



CyberEdu

La sécurité par l'enseignement supérieur des NTIC

Sensibilisation et initiation à la cybersécurité

Module 3 : les aspects réseau et applicatifs

03/11/2017

Ce document pédagogique a été rédigé par un consortium regroupant des enseignants-chercheurs et des professionnels du secteur de la cybersécurité.



Il est mis à disposition par l'ANSSI sous licence Creative Commons Attribution 3.0 France.



CyberEdu

La sécurité par l'enseignement supérieur des NTIC

Plan du module

- 1. La sécurité du protocole IP**
- 2. Sécurisation d'un réseau**
- 3. Les bases de la cryptographie**
- 4. La sécurité des applications web**

1. La 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. La 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 » inconnu à l'époque, personne n'imaginait que ces protocoles pourraient être détournés à des fins malveillantes ;
- **Aucun mécanisme de sécurité n'est donc implémenté au sein de ces protocoles.**

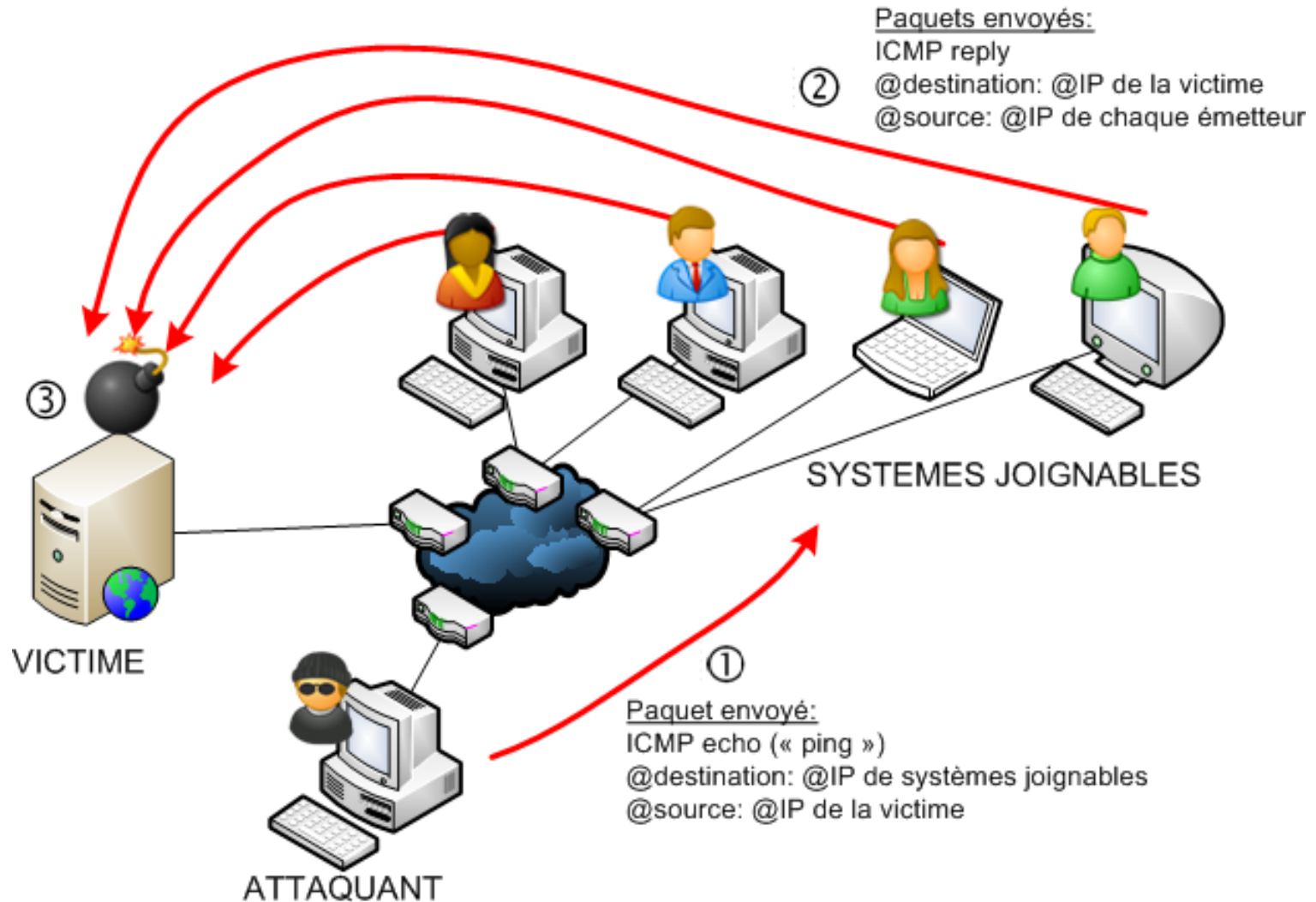
Quelques exemples de faiblesses de ces protocoles

