niveau physique→ encapsulation dans divers technologies : Ethernet, WiFi

15

les spécificité des protocoles IP, TCP et UDP

- Internet Protocol (IP) : protocole de la couche réseau fournissant une méthode d'adressage et de routage
  - Travaille en mode datagramme et non fiable
- Transport Control Protocol (TCP) : protocole de la couche transport
  - Travaille en mode connecte et en mode fiable
  - Assure le contrôle de séquencement, de flux et la congestion
  - Utilise par les applications nécessitant des données intègres (web, messagerie, transfert de fichier)
- User Data Protocol (UDP) : protocole de la couche transport
  - Travaille en mode datagramme et non fiable
  - N'assure aucun contrôle (ni séquencement, ni flux, ni congestion)
  - Utilise par les applications privilégiant la rapidité de transmission, supportant des pertes (DNS, DHCP)

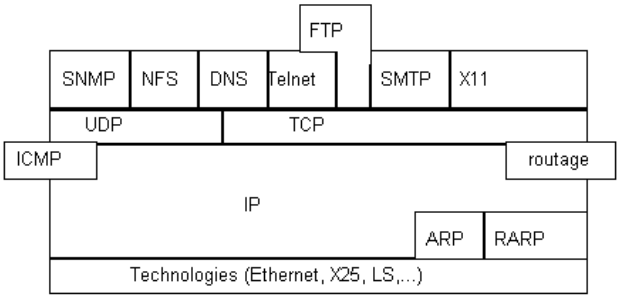
16

Définitions

- **Le séquencement : contrôle de l'ordre des paquets**
- **Le contrôle de flux : l' émetteur adaptation son débit au récepteur**
- **Le contrôle de congestion : éviter la saturation du réseau**
- **Mode fiable : utilisation d' accuses de réception; retransmission si perte ou erreur**
- **Mode connecte (vs. mode non connecte ou mode datagramme) : prise de contact par l' émetteur avant l'envoi des données pour s'assurer de la disponibilité du récepteur**

17

Les protocoles



18

Les applications sur Internet

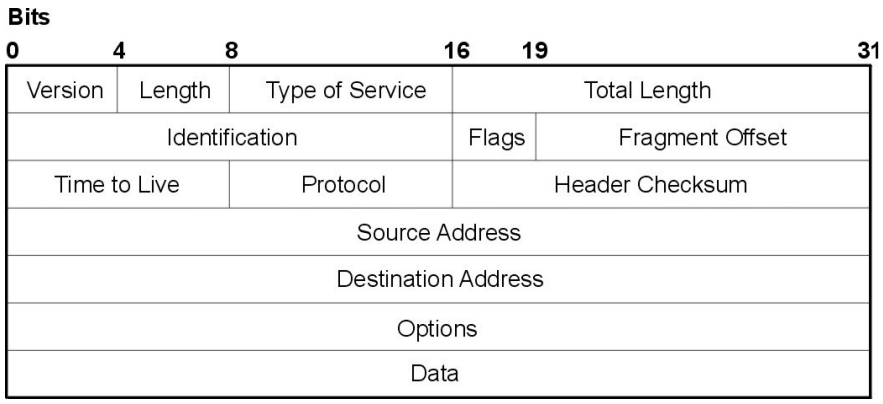
- **Les applications fonctionnent en mode client/serveur**
  - **Serveur : programme qui rend un service ; il attend les requêtes d'un client pour exécuter le service**
  - **Client : programme qui permet de solliciter un service**

19

La couche réseau : IP

20

En-tête IP



Les champs

- **version (4bits) : version du protocole (IPv4, IPv6)**
- **length (4bits) : taille de l'entête en mot de 4octets**
- **Type of service (8bits) : gestion de la QoS**
- **Total length (16bits) : taille de l'entête et de la data**
- **identification (16bits) : reconstitution des fragments ayant le même identifiant**
- **Flag (3bits) : 0-DF-MF (Réservé - Don't Fragment – More Fragment)**
- **Fragment offset (13bits) : position du fragment**
- **Time To Live (8bits) : durée de vie du paquet**
- **Protocol (8bits) : type de donnée de la data**
- **Checksum (16bits) : détection d' erreur**
- **Source Address (32bits) : @IP source**
- **Destination Address (32bits) : @IP destination**
- **Option (multiple de 32bits) : champs optionnel**

L'adressage

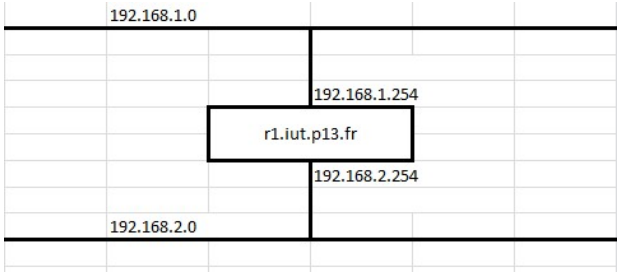
- Adresse IP codé sur 32 bits.
- Adresse IP représenté par 4 nombres séparés de points (exemple : 192.168.1.43)
- découpée en 2 parties : numéro de réseau et numéro de machine dans le réseau

Classe A	1.0.0.1 à 127.255.255.255	Premiers bits = 0
	127 réseaux - 16777216 machines	
Classe B	128.0.0.1 à 191.255.255	Premiers bits = 10
	16384 réseaux - 65536 machines	
Classe C	192.0.0.0 à 223.255.255.255	Premiers bits = 110
	2097152 réseaux - 256 machines	

- deux adresses machines spéciales :
  - tous les bits à 0
  - tous les bit à 1
- Un adresse Réseau spéciale : 127.X.X.X

GATEWAY

- Une Adresse IP est associé à une interface.
- le routeur r1.iut.p13.fr a deux interfaces, il a donc deux adresses IP



- Passage d'une interface à une autre par routage

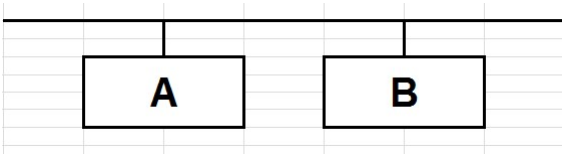
ASSOCIATION IP/MAC

- **Comment associer une adresse IP à une adresse physique ?**
  Résolution dynamique ARP

25

Résolution dynamique : le protocole ARP

- **Exemple de résolution dynamique : Ethernet, protocole ARP (Address Resolution Protocol)**



1. **Message (broadcast Ethernet) de A**
   Question = Etant donné IP(B) que vaut Eth(B)
2. **Message de B à A**
   Réponse = Voici mon adresse Ethernet Eth(B)

26

Cache ARP

- Un paquet IP envoyé => un échange ARP (1broadcast+1réponse)
  =>Trafic énorme
- Solution : chaque machine conserve les dernières transactions dans un cache

27

La couche transport

- Le protocole TCP
- Le protocole UDP

28

la couche transport

- Réalise la connexion de bout en bout entre deux processus
- Efface les faiblesses rencontré sur le réseau

- Deux protocoles essentiels fonctionnant en mode client/serveur
  - TCP : Transmission Control Protocol
  - UDP : User Datagram Protocol

Les sessions TCP et UDP

- Les paquets d'une même session son définit par le quadruplet (@IPsrc, port src, @IPdst, port dst)

- Exemple : Connexion sur un serveur HTTP
  - Client : @IP = 192.168.1.1, port =45659
  - Serveur : @IP = 192.168.1.2, port = 80
  - session = (192.168.1.1;45659;192.168.1.2; 80)

Les ports TCP et UDP

- Codés sur 2 octets
- Pour les services connus définit entre 0 et 1023
- Pour les clients les ports sont choisi de 1024 a 65535

31

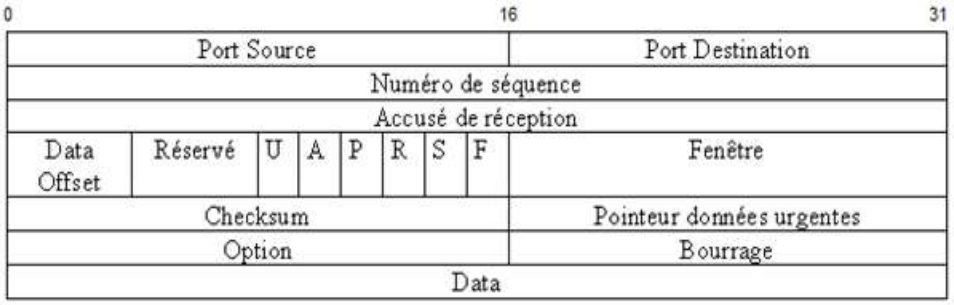
Le protocole TCP (RFC 793)

