

# Remise a niveau en réseaux

■ Date de création	@5 mai 2025 09:41
■ Matière	RaN

<mailto:mhid.imustapha@gmail.com>

(découpage pas fait pendant la formation, découpage IP | BJORNULF FRODE il fait)

OSI : 1983 | Norme internationale pour une architecture multicouches

7 couches | Permet la connexion de matériels hétérogènes  
la couche discute avec celle d'en dessous et du dessus.

0-1bit = 0volt et 0.5volt / lumiere pas lumiere

facile a dev et modif

nombre de ports = 16bits / 65535

## OSI ( | topologie de l'information)

1. Physique | bit
2. Liaison | trame
3. Réseau | paquet
4. Transport | segment
5. Session | message
6. Présentation | données
7. Application | données

1 = physique pur; transport de média; cable, wifi..

2 = Adresse MAC (Une adresse MAC (Media Access Control) est un identifiant physique unique attribué à la carte réseau d'un appareil) (ici qu'on décide qui parle et quand il parle | regle de support de media, on organise CIFS et DIFS)

**ARP** : adresse résolution protocole IP en MAC

3 = adresse logique (une adresse logique est un identifiant unique attribué à un appareil sur un réseau, comme une adresse IP) (Routeur/Routage) (Ping : protocole ICMP) (ya pas de port)

4 = encapsulation / transport des segments / fiabilité de la transmission (TCP (fiable) et UDP (rapide, non fiable)) les ports / matériels firewall (seul capable de lire les ports (et MAC/IP) / DNS(domain name services) = UDP port 53 (mais possible en TCP) (SQL port 1433) **(65k ports FOIS 2)**

5 = gère la ressource sur le réseau elle même / gère également le mode connectée (notion d'authentification)

6 = assure la mise en forme de la données / peut aussi avoir des fonctions de compression et de chiffrement (la couche de "présentation")

7 = assure l'interface user correction / gere la com entre apps (tjr authentification comme 5 et 6)

## TCP/IP

Simplification du OSI

Evolutif / efficace

couche 1/2 = Accès Réseau (gestion du supp de transmission et acheminement des infos dans le LAN)

3 (reseaux) = internet (paquets entre réseaux)

transport = transport (remise des infos au destinataire)

5 6 7 = application (SMTP / FTP / DNS... app standart du réseaux) protocoles applicatifs

## Analyste SOC / Normes :

normes couches 1/2 :

trame 802.3 : c'est l'ethernet (1882)

format (ethernetv2) : 8octets préambule(début trame/ permet synchronisation 10101010.. tant que c'est pas 11) / 6macdestination / 6macsource / 2ethertype(protocol pour savoir quel truc : 0x0800 : IP 802.1q encapsulationVLAN 0x0806 ARP) / 46-1500 données / 4 CRC(ça vérifie les bonnes valeurs)

802.11 : wifi

802.16 : wimax

802.1X : authentification triple A (filaire et wifi)(très fort)(Radius) reprise par Sisco (tacsas et non pas takas)

normes couches 2 :

48bits / 6octets pour l'adresse MAC (octet = octale = octogone = 8 cotés = 8bits)

une adresse MAC est composée de 3 blocs que 3 octets

3ers c'est via l'IEEE (id matériel(constructeur)) et les 3ders c'est la machine physique elle meme

FF:FF:FF:FF:FF:FF (tout les bits sur 1)

802.X : sous couche MAC

accès au supp physique / attribution des noms(adressage phys) / controle d'erreur(frame check sequence) verifie intégrité de la trame d'info entre recep et emetteur (le CRC controle rodondence cyclique)

VLAN **802.1q**: création de gpe de taff indé de l'infra phys : appartenance a la vlan est def par un gestionnaire de réseau / vlan indé de l'emplacement phys décapsulation dot1q.