- **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. La sécurité du protocole IP

b. Exemple d'attaque par réflexion



1. La sécurité du protocole IP

b. Exemple d'attaque par réflexion

But de l'attaque

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

Quelles sont les caractéristiques de l'attaque ?

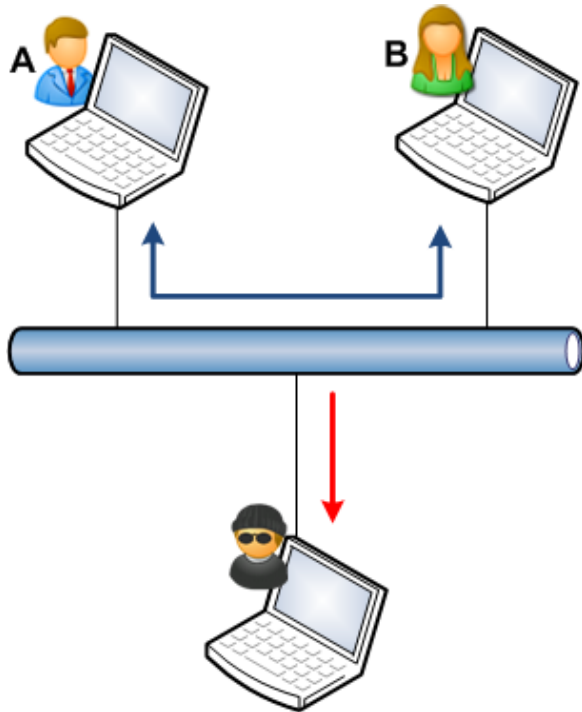
- usurpation d'adresse IP ;
- réflexion de trafic en ayant recours à des systèmes tiers « innocents ».

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.

