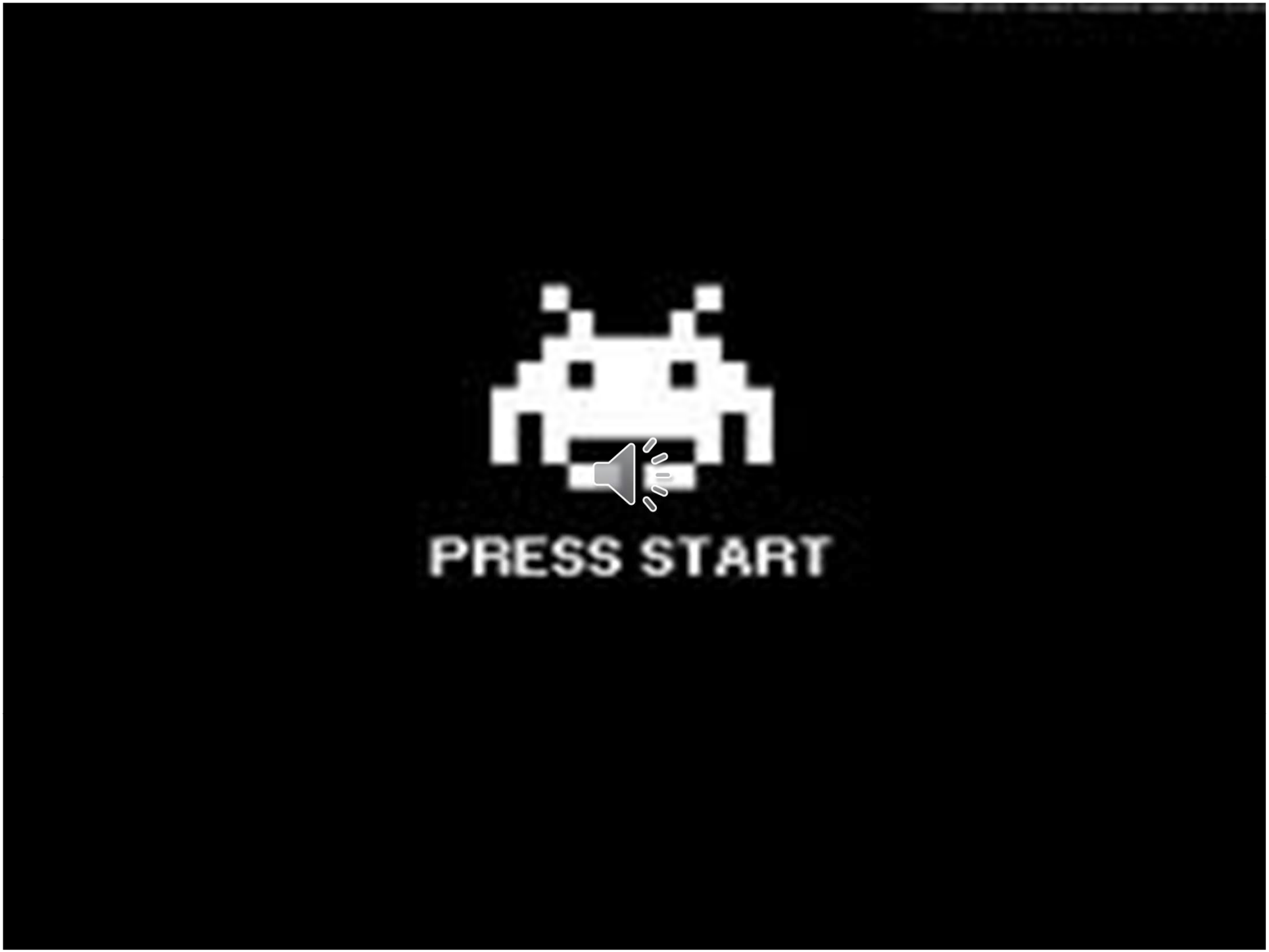

PRESS START

Xposé logiciel, système et **réseau**

Le tunneling

Le **tunneling** est un outil de plus en plus **sollicité**

- Solutions réseau
- Solutions logiciel
- Sécurité
- Insécurité

Les choses que nous allons aborder

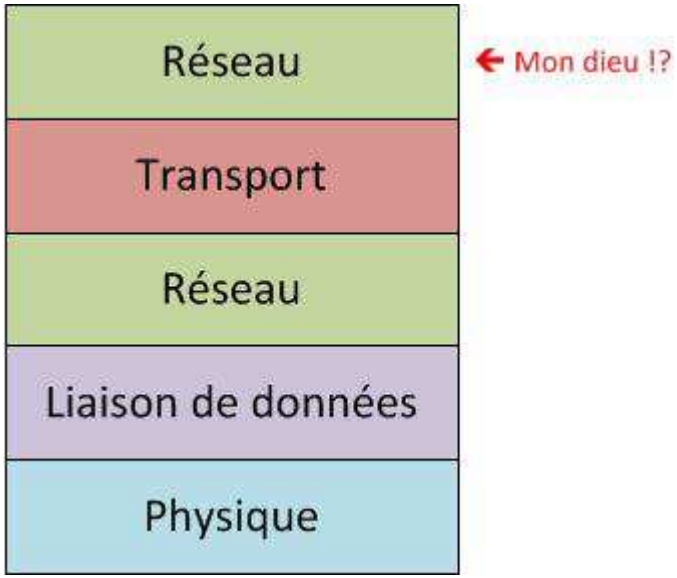
- Le **tunneling** c'est quoi ?
- L'étude de l'outil **SSH**
- **Pourquoi** et **comment** utiliser le tunneling

Le tunneling

- Tout d’abord, qu’est ce qu’un **tunnel** :

Le tunneling

- Un **tunnel** pour un fan du modelé OSI :



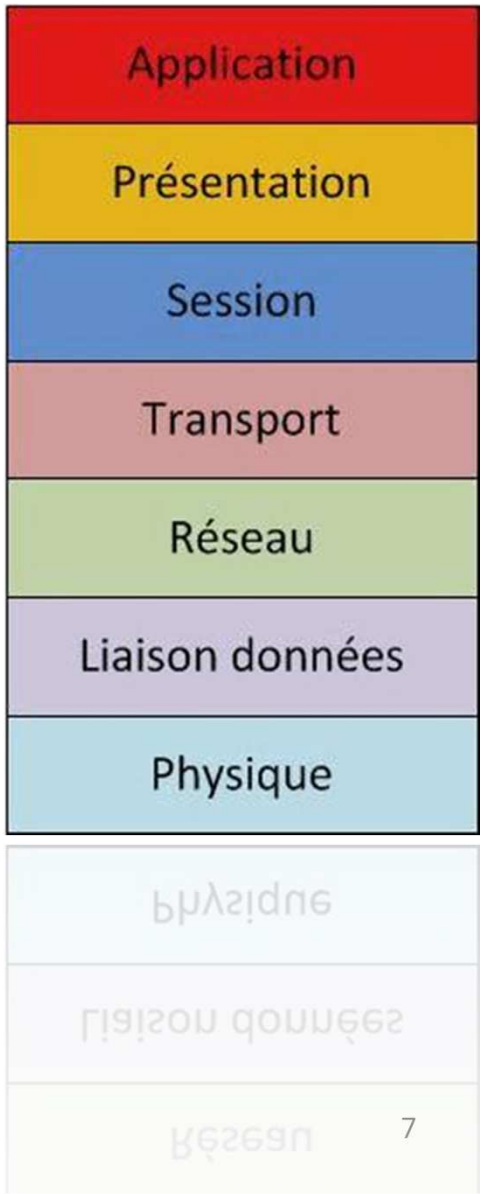
Mon dieu !?

