Master Informatique – parcours SIS
sebastien.duval@loria.fr

Sécurité des Systèmes d'Information

Chapitre 3 : Les aspects réseau et applicatifs

2025/2026

Supports basés sur le cours de Laurent Vigneron

Support de cours basé sur un document pédagogique mis à disposition par l'ANSSI sous licence Creative Commons Attribution 3.0 France.

Plan du chapitre

1. Sécurité du protocole IP
2. Sécurité du protocole DNS
3. Sécurité d’un réseau
4. Sécurité de l’IoT
5. Sécurité des applications web
6. Le sur-protocole TLS
7. Active Directory, centralisation et virtualisation

1. Sécurité du protocole IP

a) Préambule
b) Exemple d’attaque par réflexion
c) Exemples d’écoute de trafic
d) Exemple de modification du routage des datagrammes IP
e) Sécurisation du protocole IP

1. Sécurité du protocole IP

D'après vous ?

Le protocole IP n'est pas sécurisé a priori.

Quelles actions malveillantes peuvent en être les conséquences ?

1. Sécurité du protocole IP

a. Préambule

Lorsqu'ils ont été conçus, le protocole IP et les protocoles associés (TCP, UDP, ICMP, routage…) n'ont pas pris en compte la sécurité.

- « Concept sécurité » ne leur est pas venu à l'idée (c'étaient des bidouilleurs, Internet n'a jamais été réfléchi).
- **Aucun mécanisme de sécurité n'est donc implanté au sein de ces protocoles.**

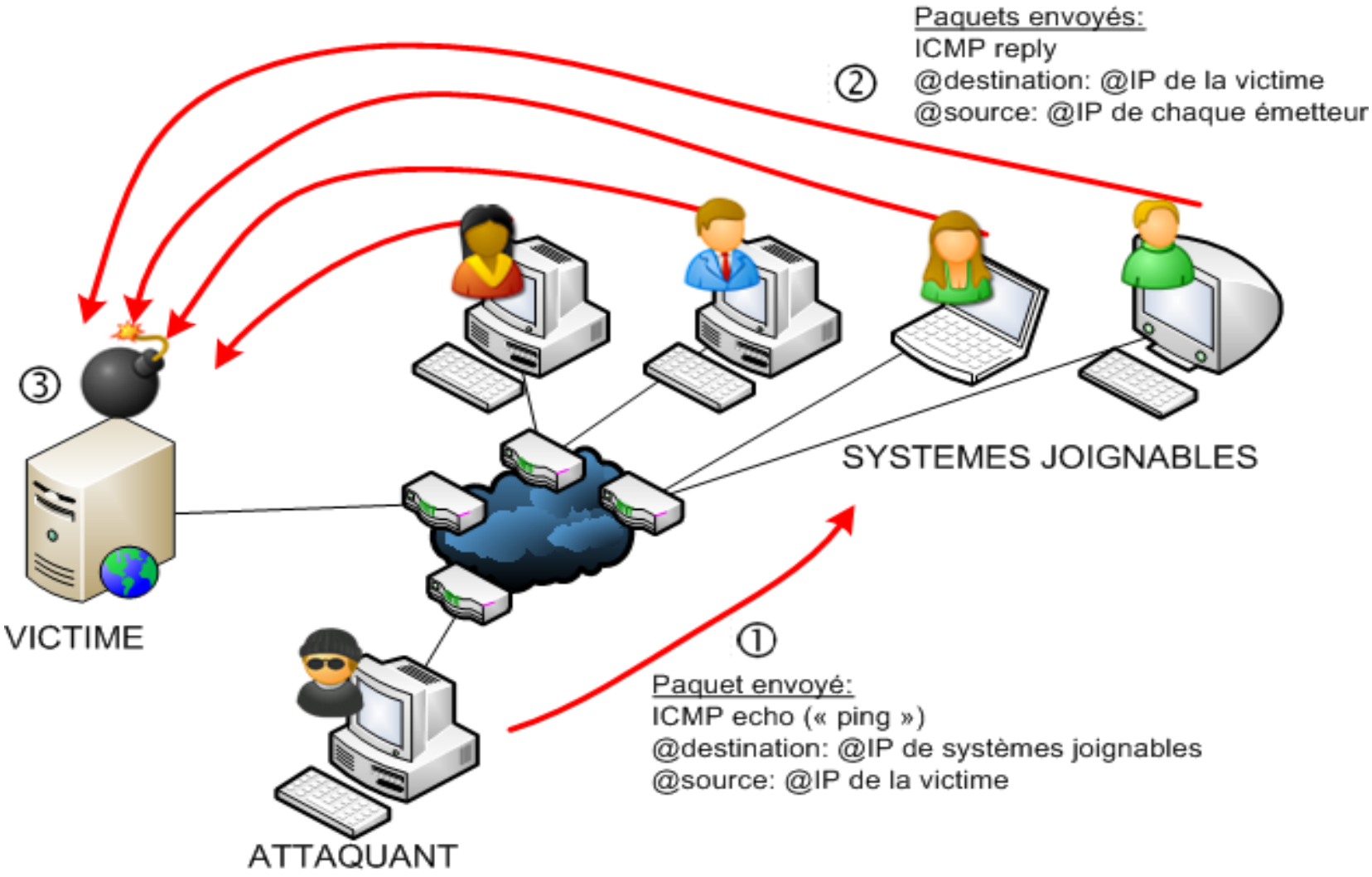
Quelques exemples de faiblesses de ces protocoles

- **Absence d'authentification des émetteurs et récepteurs** d'un datagramme : usurpation d'adresse IP possible.
- **Absence de chiffrement des données**, celles-ci sont donc transmises en clair. Un hacker positionné sur un réseau peut donc écouter les connexions et accéder aux données.
- **Le routage des datagrammes peut être modifié** de façon à rediriger les datagrammes vers un autre destinataire.
- Note : l'exploitation de ces faiblesses nécessite des prérequis techniques, i.e. elles ne sont pas systématiquement applicables à tous les réseaux.

Les diapositives suivantes illustrent ces faiblesses.

1. Sécurité du protocole IP

b. Exemple d'attaque par réflexion



1. Sécurité du protocole IP

b. Exemple d'attaque par réflexion

But de l'attaque

- porter atteinte aux performances d'un système cible (déni de service).

Quelles sont les caractéristiques de l'attaque ?

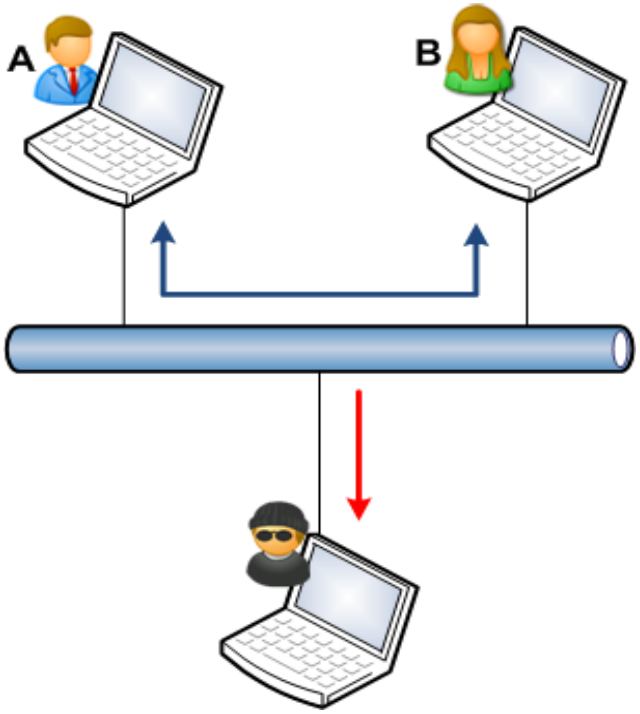
- usurpation d'adresse IP ;
- réflexion de trafic en ayant recours à des systèmes tiers « innocents ».

Séquences de l'attaque

1) Un attaquant envoie des paquets PING à des systèmes tiers joignables en indiquant l'@IP de la future victime comme @IP source.
2) Chaque système pense ainsi recevoir un PING de la part d'un système distant, et chacun va répondre à ce PING.
3) Avec suffisamment de ressources, l'attaquant sera en mesure de faire générer suffisamment de trafic pour affecter les performances de la victime.

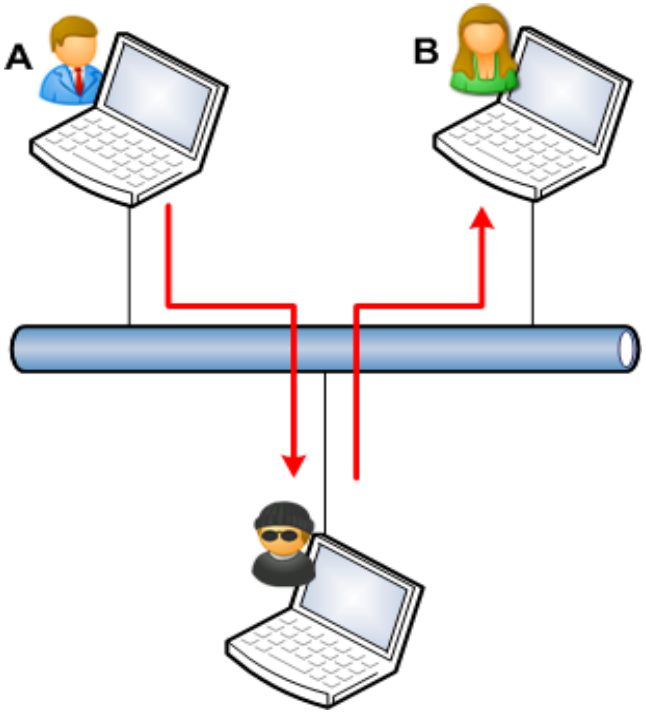
1. Sécurité du protocole IP

c. Exemples d'écoute de trafic



Écoute passive

L'attaquant est en mesure d'écouter les conversations entre A et B (atteinte à la ***confidentialité*** des échanges).

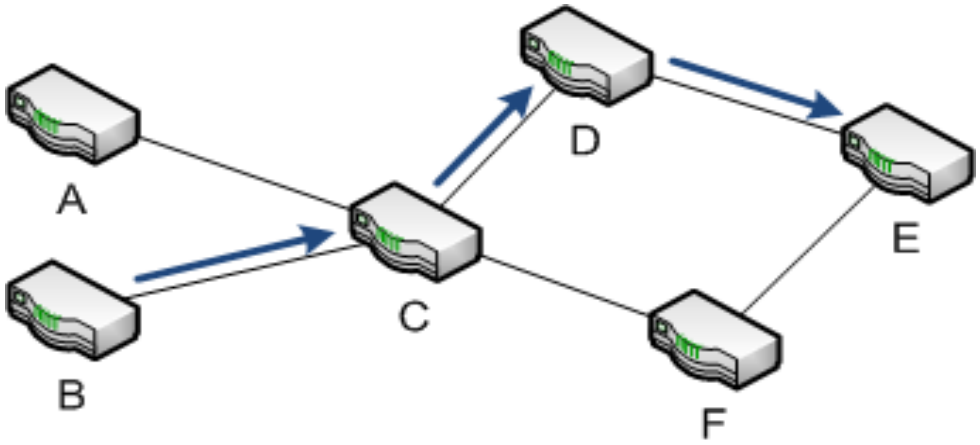


Écoute active

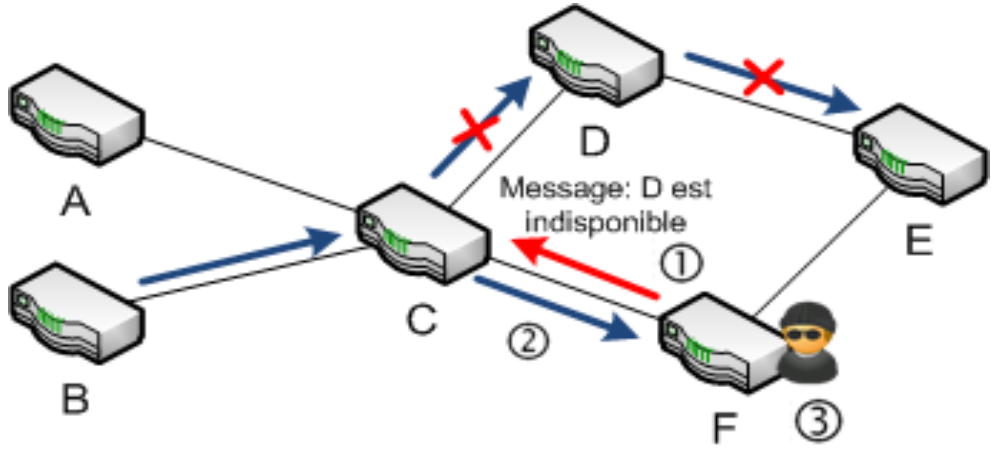
L'attaquant est en mesure de s'insérer dans la conversation entre A et B sans que ceux-ci le sachent (atteinte à la ***confidentialité*** et à l'***intégrité*** des échanges).

1. Sécurité du protocole IP

d. Exemple de modification du routage des datagrammes IP



Chaque routeur possède une table de routage qui indique vers quel routeur voisin transmettre les datagrammes. Cette table peut être mise à jour dynamiquement en fonction des évènements réseaux (protocoles BGP, RIP, OSPF, etc.).



But de l'attaque : **dérouter les paquets** à destination du réseau E, vers le réseau F maîtrisé par l'attaquant.

Méthode :

1) L'attaquant utilise une faiblesse du protocole de routage pour indiquer au routeur C que le routeur D est indisponible, et que le routeur F peut router les paquets vers E
2) Le routeur C transfère donc à F les paquets pour E, afin qu'ils puissent être routés à destination.
3) Selon le but visé par l'attaquant, celui-ci peut décider de router ou non les paquets vers E.

1. Sécurité du protocole IP

e. Sécurisation du protocole IP

Ainsi, il est nécessaire de **mettre en œuvre des mécanismes de sécurité complémentaires** afin de réduire et maîtriser les risques émanant des protocoles historiques régissant les réseaux.

Exemple de mécanismes :

- Chiffrement des communications
- Authentification des entités
- Cloisonnement réseau
- Filtrage
- Dimensionnement adapté des infrastructures
- Règles de renforcement des configurations des équipements
- Supervision des équipements
- etc.

2. Sécurité du protocole DNS

a) Contexte
b) Méthodes de résolution
c) Risques
d) Solutions

2. Sécurité du protocole DNS

D'après vous ?

C'est quoi déjà le protocole DNS ?

Quelle conséquence si ce protocole est vulnérable ?

2. Sécurité du protocole DNS

a. Contexte

Intérêt vital du trafic web

- Sécurité (origine de nombreux incidents)
- Performance (le ressenti utilisateur vis-à-vis des applications métiers peut avoir un impact sur l'activité et la productivité)

Domain Name System (DNS)

- Permet au trafic web de fonctionner
- Traduction des noms de domaines Internet en adresses IP

Problème ?

- L'un des protocoles les moins supervisés
- Correction du résultat prioritaire à qui s'en charge, et en combien de temps
- *Q : qui assure pour vous la résolution des noms de domaines ?*

2. Sécurité du protocole DNS

b. Méthodes de résolution

Utilisation d'un DNS central (vs. plusieurs DNS locaux)

- Souvent le cas pour les entreprises (même internationales)
- Facilite la résolution des noms internes, sans gérer une multitude de serveurs locaux
- Mais ne tient pas compte de la localisation du demandeur, conséquences :
  - Dégradation possible des performances (téléphonie IP, visioconférence)

→ besoin de réécrire la résolution de nom en fonction de la localisation géographique réelle de l'utilisateur

(plus simple à faire dans le Cloud qu'avec une multitude de serveurs locaux)

2. Sécurité du protocole DNS

c. Risques

Un protocole central

- Essentiel pour diriger les utilisateurs vers les sites Web demandés
- Si prise de contrôle par un attaquant
  - Attaques parfaites de type phishing et à grande échelle
  - Usurpation des destinations externes (Web) et des applications internes
- Possibilité d'exfiltration de données (champs TXT) par des malwares
  - 91 % des malwares s'en servent
  - 38 % des entreprises ne priorisent pas la sécurité de leur DNS

    [Efficient IP, 2018]
- Aussi utilisé par certains anti-virus pour mettre à jour leur base de signatures (DNS tunneling)

2. Sécurité du protocole DNS

d. Solutions

Déployer des technologies comme

- DNSSEC, signature des enregistrements DNS pour certifier les données servies
- DNS-over-TLS, chiffrement du trafic DNS

Mais ajoute de la complexité dans un système simple « roulant tout seul » depuis des années...