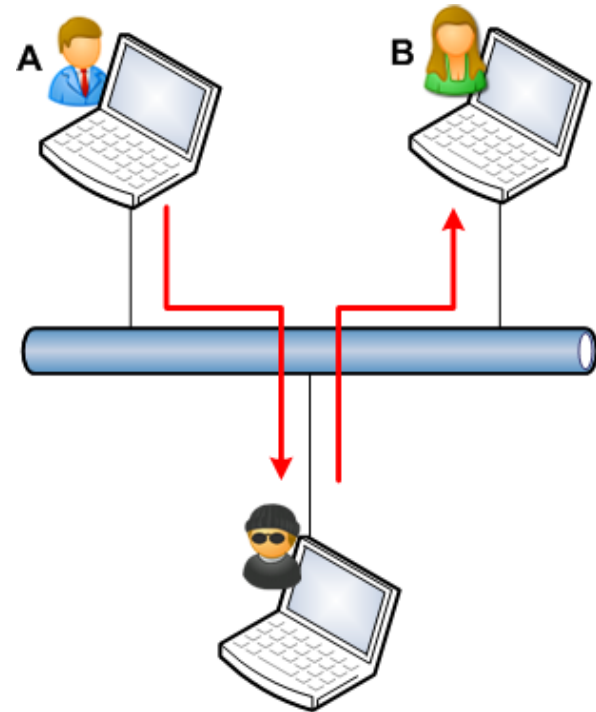
1. La sécurité du protocole IP

c. Exemples d'écoute de trafic



Ecoute passive

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

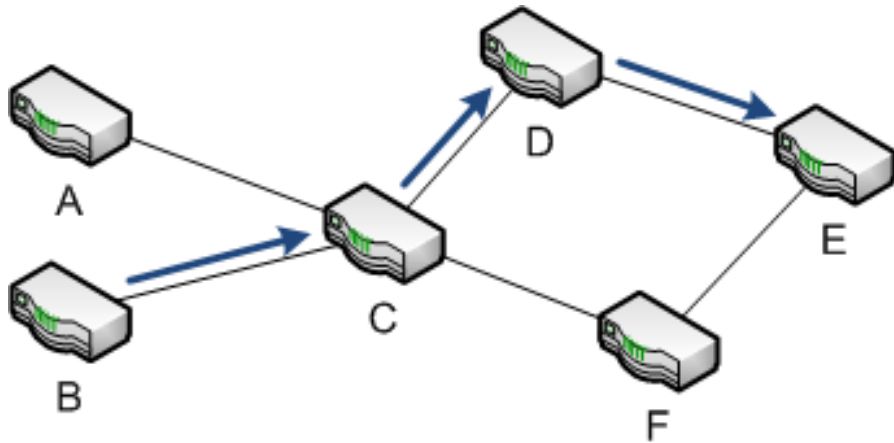


Ecoute active

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. La sécurité du protocole IP

d. Exemple de modification du routage des datagrammes IP

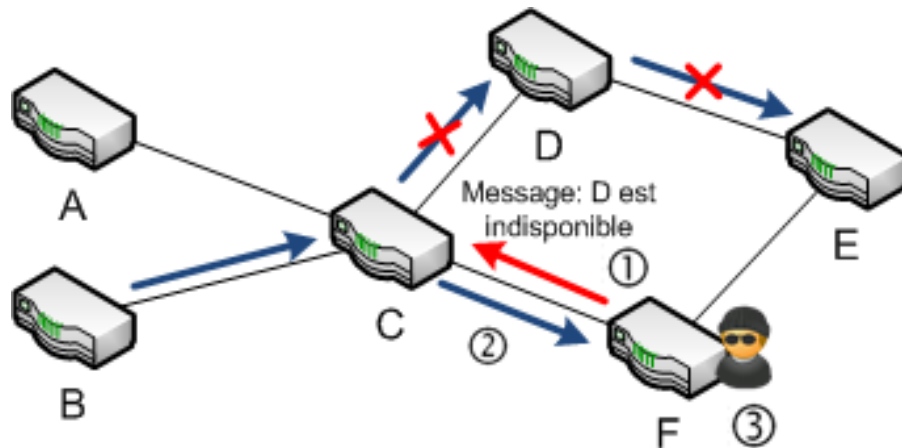


Chaque routeur possède une table de routage qui indique vers quel routeur voisin transmettre les datagrammes. Cette table peut être mise à jour dynamiquement en fonction des événements réseaux (protocoles BGP, RIP, OSPF, etc.).

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. La 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.

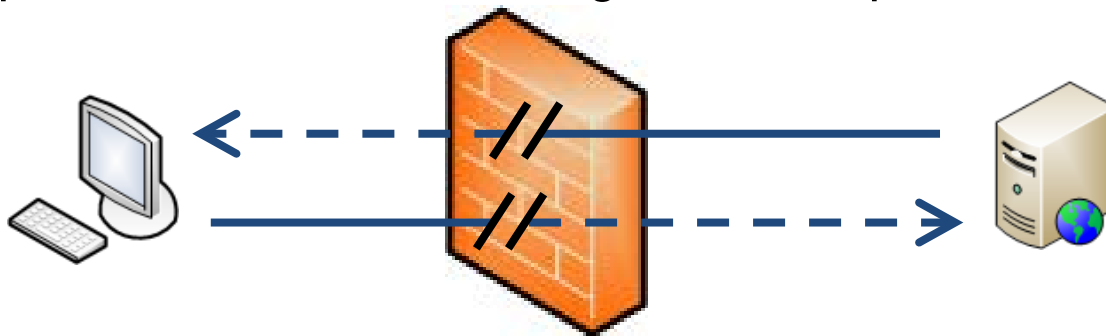
2. Sécurisation d'un réseau

- 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

2. Sécurisation 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 ;
- Implémente un **mécanisme de filtrage basé sur des règles** : il ne transmet donc que les paquets réseaux qui respectent les règles de filtrage implémentées 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 le paquet réseau ou non.