avantages vlan : perf : limite la diffusion des broadcast / sécu : separation des fluxentre les diff groupes d'user / finance : un seul equipement pour plusieurs réseaux

scapy (Kali - python) : creer des paquets/ hack vlan HOPPING

VLANs(ecure) comme dans le nom. PVLAN (p pour private) bien plus recent et mieux, chaque personne est encapsulé dans sa bulle et ça crée un pool pour chaque departement (comme Sisco aironet pour faire ça) (et askip tu peux faire des dingeries comme le Maclocking (lock certains trucs genre un switch))

couche 3 :

encapsulation IP

bourrage/padding pour atteindre les 60 octets. Le **bourrage** (ou padding) est l'ajout d'octets (souvent des zéros) à la fin d'une trame Ethernet pour que sa

taille totale atteigne **64 octets**. Cela permet de garantir la détection des collisions et la compatibilité avec tous les équipements réseau. Si la partie « données » de la trame est trop courte, on complète avec du bourrage jusqu'à ce que la trame

fasse au moins 64 octets (en-tête, données, bourrage et FCS inclus)

On peut envoyer des messages avec ping, car on peut remplir de bourrage comme on veut.

fragmentation : identification du fragment (prendre schéma de cours)

TTL (Time to Live), ça a que les couches 3 ou+ qui peuvent décrémenter le TTL. nbr de saut autorisé pour joindre qq chose

le TTL ping change en fonction de l'OS, Linux = 64 Windows = 128 Cisco = 255  
très utile en hack pour connaître l'OS

ping echo request, c'est obligé ping echo response ensuite

cloudflare 1.1.1.1 meilleur DNS

ping / tracer pour savoir le TTL

impossible de se ping entre deux machines depuis W7. Et il ne faut pas autoriser ping car on pourrait sortir des données via ping

## ARP ICMP RFC

protocoles : (couche2)ARP (couche3)ICMP DHCP DNS HTTP

ARP : Le protocole ARP (Address Resolution Protocol) permet de trouver l'adresse MAC associée à une adresse IP sur un réseau local, afin d'acheminer correctement les paquets de données

ICMP : ping entre autres (pour les recherches de pannes), véhicule des msg d'erreurs (+30 types)

**RFC** (Une **RFC** (Request for Comments) est un document officiel, numéroté et publié par l'IETF, qui définit ou décrit des protocoles, normes ou méthodes d'Internet)

RFC : 792, 950, 1256.

protocole de niveau réseau

1 octet pour le type et un pour le code (info+ sur le type) et ça donne direct le pb.

echo request ping : type 8 code 0 / echo ping reply type 0 code 0

type3 = probleme, faut chercher le code pour savoir plus

TCP(HTTP SMTP FTP)/UDP(DNS DHCP) couche 4transport

## TCP

port 65535\*2, le port 0 existe pour loopback

ports UDP importants : 53 domain / **67-68 DHCPc DHCPs** /  
**123ntp(networktimeprotocol)** / 137 netbios-ns 128 netbios / **161**  
**snmp(surveillance de l'etat des machines)** / 514 syslog

ports TCP : **20 21 ftp-data ftp(transfert de data en 20 mais id en 21)** / 22 23  
ssh telnet / **25smtp** / **80 443 http https** / **88 kerberos** / **445 MS SMB** / 143  
imap / 179 bgp / **3306 mysql** / 3389 Ms term / 5355 LLMNR

(ftp et telnet pas ouf niveau sécurité)

**TCP : three way handshake (1er condition pour communiquer en TCP)**

**(client)SYN → (serveur)ASK+SYN → ACK (SYN SYNASK ASK)**

ils sont là pour se mettre d'accord sur la taille de la fenetre

renégocie la fenetre.

RTT = temps de calcul entre deux protagonistes (Round Trip Time . Temps allée retour)

**flags TCP : URG (priorité urgence) / ACK(aquittement) / PUSH(transmettre direct couche supérieur) / RST(fermeture/erreur) / SYN(début connection) / FIN(fin connection)**

c'est que un par paquet

TCP a évolué, la window a été améliorée

## **Windows Size Value et Windows Size Scaling**

Cela permet d'adapter dynamiquement le volume de données envoyées sans accusé de réception, améliorant ainsi l'efficacité et la performance des connexions TCP sur Internet.