Inspecter les flux des DNS

- Tâche complexe (utiliser du machine learning sur de gros volumes de requêtes ?)
- Possibilité d'utiliser les atouts du Cloud (centralisation, capacité d'analyse infinie, abstraction de l'infrastructure locale)

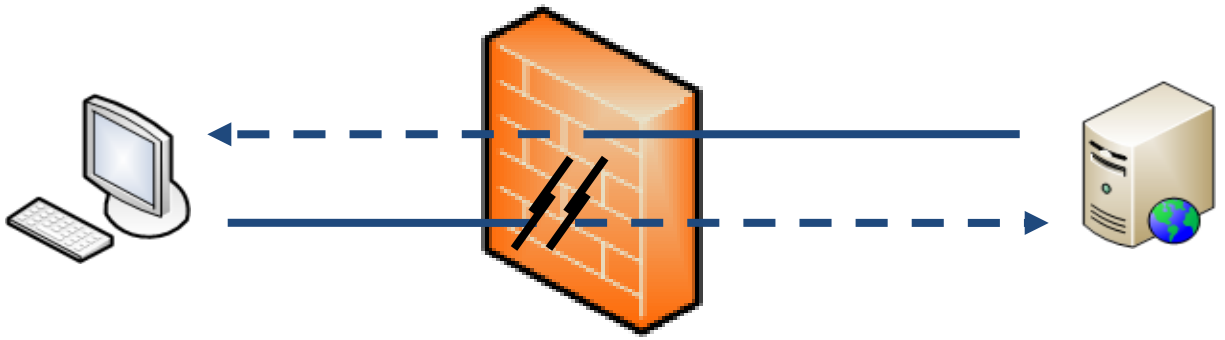
3. Sécurité d'un réseau

a) Pare-feu
b) Répartiteur de charge
c) Anti-virus
d) IDS et IPS
e) VPN
f) Segmentation
g) Exemple pratique de sécurisation avec un réseau simple
h) Réseau et virtualisation

3. Sécurité d’un réseau

a. Pare-feu

- **Équipement en coupure entre 2 ou plusieurs réseaux.**
- Inspecte les paquets réseaux entrants et sortants d’un réseau à l’autre.
- Implante un **mécanisme de filtrage basé sur des règles :** il ne transmet donc que les paquets réseaux qui respectent les règles de filtrage implantées dans la configuration du pare-feu.



Pour chaque flux entrant ou sortant, le pare-feu interroge ses règles de filtrage pour déterminer s’il doit laisser passer le paquet réseau ou non.

3. Sécurité d'un réseau

a. Pare-feu

Règles de filtrage :

- Historiquement, elles étaient basées sur les couches basses de la pile protocolaire (réseau, transport), et portaient uniquement sur les paramètres comme les adresses IP et les ports TCP/UDP.
- Les pare-feu sont également capables de filtrer selon les données de la **couche applicative** (protocole et contenu des données). Ex. : HTTP, SMTP, DNS, etc.
  - Les **proxy et reverse-proxy peuvent être vus comme des pare-feu applicatifs dédiés**. Ils permettent **d'analyser finement** les flux applicatifs (par exemple la navigation web des utilisateurs ou les flux web entrants sur un server de e-commerce).
- Un anti-virus ou un mécanisme de détection d'intrusion peut également être implanté sur le pare-feu de façon à détecter un malware en transit ou certaines attaques.

Avantage sécurité :

- L'exploitant d'un réseau peut donc restreindre le trafic entrant et sortant aux seules connexions qu'il estime légitimes. Toutes les autres connexions sont donc bloquées.

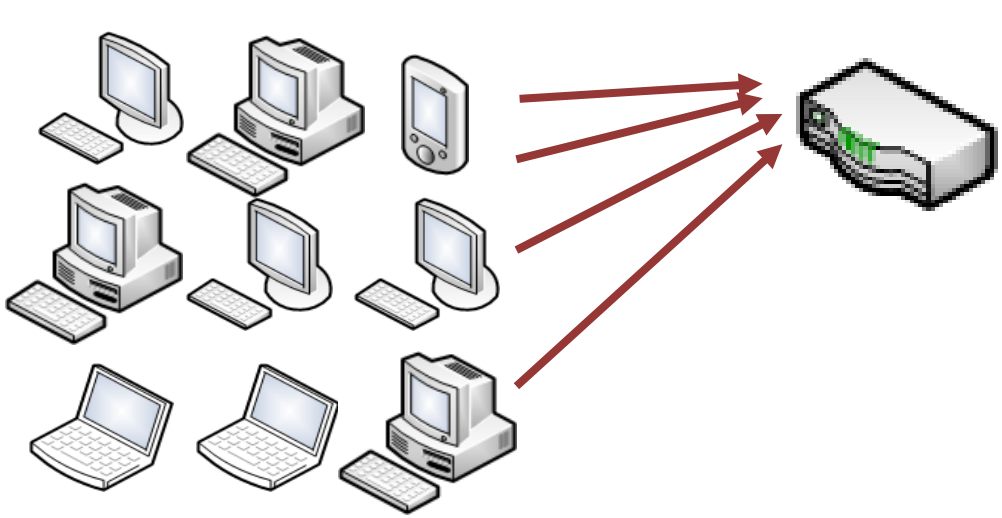
3. Sécurité d’un réseau

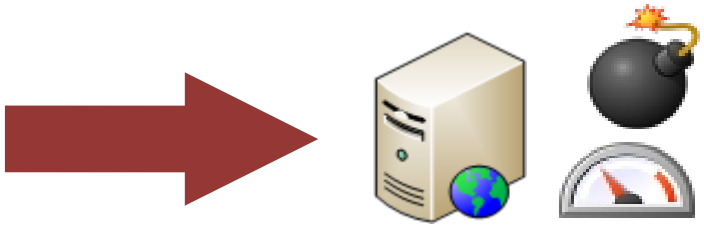
b. Répartiteur de charge

- « Load-balancer » en anglais.
- Équipement rencontré sur les grosses infrastructures où les serveurs doivent faire face à de très fortes bandes passantes et charges élevées de trafic.
- Équipement chargé de **répartir/distribuer la charge réseau** en fonction des caractéristiques de celui-ci et de la disponibilité des serveurs.
- Avantage sécurité : permet de mieux se protéger contre les **dénis de service distribués.**

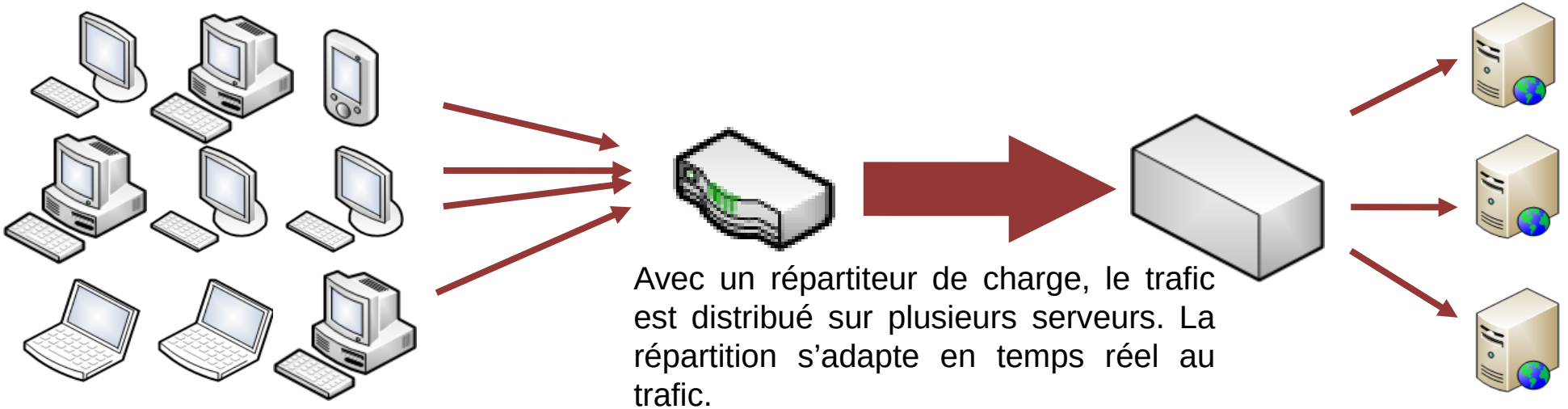
3. Sécurité d’un réseau

b. Répartiteur de charge





Sans répartiteur de charge, ce seul serveur web pourrait ne plus pouvoir faire face aux nombreuses demandes, et devenir indisponible.



Avec un répartiteur de charge, le trafic est distribué sur plusieurs serveurs. La répartition s’adapte en temps réel au trafic.

3. Sécurité d’un réseau

c. Anti-virus

Logiciel chargé de détecter et stopper les **malware connus :**

- Virus, vers, *keylogger* (enregistreur de frappe), chevaux de Troie, etc.

- Ces logiciels fonctionnent en général avec une base de données qui contient les signatures des malware connus. Ils analysent en permanence les fichiers et les exécutables du système hébergeant l’anti-virus.

- **Limite des anti-virus :** ils ne détectent (en général) que les malware déjà répertoriés par les éditeurs. Ainsi, les nouveaux virus ou les malware ciblés ne sont souvent pas détectés. D’autre part, il est impératif que l’anti-virus soit mis à jour quotidiennement.

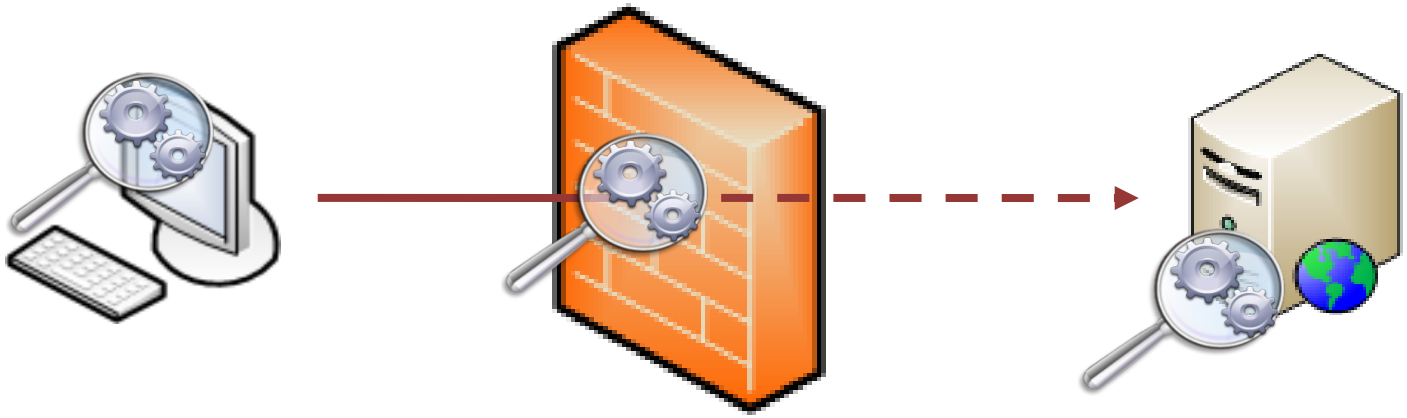
3. Sécurité d’un réseau

c. Anti-virus

Un anti-virus peut être déployé :

- En **local :** sur un système (poste de travail ou serveur) afin de détecter les virus affectant cette machine.

- En **coupure des flux réseaux :** sur un pare-feu afin d’analyser les flux réseau et détecter les malware avant même qu’ils n’atteignent leur cible. Ce fonctionnement peut être assimilé à un IDS (Intrusion Detection System), mécanisme présenté dans la section suivante.



3. Sécurité d’un réseau

d. IDS et IPS

IDS **I**ntrusion **D**etection **S**ystem

IPS **I**ntrusion **P**revention **S**ystem

Chargés d’analyser le trafic réseau pour y **détecter des tentatives d’intrusion** :

- soit en analysant le comportement des flux réseaux ;
- soit en se basant sur une base de signatures identifiant des données malveillantes (principe similaire à celui des anti-virus).

En cas de détection d’une intrusion :

- Les **IDS alertent** les administrateurs, libre à eux d’intervenir ou non.
- Les **IPS bloquent** les flux réseau concernés.

Les IDS/IPS demandent un configuration fine et maintenue :

- Ils sont en effet connus pour présenter de nombreux faux-positifs (i.e. ils détectent à tort une tentative d’intrusion).
- De plus, les IDS/IPS basés sur des signatures ne peuvent détecter que les intrusions dont les caractéristiques techniques sont déjà connues et référencées.

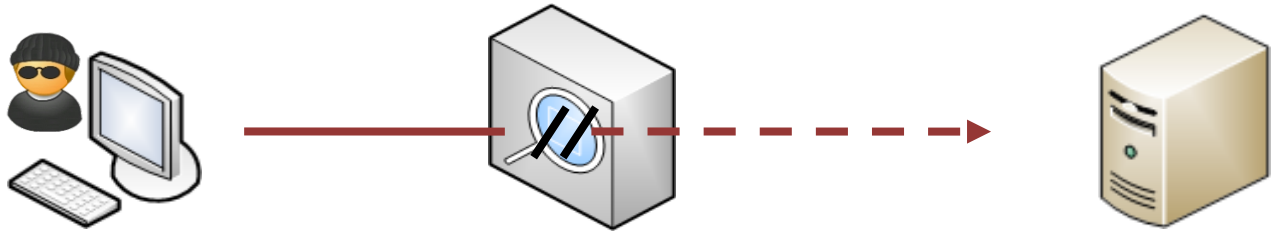
3. Sécurité d'un réseau

d. IDS et IPS

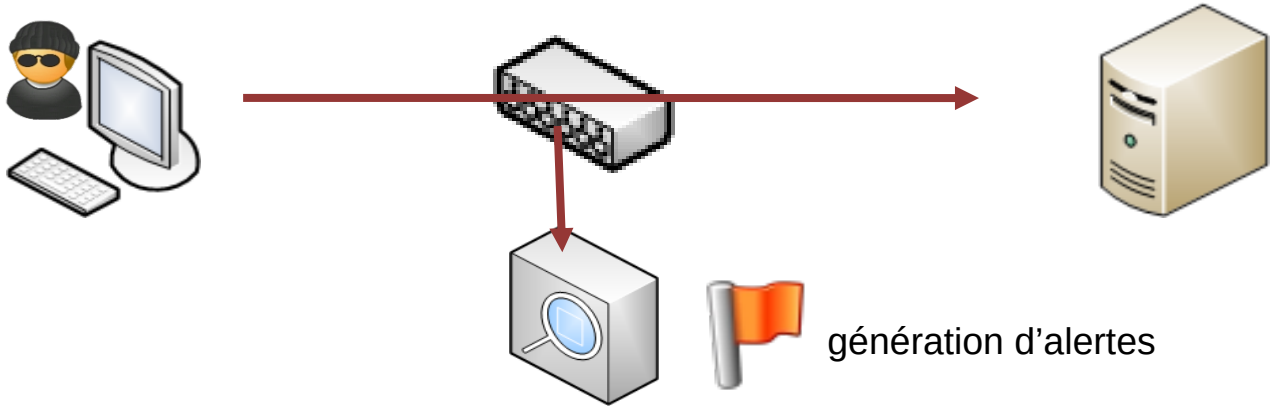
Un IDS peut être soit en coupure du flux réseaux, soit **positionné en écoute.**

Un IPS **doit forcément** être en **coupure du flux** de façon à pouvoir bloquer le trafic lorsque cela est nécessaire.