« Un protocole qui encapsule un protocole de niveau plus bas ou égal. »

Le tunneling

Rappel qui peut s'avérer utile:

- Ethernet, 802.11 → OSI 2
- IP, ICMP → OSI 3
- TCP,UDP → OSI 4
- TLS/SSL → OSI 6
- DNS, SSH, HTTP, TELNET → OSI 7



Le tunneling

➔Le tunneling c’est l’art d’utiliser des tunnels.

➔C’est un concept qui reste très simple.

➔Permet néanmoins de faire de grandes choses.

Le tunneling

Trucs faux qu'on entend souvent :

- « Tunneling c'est pareil que VPN»
- « Le JAVA c'est plus rapide que le C »

L’outil SSH

- Pourquoi je vous parle maintenant de SSH:
  - Permet de faire du tunneling
  - Va être utilisé au cours de l’exposé

- **Objectif** de cette partie: *Comprendre comment SSH permet le tunneling*

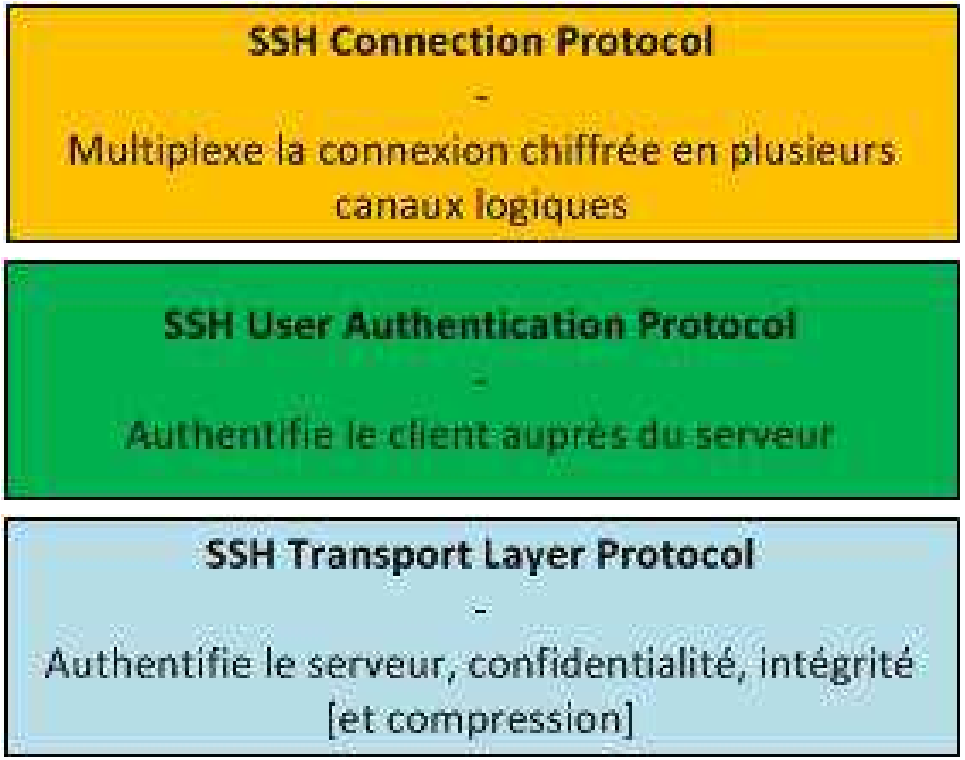
L’outil SSH

- Secure Shell (v2)
- Protocole **applicatif** (OSI 7)

- RFC 4250 -> 4255:
  - 4251 « The SSH Protocol Architecture »
  - 4252 « The SSH Authentication Protocol »
  - 4253 « The SSH Transport Layer Protocol »
  - **4254 « The SSH Connection Protocol »**

L’outil SSH – Architecture logicielle

SSH est organisé en 3 protocoles :



Plus d’explications dans les slides à venir

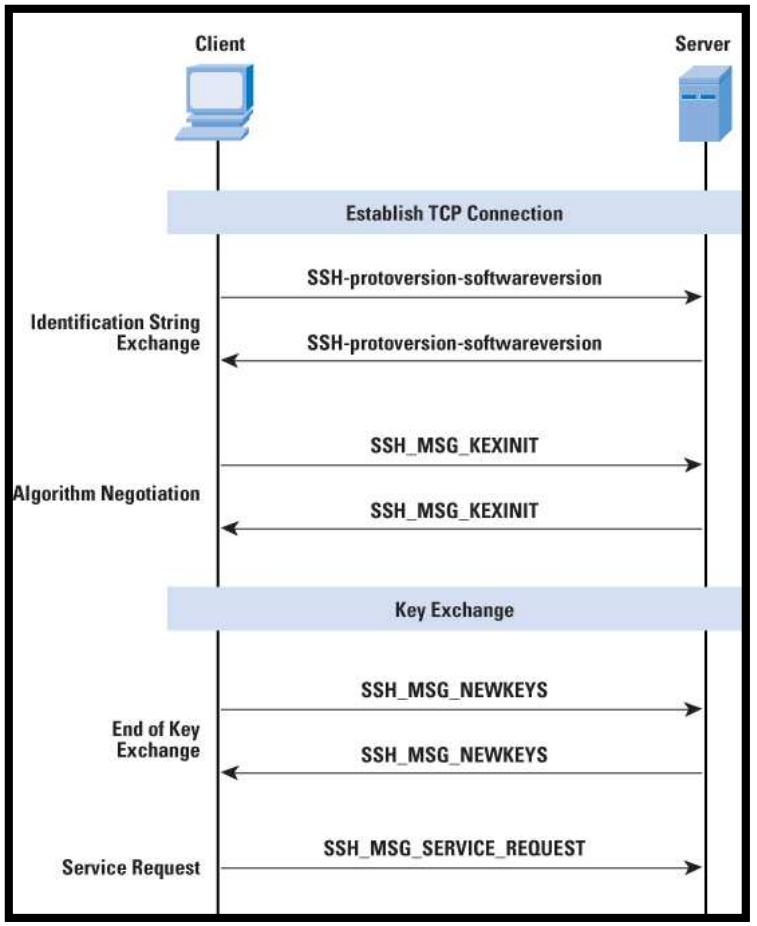
Le client est authentifié auprès du serveur SSH

Le serveur SSH est authentifié et on a notre connexion chiffrée

L’outil SSH – Architecture logicielle

- **SSH Transport Layer Protocol:**
  - Echange versions SSH
  - Echange algorithme chiffrement
  - Echange des clefs via Diffie Hellman. L’authentification du serveur se fait pendant l’échange Diffie Hellman.