Follow TCP stream ⇒ l'index stream 0 (

dans Wireshark permet d'extraire et d'afficher toutes les données échangées dans une

conversation TCP précise, comme un échange entre un client et un serveur

La fonction **Follow TCP stream** dans Wireshark permet d'extraire et d'afficher toutes les données échangées dans une conversation TCP précise, comme un échange entre un client et un serveur.

Lorsque tu vois **stream index 0**, cela signifie que Wireshark a attribué l'index 0 à la première conversation TCP détectée dans la capture : tous les paquets de cette conversation porteront le même numéro de stream (ici 0), ce qui permet de facilement isoler et analyser ce flux de communication précis.

## UDP/DHCP/DORA :

rapide, mode non connecté, référence chaque app par un num de port

Couche Application(5-6-7) : DHCP & DORA

(attaque DHCP Starvation / Rogue DHCP)

**DHCP c'est DORA → UDP 67 et 68. (67 = serv / 68 = client)**

**DORA = Discovery Offer Request Ack**

## DNS(Domain Name System)

avoir une IP dans un nom genre google.fr → 8.8.8.8

Racines DNS estempillée par le "." comme www.toto.fr

Common DNS Record Types

Type d'enregistrement	Description
<b>A IPv4</b>	Associe un nom de domaine à une adresse IPv4 (32 bits). Permet de traduire un nom en adresse IPv4
<b>AAAA IPv6</b>	Associe un nom de domaine à une adresse IPv6 (128 bits), utilisée pour la nouvelle génération d'adresses IP
<b>CNAME Canonical Name</b>	Crée un alias d'un nom de domaine vers un autre nom de domaine (pas une adresse IP). Utile pour gérer des sous-domaines ou rediriger sans changer d'adresse IP directement
<b>Mx Mail exchanger</b>	Indique les serveurs de messagerie responsables de la réception des emails pour un domaine. Doit pointer vers un nom de serveur, pas une IP

Type d'enregistrement	Description
<b>NS Nameserver</b>	Spécifie les serveurs DNS faisant autorité pour un domaine ou une zone DNS, responsables de sa résolution
<b>PTR Pointer</b>	Utilisé pour la résolution DNS inverse : associe une adresse IP à un nom de domaine (inverse du A ou AAAA). Utilisé dans les zones inverses
<b>SOA Start of Authority (server of authority)</b>	Contient les informations principales sur la zone DNS : serveur principal, contact, numéro de série, délais de mise à jour, etc. C'est l'enregistrement de base d'une zone DNS
<b>SRV Service Location</b>	Permet de localiser des services spécifiques (ex : messagerie instantanée, VoIP) en indiquant un hôte et un port pour un service donné
<b>TXT Text</b>	Permet d'ajouter des informations textuelles arbitraires dans la zone DNS, souvent utilisées pour des validations (SPF, DKIM, vérifications de propriété)

TLD = .xx donc lié a la racine DNS

Le DNS, par Stéphane Bortzmeyer - partie 1 : conférence

Creative Commons BY:SA

<http://www.iletaitunefoisinternet.fr/dns-bortzmeyer/>


<https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.youtube.com/watch%3Fv%3DQHVK666TFUI&ved=2ahUKEwjp6b2ru46NAXVsRKQEHsSsrcQwqsBeQIDAF&usq=AQyVaw3oFip8B4KKbE2W6lnzIao>

### Avec dig

```

AAAA www.bortzmeyer.org
HEADER<<- opcode: QUERY, status: NOERROR, id: 1135
gs: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 3, /
WER SECTION:
rtzmeyer.org. 10791 IN AAAA 2605:4500:2:245b::
ry time: 2937 msec
VER: 192.168.10.110#53(192.168.10.110)
N: Wed Aug 14 17:46:47 2013
SIZE rcvd: 164

```



## DNSSEC (Domain Name System Security Extensions)

l'authenticité des réponses DNS et ainsi protéger contre l'empoisonnement du cache et l'usurpation DNS (meme chose pour DDNS) mais ça date de 2015

DNS tout seul c'est pas du tout sécurisé

le "futur" c'est l'ENS ⇒ Ethereum Naming Service (viens de l'arbre de Merkle)