IPS, en coupure



IDS, en écoute



3. Sécurité d’un réseau

d. IDS, IPS et sondes réseau

Les IDS et IPS sont l’emplacement idéal pour placer des *sondes réseau*, qui créent des logs des accès réseau (origine, destination, heure, droits, …)

Ces logs doivent ensuite être agglomérés dans une grosse base de données, généralement centralisée.

Exemples de sondes réseau : Snort, Suricata, Zeek

3. Sécurité d’un réseau

e. VPN

Virtual **P**rivate **N**etwork

Un VPN est un **réseau virtuel** qui permet à **deux réseaux distants de communiquer ~~en toute sécurité~~ sans s’exposer à tout Internet**, y compris si la communication s’effectue via des réseaux inconnus et auxquels nous ne faisons pas confiance.

Exemple avec une entreprise qui possède deux sites distants qui ont besoin de communiquer entre eux via internet : comment faire passer les flux en toute sécurité via Internet que l’on ne maîtrise pas ?

3. Sécurité d’un réseau

e. VPN

Solution : grâce à des mécanismes cryptographiques, appliquer un **chiffrement des données, ainsi qu’un motif d’intégrité, à tous les flux entre les 2 sites**. On obtient ainsi un **tunnel virtuel** qui ne contient que des données chiffrées et protégées en intégrité :

- Les données qui passent sur Internet sont donc chiffrées et non-compréhensibles par un attaquant qui écouterait les flux.
- En cas de modification malveillante des flux, le mécanisme d’intégrité permettra au destinataire de déterminer que les données reçues ne sont pas intègres, et qu’il ne faut donc pas traiter ces données.

Il existe différents types de VPN, représentés sur les diapositives suivantes.

3. Sécurité d'un réseau

e. VPN

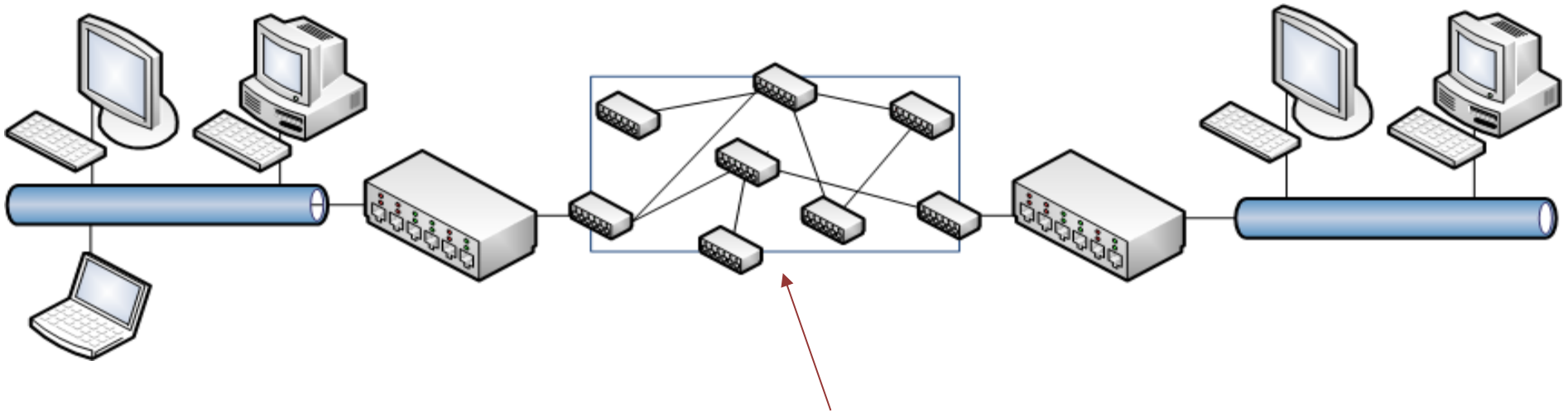
VPN de réseau à réseau, dont le tunnel est géré par les routeurs
Souvent **IPsec** – au niveau de la couche Internet

VPN entre machines
Souvent **TLS** – au niveau de la couche Transport

3. Sécurité d’un réseau

e. VPN

Il existe également des VPN qui n’ont pas recours à de la cryptographie, mais qui font appel aux infrastructures d’opérateurs. Dans ce cas, la protection du réseau est assurée par l’opérateur.



Réseau opérateur **MPLS**, dont le cœur est inaccessible aux clients se connectant sur ce réseau

3. Sécurité d’un réseau

e. VPN

Utilisations :

- **Remote connection**/desktop (connexion à distance à une machine)
- **Forward du trafic** internet à travers une machine distante
  - Bastion/proxy sur le réseau pour filtrer les flux
  - Machine d’un autre pays (ou autre IP de façon générale) pour ne pas être traqué

- Résultat : c’est *comme si les 2 machines connectées étaient sur le même réseau interne*
  - Il faut de l’**authentification**
  - Il faut que les 2 machines se fassent **confiance**

3. Sécurité d’un réseau

f. Segmentation

Un principe majeur de la Sécurité est celui du **moindre privilège :**

On ne doit donner les droits d’accès à une ressource qu’aux seules personnes/entités ayant un besoin légitime d’y accéder.

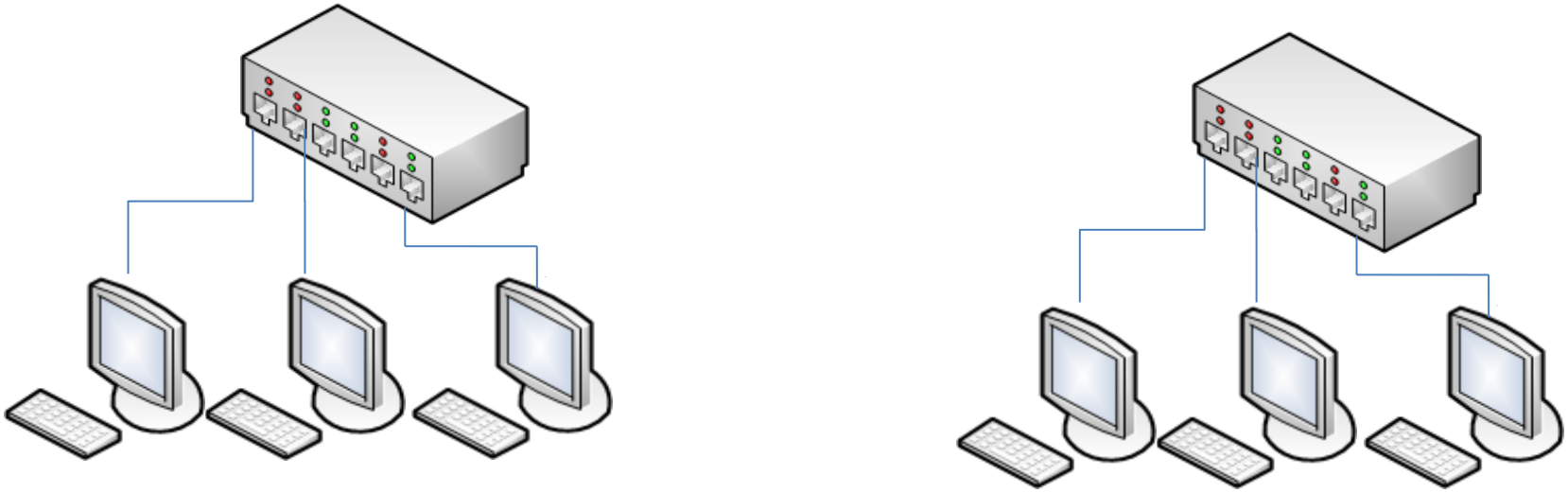
Appliqué au domaine réseau, il est donc fait **recours à de la segmentation** afin de séparer le réseau en différentes zones.

Les droits d’accès à ces zones doivent ensuite être **filtrés** afin de n’autoriser que les flux nécessaires entre chaque zone.

3. Sécurité d’un réseau

f. Segmentation

Il existe plusieurs techniques pour procéder à de la segmentation. La technique la plus évidente : implémenter deux réseaux distincts non connectés.



Implémentation de deux réseaux physiques différents, non connectés.

Avantage : **étanchéité réseau parfaite** (aucune communication possible entre ces deux zones).
Inconvénient : adapté à certains réseaux très sensibles seulement, **peu adapté aux réseaux d’entreprise** qui ont besoin de communiquer.

3. Sécurité d'un réseau

f. Segmentation

Autre technique de segmentation : **VLAN** (Virtual LAN).

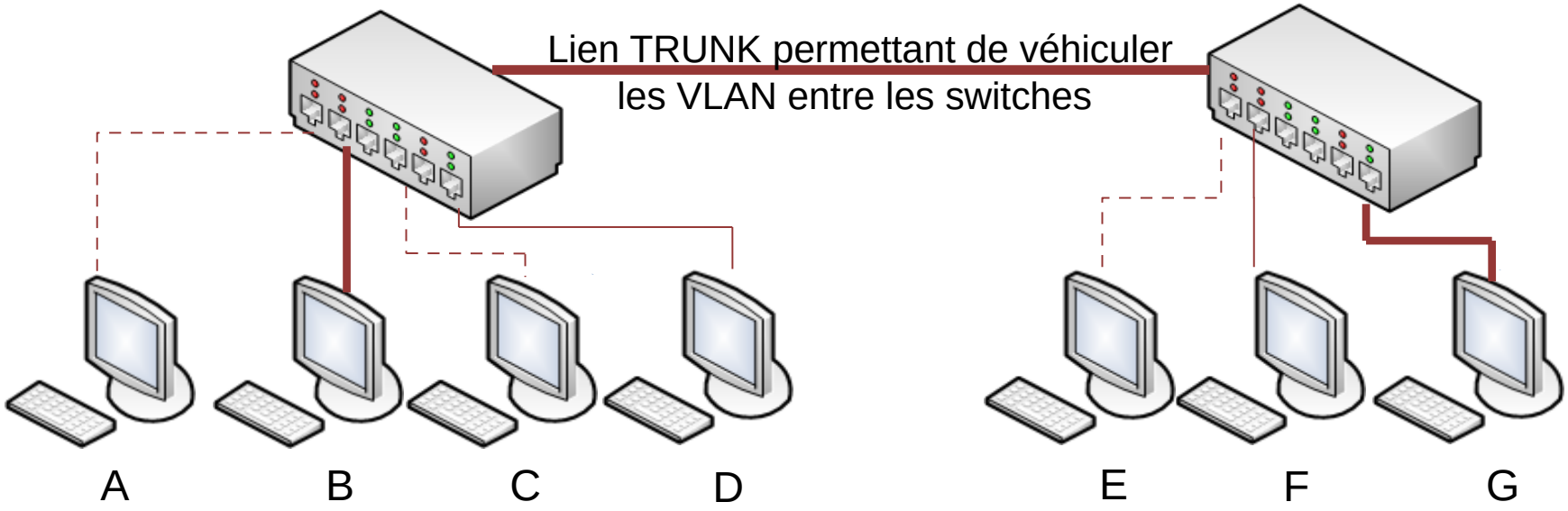
Les VLAN sont des **réseaux virtuels implémentés par les switches**. Ceux-ci **restreignent la communication entre systèmes selon des règles configurées** sur l'équipement réseau :

- La segmentation peut se faire grâce aux ports Ethernet de chaque switch (on affecte un VLAN particulier à chaque port des switches, les deux switches étant reliés entre eux par un lien TRUNK afin de véhiculer les étiquettes des VLAN).
- La segmentation peut aussi se faire grâce aux adresses MAC des systèmes.
  - Attention : les adresses MAC des cartes réseaux pouvant facilement être modifiées par les utilisateurs, le filtrage sur les adresses MAC est à considérer – logiquement – avec précaution car le niveau de sécurité effectif est limité.

Voir exemple sur la diapositive suivante.

3. Sécurité d'un réseau

f. Segmentation



Implémentation de 3 VLAN sur des réseaux distants.

VLAN 1. Les machines B et G sont segmentées des autres systèmes et peuvent communiquer entre elles deux seulement.

VLAN 2. Les machines A, C et E sont segmentées des autres systèmes et peuvent communiquer entre elles seulement.

VLAN 3. Les machines D et F sont segmentées des autres systèmes et peuvent communiquer entre elles deux seulement.

3. Sécurité d’un réseau

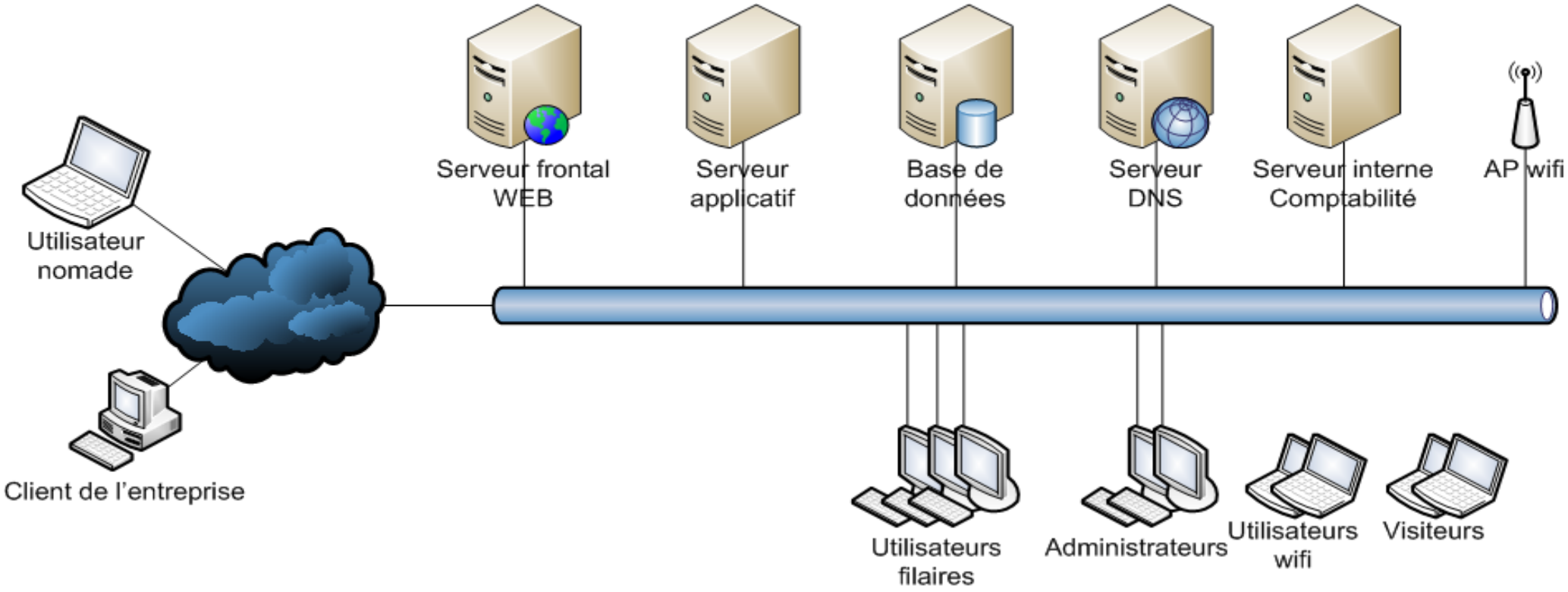
g. Exemple pratique de sécurisation avec un réseau simple

Prenons l’exemple d’un réseau d’entreprise « à plat ». Caractéristiques de cette entreprise :

- Elle fournit un **site WEB de e-commerce**.
- Certains employés se connectent sur le **réseau local filaire**, d’autres se connectent en **wifi**.
- Certains employés sont **nomades** et doivent donc se **connecter à distance**.
- Il existe deux catégories principales d’utilisateurs : les **utilisateurs « standard »** et les **administrateurs** du S.I.
- Afin de fonctionner, l’entreprise possède également des **serveurs internes** (comptabilité, wiki, etc.).
- L’entreprise souhaite permettre à ses **visiteurs** de se connecter en **wifi** afin de naviguer sur internet.