➔ **Serveur authentifié** et **Connexion chiffrée**



Source – Cisco.com

L’outil SSH – Architecture logicielle

- **SSH User Authentication Protocol:**
  - Client envoie

byte	SSH_MSG_USERAUTH_REQUEST (50)
string	username
string	service name
string	method name
....	method-specific fields

  - Plusieurs méthodes :
    - Publickey ➔ Chiffrement asymétrique. Le client dispose paire de clefs.
    - Password ➔ Le client envoie le mot de passe (chiffré car dans SSH TLP).
    - Hostbased ➔ Idem que publickey mais pour un host (permet multi client).

L’outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - Fonctionne au dessus de SSH Transport Layer Protocol.
  - La session terminal de SSH.
  - Permet la création de **tunnels**.

L’outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - L’ouverture d’une **session** (d’un canal):

```
6.1.  Opening a Session

   A session is started by sending the following message.

      byte      SSH_MSG_CHANNEL_OPEN
      string    "session"
      uint32    sender channel
      uint32    initial window size
      uint32    maximum packet size
```

Source – RFC 4254

L’outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - Par défaut, l’ouverture d’un **shell**:

```
byte      SSH_MSG_CHANNEL_REQUEST
uint32    recipient channel
string    "shell"
boolean   want reply

This message will request that the user's default shell (typically
defined in /etc/passwd in UNIX systems) be started at the other end.
```

Source – RFC 4254

Certains voient les données du shell comme les données utiles propres à SSH.

L’outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - TCP/IP Port Forwarding ➔ 
  - Deux types de port forwarding:
    - **Local** port forwarding
    - **Remote** port forwarding

L’outil SSH – Architecture logicielle

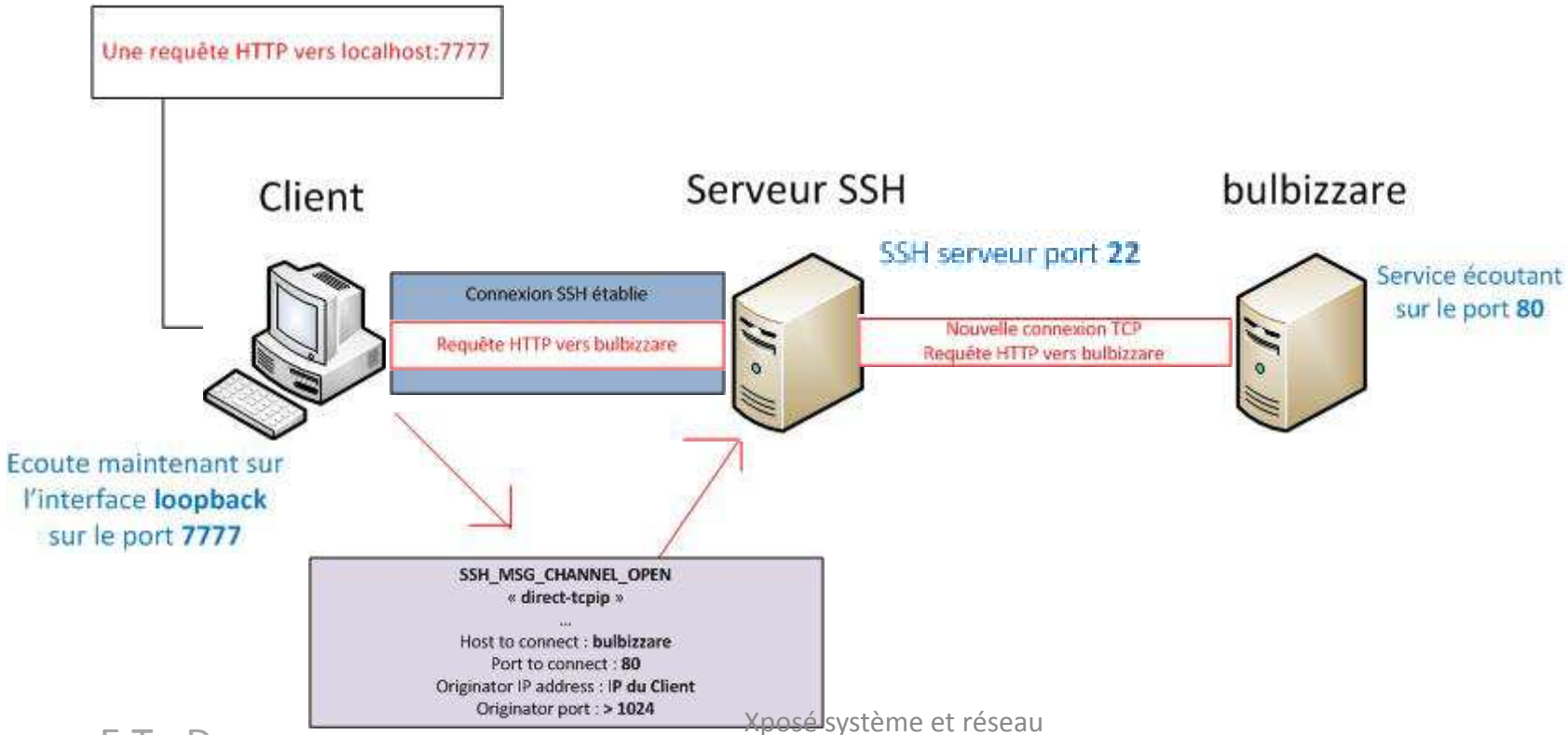
- **SSH Connection Protocol:**
  - **Local port forwarding :**
    1. Le client **C** initie une connexion SSH vers le serveur SSH **S**.
    2. Le client écoute un port X sur une interface Y.
    3. Pour chaque connexion sur Y:X le client envoi au serveur:

```
byte      SSH_MSG_CHANNEL_OPEN
string    "direct-tcpip"
uint32    sender channel
uint32    initial window size
uint32    maximum packet size
string    host to connect
uint32    port to connect
string    originator IP address
uint32    originator port
```

    4. Le client retransmet alors les données dans le canal crée.
    5. Le serveur retransmet le trafic vers « **host to connect** » sur le « **port to connect** ».