2. Sécurisation 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 peuvent également être implémentés 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 légitime. Toutes les autres connexions sont donc bloquées.

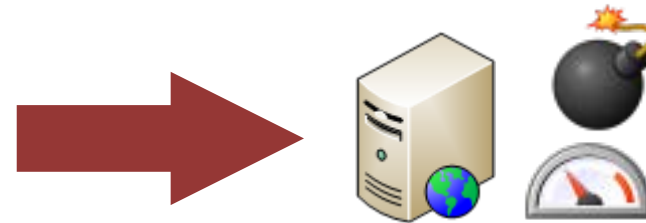
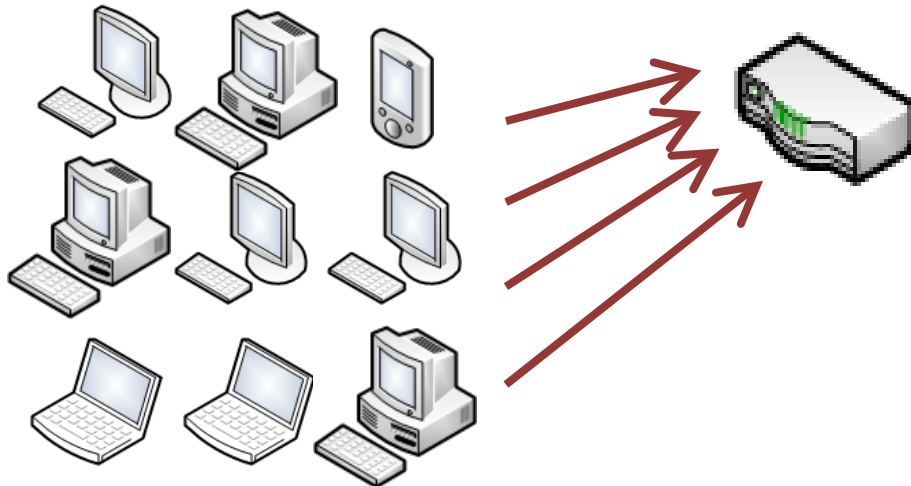
2. Sécurisation d'un réseau

b. Répartiteur de charge

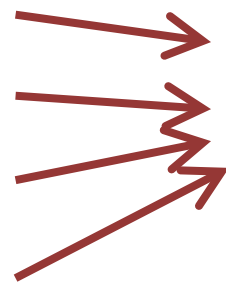
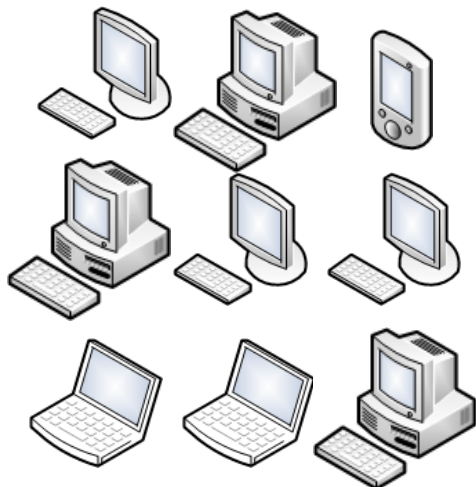
- « Load-balancer » en anglais ;
- Équipement rencontré sur les grosses infrastructures où les serveurs doivent faire face à de très fortes bandes passantes et charges élevées de trafic ;
- Équipement chargé de **répartir/distribuer la charge réseau** en fonction des caractéristiques de celui-ci et de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