3. Sécurité d’un réseau

g. Exemple pratique de sécurisation avec un réseau simple



Réseau « à plat », avant sécurisation

3. Sécurité d’un réseau

g. Exemple pratique de sécurisation avec un réseau simple

Voyons comment nous allons pouvoir sécuriser ce réseau.

- Note : il existe plusieurs façons d’améliorer la sécurité de ce réseau, nous en présentons ici uniquement les grandes lignes. Cet exercice n’est ni exhaustif ni la seule solution possible.

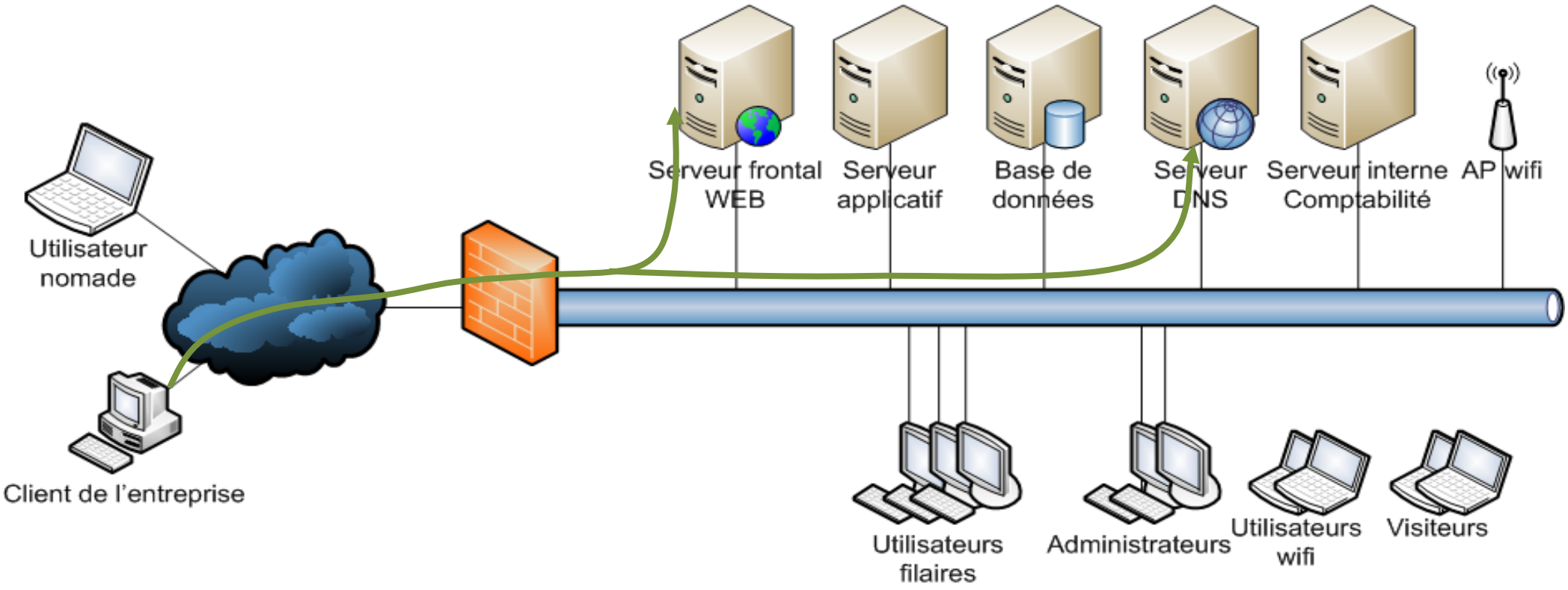
Parmi les nombreuses faiblesses architecturales de ce réseau, nous pouvons identifier au moins le problème suivant :

- Le réseau est **directement connecté à Internet**, i.e. tous les systèmes et utilisateurs peuvent communiquer avec l’extérieur (attention aux **fuites de données !**) et **tout Internet peut se connecter sur notre réseau interne.**

Corrigeons cela en implémentant un **pare-feu** en frontal qui va autoriser uniquement les flux entrants vers le serveur WEB (TCP/80 et TCP/443) et le serveur DNS (UDP/53 et TCP/53). Ainsi, Internet ne pourra plus accéder au reste du réseau interne.

3. Sécurité d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple



Réseau « à plat », avec un pare-feu en frontal

3. Sécurité d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple

Le pare-feu empêche – certes – la connexion directe entre internet et le réseau interne, mais :

- Au cas où le serveur WEB présente une **vulnérabilité**, un hacker présent sur Internet peut potentiellement **prendre la main sur ce serveur**, puis **rebondir ensuite sur le réseau interne.**

Nous allons donc **segmenter** notre réseau en **différentes zones de criticité**, notamment :

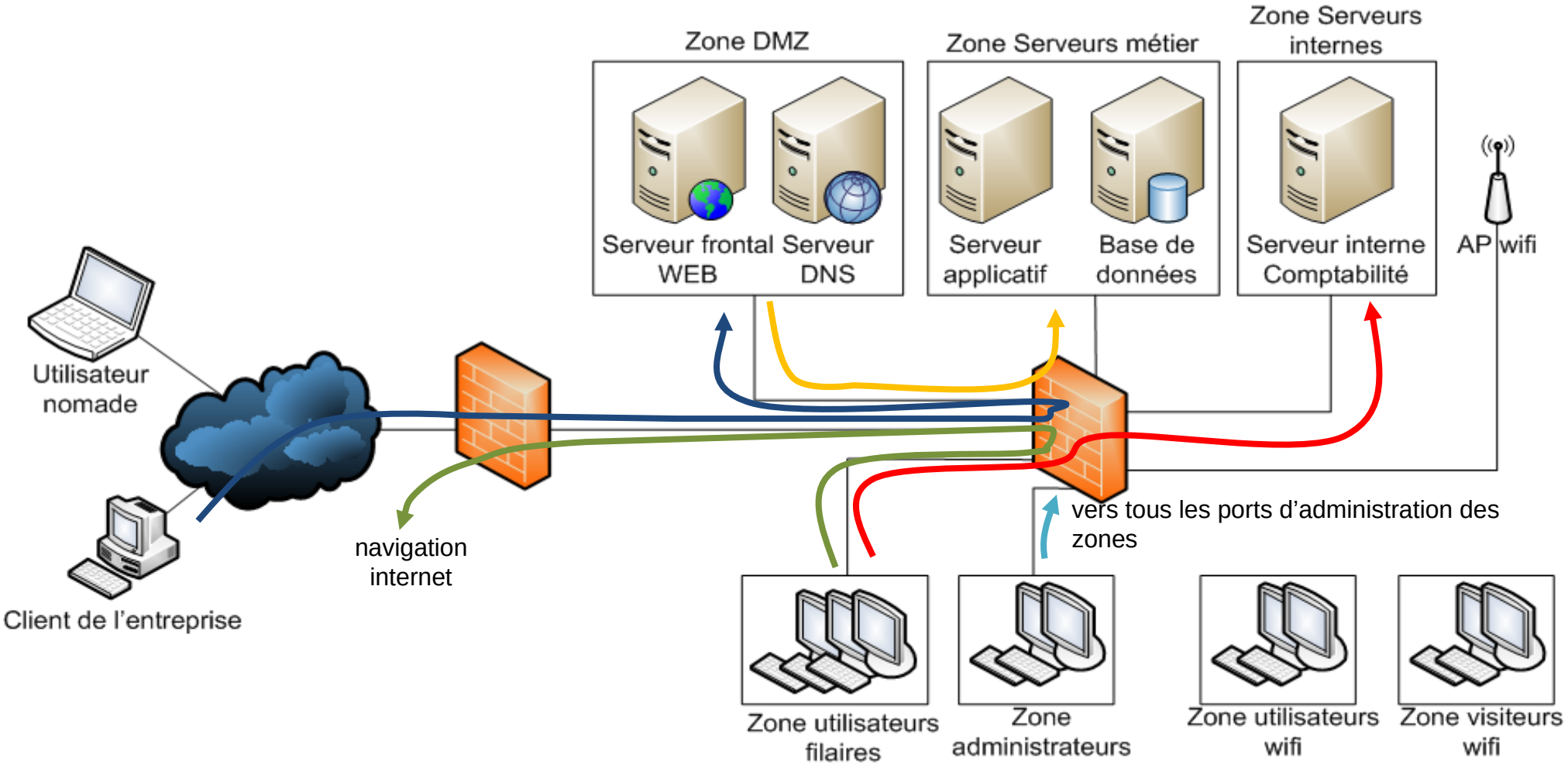
- Une **DMZ (zone démilitarisée)** destinée à héberger tous les serveurs qui doivent être accessibles depuis internet, et uniquement ceux-ci. Ainsi, en cas de faille dans le serveur web, un attaquant aurait plus de difficultés pour rebondir sur le réseau interne.
- Une zone destinée aux **serveurs internes** de l'entreprise.
- Une zone pour les **postes de travail filaires des utilisateurs.**
- Une zone pour les **postes de travail wifi des utilisateurs.**
- Une zone pour les **postes wifi des visiteurs.**
- Une zone pour les **postes de travail des administrateurs**, car ceux-ci ont besoin d'accéder à des interfaces d'administration (RDP, SSH…).

Afin que cette segmentation réseau soit efficace, nous faisons **passer tous les flux** (y compris internes) **par un deuxième pare-feu (interne)** afin que seuls les flux que nous allons configurer soient autorisés.

- Note : on observe malheureusement souvent des réseaux segmentés mais non filtrés. Cela ne sert à rien en terme de sécurité, car toutes les zones peuvent communiquer entre elles.

3. Sécurité d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple

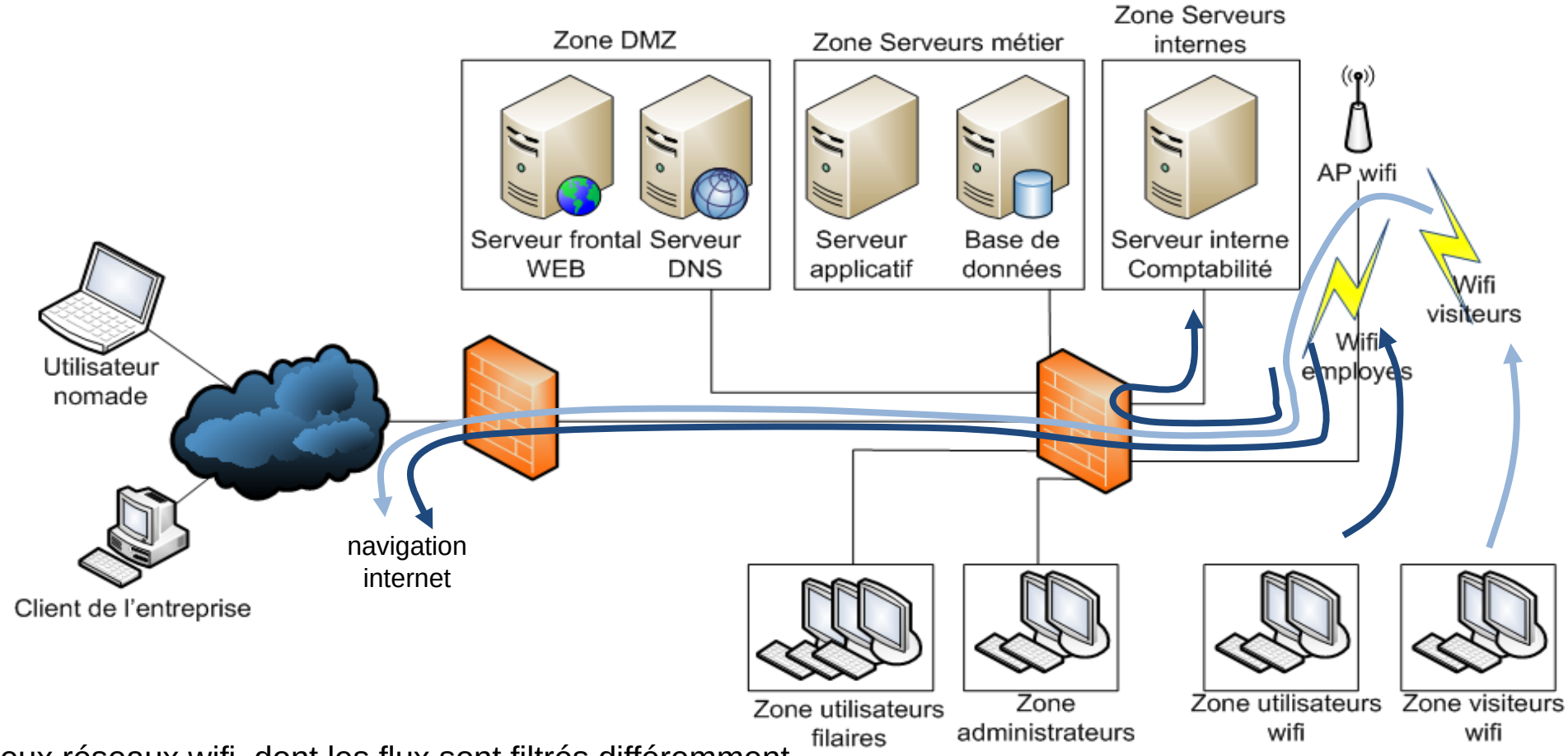


Réseau avec des zones segmentées, et un filtrage systématique via le pare-feu, y compris pour les flux internes.

3. Sécurité d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple

Le point d'accès wifi doit être accessible aux visiteurs et aux employés internes. Puisque le besoin d'accès aux ressources est différent pour ces 2 populations, nous allons donc implanter deux SSID (**deux réseaux wifi distincts**, portés par le même point d'accès, et dont le pare-feu filtrera les flux).



Deux réseaux wifi, dont les flux sont filtrés différemment.