- Application avec comme avantage l' intégrité des données
- Identifié dans le paquet IP par le champ protocole numéro 6
- Mode connecté
- Mode fiable
- Contrôle de séquencement, de flux, de congestion

PORT	NOM	SIGNIFICATION
21	ftp	transfert de fichier
22	ssh	connexion à distance
23	smtp	transfert de courrier électronique
53	domain	serveur de nom

32

En-têteTCP



33

Les champs de l' ên-tete TCP

- **Source Port (16 bits) : numéro du port qui identifie l'application émetteur**
- **Destination Port (16 bits) : numéro du port qui identifie l'application récepteur**
- **Num de sequence (32 bits) : numéro de paquet**
- **Accusé de réception (32 bits) : prochain numéro de paquet attendu**
- **Data Offset (4 bits) : taille de l' entête multiple de 32 bits**
- **Réservé (6 bits) : toujours a 0**
- **Drapeau (6 bits) : URG, ACK, SYN, PSH, RST, FIN**
- **Fenêtre (16 bits) : nombre d'octets qu'on peut recevoir a la fois avant acquittement (contrôle de flux).**
- **Checksum (16 bits) : détection d'erreurs sur l' entête et les données**
- **Urgent Pointer (16 bits) : indique le dernier octet urgent a transmettre si flag URG a 1**
- **Options (variable) : permet d'ajouter des options**
- **Bourrage (variable) : comble la taille du champs options pour avoir une entête multiple de 32bits**

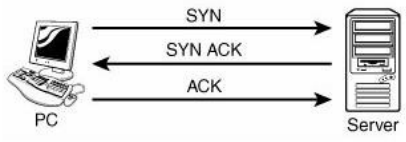
34

Les flags

- URG : indique que le champ Pointeur de donnée urgente est utilisé (data à transmettre rapidement à la couche applicative ex : interruption)
- ACK : Indique que le segment transporte un accusé de réception
- SYN : Utilisé pour établir la connexion TCP
- PSH : indique que les données peuvent être envoyer à la couche applicative sans passer par la mise en tampon
- RST :réinitialisation de la connexion.
- FIN : indique la fin de la connexion.

Ouverture d'une connexion TCP

- Le serveur ouvre une socket pour se mettre en attente
- Le client contacte l'application distante pour s'y connecter
- Segments d'ouverture en 3 phases



Fermeture d'une connexion TCP

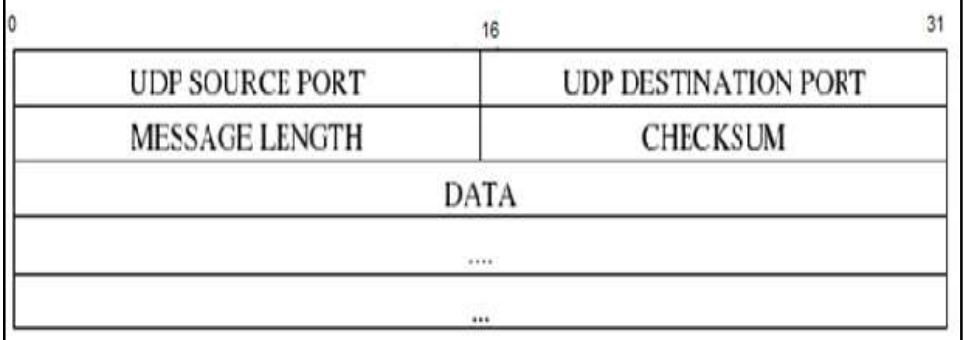
- Le client indique la fin de la connexion
- Le serveur libère les ressources
- Segment de fermeture en 3 ou 4 segments

Le protocole UDP

- Application avec comme avantage la rapidité de traitement et la réduction de trafic
- Identifié dans le paquet IP par le champ protocole numéro 17
- Pas de connexion, pas de séquencement, pas de fiabilité, pas de contrôle de flux

PORT	NOM	SIGNIFICATION
53	domaine	serveur de nom
69	tftp	transfert de fichiers
161	snmp	gestion de réseau

En-tête UDP



39

Format du datagramme UDP

- **Source Port (16 bits) : numéro du port qui identifie l'application émetteur**
- **Destination Port (16 bits) : numéro du port qui identifie l'application récepteur**
- **Length (16 bits) : Longueur de tout le datagramme**
- **Checksum (16 bits) : détection d'erreur**

40

Les services Internet

41

La couche applicative : DNS

42

Principes du DNS

- **Problématique : Les équipements communiquent grâce à leur adresse IP – Pas facile à mémoriser.**
- **Solution :**
  - **nom machine unique**
  - **Correspondance entre le nom machine et son adresse**
- **Comment : ?**

Résolution statique

- **Mise en places d'un fichier centralisé qui renseigne la correspondance nom machine/adresse IP**
- **Réplication de ce fichier vers toutes les machines du réseau**
- **/etc/hosts sous linux**

```
# Do not remove the following line, or various programs that
# require network functionality will fail..
127.0.0.1       localhost.localdomain localhost

# Messagerie
195.242.149.64  smtp1.iut.p13.fr
195.242.147.66  smtp2.iut.p13.fr

### Divers
10.196.2.45     srvbackup.iut.p13.fr
10.162.216.1    srvftp.iut.p13.fr
```

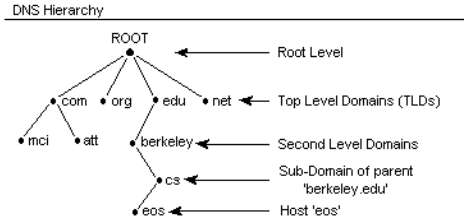
- **Responsabilité donnée à l'administrateur du domaine pour les mises à jour**
- **Inconvenant : taille du fichier devient vite ingérable**

Résolution dynamique

- Le service DNS est une base de données distribuée de maniéré hiérarchique qui permet les fonctions suivantes :
  - La conversion nom en adresse IP
  - La conversion adresse IP en nom
  - Le routage de courrier électronique
  - Et d'autre informations…

L'espace de noms hiérarchique

- DNS est basé sur la notion de domaines contenant des sous-domaines, qui contiennent à leur tour des sous-domaines, etc….
  - La racine est notée ". "
  - Top Level Domains :
    - Generique : .com, .org, .edu,
    - Pays (2lettres) : .fr, .uk, .de,
  - Secondary Level Domains : orange, paris-13,
- La combinaison du nom de la machine et des domaines auquel il est rattaché constitue le FQDN (exemple : eos.cs.berkeley.edu)
- L'AFNIC est responsable de la gestion du .fr



Les fonctions

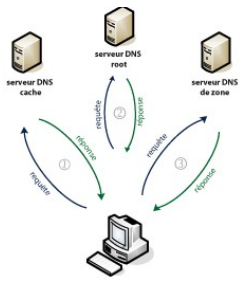
- Le serveur DNS d'une zone possède :
  - les @IP des serveurs racines
  - Les ressources de sa zone
  - les serveurs DNS de ces sous-zones
- Il peut y avoir plusieurs serveurs DNS pour une même zone (redondance, répartition de charge)
- Un serveur DNS peut travailler pour plusieurs zones (redondance, mutualisation)
- Il n'y a qu'un seul DNS primaire par zone les autres sont généralement appelé serveur secondaire

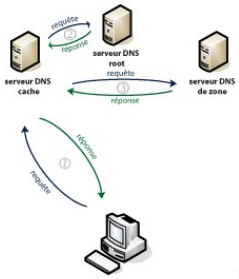
Réplication vers les DNS secondaire

- Transfert de zone :
- Les DNS secondaires (slaves) demandent le SOA de la zone au master NS a un intervalle appelé le refresh-time.
- Si le serial a été incrémente, ils demandent le transfert de zone.
- Le transfert de zone est réalisé dans une connexion TCP.
- DNS Notify (RFC 1996) : Permet à un primaire de prévenir ses secondaires d'un changement dans une zone
- Incremental zone transfer (IXFR, RFC 1995) : Permet à un secondaire de mettre a jour ses zones uniquement avec les données modifiées depuis le dernier échange avec le primaire
- DNSSEC (RFC 2535) permet de sécuriser les updates en authentifiant les clients.