2. Sécurisation d'un réseau

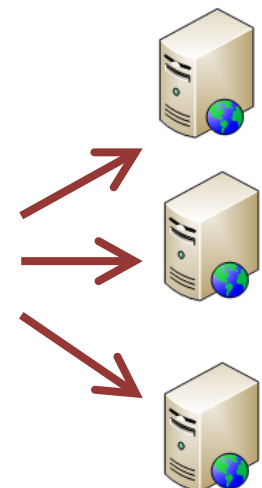
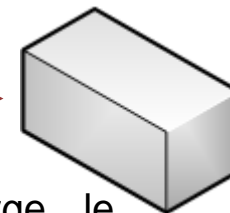
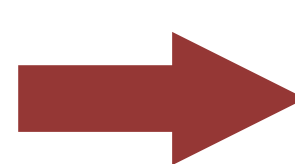
b. Répartiteur de charge



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.



2. Sécurisation d'un réseau

c. Anti-virus

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

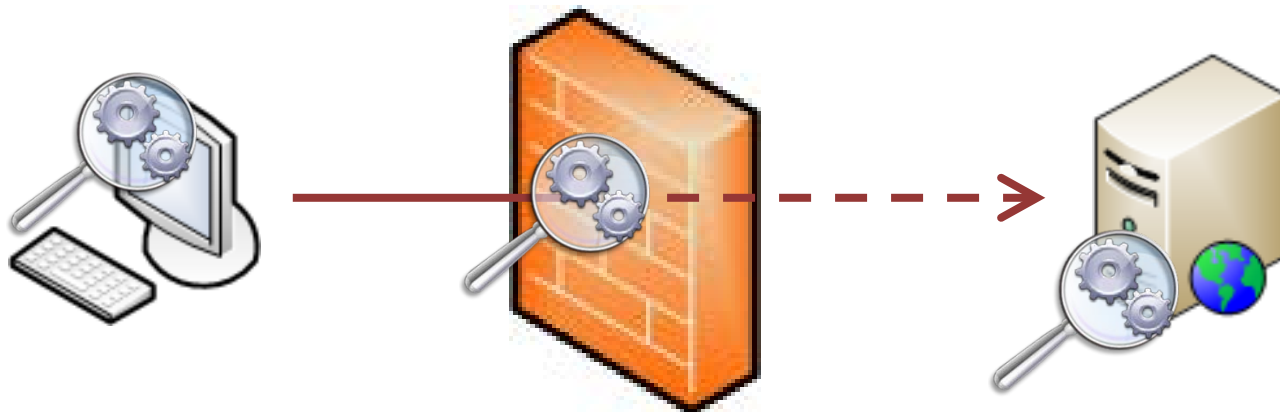
- Virus, vers, *keylogger* (enregistreur de frappe), chevaux de Troie, etc.
- Ces logiciels fonctionnent en général avec une base de données qui contient les signatures des malware connus. Ils analysent en permanence les fichiers et les exécutable du système hébergeant l'anti-virus ;
- **Limite des anti-virus** : 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.