**ENS** est l'équivalent Web3 du DNS, qui transforme des adresses cryptographiques longues et complexes en noms simples et mémorables, tout en restant sécurisé, transparent et résistant à la censure

## HTTP / HTTPS

On utilise des méthodes → commande spécifiant un type de requete, elle demande au serveur de faire une action.

en général l'action concerne une ressource identifiée par l'URL

ex : une requête GET est envoyée pour récupérer la page du site web

Methode	Description
GET	Demande la ressource au serveur, sans effet sur la ressource. Elle peut être répétée sans effet
HEAD	Demande que les infos sur la ressource sans demander la ressource elle-même
OPTIONS	Liste des méthodes prises en charge dans l'URL
POST	Envoie des données au serveur en vue d'un traitement à une ressource

## Résolution DNS

1. **Navigateur** demande l'adresse IP de « google.com ».
2. **Client DNS** (ton ordinateur) :
  - Query IPv4
  - Answer IPv4
  - Query IPv6
  - Answer IPv6
3. **Serveur DNS récursif** :
  - Vérifie son **cache**.
  - Si non trouvé, interroge la **racine DNS**.
  - La racine indique le **serveur TLD** (ici, pour .com).
  - Le serveur TLD indique le **serveur faisant autorité** pour google.com.
  - Le serveur faisant autorité répond avec l'adresse IP de google.com.
4. **L'adresse IP** est renvoyée au client.
5. **Navigateur** reçoit l'adresse IP de Google.

## 2. Connexion à Google

- **Navigateur** établit une connexion TCP (port 80 ou 443). (3Way Handshake)
- **Navigateur** envoie une requête HTTP/HTTPS (ex : GET / (GET de la racine)).



(Si on passe en HTTPS, ça peut être demandé par le serv ou le client. c'est le client hello, donc on chiffre → Client Certificat en TLS1.3 PUIS après ya le GET /)

- **Serveur Google** 200OK puis POST /

**200 OK** est un code de statut HTTP qui indique que la requête envoyée par le client (par exemple, ton navigateur) a été traitée avec succès par le serveur.

- **Serveur Google** répond avec la page web.
- **Navigateur** affiche la page.

La partie 2 (TCP + HTTP) porte le nom de l'**index stream (Index de flux)**

## Status Codes HTTP

Classe	Plage de codes	Signification générale	Exemples courants
1xx	100–199	<b>Informatif</b> – La requête est en cours de traitement	100 Continue, 101 Switching Protocols
2xx	200–299	<b>Succès</b> – La requête a été traitée avec succès	200 OK, 201 Created, 204 No Content
3xx	300–399	<b>Redirection</b> – Une action supplémentaire est nécessaire	301 Moved Permanently, 302 Found, 304 Not Modified
4xx	400–499	<b>Erreur client</b> – La requête contient une erreur	400 Bad Request, 401 Unauthorized, 403 Forbidden, 404 Not Found
5xx	500–599	<b>Erreur serveur</b> – Le serveur n'a pas pu traiter la requête	500 Internal Server Error, 502 Bad Gateway, 503 Service Unavailable

## Suivre un paquet TCP

Chemin de réseau incluant : 1 client, 2 switchs, 1 routeur standard, 1 routeur qui fait des translations d'adresse réseau NAT(Network Address Translations) et 1 serveur.

- **NAT** : permet à plusieurs appareils d'utiliser une seule IP publique.

- **Tableau de NAT** : associe les adresses internes et externes pour router correctement les réponses.

Les box utilisent du NAT PAT dynamique.

Le **PAT** est la méthode la plus courante dans les réseaux domestiques ou d'entreprise pour permettre à plusieurs appareils d'accéder à Internet via une seule adresse IP publique, en utilisant les ports pour distinguer les connexions. La table **CAM** est la « mémoire » du switch qui lui permet de savoir sur quel port envoyer une trame en fonction de l'adresse MAC de destination (même si elle peut contenir des informations, comme l'IP de la machine, le netbios, l'heure)

### Relire le schéma en fonction du point de sniffage

hub = multiprise

switch = gère les trames / analyse / sait qui parle à qui / VLAN.