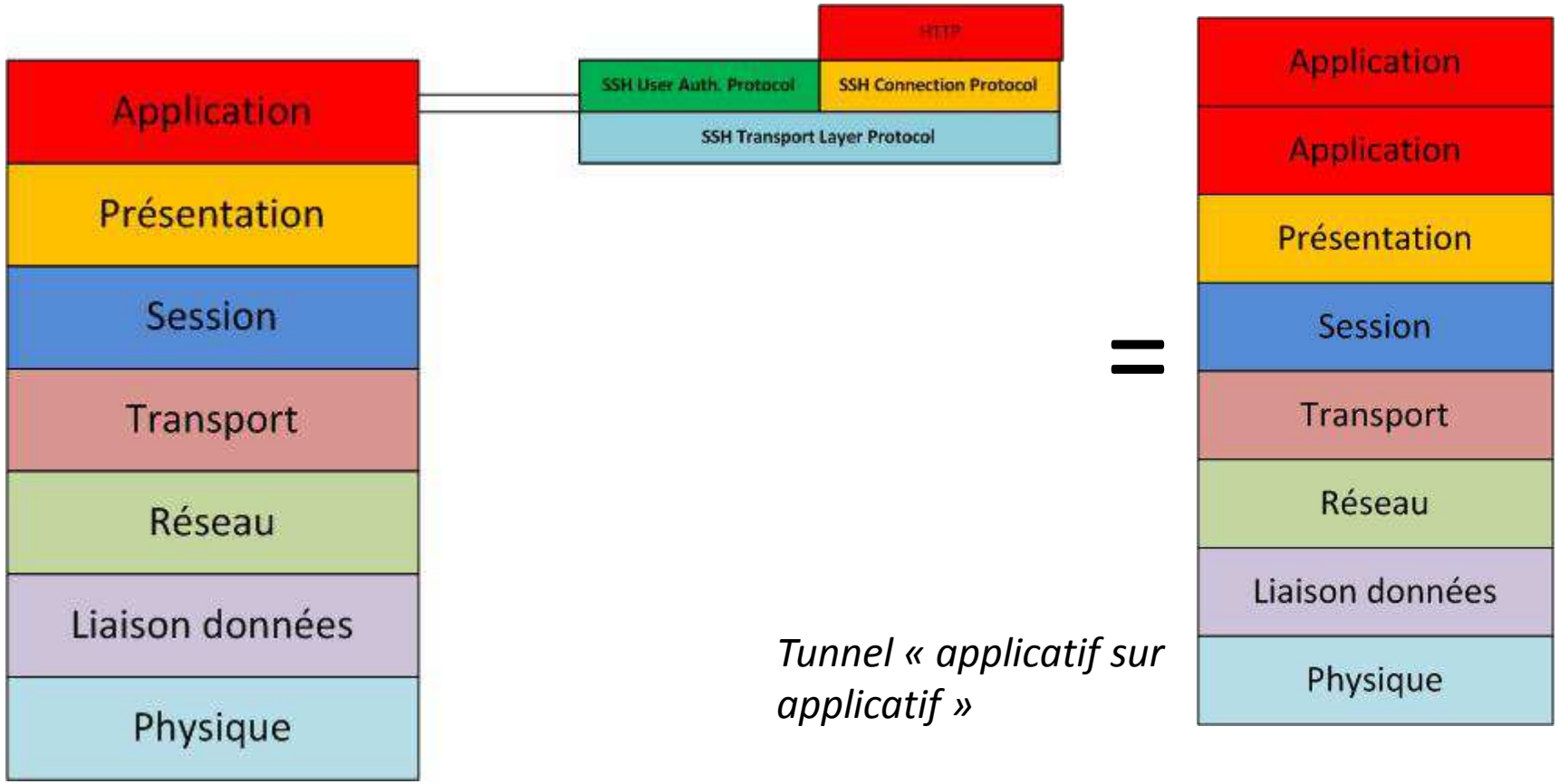
L'outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - **Local port forwarding : «** Ssh –L localhost:7777:blubizzare:80 serveurSSH**»**



L’outil SSH – Architecture logicielle

SSH Tunnel applicatif (exemple précédent):



Tunnel « applicatif sur applicatif »

L’outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - **Reverse port forwarding :**
    1. Le client **C** initie une connexion SSH vers le serveur SSH **S**.
    2. Le client envoie :

```
byte      SSH_MSG_GLOBAL_REQUEST
string    "tcpip-forward"
boolean   want reply
string    address to bind (e.g., "0.0.0.0")
uint32    port number to bind
```

    3. Le serveur écoutera alors sur son interface **X** sur le port **Y**.

L’outil SSH – Architecture logicielle

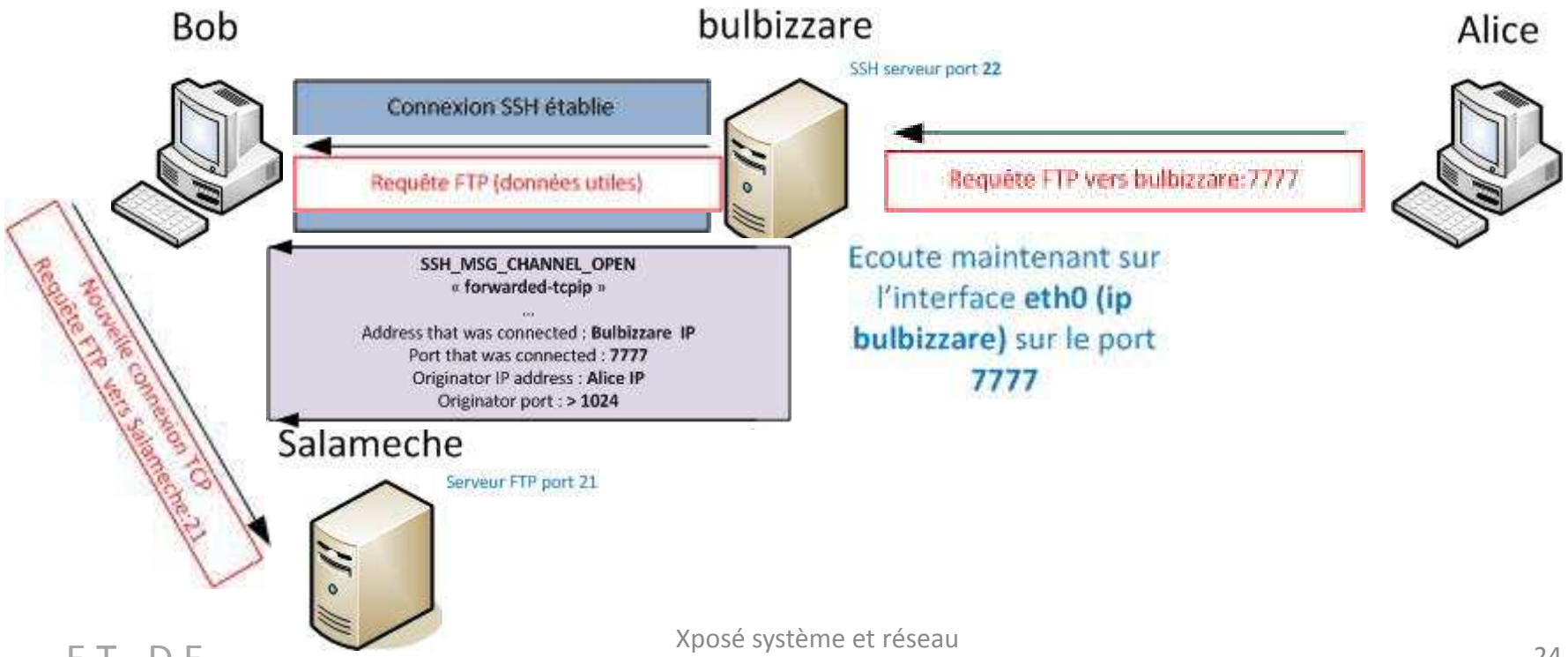
- **SSH Connection Protocol:**
  - **Reverse port forwarding (suite):**
    4. Lorsqu’une connexion est initiée sur le X:Y du serveur, ce dernier envoie au client :

```
byte      SSH_MSG_CHANNEL_OPEN
string    "forwarded-tcpip"
uint32    sender channel
uint32    initial window size
uint32    maximum packet size
string    address that was connected
uint32    port that was connected
string    originator IP address
uint32    originator port
```

    5. Un canal est ouvert vers le client, les données seront transférées vers le client qui retransmettra vers le port et l’adresse voulues (définis dans la commande ssh du client).