3. Sécurité d’un réseau

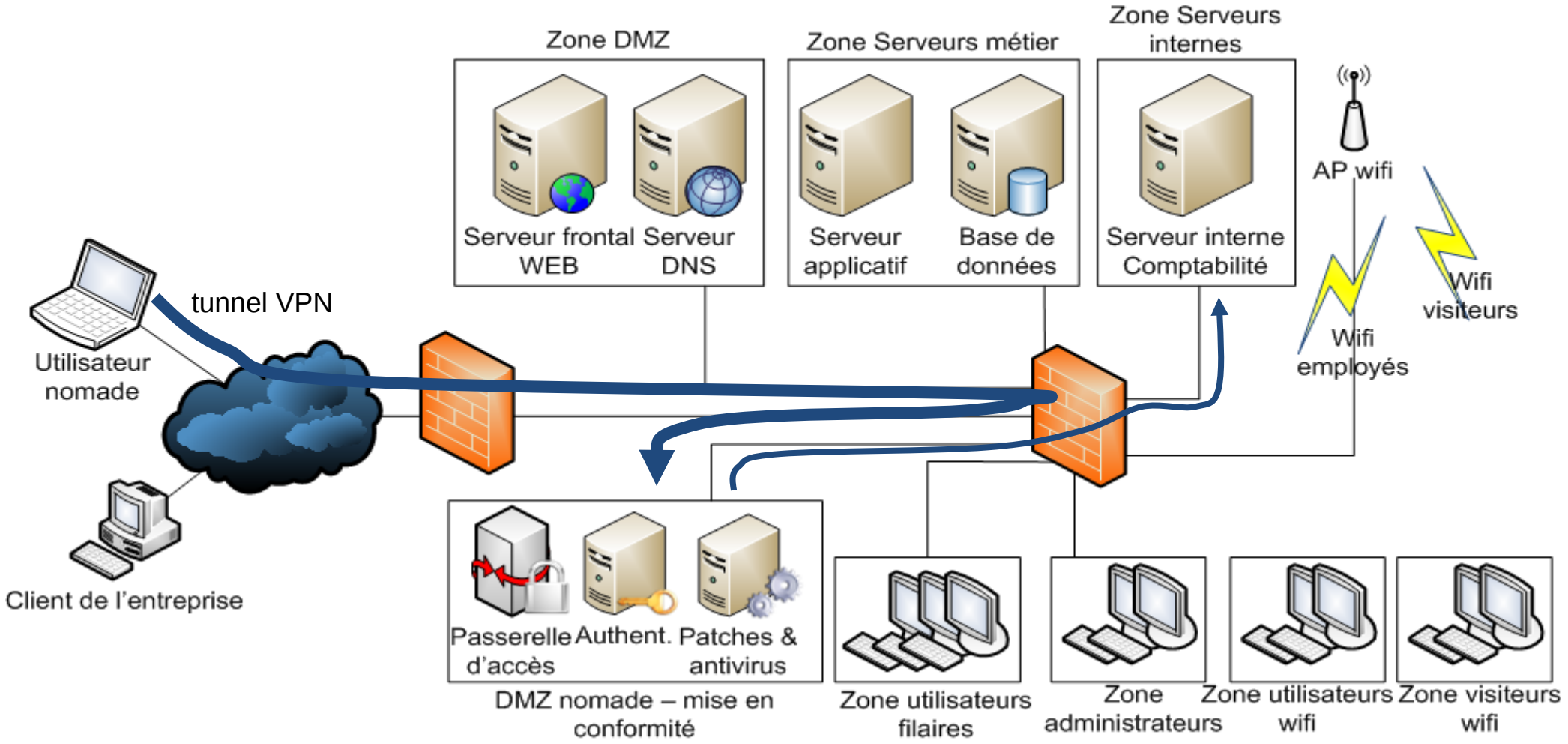
g. Exemple pratique de sécurisation avec un réseau simple

Nous devons également permettre aux **utilisateurs nomades de se connecter** au réseau interne depuis internet. Cela se fait via une DMZ spécifique, appelée zone de mise en conformité, dont le rôle est le suivant :

- Fournir l’interface d’accès au réseau interne depuis internet, en général via un **tunnel VPN**.
- **Vérifier que le poste nomade et son utilisateur sont habilités** pour se connecter à distance.
- **Vérifier le niveau de sécurité du poste** avant d’autoriser la connexion (**patches et anti-virus à jour** notamment).
- Si tout est OK, alors **autoriser les flux vers les zones internes** (et seulement celles qui sont nécessaires pour le métier), toujours en passant par le **pare-feu**.

3. Sécurité d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple



Réseau avec DMZ de mise en conformité pour les postes nomades.

3. Sécurité d’un réseau

g. Exemple pratique de sécurisation avec un réseau simple

Enfin, il serait souhaitable de **mieux filtrer le trafic WEB** entrant et sortant :

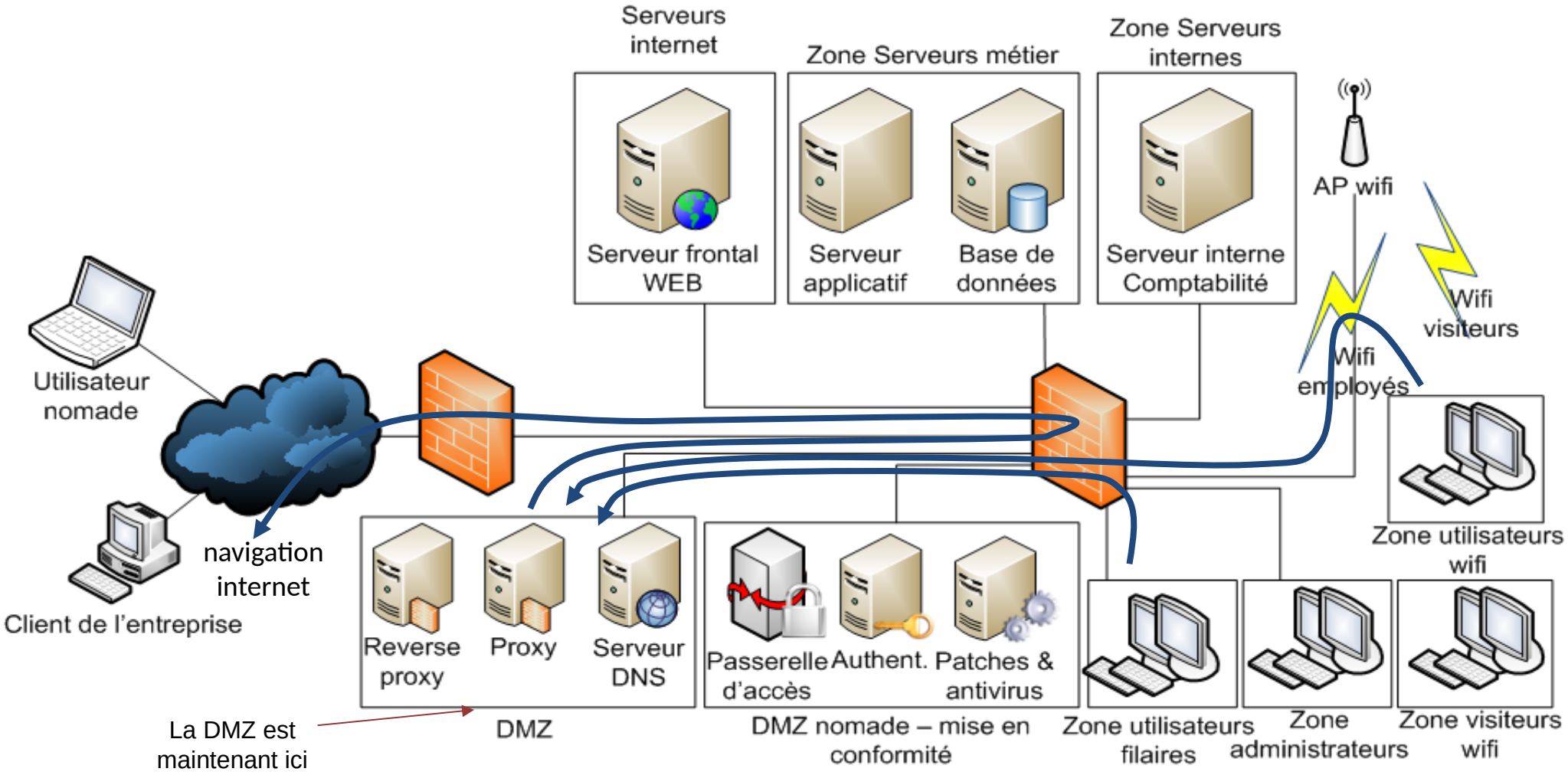
- **Trafic sortant** : définir les catégories de sites WEB que les employés sont autorisés à naviguer, implanter une liste blanche ou noire de sites autorisés/interdits.
- **Trafic entrant** : analyser les requêtes WEB d’internet vers le serveur de e-commerce afin d’intercepter les requêtes malveillantes (injection, malware, etc.).

Nous allons donc recourir à un **proxy pour analyser les flux sortants**, et **un reverse-proxy pour analyser les flux entrants**. Ces équipements étant en coupure, ils empêchent donc les postes de travail des utilisateurs d’être connectés directement à Internet tout en leur permettant de naviguer sur les sites autorisés. Même remarque pour le serveur WEB : celui-ci n’est plus connecté directement sur Internet, c’est le reverse-proxy qui est maintenant en frontal.

Puisque les proxies et reverse-proxies sont en frontal Internet, ce sont donc eux qu’il faut **placer dans la DMZ** maintenant.

3. Sécurité d'un réseau

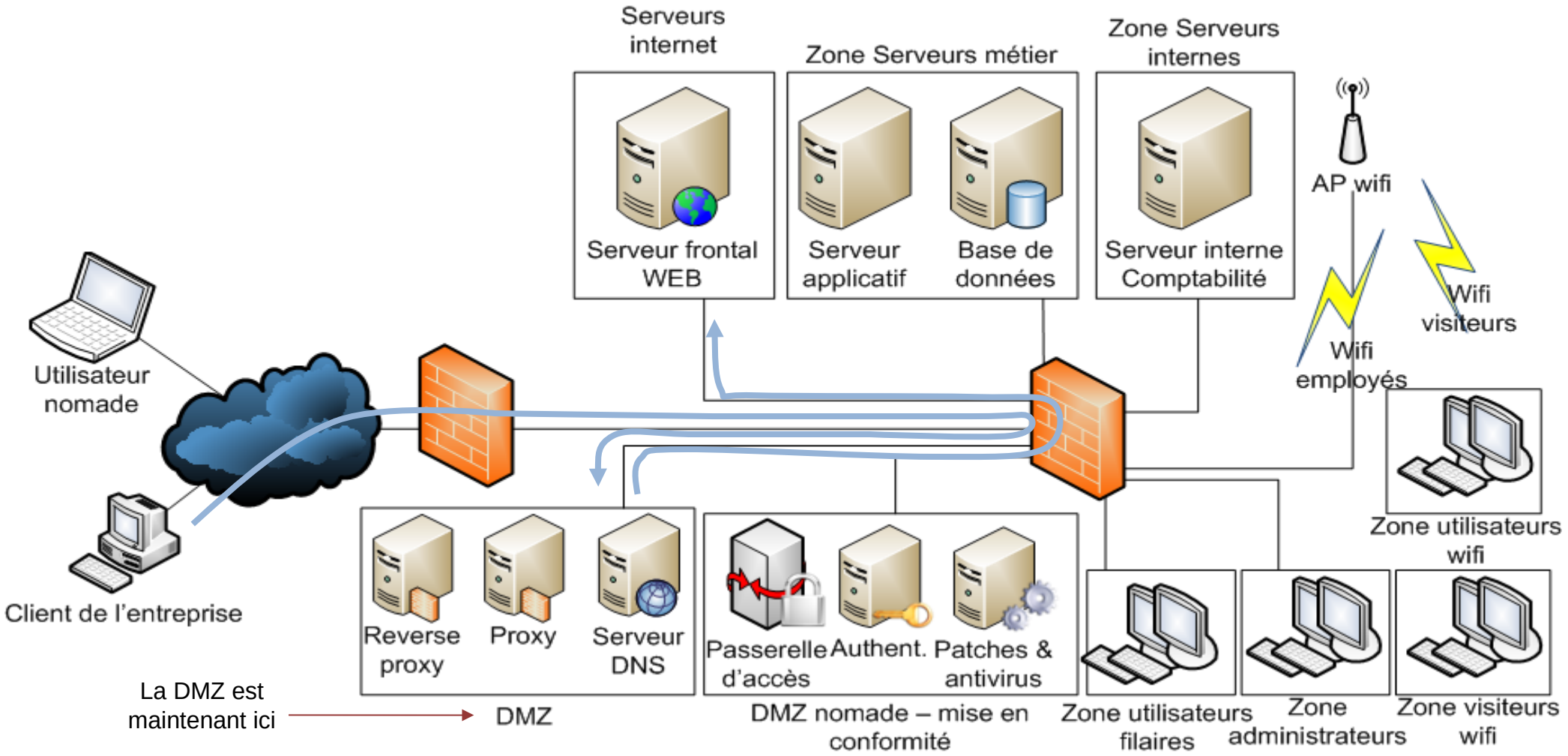
g. Exemple pratique de sécurisation avec un réseau simple



Réseau avec un proxy et un reverse-proxy en coupure des flux de/vers Internet

3. Sécurité d'un réseau

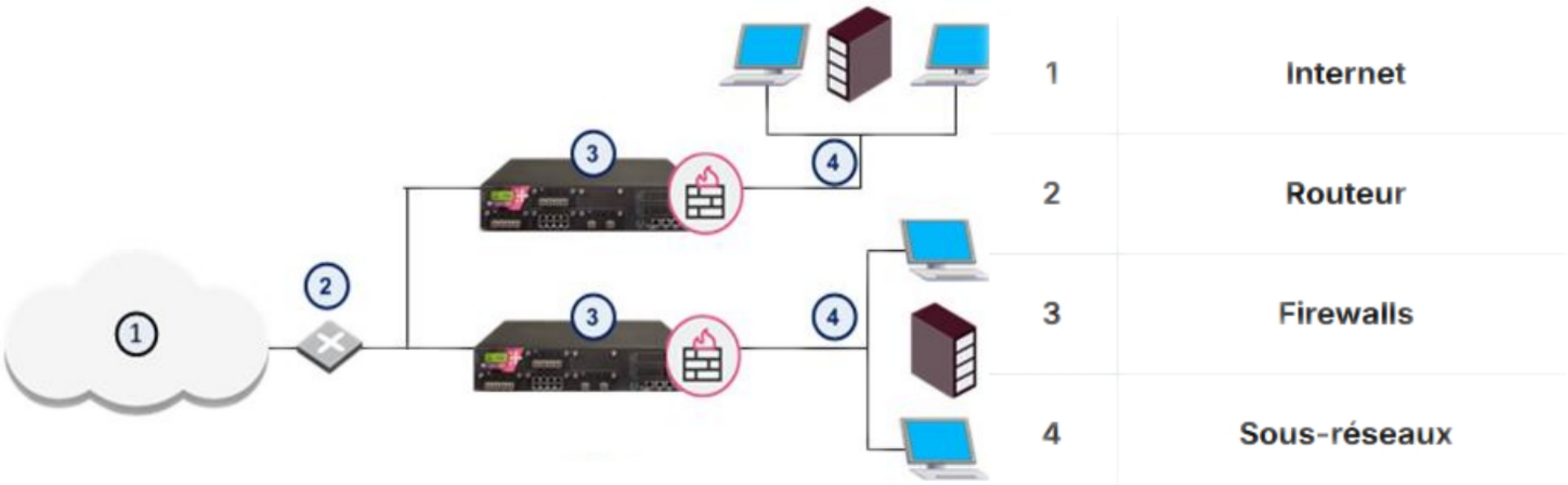
g. Exemple pratique de sécurisation avec un réseau simple



Réseau avec un proxy et un reverse-proxy en coupure des flux de/vers Internet.