Options pour voir le trafic sur le réseau Ethernet :

- Capturer directement la machine (Wireshark) (Illégal)
- Spanning du port switch de l'hôte (il faut se connecter au switch en direct et Wireshark)
- Mise en place d'un TAP (Test Access Port) (avec un Hub ou TAP pro)

## Wireshark

Pas le droit de l'utiliser en entreprise. il faut des autorisations.

lecture de fichiers de capture, ça reste un outil simple, mais pas un outil de capture.

pour le wifi il faut des intercepteur WLAN pour éviter de faire galérer Wireshark (Tshark capture)

Librairie GITLAB Wireshark Foundations, dans le wiki y'a le SampleCaptures.

<https://www.wireshark.org/docs/dfref/>

On peut récupérer sa config Wireshark dans le dossier de HELP.

Wireshark récupère tout les paquets car il détruit le filtre MAC.

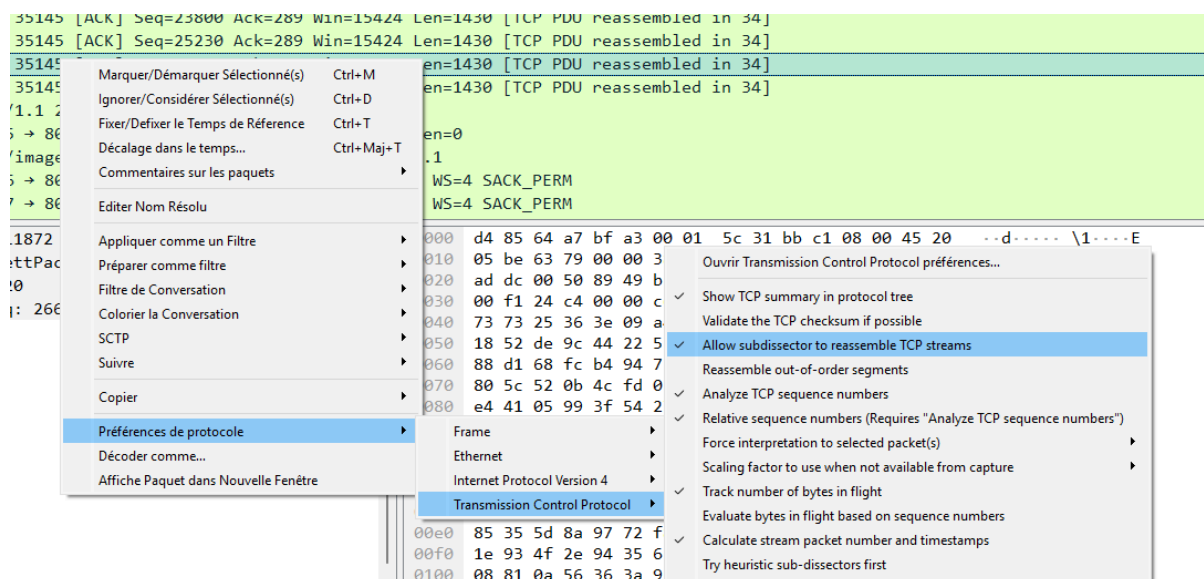
Dissection euristique de Wireshark

1<sup>ère</sup> chose à cliquer (JAUNE ROUGE (vert = note sur des paquets)) = expert info en bas à gauche

puis Statistiques → Protocol Hierarchy (c'est le résumé des protocoles) → Conversations (tableau de tout les interlocuteurs qui se sont parlés) → Endpoints (discussions finis entre deux personnes)

Promqry is a command line tool that can be used to detect network interfaces that are running in promiscuous mode.

DANS LES PAQUETS TCP DESACTIVER → CLIC DROIT → PROTOCOLE PREFERENCES → TCP → ALLOW SUBDISSECTOR (et si vieux wireshark, on coche track number of bytes in flight et calculate stream packet number and timestamps) (et desactiver Validate the TCP checksum car Wireshark galère)



Pour trouver une information, faut se demander dans quel couche OSI se trouve cette dite information.