La résolution

- Une requête est une demande de résolution de noms envoyée à un serveur DNS
- Il existe trois types de résolution :
  - Résolution itérative
  - Résolution récursive
  - Résolution mixte

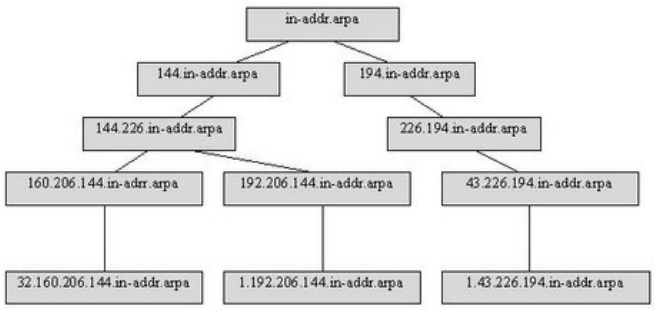




49

La résolution inversé

- Résolution inversé : Obtenir le nom réel de la machine à partir de l' adresse IP
- En inversant l'ordre des octets de l'adresse IP auquel on ajoute .in-addr.arpa

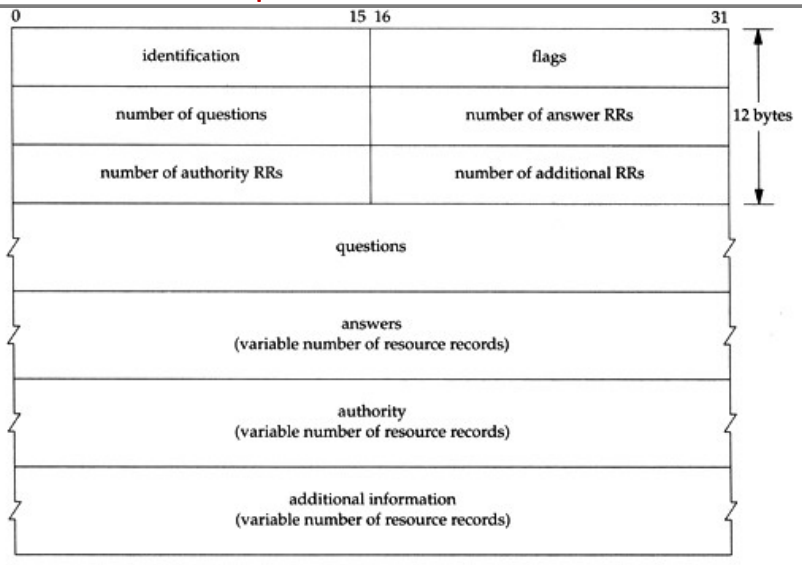


50

RR

- Les différents types d'enregistrement (Resources Record)
  - SOA : paramètre global de la zone
  - A : adresse IP associée a un nom
  - AAAA : adresse IPv6 associée a un nom
  - NS : serveur de nom ayant autorité pour le domaine
  - PTR : nom réel de la machine auquel est associé l'adresse IP
  - MX: serveur de messagerie pour la zone
  - CNAME : alias vers un autre nom

La Format des requête DNS



Les champs de l' ên-tete

- Identification (16 bits) : l'id permet d'associer la demande à la réponse
- Flag (16 bits) : voir encadré plus bas
- Number of questions (16bits) : Nombre de questions
- Number of responses RRs (16bits) : Nombre de réponses
- Number of authority RRs : Nombre d' entrée dans la section Authority
- Number of additional RRs : Nombre d' entrée dans la section Additionnelle

Le champs flags

- QR (1bit) : indique s'il s'agit d'une requête (0) ou d'une réponse (1)
- opcode (4bits) : type de la requête 0 (requête standard) - 1 (requête inversée) - 2 (status d' une requête serveur).
- AA (1bit) : Le serveur est authoritaire pour le domaine (Authoritative Answer)
- TC (1bit) : message tronqué (seulement 512 octets sont retournés)
- RD (1bit) : demande de résolution récursive
- RA (1bit) : recursion accepté
- zero (3bits) : Réservé
- rcode (4bits) : code retour 0 (ok) 1 (erreur dans la requête) – 2 (problème sur le serveur) - 3 (nom de domaine n'existe pas)

Exemple de transaction

```
Domain Name System (query)
    [Response In: 417]
    Transaction ID: 0x48a0
  Flags: 0x0100 (Standard query)
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data: Unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  Queries
```

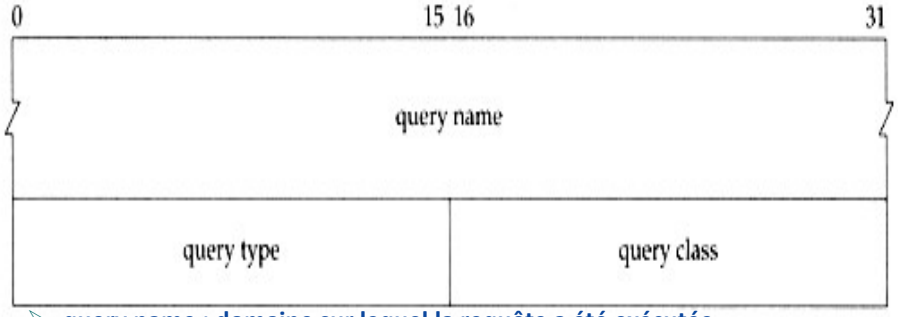

55

Exemple de transaction

```
Domain Name System (response)
    [Request In: 415]
    [Time: -0.618407000 seconds]
    Transaction ID: 0x48a0
  Flags: 0x8102 (Standard query response, Server failure)
      1... .... .... .... = Response: Message is a response
      .000 0... .... .... = Opcode: Standard query (0)
      .... .0.. .... .... = Authoritative: Server is not an authority for domain
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... 0... .... = Recursion available: Server can't do recursive queries
      .... .... .0.. .... = Z: reserved (0)
      .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
      .... .... ...0 .... = Non-authenticated data: Unacceptable
      .... .... .... 0010 = Reply code: Server failure (2)
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  Queries
```


56

Le champs questions



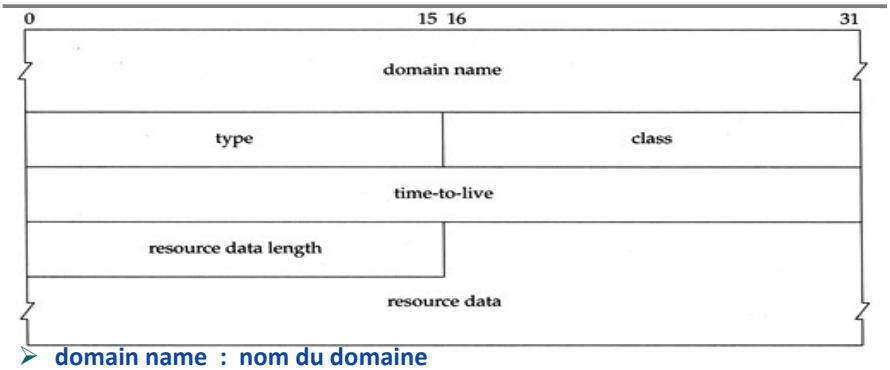
- query name : domaine sur lequel la requête a été exécutée
- query type (16bits) : type d' enregistrement A (01) – NS (02) – SOA (06) – PTR (12) – MX (15)
- query class (16bits) : protocol Internet (01)

- Représentation du nom de domaine orange.fr

  6orange2fr0 qui sera codé en hexa 06 6f 72 61 6e 67 65 02 66 72 00

57

RRs



- domain name : nom du domaine
- type (16bits) : type d'enregistrement A (01) – NS (02) – SOA (06) – PTR (12) – MX (15)
- class : (16bits) : protocol Internet (01)
- TTL (32bits) : durée de validité dans le cache
- length (16bits) : longueur des données qui vont suivre
- data : données

58

Configuration : méthode de résolution

- Le résolveur :
  - dans quel domaine chercher un hôte
  - ordre des serveurs de nom à utiliser
- /etc/resolv.conf

```
search p13.fr
name server 192.168.1.253
name server 192.168.1.252
```

- ordre de la méthode résolution
- /etc/nsswitch.conf

```
hosts:     files dns
```


59

Configuration : définitions des zones

```
/etc/named.conf
options {
   directory "/var/named";
};
zone p13.fr {
        type slave;
        file "maitre/p13.fr";
};
zone 10.10.in-addr.arpa {
        type master;
        file "maitre/10.10";
};
```


60

Configuration : Description des zones

/var/named/maitre/p13.fr

```
$TTL 180
@               IN      SOA     dns.p13.fr.  root.dns.p13.fr. (
2021071401 ; Serial
28800 ; Refresh
3600 ; Retry
604800 ; Expire
38400 ; TTL
)
                        IN      NS      dns1.p13.fr.
                        IN      A       10.10.0.1
ftp11                   IN      A       10.10.1.1
ftp12                   IN      A       10.10.1.2
```