2. Sécurisation 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.



2. Sécurisation 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 une configuration fine et maintenue :

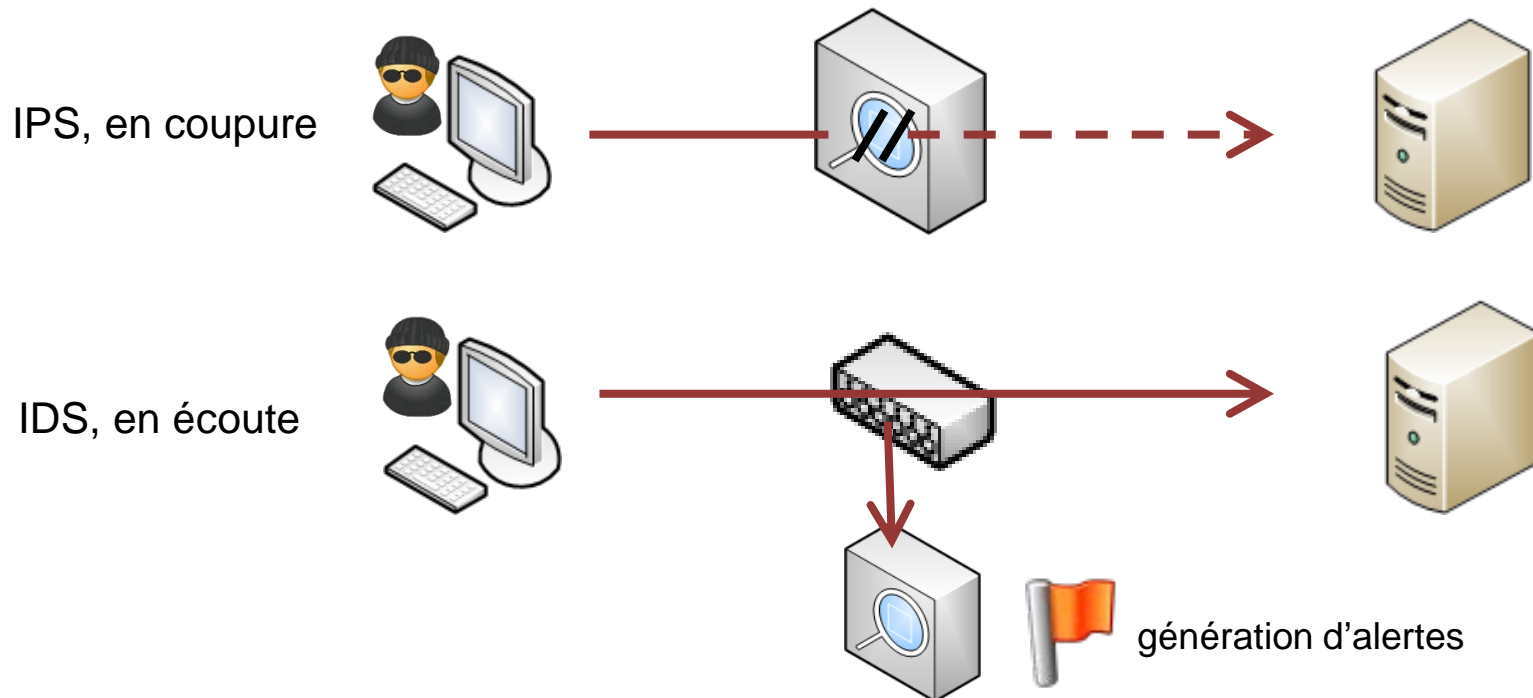
- Ils sont en effet connus pour présenter de nombreux faux-positifs (i.e. ils détectent à tort une tentative d'intrusion) ;
- De plus, les IDS/IPS basés sur des signatures ne peuvent détecter que les intrusions dont les caractéristiques techniques sont déjà connues et référencées.

2. Sécurisation 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.



2. Sécurisation d'un réseau

e. VPN

VPN **V**irtual **P**rivate **N**etwork

Un VPN est un **réseau virtuel** qui permet à **deux réseaux distants de communiquer en toute sécurité**, 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 et 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 ?