3. Sécurité d’un réseau

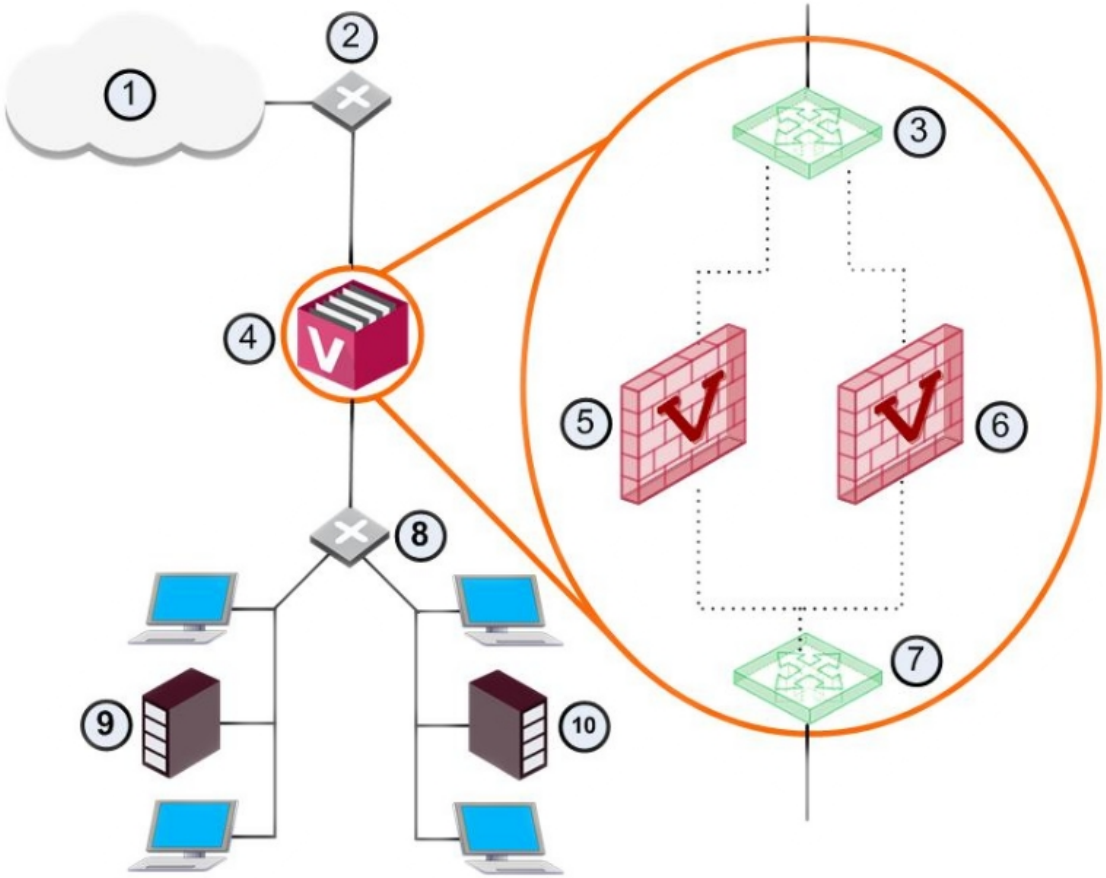
h. Réseau et virtualisation : pare-feu virtualisé VSX (Virtual System Extension) de chez Checkpoint



Exemple de réseau sans VSX

3. Sécurité d’un réseau

h. Réseau et virtualisation : pare-feu virtualisé VSX (Virtual System Extension) de chez Checkpoint

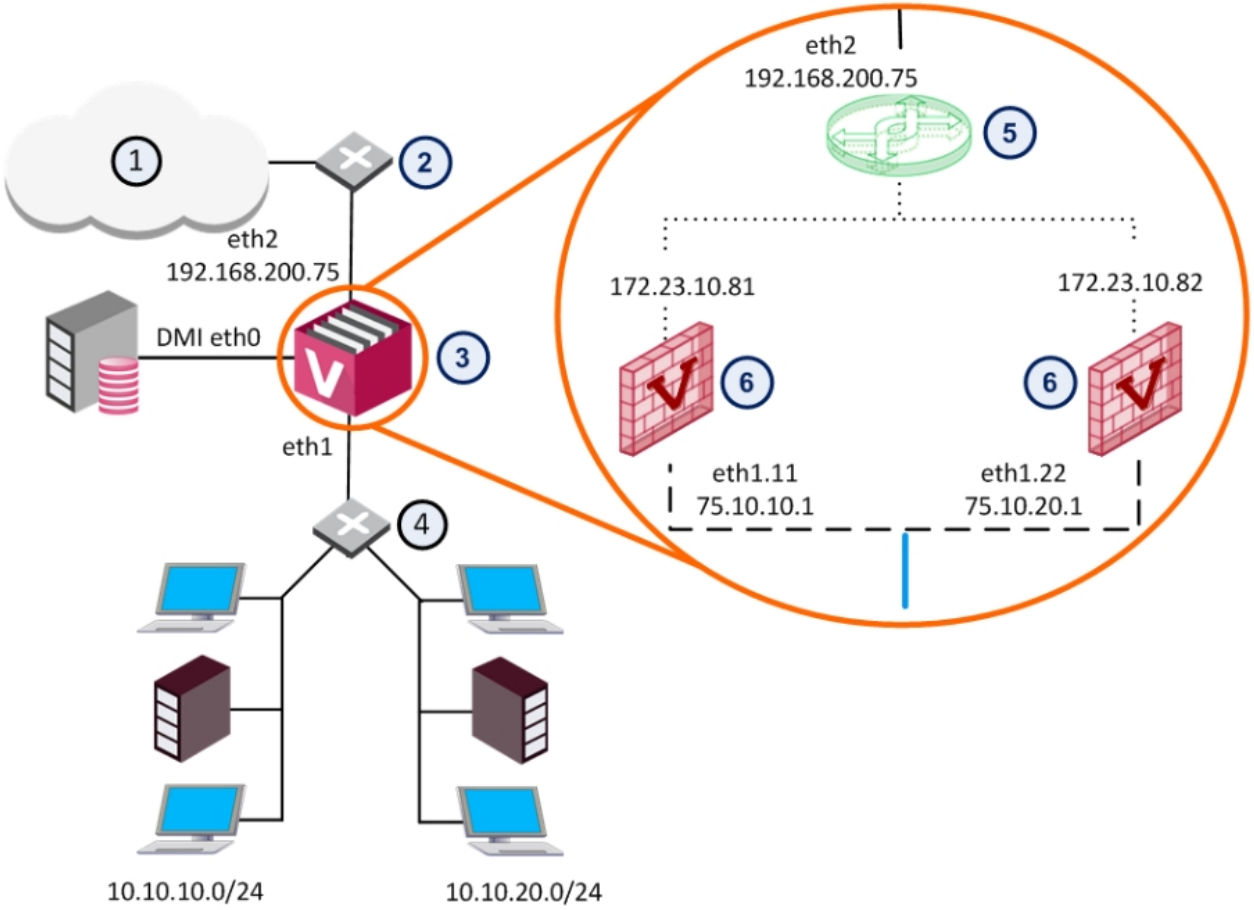


- 1 pare-feu physique
- Plusieurs pare-feux virtuels
  - Trafic pour la VLAN 9 filtré par le pare-feu 5
  - Trafic pour la VLAN 10 filtré par le pare-feu 6
- → centralisation et uniformisation des pare-feux

Schéma de base du pare-feu VSX de chez Checkpoint

3. Sécurité d'un réseau

h. Réseau et virtualisation : pare-feu virtualisé VSX (Virtual System Extension) de chez Checkpoint



1	Internet
2	Routeur
3	Firewall VSX
4	Switch
5	Switch VSX
6	Security Gateways VSX

Schéma un peu plus détaillé avec VSX

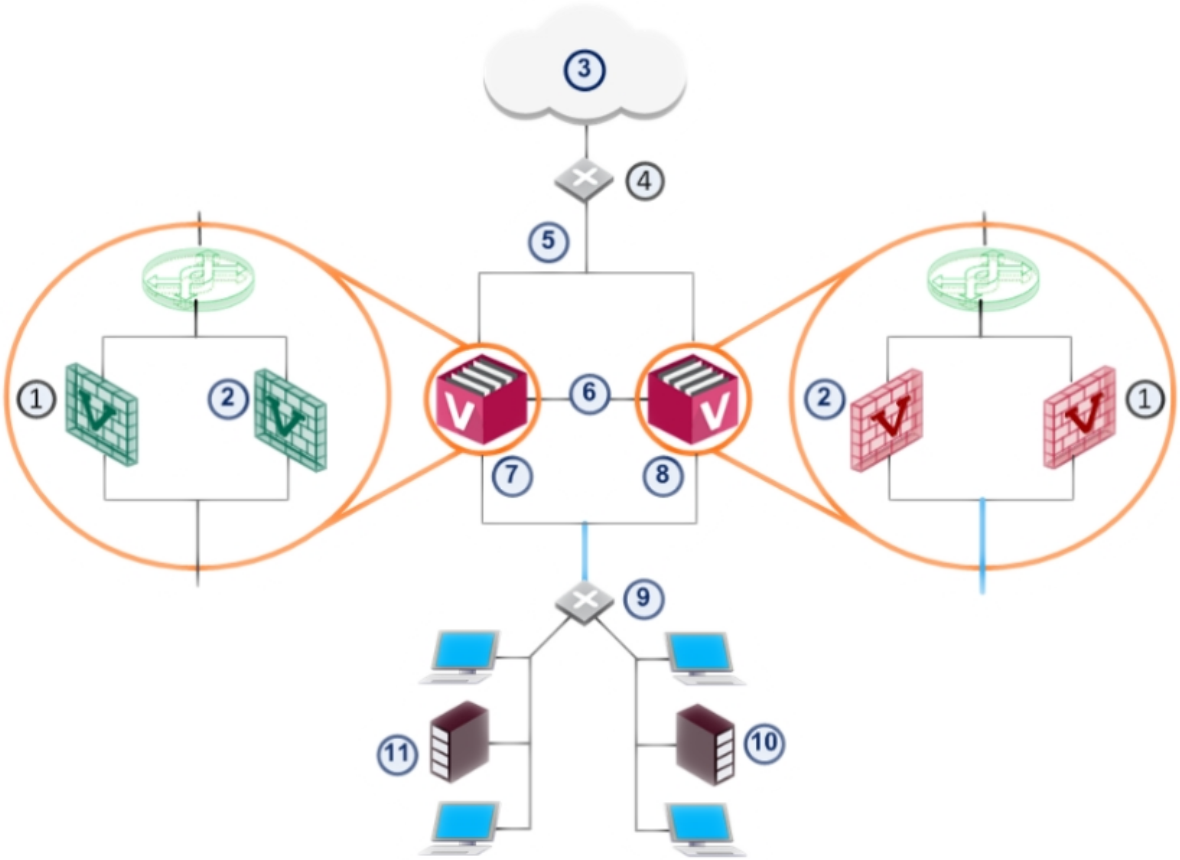
3. Sécurité d’un réseau

h. Réseau et virtualisation : pare-feu virtualisé VSX (Virtual System Extension) de chez Checkpoint

- Fait pare-feu + routeur
- Plusieurs pare-feux intégrés
- On peut y ajouter un antivirus, un IDS, un IPS

3. Sécurité d'un réseau

h. Réseau et redondance : pare-feu virtualisé VSX (Virtual System Extension) de chez Checkpoint

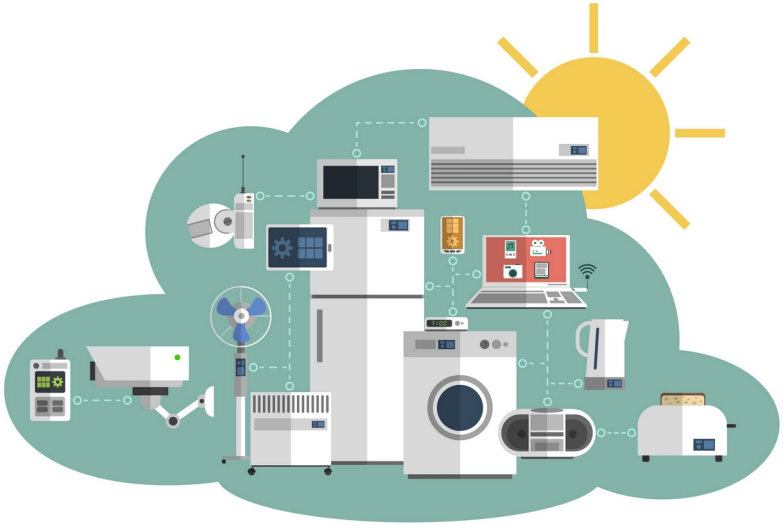


- 2 pare-feux physiques
- 2x2 pare-feux virtuels
- Lien physique de synchronisation TRUNK entre les deux pare-feux physiques
  - 7 et 8 partagent le même état
  - Possibilité de répartir la charge

Exemple de redondance : cluster de firewalls

4. Sécurité de l'IoT

a) Contexte
b) Problèmes
c) Le cas MQTT
d) Bilan



4. Sécurité de l'IoT

a. Contexte

IoT : Internet of Things (Internet des objets)

- Augmentation très rapide du nombre de dispositifs dans la vie courante, personnelle et professionnelle.
- Tendance au tout connecté.
  - Exemple : bâtiments intelligents (smart building)
- Convergence des technos de l'information (IT) et opérationnelles (OT), avec prolifération de dispositifs IoT.
- Matériel peu cher et facile à installer (capteurs, déclencheurs, contrôleurs, éclairages, caméras).
- Utilisation par connexion filaire ou sans fil.

→ De nouveaux défis en matière de cybersécurité !

4. Sécurité de l’IoT

b. Problèmes

Manque de sécurité

- Dispositifs souvent dépourvus de sécurité.
- Vulnérabilités découvertes de plus en plus souvent.

Mais tout n’est pas à la charge des constructeurs IoT

- Mauvaises pratiques (codes d’identification par défaut ou simples).
- Trafic non chiffré.
- Manque de segmentation du réseau.

4. Sécurité de l’IoT

D’après vous ?

Si ma brosse à dents connectée est piratée, c’est grave ?

4. Sécurité de l’IoT

b. Problèmes

Exemple : caméras de surveillance de dernière génération

- Livrées avec des protocoles faibles (Telnet, FTP, SSDP) activés par défaut
- Utilisent souvent RTP (protocole de communication temps réel non chiffré) et RTSP (protocole de streaming temps réel).
- Installées, configurées et déployées par des non experts en cybersécurité.
- Risque d’intrusion et de sabotage devenu critique pour les entreprises.

Exemple : sites du dark Web qui donnent un flux vidéo live de caméras pour surveiller les bébés.

4. Sécurité de l’IoT

b. Problèmes

Buts des attaques

- Tromper les utilisateurs en remplaçant les informations par d’autres (séquence vidéo pré-enregistrée).
- Permettre d’atteindre des vulnérabilités du monde physique.
- Ouvre la porte sur des intrusions dans les systèmes des entreprises.

- Possibilité d’utiliser SRTP sur la sécurité de la couche transport (TLS).
- Existence de protocoles sécurisés alternatifs à HTTP, FTP et Telnet.
- Mais pas toujours disponibles sur les périphériques IoT, ou pas configurés par défaut.
- Manque de compétences des utilisateurs finaux pour les configurer.

4. Sécurité de l’IoT

c. Le cas MQTT

Communication des objets connectés

- *Message Queuing Telemetry Transport*, ancienne initiative, devenue un standard depuis.

Détails :

- Sert à faire communiquer nos objets connectés simplement et efficacement.
- Infrastructure composée de 3 éléments
  - Le producteur d’information (*producer*)
  - Le consommateur d’information (*consumer*)
  - Un orchestrateur d’ensemble nommé Broker
- Fonctionnement sur le principe d’abonnement à un objet producteur, qui pousse l’information, distribuée par le broker aux objets abonnés au sujet.
- Nombreuses combinaisons d’abonnements

4. Sécurité de l’IoT

c. Le cas MQTT

Risques

- Des brokers publics pour fournir des environnements de tests

  dev.rabbitmq.com, *iot.eclipse.org*, *test.mosquitto.org*
- Mais parfois aussi utilisés en production
- Donc des informations sensibles transitent via ces infrastructures ouvertes à tous

Défauts de MQTT

- Manque d’utilisation du chiffrement (pourtant MQTTS avec TLS existe, mais utilisé par peu de brokers)
- Manque de confidentialité : un compte inscrit à un broker accède à presque tout (peu de ségrégation ou cloisonnement)
- Manque d’authentification : connexion anonyme courante.

4. Sécurité de l'IoT

d. Bilan

- Anciennes solutions de sécurité inadaptées pour les réseaux d'aujourd'hui.
- IoT : développé par beaucoup de gens qui n'y connaissent rien en informatique, encore moins en sécurité : encore plus besoin de **formation** !

Problème de fond : les objets connectés ont peu de ressources, si peu que même la cryptographie est difficile à ajouter : sans cryptographie, pas de sécurité !

- Cryptographie à clef publique : rare (trop chère), mais peu utile. On s'en sert surtout pour partager un secret initial, mais on peut le faire en dur, dès la mise en service de l'objet.
- Cryptographie à clef privée : rare (erreur), chère (perte de ressources utiles), imparfaite sur ces objets (cf. Ch4, « Aller plus bas »).

(Pas question de coder comme des bourrins en important d'immenses librairies, n'est-ce pas?)

5. Sécurité des applications web

a) Usurpation d’identité via les cookies
b) Injection SQL

5. Sécurité des applications web

a. Usurpation d'identité via les cookies

Comme toutes les applications, les applications web sont sujettes à des vulnérabilités. Nous allons en voir deux d'entre elles :

- une faiblesse basée sur les cookies ;
  - Ce qui permet – par exemple – à un attaquant de contourner un mécanisme d'authentification.
- une faiblesse basée sur un code source mal développé.
  - Ce qui permet – par exemple – à un attaquant de contourner un mécanisme d'authentification, d'accéder à des données pour les divulguer ou les corrompre.