/var/named/maitre/10.10

```
$TTL 180
@               IN      SOA     dns.p13.fr. root.dns.p13.fr. (
2021111602
28800
3600
604800
38400
)
                        IN      NS      dns.p13.fr.
1.1                     IN      PTR     ftp11.p13.fr.
1.2                     IN      PTR     ftp12.p13.fr.
```


Outils d'interrogation

Vérifier les serveurs de noms en utilisant les outils :

host est un simple outils qui permet d'effectuer des consultations ascendantes et inverse

```
[root@pc1 ~]#host pc2.p13.fr
pc2.p13.fr has address 192.168.1.2
[root@pc1 ~]#host 192.168.1.2
192.168.1.2,in-addr.arpa domain name pointer pc2.p13.fr
```

dig peut être utilisé pour donner une sortie détaillée. Il fonctionne uniquement avec les noms d'hôtes fqdn

```
[root@pc1 ~]#dig –x  192.168.1.2 +short
pc2.p13.fr

[root@pc1 ~]#dig mx  p13.fr +short
10 mail.p13.fr
20 mail.p13.fr
[root@pc1 ~]#dig pc2.p13.fr +short
192.168.1.2
```

La couche applicative : SMTP/IMAP/POP3

63

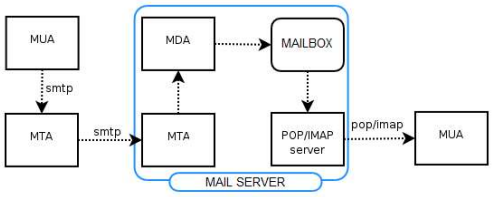
Présentation

- SMTP (Simple Mail Transfer Protocol) a été crée au début d' Internet et a commencé à être utiliser à grande échelle dans les années 1980.
- Envoi de courriers depuis le client vers le serveur
- Relai du courrier d'un serveur a un autre
- Ne nécessite pas d'authentification obligatoire à l' origine mais pour limiter le SPAM aujourd'hui il est requis
- Serveur en écoute sur le port TCP 25
- Basé sur TCP
- Protocole client/serveur en mode texte
- fonctionnement simple en mode requête/réponse
- Format d' une adresse électronique :
  - partie-locale@nomdedomaine
  - Exemple : toto@p13.fr
- destinataire :
  - direct : To
  - en copie conforme : CC
  - en copie conforme cachée : BCC

64

Terminologie

- **Plusieurs Processus collaborent afin d'acheminer les mails :**
  - **MUA (Mail User Agent) => client de messagerie (outlook, thunderbird) , transmet le message au MTA**
  - **MTA (Mail Transfert Agent) => Agent de transfert de courrier électronique (postfix, sendmail, exchange)**
  - **MDA (Mail Delivery Agent) => Agent de distribution de courrier à la BAL du destinataire (procmail, mail)**



- **Quand il le souhaite, le client de messagerie du destinataire se connecte et s'authentifie auprès du serveur POP ou IMAP qui relève le message dans la BAL, et le transmet au client**

Interaction avec le DNS

- **Pour acheminer le courrier, le serveur analyse dans un premier temps la partie de l'adresse située à droite du @, pour trouver le domaine du destinataire. Si ce domaine le concerne:**
- **Le serveur SMTP cherche alors la BAL du destinataire en regardant la partie de l'adresse située à gauche du @.**
- **Si le domaine du destinataire ne le concerne pas, il va chercher le serveur SMTP qui gère ce domaine, au moyen des champs MX du DNS du domaine destinataire et transmet le message à ce serveur.**

L'en-tête

Le courrier contient toute une partie "cachée" qui nous permet de savoir quel chemin il a suivi pour arriver dans votre BAL

From toto@orange.com Thu Jul 24 10:55:42 2014
X-Apparently-To: tata@yahoo.fr via 188.125.84.180; Thu, 24 Jul 2014 10:32:57 +0000
Return-Path: <toto@orange.com>
Received-SPF: pass (domain of orange.com designates 80.12.204.245 as permitted sender)

X19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19fX19f
X19fX19fX19fX19fX19fGRlcyBpbmZvX-Originating-IP: [80.12.204.245]
Authentication-Results: mta1092.mail.ir2.yahoo.com from=orange.com; domainkeys=neutral (no sig); from=orange.com; dkim=neutral (no sig)
Received: from 127.0.0.1 (EHLO relais-inet.francetelecom.com) (80.12.204.245)
by mta1092.mail.ir2.yahoo.com with SMTPS; Thu, 24 Jul 2014 10:32:57 +0000
Received: from omfeda07.si.francetelecom.fr (unknown [xx.xx.xx.200])
by omfeda11.si.francetelecom.fr (ESMTP service) with ESMTP id DF6141B96D7
for <tata@yahoo.fr>; Thu, 24 Jul 2014 11:55:51 +0200 (CEST)
Received: from PUEXCH51.nanterre.francetelecom.fr (unknown [10.101.44.31])
by omfeda07.si.francetelecom.fr (ESMTP service) with ESMTP id 208E3158126
for <tata@yahoo.fr>; Thu, 24 Jul 2014 11:55:44 +0200 (CEST)
Received: from PUEXCB1A.nanterre.francetelecom.fr ([10.101.44.11])
by PUEXCH51.nanterre.francetelecom.fr ([10.101.44.31]) with mapi; Thu, 24 Jul2014 11:55:44 +0200

From: <toto@orange.com>
To: « tata@yahoo.fr" <tata@yahoo.fr>
Date: Thu, 24 Jul 2014 11:55:42 +0200
Subject: [SUJET] : TEST
Thread-Topic: [SUJET] : TEST
Thread-Index: Ac+nJWkNFJTwLTc7TGGcQbPgWjr9hw==
Message-ID: <25656_1406195751_53D0D827_25656_243_1_54851BE46C553B4D92BDAA11CF289033A508D308B8@PUEXCB1A.nanterre.francetelecom.fr>
Accept-Language: fr-FR
Content-Language: fr-FR
X-MS-Has-Attach:
X-MS-TNEF-Correlator:
acceptlanguage: fr-FR
Content-Type: multipart/alternative; boundary="_000_54851BE46C553B4D92BDAA11CF289033A508D308B8PUEXCB1Anante_"
MIME-Version: 1.0
X-PMX-Version: 6.0.3.2322014, Antispam-Engine: 2.7.2.2107409, Antispam-Data: 2014.6.25.123022
Content-Length: 4438

67

Les champs de l'en-tête

- **Return-Path: C'est l'adresse qui sera utilisée :**
  - **Pour la réponse (répondre à l'expéditeur)**
  - **Le renvoi du message s'il ne peut arriver au destinataire (MAILER-DAEMON)**
- **Received: Chaque MTA qui reçoit le message y inscrit le nom du MTA qui le lui a envoyé, ainsi que le sien**
- **Message-ID: C'est un identifiant unique du message (attribué par le premier MTA)**
- **From: adresse de l'expéditeur.**
- **To: adresse du destinataire**
- **Subject: L'objet du message**
- **Date: La date d'émission écrite par le MUA de l'émetteur**
- **MIME-Version: Version du mode de codage des données.**
- **Content-Type: Type de codage utilisé (multiplepart/alternative => codé en plusieurs parties de maniére différentes)**
- **X-... champs non officiels (non prit en compte par les MTA)**
- **charset= Jeu de caractères utilisé.**
- **Content-Transfer-Encoding: Codage sur 7 ou 8 bits.**

68

Le format MIME

- **MIME : Multipurpose Internet Mail Extension**
  - **Proposé par les laboratoires Bell en 1991**
  - **Palier à la limitation des mails qui ne sont encodé qu'en ASCII 7bits**
  - **Permet l'échanges de courriers écrits avec des jeux de caractères différents, et de joindre des fichiers (objets-documents) :**
  - **Le message est encodé à l' émission, décodé à la réception, via l'indication du type MIME utilisé**

Type MIME	Type de fichier	Extension associée
application/acad	Fichiers AutoCAD	dwg
application/clariscad	Fichiers ClarisCAD	ccad
application/drafting	Fichiers MATRA Prelude drafting	drw
application/dxf	Fichiers AutoCAD	dxf
application/i-deas	Fichiers SDRC I-deas	unv
application/iges	Format d'échange CAO IGES	igs,iges
application/octet-stream	Fichiers binaires non interprétés	bin
application/oda	Fichiers ODA	oda
application/pdf	Fichiers Adobe Acrobat	pdf

Exemple MIME

On spécifie la version

MIME-Version : 1.0

On annonce ce qui vient dans le corps

Content-Type : Multipart/related;

charset="ISO-8859-1";

type="multipart/alternative« ;

On indique le séparateur des différentes partie

Boudary="Boundary-00=_JXS9QL8000000000000000"

Transmission d'une image

-- Boundary-00=_JXS9QL8000000000000000

content-Type : image/gif;

name="image.gif";

Content-Transfer-Encoding : Base64

Content-ID : <D0FEE56-3E5A-AEC1-342242>

Transmission d'une séquence HTML

-- Boundary-00=_JXS9QL8000000000000000

content-Type : Text/HTML;

charset="ISO-8859-1"

Content-Transfert-Encoding : quoted-printable

<HTML>.....</HTML>

Telnet –SMTP

telnet smtp.domaineX.fr 25
Trying IP.IP.IP.IP... smtp.domaineX.fr.
Escape character is '^]'.
220 smtp.domaineX.fr <version> Ready at <date>
HELO domaineY.fr
250-smtp.domaineX.fr
MAIL FROM:<auteur@domaineY.fr>
250 Sender ok
RCPT TO:<destinataire@domaineX.fr>
250 Recipient ok.
DATA
354 Enter mail, end with "." on a line by itself
Subject: Test
Corps du texte
.
250 Ok
QUIT
221 Closing connection
Connection closed by foreign host.