2. Sécurisation 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.

2. Sécurisation d'un réseau

e. VPN



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

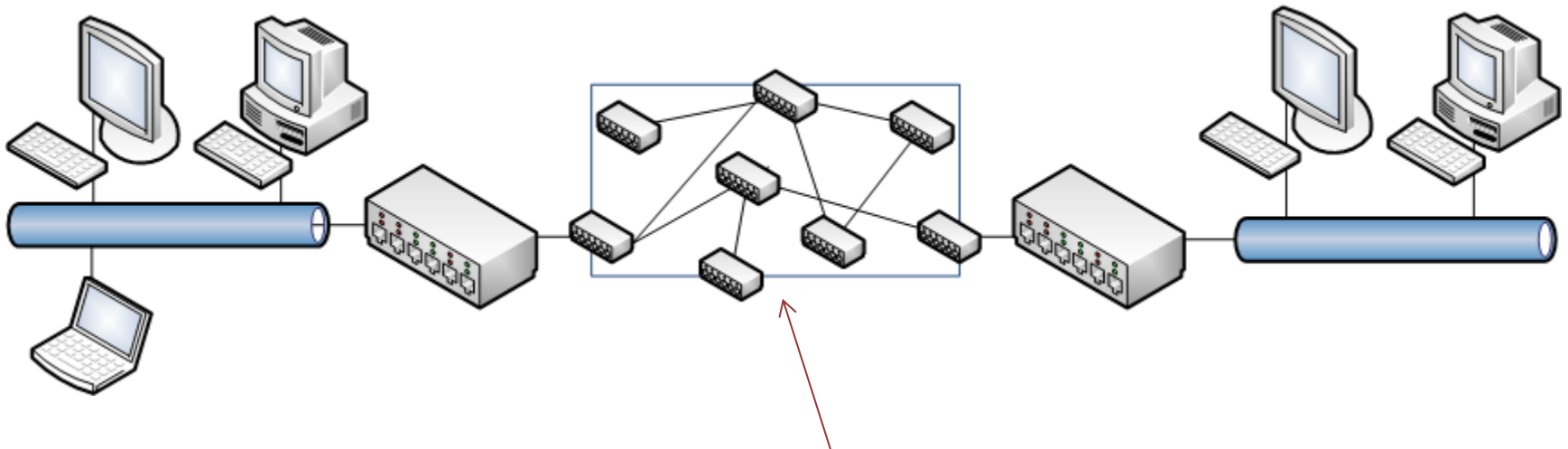


VPN entre systèmes
TLS – au niveau de la couche Transport

2. Sécurisation 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

2. Sécurisation 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.

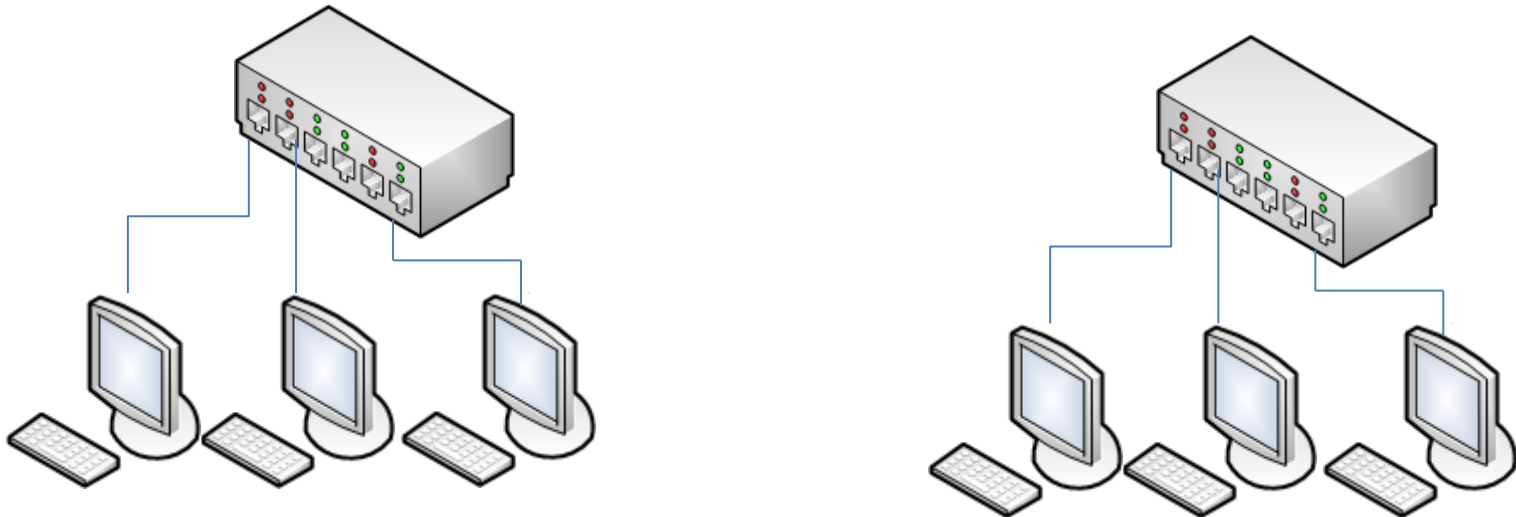
Appliqué au domaine réseau, il est donc fait **recours à de la segmentation** afin de séparer le réseau en différentes zones.