L'outil SSH – Architecture logicielle

- **SSH Connection Protocol:**
  - **Reverse port forwarding :«** ssh –R ipbulbizzare:7777:ipSalameche:21**»**



L’outil SSH – Architecture logicielle

Ce qu’il faut retenir :

➔Local port forwarding

➔Remote port forwarding

➔Connexion authentifiée et chiffrée

➔SSH est un outil puissant pour faire du tunneling.

Pourquoi et comment utiliser le tunneling

- Par souci de sécurité
- Pour étendre son réseau
- Pour outrepasser la sécurité

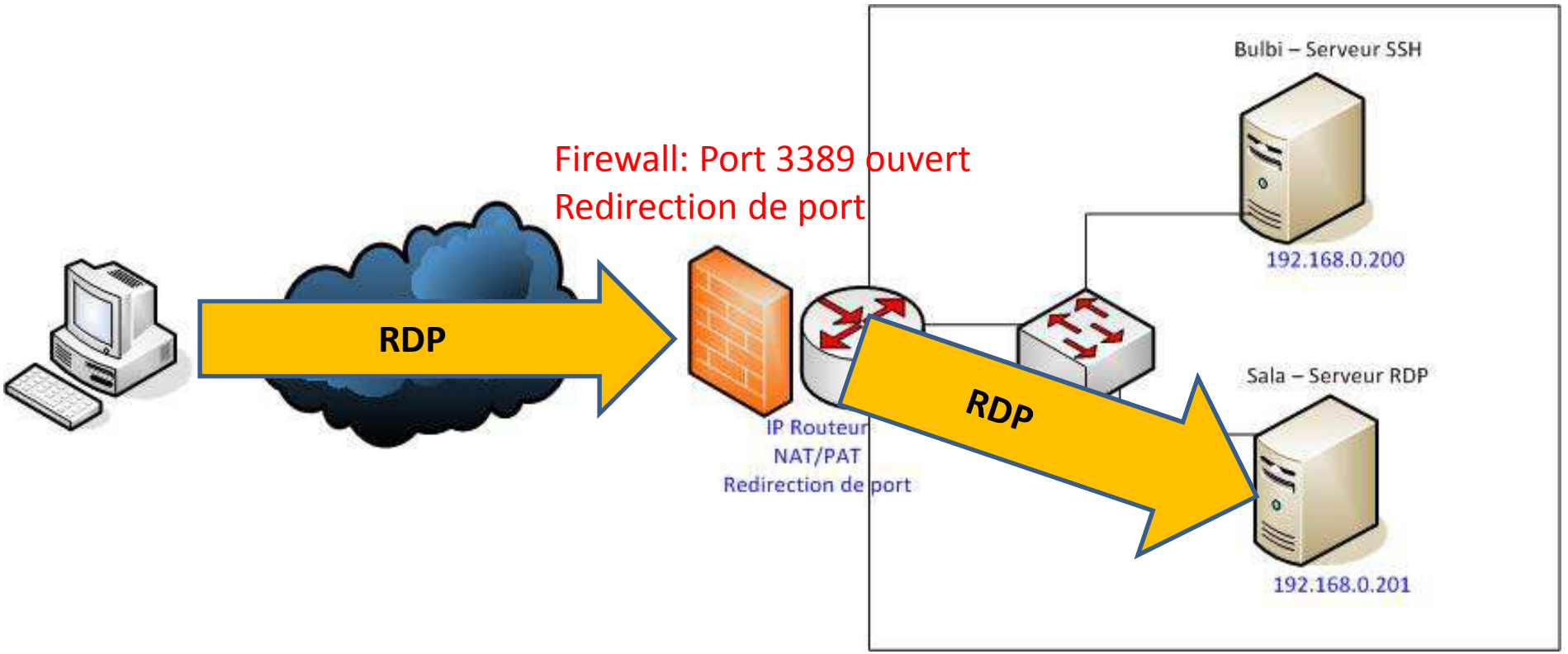
* *Liste non exhaustive*

Pourquoi et comment utiliser le tunneling

- **Souci de sécurité**
  - Un protocole qui implémente un algorithme de chiffrement un peu dépassé.
  - Un protocole sensible aux exploits.
  - Un protocole développé par Microsoft ?
  - Exemple : **RDP** [Remote Desktop Protocol]

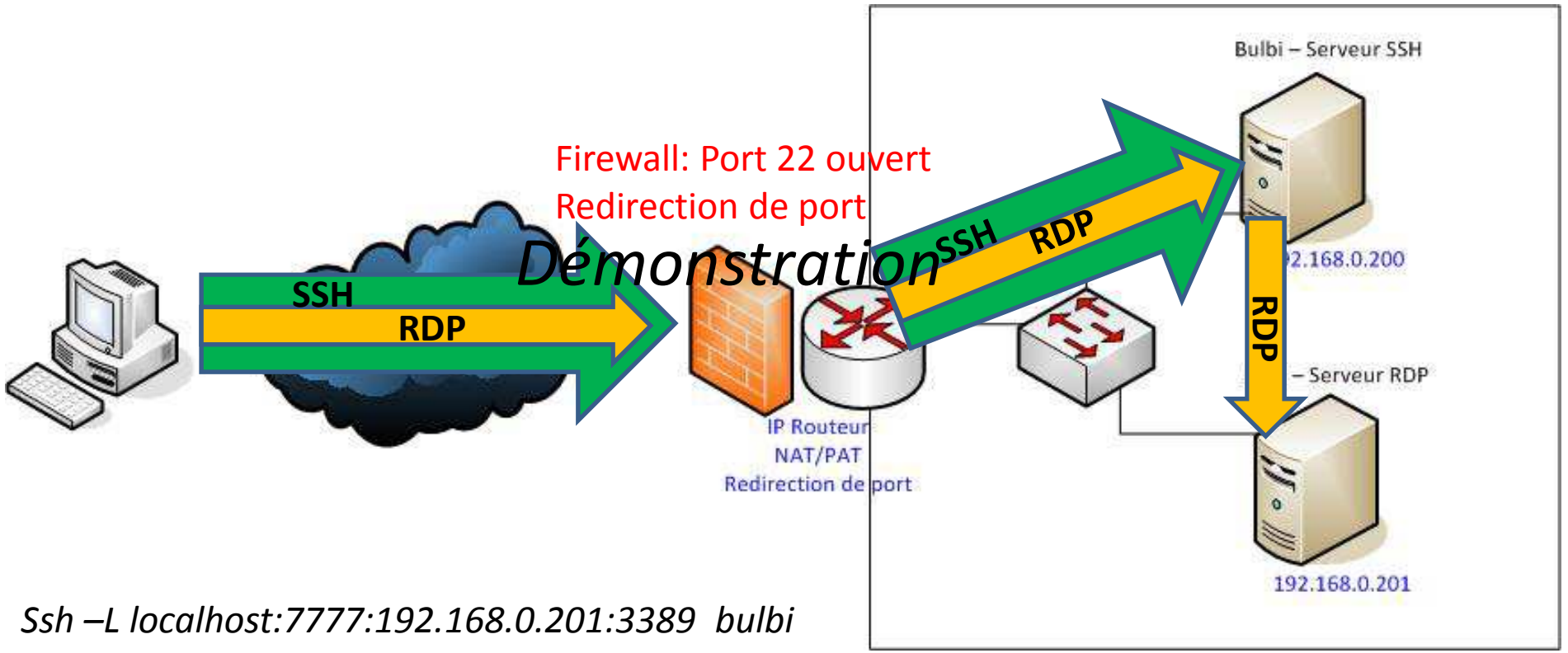
Pourquoi et comment utiliser le tunneling