Valeur du code retour :
2xx réponse positive
3xx réponse positive intermédiaire
4xx timeout
5xx réponse négative

POP3

- **Post Office Protocol version 3**
- **Serveur en écoute sur le port TCP 110**
- **basé sur TCP**
- **Gère le dialogue de récupération de contenu de la BAL**
- **Authentification (en claire)**
- **Protocole client/serveur en mode texte**
- **fonctionnement simple en mode requête/réponse**
- **Présent sur le serveur jusqu'à téléchargement par le client**
- **Par défaut le client demande l'effacement des messages sur le serveur POP3 dés qu'ils ont été téléchargés**
- **Aucun classement ne peut être réalise sur le serveur**

Fonctionnement

- Les messages sont contenus sur le serveur dans une file
- POP3 est capable de les délimiter, de les lister, de les compter, de calculer leur taille, d'extraire une partie du message, de récupérer un message ou de le supprimer
- Tout le reste se fait sur le poste client (trie, archivage)

1	texte complet du message
2	texte complet du message
3	texte complet du message
4	texte complet du message
5	texte complet du message
N	texte complet du message

Les commandes

- USER : login de votre compte
- PASS : mot de passe de votre compte
- LIST : liste les messages
- DELE n : efface le message numéro n (effectif après le quit)
- RSET : annule les commandes DELE
- STAT : nombre de messages et taille de l'ensemble des messages
- TOP n l : affiche les l premières ligne du message numéro n
- RETR n : récupère le message numéro n
- NOOP : ne rien faire (évite de perdre la connexion)
- QUIT : quitter la session

Telnet – POP3

telnet pop.domaineX.fr 110
Trying IP.IP.IP.IP...
Connected to pop.domaineX.fr.
Escape character is '^]'.
+OK ready
user toto
+OK User name (toto) ok. Password, please.
pass mypassword
+OK Logged in.
stat
+OK 5 8958
list
+OK 5 messages
1 7962
2 1528
3 3513
4 1627
5 4328
quit
+OK Logging out.
Connection closed by foreign host.

IMAP

- **Interactif Message Access Protocol**
- **Serveur en écoute sur le port TCP 143**
- **basé sur TCP**
- **Comme POP3 il gère le dialogue de récupération de contenu de la BAL**
- **Synchronisation des messages, et non un téléchargement systématique de tout le message avec ses pièces jointes comme POP3**
- **Création de dossiers sur le serveur, déplacement des messages d'un dossier vers un autre**
- **Lecture des messages en les laissant sur le serveur**
- **Les messages restent sur le serveur sans être rapatrié**
- **On peut aussi les rapatrié pour les lire en mode hors-ligne**

FONCTIONNEMENT

- IMAP manipule des messages qui se trouvent dans des répertoire ou sous-répertoire appelé namespaces.
- Chaque message est représenté par deux numéros; le numéro de séquence et l'uid
  - Le numéro de séquence est valable que dans le bon répertoire et indique l'emplacement du message dans la file, il peut changer au cours du temps
  - L' UID est unique à chaque message et ne change jamais au cours du temps
- Tous se fait à distance sur le serveur (trie, archivage)
- Outils puissant et complexe permet de faire tous ce que fait POP3 avec une granularité plus fine

Gestion des répertoires

- Lister les répertoires

```
list "" *
list inbox *
list "Archive_2013" *
```

- Créer un nouveau repertoire

```
create INBOX.Archive_2014
create "A_classer"
```

- Supprimer un répertoire

```
delete INBOX.Archive_2014
```

- Renomme un répertoire

```
rename "A_classer" "A_ranger"
```

Les commandes

- Les commandes doivent être précédé d'un index
- <tag> login user password : authentification
- <tag> LIST repertoire mot_cle : cherche la référence dans la box
- <tag> select repertoire : choix de la boite aux lettre
- <tag> search requête : recherche de mail basé sur des critéres
- <tag> fetch num_mess action : syntaxe complexe qui permet d' extraire des informations d'un ensemble de messages
- <tag> STORE num_mess choix-item item-value :
- <tag> logout : quitter la session

Search – uid search

- From <adr_email>
- To <adr_email>
- Since <date>
- Before <date>
- Subject <string>
- Body <string>
- Not <query>
- or <query>

a1 **SEARCH FROM "tata@domaine.com" BEFORE 27-Jul-2014**
** SEARCH 1 5*
a1 OK Completed

fetch – UID fetch

- **BODY[TEXT] : Le corps du message**
- **BODY[HEADER] : En-tête du mail**
- **BODY[HEADER.FIELDS (<list>)] : Choix des champs de l'en-tête**
- **BODY[] : Affichage du mail (en-tête+corps)**
- **BODY.PEEK : comme BODY[] sans positionné le flag /seen**
- **FLAGS : L'ensemble des flags du message**
- **UID : L' UID du message**

a1 FETCH 1 BODY[HEADER.FIELDS (to)]
** 1 FETCH BODY[HEADER.FIELDS (To)] {24}*
To: toto@domaine.com
)
a1 OK Completed

a2 FETCH 1 (BODY[HEADER.FIELDS (Subject From)])
1 FETCH (BODY[HEADER.FIELDS (SUBJECT FROM)] {76}
From: ta tatata <tata@domaine.com>
Subject: Re: [Fwd: Re: bon travail]
)
a2 OK Fetch completed.

a3 fetch 1 body[text]
** 1 FETCH (FLAGS \Seen) BODY[TEXT] {28}*
ceci est mon corps de mail
)
a3 OK Completed

81

Les flags

Permettent de déterminer le statut du mail.

A la connexion nous avons la présentation des différents flags

- **answered : message auquel on a répondu**
- **flagged : message auquel il faut porter une attention particulière**
- **draft : message commencer mais non terminer.**
- **deleted : message qui sera supprimé plus tard**
- **seen : message indiqué comme lu**
- **Recent : nouveau message reçu depuis la dernière session (non modifiable)**

Les outils de messagerie classique gère automatiquement la gestion des flags mais pas le telnet

82

Telnet - IMAP

telnet imap.domaineX.net 143

** OK IMAP4 server ready*

a1 login paul 123

a1 OK User logged

a2 list "*" "*"

** LIST (\HasChildren) "." "INBOX"*
** LIST (\HasChildren) "." "INBOX.DRAFT"*
** LIST (\HasChildren) "." "INBOX.OUTBOX"*
** LIST (\HasChildren) "." "INBOX.SPAM"*
** LIST (\HasChildren) "." "INBOX.TRASH"*
a2 OK Completed (0.000 secs 6 calls)

a3 select INBOX

* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen *)]
6 EXISTS
1 RECENT
* OK [UNSEEN 5] First unseen.
* OK [UIDVALIDITY 1216295789] UIDs valid
OK [UIDNEXT 8] Predicted next UID
a3 OK [READ-WRITE] Select completed.

8 logout

** BYE LOGOUT received*
8 OK Completed

83

POP3 vs IMAP

AVANTAGES POP3

- Simple
- Recherche de mail rapide (en local)
- Utilisation minimale des ressources du serveur.

INCONVENIENTS POP3

- Gestion des sauvegardes par l'utilisateur.

AVANTAGES IMAP

- Changement de client facilité
- Gestion des messages sur le serveur
- Orienté mobilité

INCONVENIENTS IMAP

- Quota disque sur serveur.