Edit → preference / apparence → column pour changer les colonnes

Il y a deux default dans Wireshark : les fichiers > a 100Mo et un flux beaucoup trop rapide par rapport a la machine

(pour les gros fichiers il y a Cascade Pilot (Payant) et Zui Brim data (gratuit))

(et quand le flux est trop fort, Dropped: x / ACKed Lost Segment / Previous Segment Not Captured )

deux filtres fort : "contains" et "matches"

tel que frame contains xx (si on est sur de l'ortho)

ou frame matches "(?) xxx" et là ya pas de probleme de casse.

ou encore frame matches "(?i) (xxx|xx|xxxx)".

ou encore frame matches "bon.our" ou "bon.{1,3}r" et il cherche les mots avec les lettre bon +1 à 3 lettres random et fini par r

## Attaques (+ Défenses)

ManInTheMiddle / ARP Spoofing (usurpation d'identité) → Problèmes ARP (flood) (empoisonne les caches ARP) (arp-a pour voir le cache ARP) (BetterCap pour faire ça)

attaque : arpspoof -i eth1 -t xxx.xxx.x.xx(cible) xxx.xxx.x.x(celui que tu usurpes) (il bombarde une machine pour lui dire qu'il est une autre machine)

et on bombarde aussi l'autre machine pour se faire passer pour l'autre

et on active le mode routeur de notre machine : echo 1 > /proc/sys/net/ipv4/ip\_forward

(pour trouver = 2fois meme IP avec 2MAC différentes)

---

### Attaque par MAC Flooding

Noyer le switch pour qu'il arrive plus a corréler et ducoup ça deviens comme un hub et il envois de partout. En ARP/TCP (ou autres). la commande c'est "macof"

Le switch perds sa table de corrélation CAM → hub

---

### DHCP Starvation

Exploite la faiblesse DHCP (pas d'authentification)

Envoie massivement des DHCP avec des MAC spoofées

Epuise les adresses disponible dans le DHCP

Empeches les nouvelles machines de rejoindre le réseau

Peut préparer une attaque de type rogue DHCP

---

### VLAN hooping

Permet de contourner l'isolation entre VLANs si mal configuré. l'atta envois une trame avec deux balises VLAN, la balise externe correspond au vlan natif / la balise interne cible un vlan interdit

le premier switch retire la balise externe et transmet la trame

le second lit la balise interne et achemine la trame vers un vlan non autorisé

---

Problèmes des protocoles d'authentification en clair

des protocoles envoient les id sans chiffrements (FTP / HTTP / POP3..)

un attaquant peut sniffer les paquets et lire les mdp en clair / Ces protocoles sont vulnérables aux attaques MITM et sniffing passif

Ils sont obsolètes et remplacés par :

Telnet → SSH | FTP → SFTP ou FTPS | HTTP → HTTPS | POP3 → POP3S

---

Défense !

Avant d'installer un périphérique réseau, il faut regarder physiquement si il n'y a pas de problèmes.

injecter le firmware par nous même, sur le site du constructeur.

mettre en horloge de temps synchronisé à notre entreprise (server NTP)

après on peut installer son switch.

compte admin : (sans le nom admin dans le nom ni le mdp)

mécanisme de chiffrement dans le périphérique à activer.

Pour stopper tout MITM sur le switch : Port security

port security mac address auto apprentissage. On peut alerter ou stopper le port ou stopper le switch

tous les ports qui ne sont pas mac lockés y'a personne dessus, all ports switch off

DHCP Snooping :

Cela permet qu'un seul port émet les offres. DHCP Snooping trust port xx

DAI(Dynamic ARP Inspection) : le switch est capable de contrôler les paquets du réseau

quel mac sur quel ports et connaître le serveur DHCP. Le switch peut savoir si le paquet est un faux grâce à ça.

Pour réussir le TP → Utiliser le Retenir+++++

**Mots clefs :**

Trouble Shooting TS (commence par en bas et on remonte)

OSI

TCP/IP

VLAN

Ressources :

FRAMEIP.COM

Wireshark

<https://academy.binance.com/fr>

RTFM BOOK (noir) redhat

BTM BOOK (noir) bluehat

<https://www.wireshark.org/docs/dfref/>