- **Souci de sécurité [RDP] :**



Pourquoi et comment utiliser le tunneling

- **Sécuriser RDP**:



Ssh –L localhost:7777:192.168.0.201:3389 bulbi

Pourquoi et comment utiliser le tunneling

- **Etendre son réseau [VPN]:**
  - Relier des LAN distants sans utiliser de liaisons louées

Pourquoi et comment utiliser le tunneling

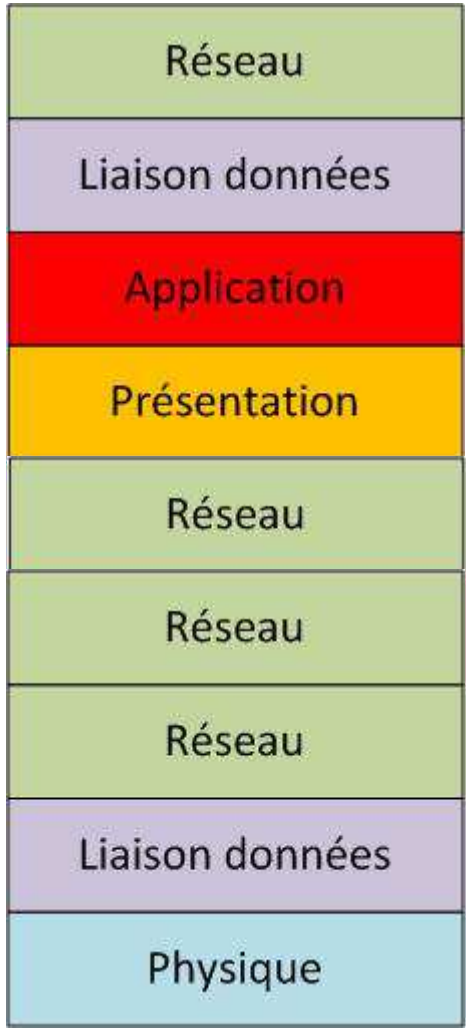
- **Etendre son réseau [VPN] :**
  - Encapsuler les **paquets IP.**

  Plusieurs façons :

OpenVPN

GRE, IPSec Nat Traversal

IP/IP



Pourquoi et comment utiliser le tunneling

- **Etendre son réseau [VPN] :**

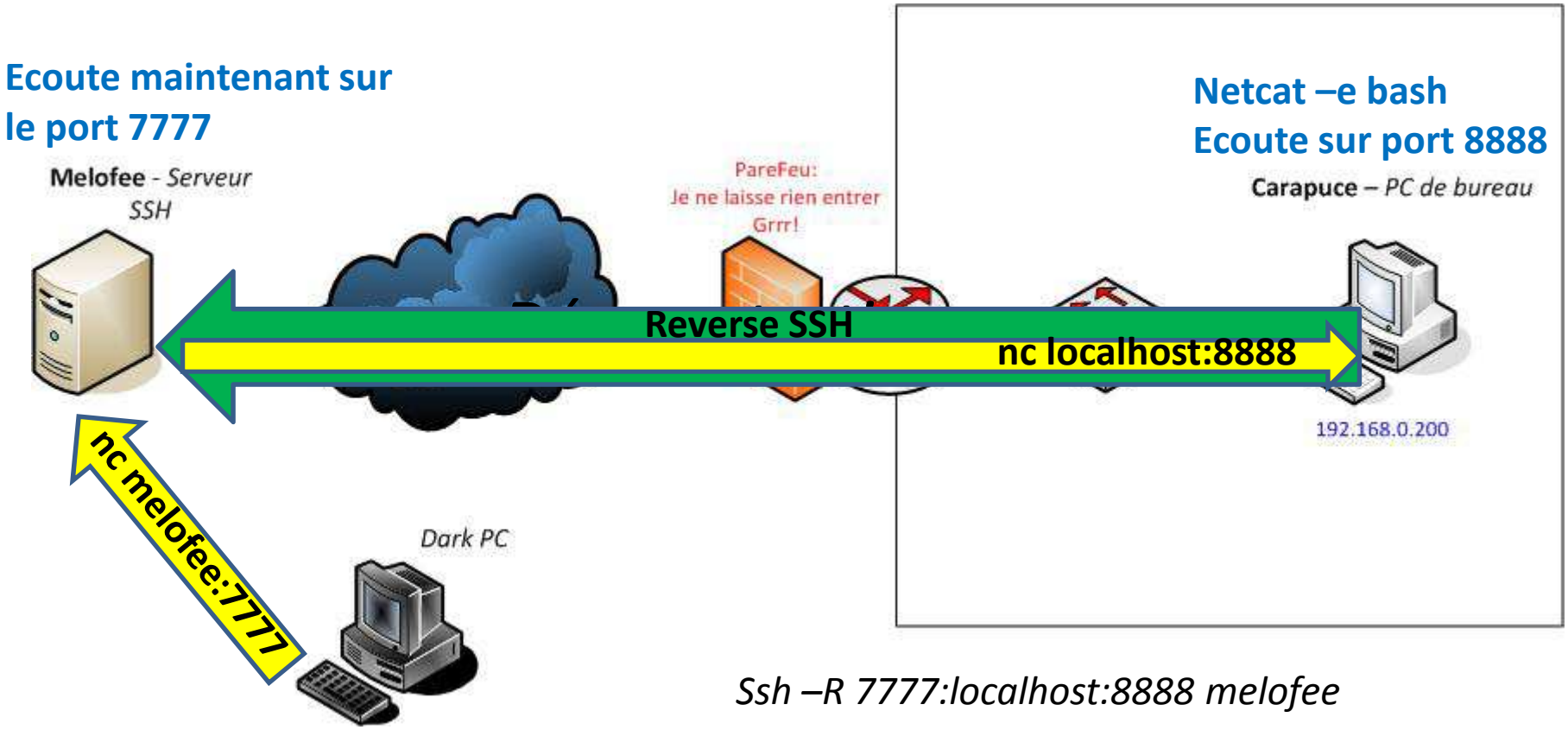
Démonstration

Pourquoi et comment utiliser le tunneling

- **Outrepasser la sécurité :**
  - *Reverse Tunneling* [Firewall bypass]
  - *Iodine* [Dns Tunneling]