84

WEBMAIL

- **Entre le client et le serveur, un seul protocole HTTP**
- **La gestion du courrier se fait à travers des pages html**
- **Avantage :**
  - **Le nomadisme**
  - **Aucune trace sur le poste client**
- **Inconvenant :**
  - **Plus lent qu'une messagerie traditionnelle**
  - **Publicité**

85

Fichier de configuration

/etc/postfix/main.cf

mail_owner = postfix

myhostname = nomserveur.mondomaine.fr

myorigin = $mydomain

inet_interfaces = $myhostname, localhost

mydestination = $myhostname, localhost. $mydomain, $mydomain, smtp. $mydomain, mail$mydomain

unknow_local_recipient_rejet_code = 450

mynetworks = X.X.X.0/24, 127.0.0.0./8

relay_domains = $mydestination

masquerade_domains = mondomaine.fr

masquerade_exceptions = root

alias_maps = hash:/etc/postfix/aliases

smtpd_banner = $myhostname

mailbox_command = /usr/bin/procmail

smtpd_recipient_restrictions = permit_mynetworks, hash:/etc/postfix/access, reject_unauth_destination

86

Définition des variables

myhostname : nom FQDN de la machine

mydomain : domaine locale

myorigin : domaines pouvant envoyer des mails

mydestination : domaines qui sont accepté pour la délivrance locale des mails

unknown_local_recipient_reject_code : code erreur à envoyer si l' émetteur ne fait pas partie de la liste mydestination

mynetworks : liste des réseaux autorisé à être relayer

relay_domains : domaines pouvant être relayé

alias_maps : fichier des alias

smtpd_banner : baniére smtp de postfix

mailbox_command : distribution du courrier

smtpd_recipient_restrictions : restriction sur les destinataire (doivent appartenir aux domaines qui sont gérés par cette machine)

Les services de transfert de fichiers: FTP

FTP

- File Transfert Protocol
- Port 21 basé sur TCP
- Permet le transfert de fichiers (upload/download) entre deux machines
- Authentification (en claire)
- Protocole client/serveur en mode texte
- fonctionnement simple en mode requête/réponse
- Gestion à distance des fichiers
  - Créer
  - Supprimer
  - Renommer
  - lister

89

Fonctionnement

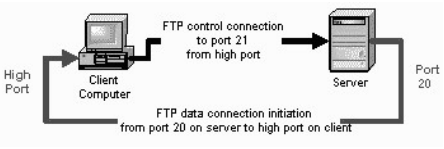
- FTP fonctionne au-dessus du protocole de niveau transport TCP.
- Démarrage d'une session : ftp nom_serveur
- Lors d'une connexion FTP, deux canaux sont ouverts.
  - Un canal de contrôle
    - Pour les commandes et les réponses
    - Cree quand la connexion au serveur est établie
    - Port 21 en général
    - Toutes les commandes effectuées sur le canal de contrôle suivent les recommandations du protocole Telnet
  - Un canal pour les données
    - Cree a la demande pour chaque transfert de données
    - Rompu a la fin de chaque transfert de données
    - Port 20 en général

90

Deux Modes

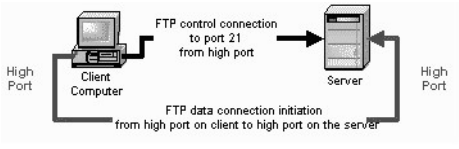
Le mode ACTIF

- Mode par défaut
- Le client établi dans un premier temps une session TCP sur le port 21 du serveur .
- Le serveur établi une session TCP avec comme port source 20 un port aléatoire du client



Le mode PASSIF

- Le client établi dans un premier temps une session TCP sur le port 21 du serveur .
- Une fois la session établie et l'authentification FTP acceptée, on demande au serveur le passage en mode passif via la commande PASV.
- Le client reçoit le numéro du port sur lequel il peut se connecter pour le transfert de donnée



Deux types d' accès FTP

- FTP anonyme permet les connections publique en tant qu' anonyme
  - Pratique
  - Sécurité amoindrie
  - couple anonymous/email
  - Lecture seule
- FTP protégé ne permet que des personnes authentifié
  - Un peu plus sécure
  - Authentification requis
  - Authentification en clair
  - Droit en Lecture/écriture

Commandes FTP

- username (USER) : nom d'utilisateur
- password (PASS): mot de passe
- cd : changement de répertoire distant
- lcd : changement de répertoire local
- mkdir/rmdir : suppression d'un répertoire
- delete : suppression d'un fichier
- get/put : sens du transfert
- ascii/bin : type de fichier à transférer
- prompt : commande multiple m..
- bye : fermeture de session

93

Les reponses FTP

- Les réponses du serveurs permettent d'assurer la synchronisation entre client et serveur
- Pour chaque commande reçu du client le serveur effectue renvoie un code réponse
- Les codes réponses sont constituées de 3 chiffres XXX
- 1xx action en cours attendre une autre nouvelle réponse
- 2xx demande exécuté avec succès
- 3xx demande d'information complémentaire
- 4xx erreur temporaire – action peut être renouvelée
- 5xx erreur de syntaxe
- Exemple :
- 226 => transfert terminé ok
- 331 => attente du password
- 430 => authentification ko

94

session - FTP

ftp ftp.free.fr
Connecté à ftp.proxad.net.
220 Welcome to ProXad FTP server
Utilisateur (ftp.proxad.net:(none)) : anonymous
331 Please specify the password.
Mot de passe :
230 Login successful.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
MPlayer
lost+found
mirrors
nzb
pub
tmp
226 Directory send OK.
ftp : 61 octets reçus en 0,00 secondes à 61000,00 Ko/s.
ftp> cd pub
250 Directory successfully changed.
ftp> ls*200 PORT command successful. Consider using PASV.*
150 Here comes the directory listing.
Distributions_Linux
Games
226 Directory send OK.
ftp : 158 octets reçus en 0,00 secondes à 158000,00 Ko/s.

ftp> cd Games
250 Directory successfully changed.
ftp> cd freenews
250 Directory successfully changed.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
freenews-live-dvd-i386.iso
freenews-live-dvd-i386.md5sum
226 Directory send OK.
ftp : 93 octets reçus en 0,00 secondes à 93000,00 Ko/s.
ftp> get freenews-live-dvd-i386.md5sum
200 PORT command successful. Consider using PASV.
150 Opening BINARY mode data connection for freenews-live-dvd-i386.md5sum (33 bytes).
226 Transfer complete.
ftp : 33 octets reçus en 0,01 secondes à 3,30 Ko/s.
ftp>

Les services de transfert de fichiers: SFTP

SFTP

- SFTP Signifie SSH File Transfert Protocol ou Secure File Transfert Protocol
- SFTP fait partie de SSH ou Secure Shell
- Client SFTP se comporte comme un client FTP classique:
  - Vue sur les répertoires et fichiers; Déposer, extraire…des fichiers
  - Mêmes commandes que FTP
- Les systèmes Linux proposent des packages standard d'une implémentation open source de SSH (OpenSSH).
- SFTP est un protocole protégé à l'aide des techniques cryptographiques:
  - Tout le trafic entre le client et le serveur est entièrement chiffré depuis le processus d'identification jusqu'à l'envoi de fichiers
  - SFTP convient très bien à l'échange sécurisé de fichiers sur Internet

97

SFTP: SSH

- SSH permet de sécuriser les communications des réseaux en utilisant la cryptographie
  - SSH est composé d'un ensemble d'outils permettant des connexions sécurisées entre les machines. Ces outils ont pour but de remplacer les utilitaires de connexions classiques n'utilisant pas de chiffrage.
    - Remplace: rcp, rlogin, telnet, ftp.
  - SSH chiffre et compresse un canal de communication qui sécurise les données transmises (permet d'éviter les sniffeurs réseaux)
  - Non seulement le mot de passe est chiffré lors de la connexion mais les informations circulant sur le réseaux entre les deux machines le sont.
- SSH un protocole:
  - SSH existe en deux versions majeures:
    - Version 1: version vulnérable des failles de sécurité
    - Version 2: beaucoup plus sûre et possède le protocol de transfert de fichiers complet (SFTP).
- OpenSSH du projet OpenBSD apparait en 1999, aujourd'hui c'est le plus utilisé
  - Produit en accord avec la législation Française sur la cryptographie

98

Fonctionnement SSH

Authentification des serveurs

- **Le principe d'authentification du serveur se fait par chiffrement à clé publique du protocole SSH.**
- **Le client doit donc connaitre la clé publique du serveur sur lequel il veut se connecter avant toute connexion. Ainsi, il existe un mécanisme pour la machine cliente pour stocker les clés d'un serveur afin de les réutiliser ensuite.**
- **Il pourra ainsi vérifier la clé d'un serveur à chaque nouvelle connexion avec celle enregistrée lors de la première connexion.**
- **Cela permet d'éviter les attaques du type main-in-the-middle.**