5. Sécurité des applications web

a. Usurpation d'identité via les cookies

Les cookies sont des fichiers gérés par les navigateurs web afin de stocker (et réutiliser) des informations concernant l'utilisateur, par exemple :

- son identifiant ;
- ses préférences d'affichage et de disposition de la page web.

Les cookies sont nécessaires pour toutes les pages web dynamiques qui ont besoin d'identifier ou d'authentifier l'utilisateur, en permettant notamment la mise en œuvre de sessions :

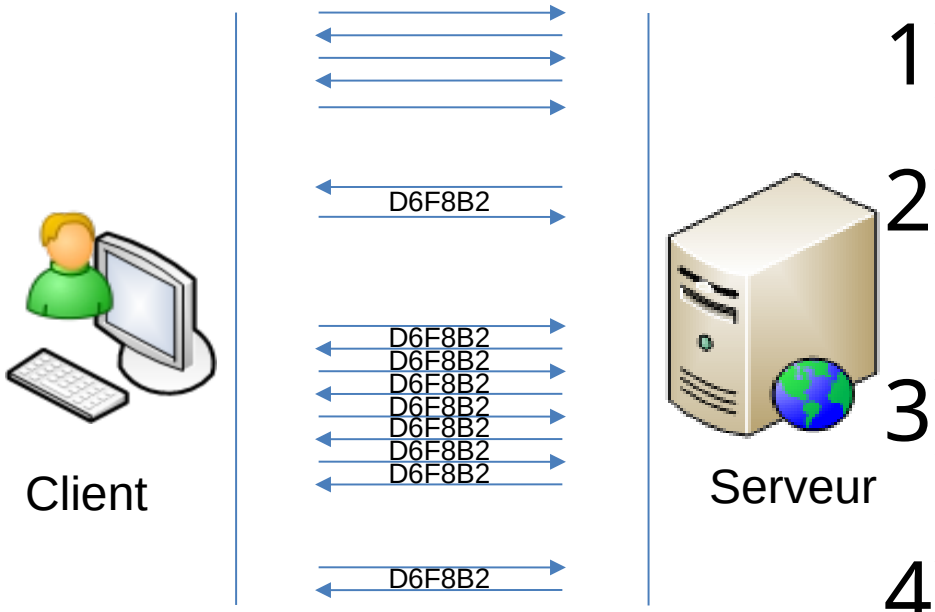
- les sites marchand (afin d'afficher le panier de l'utilisateur connecté) ;
- les sites bancaires (afin d'afficher le solde du compte de l'utilisateur connecté et non pas celui d'un autre client) ;
- les sites « en général » (afin d'afficher des publicités ciblées sur notre navigation).

Il est possible – sous certaines conditions – d'usurper l'identité d'un utilisateur sur un site web si on arrive à récupérer son cookie d'authentification.

5. Sécurité des applications web

a. Usurpation d'identité via les cookies

Fonctionnement habituel d'une connexion sur un site web nécessitant une authentification par cookie (site marchand, site bancaire, etc.) :



1) Phase d'authentification (via un mot de passe en général).

2) Phase de génération du cookie d'identification*. L'utilisateur est maintenant connecté « à son compte ».

3) Phase de « navigation » : le cookie est inclus dans tous les échanges afin que le serveur puisse identifier la connexion de l'utilisateur.

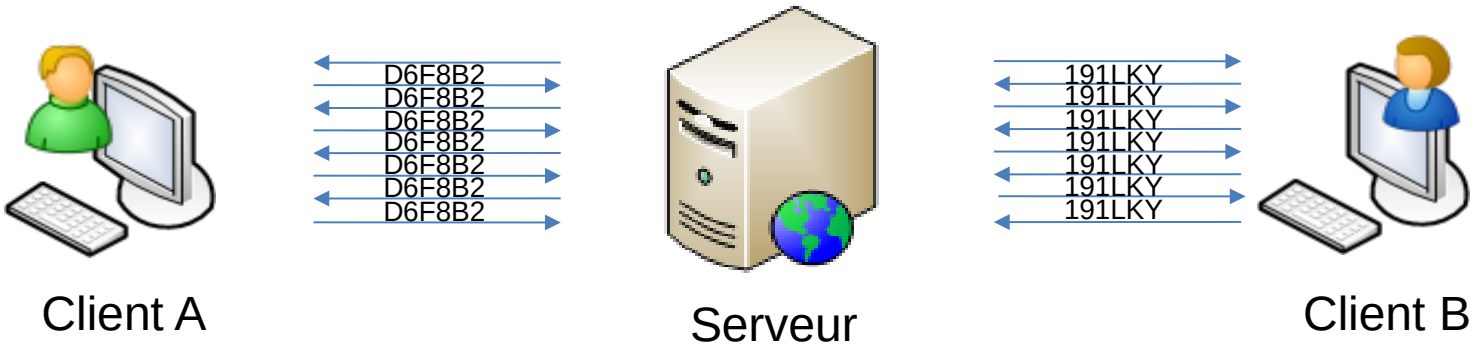
4) Phase de déconnexion : la session de l'utilisateur est maintenant clôturée. Le cookie est invalidé.

- Un cookie d'authentification est en fait une chaîne de caractères aléatoire et **unique**, suffisamment longue pour qu'elle ne puisse pas être générée deux fois par erreur.
  Exemple d'un cookie d'identification : D6F8B2BE3ED3040D9A3C10-D6F8B2A305D048B9

5. Sécurité des applications web

a. Usurpation d'identité via les cookies

A tout moment d'une connexion, chaque utilisateur du site web possède donc son propre cookie, unique à lui. Le serveur est donc en mesure d'identifier à qui appartient chaque connexion, et donc d'afficher les pages web qui lui sont propre.

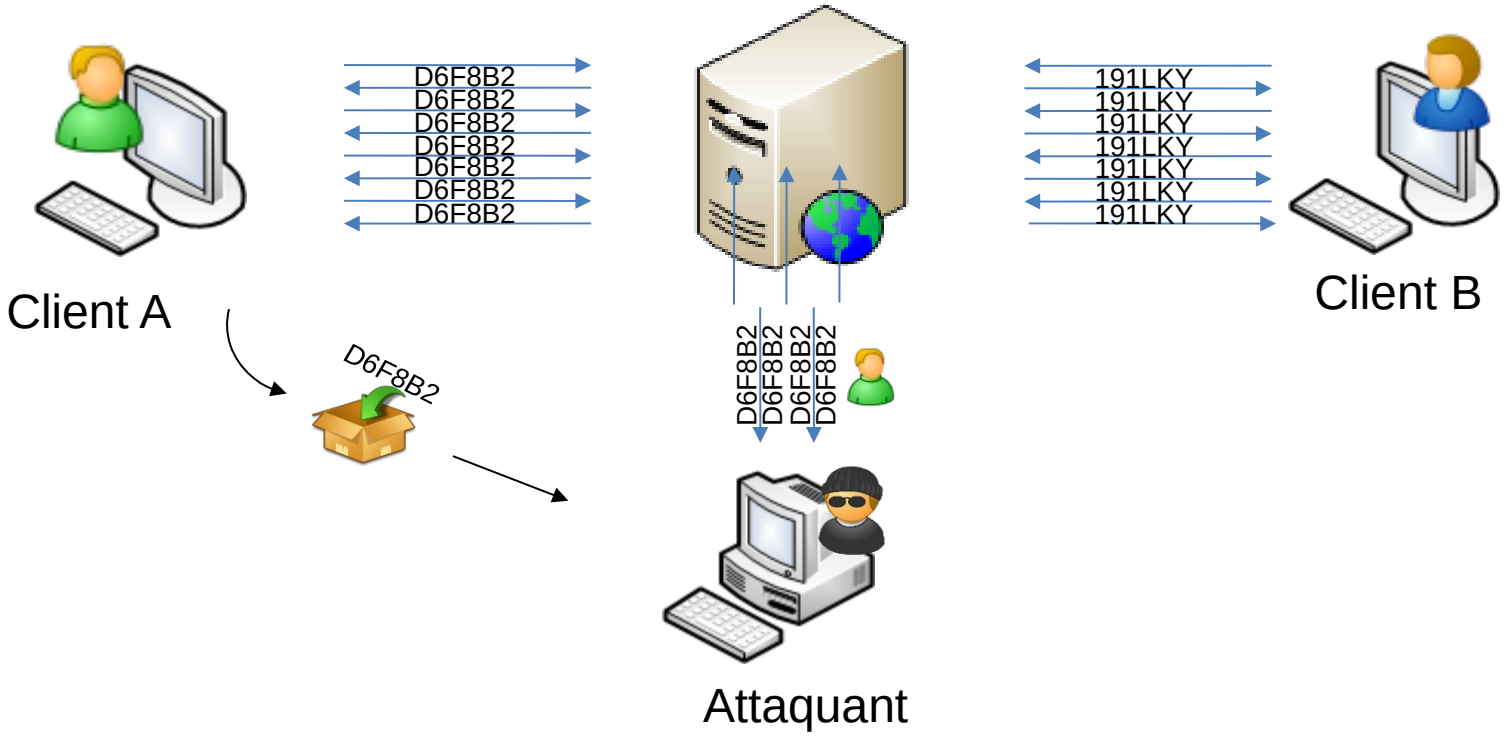


5. Sécurité des applications web

a. Usurpation d'identité via les cookies

Mais que se passe-t-il si un attaquant arrive à dérober le cookie d'authentification d'un utilisateur et se connecte au même serveur ?

- Il se fait passer pour l'utilisateur dont il a dérobé le cookie au près du serveur applicatif ! Il usurpe donc l'identité de la victime et accède à son compte.



5. Sécurité des applications web

a. Usurpation d'identité via les cookies

L'attaquant peut dérober un cookie d'authentification par différents moyens :

- soit en écoutant le trafic réseau HTTP et en interceptant les données applicatives, dont le cookie ;
  - Moyen de protection : l'utilisateur doit **s'assurer que le site auquel il est connecté utilise du HTTPS** (le cookie est donc chiffré pendant le transport).

- soit en dérobant le cookie sur le poste de travail en utilisant une vulnérabilité du système ;
  - Moyen de protection : l'utilisateur doit **sécuriser son système d'exploitation et ses logiciels** correctement (services inutiles désactivés, installation des mises à jours de sécurité, anti-virus, etc. voir le chap.2 pour plus d'informations).

- soit en dérobant le cookie sur le poste de travail via des méthodes d'ingénierie sociale ciblées sur l'utilisateur ;
  - Moyen de protection : l'utilisateur doit **être sensibilisé aux méthodes d'ingénierie sociale** (phishing, spam, etc.) afin de « ne pas tomber dans le panneau »

- soit en dérobant le cookie via une faille sur le serveur ;
  - Moyen de protection : l'exploitant du serveur doit **suivre les bonnes pratiques de sécurisation et du maintien en condition de sécurité** du serveur, ainsi que les **bonnes pratiques de développement applicatif**.

5. Sécurité des applications web

a. Usurpation d'identité via les cookies

- Il existe des mécanismes pour utiliser des cookies d'authentification en toute sécurité
- Il existe des alternatives aux cookies (le plus courant est le JWT, Java Web Token, un peu mieux sécurisé)

5. Sécurité des applications web

b. Injection SQL

- Une attaque par injection SQL permet à un **attaquant d'interagir directement avec la base de données** d'un site web (alors que l'accès à cette base est bien entendu interdit).

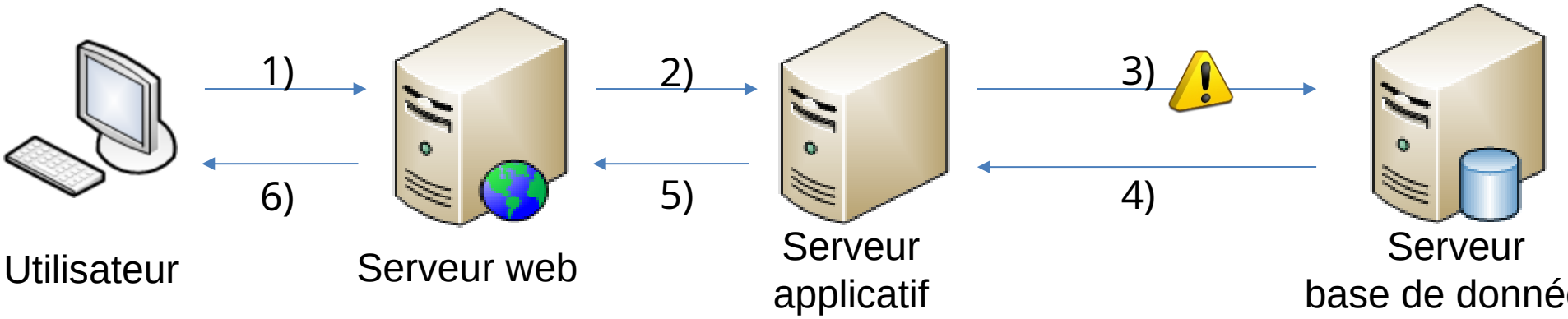
- L'objectif de ce type d'attaque est en général de **contourner le mécanisme d'authentification, d'accéder ou de modifier frauduleusement les données** confidentielles de la base (mots de passe, téléphones, numéro de carte bancaire, etc.).

- Il existe de multiples variantes possibles, la diapositive suivante présente un exemple de contournement d'authentification d'une page web.

5. Sécurité des applications web

b. Injection SQL

Architecture logicielle standard d'un site web faisant appel à une base de données



1) Le navigateur client demande l'affichage d'une page.
2) Le serveur web transfère la demande au serveur applicatif.
3) Le serveur applicatif génère une requête SQL afin de récupérer les informations nécessaires.
4) Le serveur base de données retourne le résultat de la requête au serveur applicatif.
5) Le serveur applicatif transmet au serveur web les informations nécessaires à la création de la page à afficher.
6) Le serveur web envoie les pages HTML au navigateur client.

5. Sécurité des applications web

b. Injection SQL

- L’objectif d’une attaque de type injection SQL consiste à détourner la requête SQL de l’étape 3 (diapositive précédente), et – en fonction du contexte – créer sa propre requête SQL malveillante.

- La diapositive suivante illustre comment une telle attaque peut être menée à partir d’un navigateur client.

5. Sécurité des applications web

b. Injection SQL

Formulaire WEB :

Entrez votre identifiant et votre mot de passe puis cliquez sur Connexion :

$user contient le login renseigné dans le formulaire par l'utilisateur.
$mdp contient le mot de passe.

La requête SQL permettant de vérifier le login et le mot de passe est la suivante :

```
select count(*) from user where user='$user' and mdp='$mdp'
```

Ainsi, une requête légitime serait la suivante :

```
select count(*) from user where user='thomas' and mdp='cykUfl9an'
```

5. Sécurité des applications web

b. Injection SQL

Formulaire WEB :

Entrez votre identifiant et votre mot de passe puis cliquez sur Connexion.

Mais que se passe-t-il si un attaquant rentre précisément les chaînes de caractères suivantes ?
Login : azerty
Mot de passe : **abcd' or 1=1/***

La requête SQL `select count(*) from user where user='$user' and mdp='$mdp'`

devient donc :

```
select count(*) from user where user='azerty' and mdp='abcd' or 1=1/*'
```

Cette condition est toujours vraie !

5. Sécurité des applications web

b. Injection SQL

- La condition étant toujours vraie, la requête est donc toujours valide, quel que soit le mot de passe renseigné par l'attaquant !
  - Les caractères /* sont utilisés pour ignorer la fin de la requête légitime.

- La faiblesse réside ici dans le code applicatif : les **données** renseignées par l'utilisateur (i.e. un attaquant dans notre scénario) **ne sont pas vérifiées/validées ;** elles sont au contraire utilisées telles quelles sans aucune vérification préalable qu'elles sont « inoffensives ».