Pourquoi et comment utiliser le tunneling

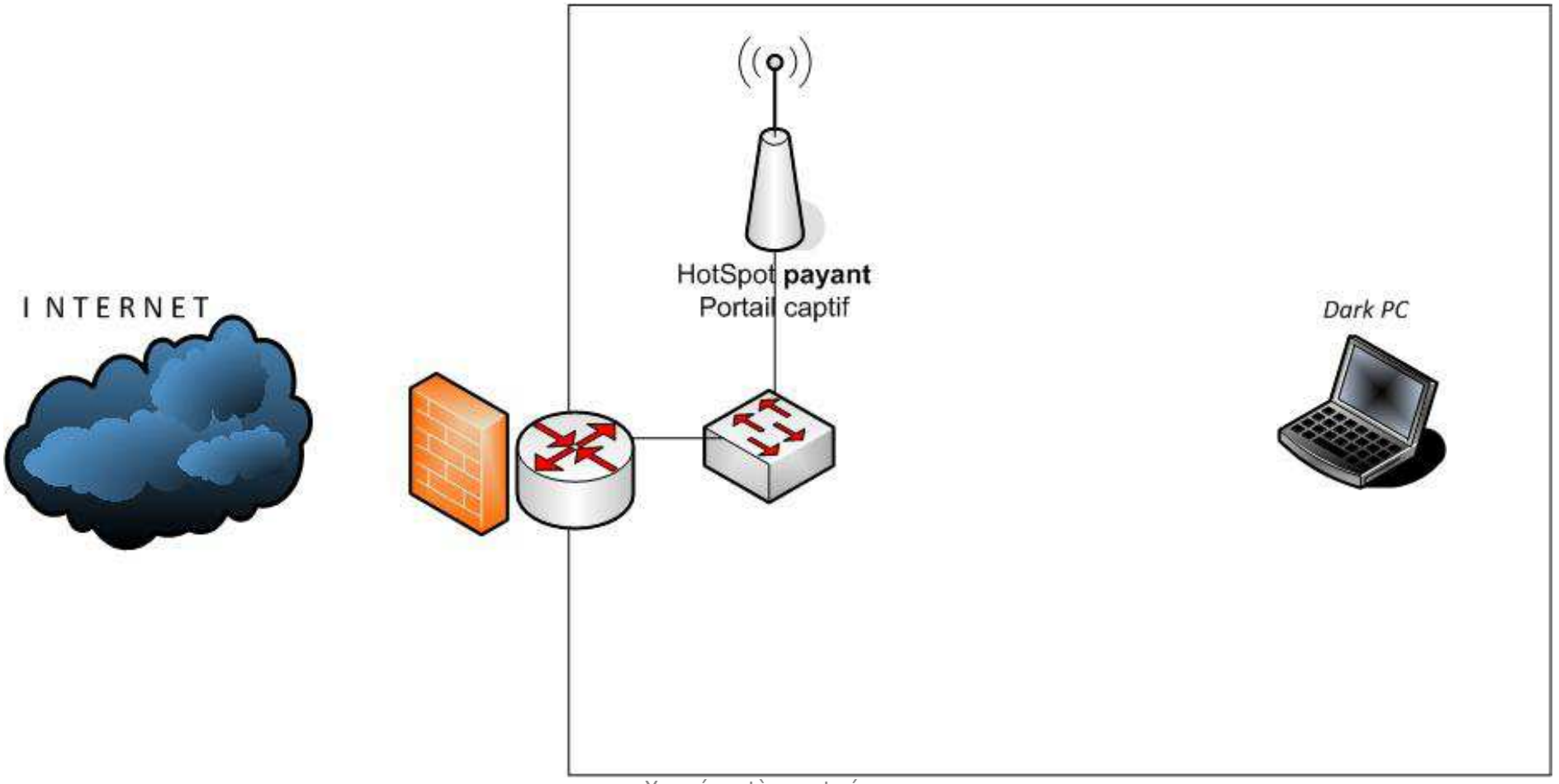
- Outrepasser la sécurité « **Reverse Tunneling** »:



Ssh –R 7777:localhost:8888 melofee

Pourquoi et comment utiliser le tunneling

- Outrepasser un **portail captif « Iodine »**:



Pourquoi et comment utiliser le tunneling

- Rappel sur les portails captifs

Cela est obtenu en interceptant tous les paquets qu'elles qu'ils soient leur destinataire jusqu'à ce que l'utilisateur ouvre son navigateur web et essaie d'accéder à Internet.

*La technique des portails captifs (**captive portal**) consiste à* ~~Et essaie d'accéder à Internet..~~ *HTTP à un réseau de consultation à afficher une page web spéciale… **blabla***

Et essaie d'accéder à Internet

Requête HTTP ?

Pourquoi et comment utiliser le tunneling

- Rappel sur les portails captifs

Le portail captif intercepte les requêtes HTTP et force alors le client à naviguer sur le serveur WEB du portail captif.

Pourquoi et comment utiliser le tunneling

- Rappel sur les portails captifs

Pas d'adresse IP ➔ Pas de requête HTTP

Pourquoi et comment utiliser le tunneling

- Rappel sur les portails captifs:

Conclusion :

Les requêtes **DNS** sont **autorisées** à sortir, même quand nous ne sommes **pas authentifié**.