Authentification des utilisateurs

- **Une fois que la connexion sécurisée mise en place entre le client et le serveur, le client doit s'identifier sur le serveur afin d'obtenir un droit d'accès.**
  - **Par mot de passe:** le client envoie un nom d'utilisateur et un mot de passe au serveur au travers de la communication sécurisée et le serveur vérifie si l'utilisateur concerné a accès à la machine et si le mot de passe fourni est valide
  - **Par clés publiques:** Si l'authentification par clé est choisie par le client, le serveur va créer un challenge et donner un accès au client si ce dernier parvient à déchiffrer le challenge avec sa clé privée
  - **Par hôte de confiance:** Système équivalent aux systèmes utilisant rhost ou hosts.equiv en utilisant les clés publiques des serveurs
  - **Par Kerberos, SmartCard, PAM**

Les protocoles du web : HTTP

Le protocole HTTP

- HyperText Transmission Protocol
- Port 80 basé sur TCP
- Transfert de fichier avec support du format des données
- Modèle Requête/Réponse en mode texte
- Prise en charge de la gestion MIME
- Formatage de l'affichage basé Langage d' affichage HTML (rôle du navigateur)

Généralité

- L' évolution du protocole HTTP
  - **1996 version 1.0**
    Etablissement de connexion, requête du client, réponse du serveur et fermeture de connexion.
  - **1999 version 1.1**
    Amélioration du temps de traitement (persistance)
  - **2012 version 2.0** Amélioration du téléchargement (multiplexage de transfert)

Requête HTTP 1.0

méthode, document demandé, version du protocole utilisée.

en-tête : (facultatif) informations supplémentaires

[ligne vide]

Corps : (facultatif) paramètres de la requetés

105

Exemple

GET /renseignements.html HTTP/1.0

Host: www.domaine.com

Referer: http://www.domaine.com/

User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:31.0) Firefox/31.0

106

Réponse HTTP 1.0

version du protocole, code réponse, texte réponse

en-tête : (facultatif) informations supplémentaires

[ligne vide]

Corps :le document demandé

107

Exemple

HTTP/1.0 200 OK

Date: Fri, 01 Aug 2014 22:19:39 GMT

Server: Apache/0.8.4

Content-Type: text/html

Content-Length: 42

Expires: Sat, 02 Aug 2014 22:19:39 GMT

Last-modified: Fri, 01 Aug 2014 14:21:40 GMT

<html>Hello world!</html>

108

requête/réponse HTTP 1.1

```
GET / HTTP/1.1
Host: www.unexemple.fr
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; fr; rv:1.9.2.13)
Gecko/20101206 Ubuntu/10.10 (maverick) Firefox/3.6.13
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
```

```
HTTP/1.1 200 OK
Date: Fri, 04 Feb 2011 11:01:17 GMT
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Content-Length: 32

<html>exemple de réponse.</html>
```


En-tête

Nom	Description
Accept	type MIME accepté par le client
Accept-Charset	jeu de caractére accepté par le client
Accept-Encoding	codage accepté par le client
Accept-Language	Langage accepté par le client
Content-Type	type MIME contenu dans le corps
Content-Language	langage contenu dans le corps
Content-Encoding	codage contenu dans le corps
Content-Length	longueur du corps
Date	début du transfert
Referer	URL d'où provient la requête
User-Agent	information sur le client
Server	information sur le serveur

Les méthodes HTTP

- GET : Permet d'obtenir une ressource
- HEAD : Permet d'obtenir que l'en-tête de la ressource (rafraichissement du cache)
- POST : Envoi de données vers le serveur (compléter un formulaire)
- TRACE : La réponse contient en corps de message la requête reçue par le serveur (debug proxy)
- OPTIONS : Informations sur le serveur
- PUT : Remplacer une ressource
- DELETE : Supprimer une ressource

Les codes réponses

- La réponse du serveur vient avec un code (3chiffres) indicateur du résultat de la requête :
- 1xx : information
- 2xx: succès
- 3xx : redirection
- 4xx : erreur du client
- 5xx : erreur du serveur

Multi-homing

- **Plusieurs sites sur un même serveur http**

Grâce à l' en-tête « Host », les sites web sont accessibles sur la même adresse IP et avec le même numéro de port.

« Host » correspond au nom des différents serveur virtuel

Exemple :

http://www.gnu.fr

http://info.gnu.fr

www.gnu.fr et info.gnu.fr sont des alias de la même adresse IP

HTML

- **Inventé au CERN dans les années 80**
- **Langage à balises hypertexte**
- **Interprété par les navigateurs**
- **Permet l'insertion d'objet tel que des**
  - **Images, animation flash**
  - **Sons**
  - **Script**
  - **Applet**

Fichiers de configuration

/etc/apache/conf/http.conf

Le fichier httpd.conf contient la configuration général du serveur HTTP

```
ServerType standalone
LockFile /var/lock/apache.lock
PidFile /var/run/apache.pid
ScoreBoardFile /var/run/apache.scoreboard
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150
MaxRequestsPerChild 100
ExtendedStatus On
Port 80
User www
Group www
ServerAdmin webmaster@mondomaine
NameVirtualHost 193.51.14.15
ServerName www.mondomaine
ServerRoot /etc/apache
```

/etc/httpd/conf/srm.conf

Le fichier srm.conf contient la configuration d'utilisation (racine des documents ,compte utilisateurs,..)

```
DocumentRoot /home/httpd/html
UserDir mypublic
```


115

Virtuel Server

- **Cette partie déclare un second serveur web qui sera virtuel car il utilise la même interface physique que le premier serveur web mais possédé un nom d'hôte différent.**

```
<VirtualHost gestion.mondomaine>
  DocumentRoot /home/gestion/web
  ServerName gestion.mondomaine
  ErrorLog /var/log/apache/gestionlogs
  <Directory /home/gestion/web/ >
    Options Indexes Includes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from 192.168.1.0/255.255.255.0
  </Directory>
</VirtualHost>
```


116

Les protocoles du web : HTTPS

117

Présentation HTTPS

- **HTTPSecured: Association de HTTP avec une couche de chiffrement SSL ou TLS.**
- **Vérification de l'identité du site grâce à un certificat d'authentification**
  - **Certificat émis par une autorité tierce, réputée fiable (et faisant généralement partie de la liste blanche des navigateurs internet)**
  - **Il peut permettre la validation de l'identité du visiteur s'il utilise un certificat d'authent client**
- **HTTP vs HTTPS:**
  - **Commence avec « https:// »**
  - **Utilisation du port 443 par défaut**
  - **Sécurisation eavesdropping (attaques par écoute)**
- **Confidentialité et intégrité des données**
  - **Il garantit théoriquement la confidentialité et l'intégrité des données envoyées par l'utilisateur (notamment des informations entrées dans les formulaires) et reçues du serveur**

118

Présentation HTTPS

- **La Généralisation du protocoles HTTPS est faite**
  - Les navigateurs ont commencé à signaler les sites non sécurisés à partir de janvier 2017
  - Mozilla réserve ses nouvelles fonctionnalités aux sites sécurisés
  - A partir de la version Firefox 83, on peut choisir le mode https uniquement.
  - Google Chrome passe à https par défaut.
- **Sécurisation complète des données, empêche les actions malveillantes de part et d'autres des tunnels**
- **Certificat de type X.509**