- Comment s'en protéger ?
  - **Valider systématiquement chaque donnée** extérieure avant de l'utiliser.
  - Recourir à des requêtes préparées (connues sous le nom de « **prepared statements** »), qui ont l'avantage d'être plus résistantes aux injections.
  - D'une façon générale, **respecter les bonnes pratiques de développement** recommandées par l'industrie concernant le code PHP, Java, etc.

6. Le sur-protocole TLS

a) Principe de TLS
b) Versions de TLS
c) Sécurité de la version actuelle TLS v1.3
d) Utilisations de TLS

6. Le sur-protocole TLS

a. Principe de Transport Layer Security (TLS)

- **TLS** : surcouche de protocole réseau → plug-in sur **n'importe quel protocole existant** (HTTP, DNS, SMTP, etc)
- Crée un canal sécurisé cryptographique
- Protocole complexe avec de nombreuses étapes :
  - **Handshake :** « Bonjour, je veux me connecter, voici les méthodes de connexion sécurisée que je supporte. »
    « Bonjour client, je choisis telle méthode. »
    (très schématisé, c'est beaucoup plus complexe)
  - **Initialisation des paramètres cryptographiques (cf. Ch4)**
  - **Vérification du certificat électronique (cf. Ch4 et 6)**
  - **Le canal sécurisé est établi**

6. Le sur-protocole TLS

b. Versions de TLS

- SSL 1.0 (-) : ~~historique~~
- SSL 2.0 (1995) : ~~historique~~
- SSL 3.0 (1996) : ~~historique~~
- TLS 1.0 (1999) : ~~historique~~
- TLS 1.1 (2006) : ~~historique~~
- TLS 1.2 (2008) : *sécurité* ***passable*** (supporté par > 90 % des sites Web)
- **TLS 1.3** (2018) : *sécurité* ***suffisante*** (supporté par > 70 % des sites Web)

6. Le sur-protocole TLS

c. Sécurité de la version actuelle de TLS (TLS 1.3)

- cf. Travaux de l'équipe Inria Prosecco (dirigé par Karthikeyan Bhargavan)
- TLS 1.3 corrige de nombreuses failles de TLS 1.2
- Il reste des failles connues et non-corrigées dans TLS 1.3
  - Failles difficiles à exploiter
  - Failles difficiles à corriger (avec les solutions actuelles)
  - Il y aura sans doute un TLS 1.4 à l'avenir

6. Le sur-protocole TLS

d. Utilisations de TLS

- HTTP**S** : sécurisation du Web
- DNS-over-**TLS** : sécurisation de DNS
- Beaucoup d'autres
- Applicable partout
- Implémenté dans tous les systèmes d'exploitation (OpenSSL sur les gros OS, mbedTLS sur OS pour petits objets, ...)
- Pas implémenté sur les objets connectés sans système d'exploitation

- Il y a des alternatives pour certains usages : SSH a son propre protocole, GIT a son propre protocole, etc

7. Active Directory, centralisation et virtualisation

a) Active Directory – Annuaire, gestionnaire de comptes centralisé
b) GPO – centralisation des autorisations
c) Postes de travail virtualisés – VM et conteneurs
d) Intégration de la supervision et de sécurisation des communications

7. Active Directory, centralisation et virtualisation

a. Active Directory – Annuaire, gestionnaire de comptes centralisé

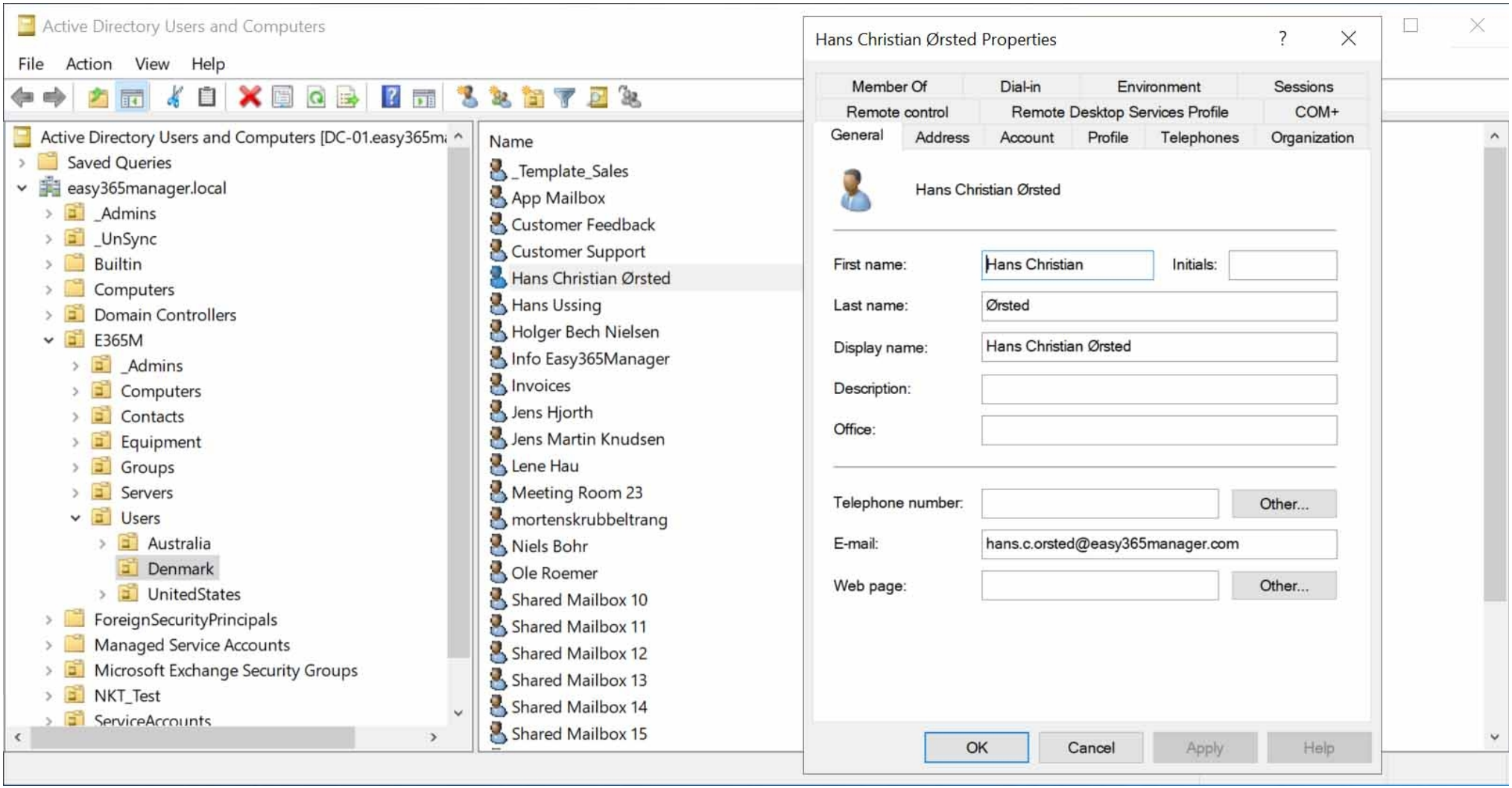
Active Directory
Technologie (surtout sous Windows) d'annuaire des comptes d'un SI.
Déployé sur un serveur central du SI.

- → Gestion centralisée des **droits d'accès** (groupes)
- → Gestion centralisée des **logins**
- → Gestion centralisée des **machines autorisées** sur le réseau
- → Gestion centralisée de la **sécurisation des communications internes** du SI
- → Gestion centralisée des **autorisations**
- → Déploiement automatique des **sessions utilisateur** sur les machines du SI
- → Intégration de nombreux **outils de sécurisation** sur ce nœud central

On en regarde une partie dans cette section, le reste il faudra découvrir sur le tas !

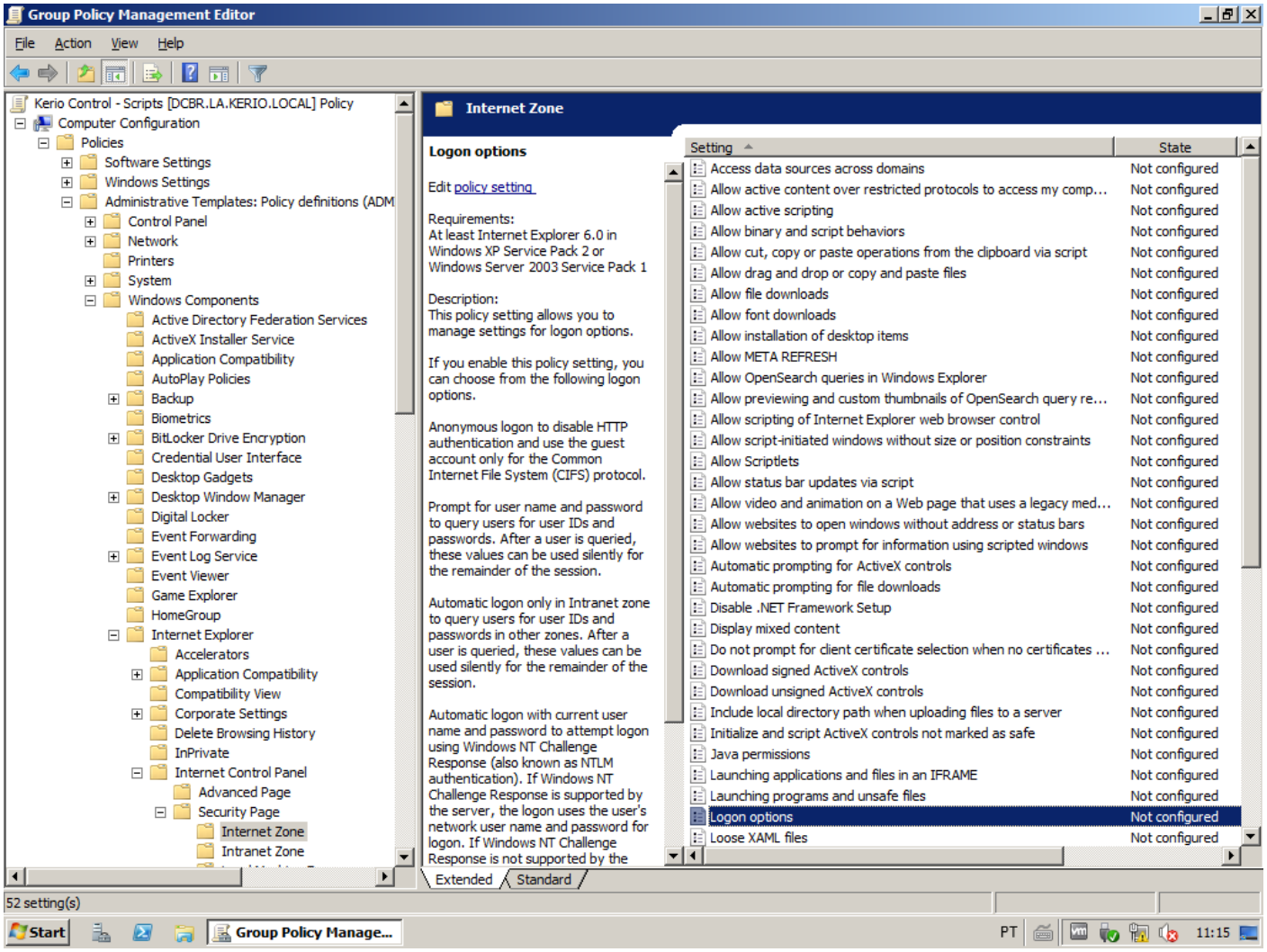
7. Active Directory, centralisation et virtualisation

a. Active Directory – Annuaire, gestionnaire de comptes centralisé



7. Active Directory, centralisation et virtualisation

b. GPO – centralisation des autorisations



7. Active Directory, centralisation et virtualisation

c. Postes de travail virtualisés – VM et conteneurs

- Pas de poste de travail physique dédié à un utilisateur : on déploie un poste de travail virtuel à la volée (vous avez l'habitude avec l'AD des postes de salles de TP).
- Login sur n'importe quelle machine → envoyé vers l'authentification + autorisation centralisée sur le serveur de l'AD → déploiement d'un poste de travail virtuel sur la machine
  - Postes de travail uniformisés
  - Postes de travail facilement mis à jour
  - Pas de dépendance aux machines physiques
  - Centralisation → simplification des authentifications + autorisations

7. Active Directory, centralisation et virtualisation

c. Postes de travail virtualisés – VM et conteneurs

Machine virtuelle (VM)

Émulation d'une machine :
- Couche physique
- OS
- Applis
- …

Usage :
- Test d'une techno sur la machine qu'on veut (OS spécifique, téléphone spécifique, …)
- Indépendant de la couche physique réelle

Conteneur (Docker généralement)

Émulation d'un OS :
- OS
- Applis
- …

Usage :
- Beaucoup - cher qu'une VM
- Déploiement de plusieurs conteneurs sur la même machine physique
- Rapide à déployer à la volée

7. Active Directory, centralisation et virtualisation

d. Intégration de la supervision et de sécurisation des communications

Supervision

- Tous les accès transitent par l'AD : nœud central de la sécurité d'un parc informatique Windows
- Pratique pour superviser ce qui se passe dans le SI (logins, machines actives, tentatives de modification des droits d'accès, …)
- On verra la supervision un peu plus en détails au ch6, Section SIEM (Security Information and Event Manager), qu'on peut intégrer à un AD

Sécurisation des communications

- L'AD joue le rôle d'autorité centrale interne (CA) (cf. ch4 Cryptographie), génère et gère les clefs de sécurisation cryptographiques pour toutes les machines du SI, en particulier peut intégrer une infrastructure de clefs publiques (PKI) (cf. ch7 PKI)
- L'AD centralise les tunnels sécurisés d'accès (VPN, SSH, Remote Desktop = RDP)

Et bien plus

- Intégration d'antivirus, de gestionnaire [automatisé] d'incidents, …

Bref, l'AD est le nœud central de la sécurité d'un SI Windows.

Références liées au chapitre

Matériels réseau

https://cyber.gouv.fr/publications/recommandations-de-configuration-des-commutateurs-et-pare-feux-siemens-scalance
https://cyber.gouv.fr/publications/recommandations-de-configuration-des-commutateurs-et-pare-feux-hirschmann
https://cyber.gouv.fr/publications/recommandations-pour-une-configuration-securisee-dun-pare-feu-stormshield-network

Cloisonnement

https://cyber.gouv.fr/publications/profil-de-fonctionnalites-et-de-securite-sas-et-station-blanche-reseaux-non-classifies
https://cyber.gouv.fr/publications/recommandations-de-deploiement-du-protocole-8021x-pour-le-controle-dacces-des-reseaux
https://cyber.gouv.fr/publications/recommandations-pour-choisir-des-pare-feux-maitrises-dans-les-zones-exposees-internet

Protocoles réseau

https://cyber.gouv.fr/publications/recommandations-de-securite-relatives-tls
https://cyber.gouv.fr/publications/recommandations-relatives-aux-architectures-des-services-dns

Internet

https://cyber.gouv.fr/publications/recommandations-pour-lhebergement-des-si-sensibles-dans-le-cloud
https://cyber.gouv.fr/publications/recommandations-relatives-linterconnexion-dun-si-internet

Web et Dev

https://cyber.gouv.fr/publications/securiser-un-site-web
https://cyber.gouv.fr/publications/mise-en-oeuvre-securisee-dun-cms
https://cyber.gouv.fr/publications/regles-de-programmation-pour-le-developpement-securise-de-logiciels-en-langage-c

Pour aller plus loin…

Sécurité réseaux, Jérôme François, Patrick Étienne

Sécurité des CPS, Abdelkader Lahmadi (option)

Cryptographie avancée, Marine Minier (option)

Sécurité des Applications Web, Sébastien Duval

Support de cours basé sur un document pédagogique mis à disposition par l'ANSSI sous licence Creative Commons Attribution 3.0 France.