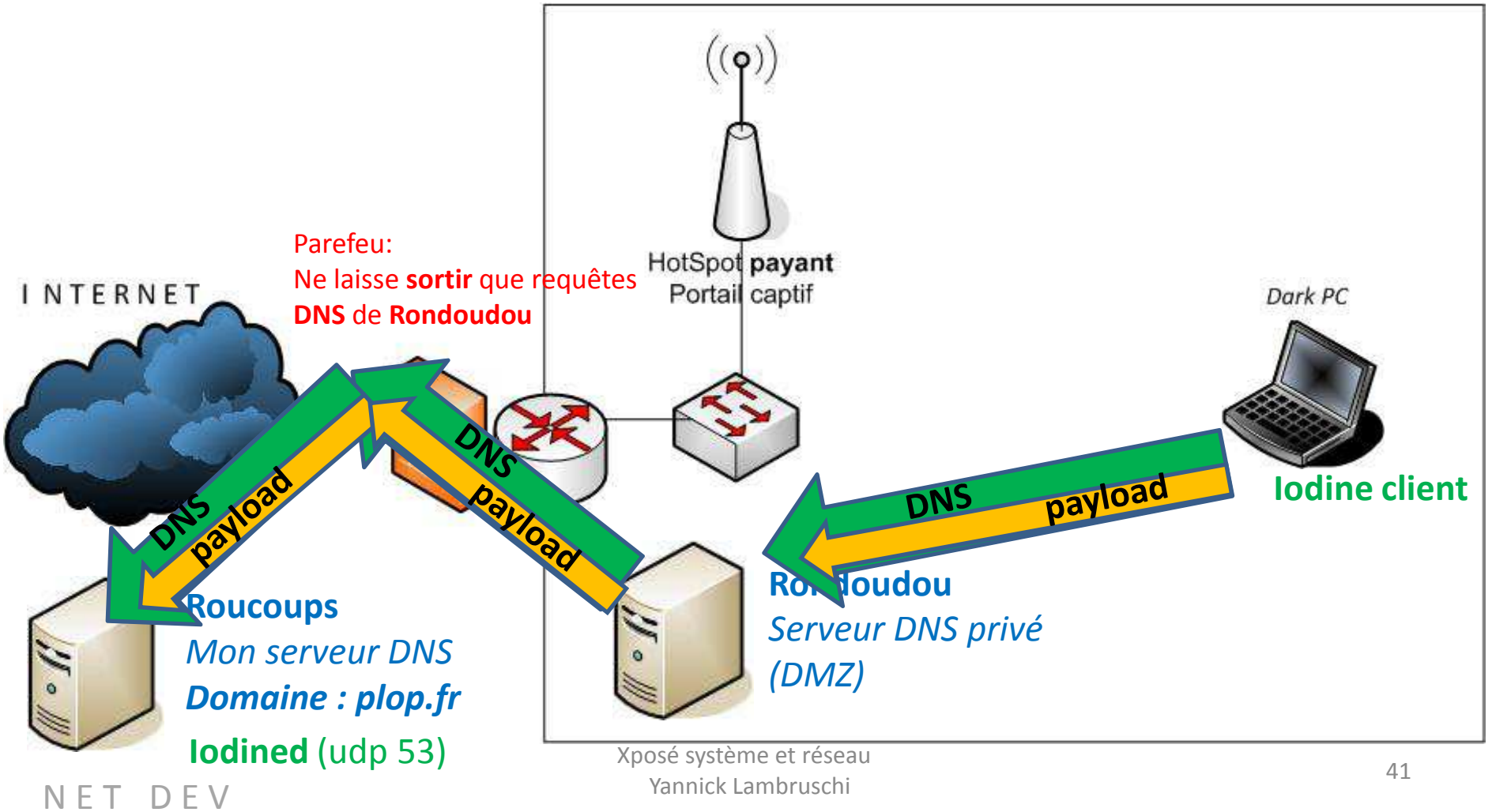
Pourquoi et comment utiliser le tunneling

- Outrepasser un **portail captif « Iodine »**:

Il suffit d'encapsuler nos données utiles dans un datagramme udp avec pour port destination 53

Pourquoi et comment utiliser le tunneling

- Outrepasser un **portail captif « Iodine »**:



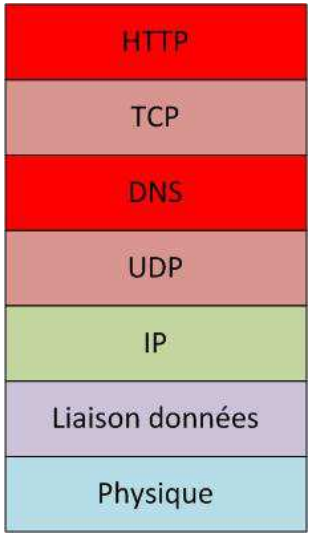
Pourquoi et comment utiliser le tunneling

- Outrepasser un **portail captif « Iodine** »:

Démonstration

Pourquoi et comment utiliser le tunneling

- Outrepasser un **portail captif « Iodine »**:
  - **Inconvénients** :
    - On détruit le modèle OSI, on en paye les frais:
      - Surcharge d'entête
      - Faible débit
      - Perte de fiabilité



    - **Analyse statistique** peut le détecter. Heureusement il y a **Heyoka** http://heyoka.sourceforge.net/

Webographie

- RFC4250->4255
- http://en.wikipedia.org/wiki/Generic_Routing_Encapsulation
- http://en.wikipedia.org/wiki/IPsec
- http://fr.wikipedia.org/wiki/Secure_Shell
- De la doc Cisco
- http://www.frameip.com

O V E R

Aluminum	Silicon	Phosphorus	Sulfur	Chlorine	Argon
2 8 3 **Al_{13}** 660.32° 2519°	2 8 4 **Si_{14}** 1414° 3265°	2 8 5 **P_{15}** 44.15° 280.5° 721°	2 8 6 **S_{16}** 115.21° 444.60° 1041°	2 8 7 **Cl_{17}** -101.5° -34.04° 143.8°	2 8 8 **Ar_{18}** -189.35° -185.85° -122.28°
+3	+2+4-4	+3+5-3	+4+6-2	+1+5+7-1	0
26.981538	28.0855	30.973761	32.066	35.4527	39.948
0.000277%	0.00326%	0.000034%	0.00168%	0.000017%	0.000329%
Gallium	**Germanium**	**Arsenic**	**Selenium**	**Bromine**	**Krypton**
2 8 18 3 **Ga_{31}** 29.76° 2204°	2 8 18 4 **Ge_{32}** 938.25° 2833°	2 8 18 5 **As_{33}** 817t° 614s° 1400°	2 8 18 6 **Se_{34}** 221° 685° 1493°	2 8 18 7 **Br_{35}** -7.2° 58.8° 315°	2 8 18 8 **Kr_{36}** -157.36° -153.22° -63.74°
+3	+2+4	+3+5-3	+4+6-2	+1+5-1	0
69.723	72.61	74.92160	78.96	79.904	83.80
1.23×10^{-7}%	3.9×10^{-7}%	2.1×10^{-8}%	2.03×10^{-7}%	3.8×10^{-8}%	1.5×10^{-7}%
Indium	**Tin**	**Antimony**	**Tellurium**	**Iodine**	**Xenon**
2 8 18 18 3 **In_{49}** 156.60° 2072°	2 8 18 18 4 **Sn_{50}** 231.93° 2602°	2 8 18 18 5 **Sb_{51}** 630.63° 1587°	2 8 18 18 6 **Te_{52}** 449.51° 988°	2 8 18 18 7 **I_{53}** 113.7° 184.4° 546°	2 8 18 18 8 **Xe_{54}** -111.75° -108.04° 16.58°
+3	+2+4	+3+5-3	+4+6-2	+1+5+7-1	0
114.818	118.710	121.760	127.60	126.90447	131.29
6.0×10^{-10}%	1.25×10^{-8}%	1.01×10^{-9}%	1.57×10^{-8}%	2.9×10^{-9}%	1.5×10^{-8}%
Thallium	**Lead**	**Bismuth**	**Polonium**	**Astatine**	**Radon**
2 8 18 32 18 3 **Tl_{81}** 304° 1473°	2 8 18 32 18 4 **Pb_{82}** 327.46° 1749°	2 8 18 32 18 5 **Bi_{83}** 271.40° 1564°	2 8 18 32 18 6 **Po_{84}** 254° 962°	2 8 18 32 18 7 **At_{85}** 302°	2 8 18 32 18 8 **Rn_{86}** -71° -61.7° 104°
+1+3	+2+4	+3+5	+2+4		0
204.3833	207.2	208.98038	[209]	[210]	[222]
6.0×10^{-10}%	1.03×10^{-8}%	4.7×10^{-10}%			

Merci de votre attention

Des questions ?