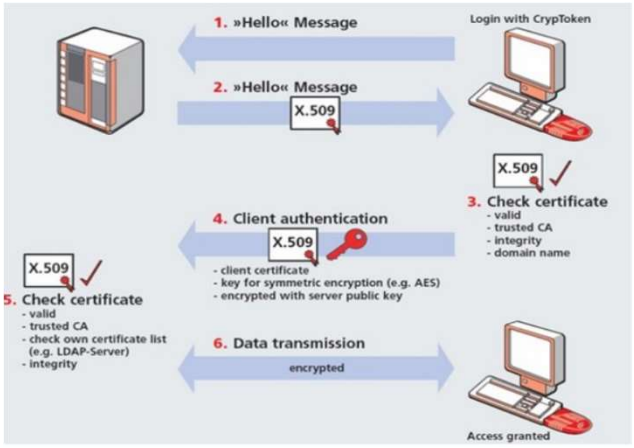
119

Présentation HTTPS

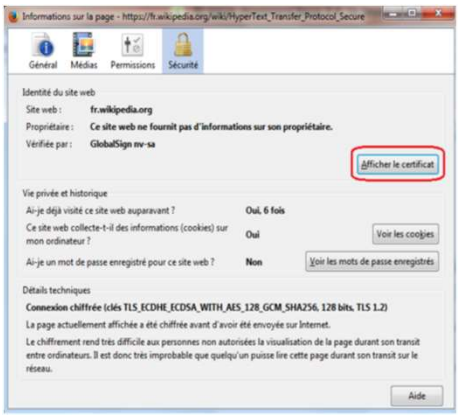
- **Structuration d'un certificat:**
  - Version
  - Numéro de série
  - Algorithme de signature du certificat
  - DN du délivreur (autorité de certification)
  - Validité (dates limite)
    - Pas avant
    - Pas après
  - DN du détenteur du certificat
  - Informations sur la clé publique :
    - Algorithme de la clé publique
    - Clé publique proprement dite
  - Identifiant unique du signataire (optionnel, X.509v2)
  - Identifiant unique du détenteur du certificat (optionnel, X.509v2)
  - Extensions (optionnel, à partir de X.509v3)
    - Liste des extensions
  - Signature des informations ci-dessus par l'autorité de certification

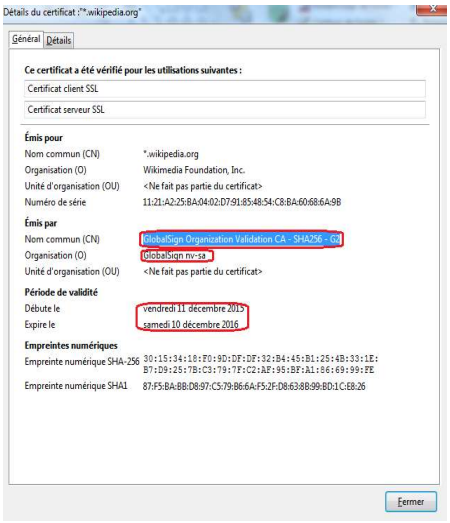
120

Présentation HTTPS



Présentation HTTPS



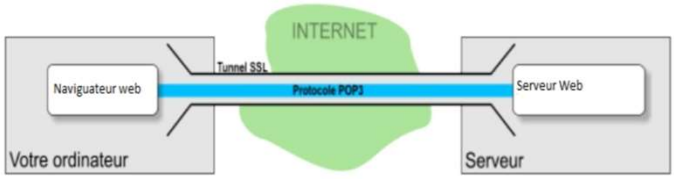


Fonctionnement

- SSL, devenu TLS après l'achat du brevet par IETF, est un protocole de sécurisation des échanges de données sur internet.
- TLS assure 3 choses:
  - Confidentialité: empêchant l'espionnage des informations échangées
  - Intégrité: impossibilité de truquer les informations
  - Authentification: assure l'identité
- TLS consiste en 2 protocoles:
  - TLS Handshake Protocol: Négociation des clés et protocoles de chiffrement communs avant de communiquer
  - TLS Record protocol: chiffrement des informations et contrôles

123

Fonctionnement



124

Fontionnement

- **Négociation TLS (Handshake):**
  - Au début de la communication le client et le serveur s'échangent :
    - La version SSL
    - La liste des méthodes de chiffrement (sym et asym) et de signature
    - Les méthodes de compression
    - Des nombres aléatoires
    - Les certificats
  - Le client et le serveur essaient d'utiliser le protocole de chiffrement le plus puissant
  - Diminution du niveau jusqu'à trouver un protocole commun aux deux
  - Début de l'échange de données

Fonctionnement

- **Début de communication (Record)**
  - Expéditeur
    - Découpage de données en paquets
    - Compression des paquets
    - Signature par cryptogramme
    - chiffrement des données
    - Envoi
  - Récepteur
    - Déchiffrage des données
    - Vérification de la signature
    - Décompression des paquets
    - Réassemblage des paquets

Fonctionnement

- **TLS utilise:**
  - **Chiffrement asymétrique (Rsa, Diffie-Hellman): clés de session**
  - **Chiffrement symétrique (DES, 3DES, IDEA, RC4…): chiffrement des données**
  - **Signature cryptographique des messages (HMAC, Utilisant MD5, SHA…) pour vérifier la contrainte d'intégrité**

- **La liste des système utilisés est visible depuis le navigateur sur une page https.**

127

Déploiement

- **Déploiement à la racine du serveur**
- **SSLv2, SSLv3 & TLS V1.0 sont parfaitement gérés par la plupart des navigateurs**
- **Récupération du type de ressource grâce à Apache : création automatique d'une variable d'environnement interne nommé HTTPS de valeur «ON»**
- **Limiter l'accès de certains fichiers au protocole HTTPS:**
  - **Dans le cas d'un serveur virtuel:**

```
<virtualHost….:80>
#...
RedirectPermanent page.html https://www.monsite.com/page.html
</virtualHost>
```


128

Déploiement

- **Limiter l'accès de certains fichiers au protocole HTTPS:**
  - **Depuis un fichier .htaccess:**
    - RewriteEngine On
    - RewriteCond %{HTTPS}!=on
    - RewriteRule secure[/_].*https://%{SERVER_NAME}%{REQUEST_URI}[R,L]
  - **Via PHP:**

```
<?php
 if (!isset($_SERVER['HTTPS']) || !_server['HTTPS'] != 'on') {
header('Location: https//' . §_SERVER['SERVER_NAME'] .
§_SERVER['REQUEST_URI']);
exit;
}
?>
```


Faiblesses et renforcements

- **Faiblesses HTTPS:**
  - **Attaque du type man in the middle**
    - **Présenter par Moxie Marlinspike à la BHC 2009**
    - **Attaque invisible et très éfficace**
    - **Changement des liens https:// en htt://**
  - **Failles de sécurité internes**
    - **Présentées par Mark Zusman (Intrepidus Group) et Alexander Sotirov (phmsecurity.com) à la BHC 2009**
    - **Facilité d'obtention d'un certificat (phishing, etc)**
    - **Contournement des certificats EV (Extended Validation)**

Faiblesses et renforcements

- **Standard RFC2246:**
  - « An Upper limit of 24 hours is suggested for session ID lifetimes, since an attacker who obtains a master_security may be abale to impersonate the compromised party until the corresponding session ID is retire »
  - Certaines versions d'IE renégocient toutes les 2 minutes
- **Renforcements:**
  - Redirection systématique vers HTTPS depuis le serveur
  - Mises à jour des navigateurs pour forcer le protocole
  - Plus grande vigilance et repise en question de VeriSign
- **HTTPS cohabite avec S-HTTP:**
  - S-HTTP est flexible et s'intègre plus au protocole HTTP
  - HTTPS est beaucoup plus simple à mettre en place ce qui en fait le leader incontesté

Conclusion

- **TLS est nécessaire pour:**
  - Une boutique en ligne (commandes et cartes de crédit)
  - Proposer une connexion protégée
  - Traiter des données sensibles
  - Répondre aux normes relatives à la confidentialité et à la sécurité des données
  - Protéger la confidentialité des données
  - Acquérir la confiance des utilisateurs

Annexe

Guide de survie sous Linux

- **Les fichiers de configuration des services sont contenus dans le répertoire /etc**
  - **/etc/vsftpd.conf**
  - **/etc/apache2/apache2.conf**
- **Le démarrage et l' arrêt des services est geré par un script contenu dans le répertoire /etc/init.d.**
  - **/etc/init.d/apache2 start pour démarrer le serveur web apache**
  - **/etc/init.d/apache2 stop pour arrêter le serveur apache**
  - **/etc/init.d/apache2 restart pour redémarrer le serveur apache**
  - **/etc/init.d/apache2 status pour connaitre le statut du serveur apache**
  - **/etc/init.d/apache2 reload pour prendre en compte les modification**
- **La journalisation des logs se trouve dans le répertoire /var/log**
  - **/var/log/syslog : Tous les évènement système (démarrages, arrêts, erreurs...)**
  - **/var/log/apache2/error.log : Toutes les erreurs**
  - **/var/log/apache2/access.log : Toutes les requêtes traités par le serveur**
  - **/var/log/vsftpd.log :**

La commande netstat

- **Netstat permet de visualiser l'affichage des connexions reseaux**
  - **-a : affiche les sockets d' ecoute des serveurs**
  - **-at : affiche les sockets TCP**
  - **-au : affiche les sockets UDP**
  - **-n affiche les adresses IP (pas de resolution DNS)**

La commande ps

- **ps permet de visualiser l'affichage les processus en cours**

-e affiche tous les process

-f affiche toutes les informations