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

2. Sécurisation 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.

2. Sécurisation d'un réseau

f. Segmentation

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

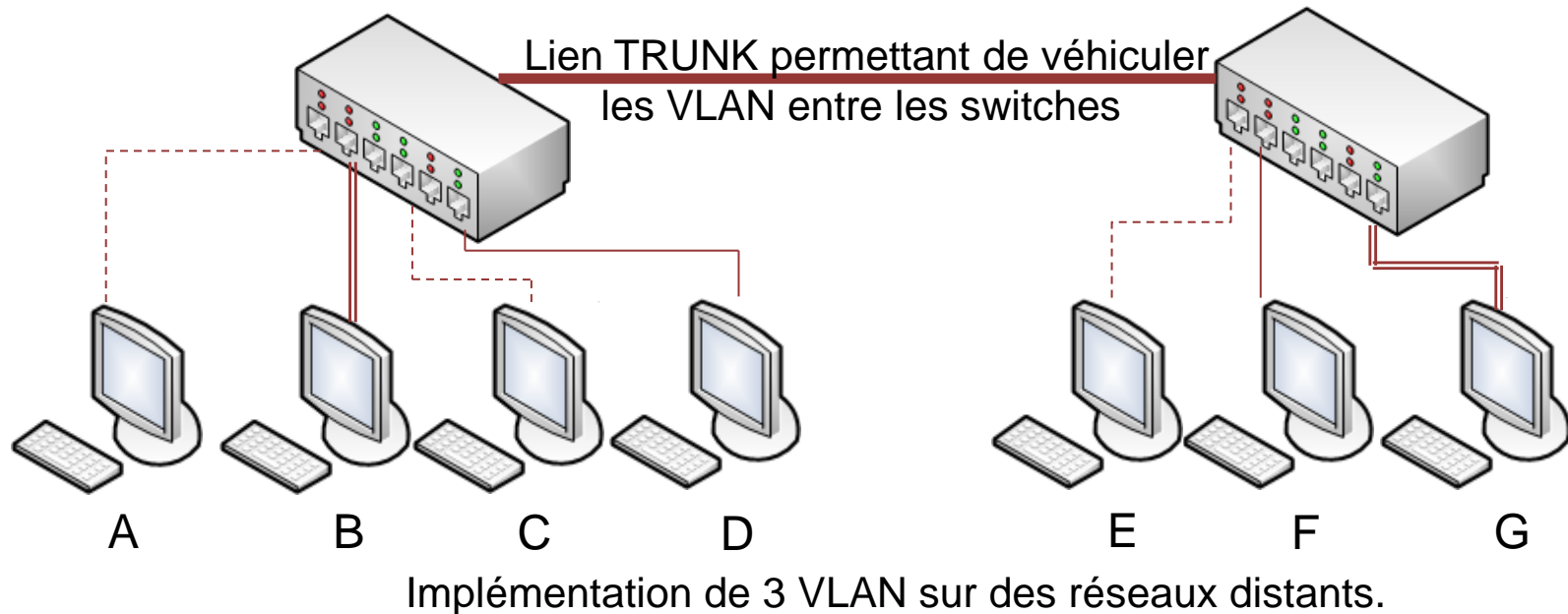
Les VLAN sont des **réseaux virtuels implémentés par les switches**. Ceux-ci **restreignent la communication entre systèmes selon des règles configurées** sur l'équipement réseau :

- 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 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.

2. Sécurisation d'un réseau

f. Segmentation



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.

2. Sécurisation d'un réseau

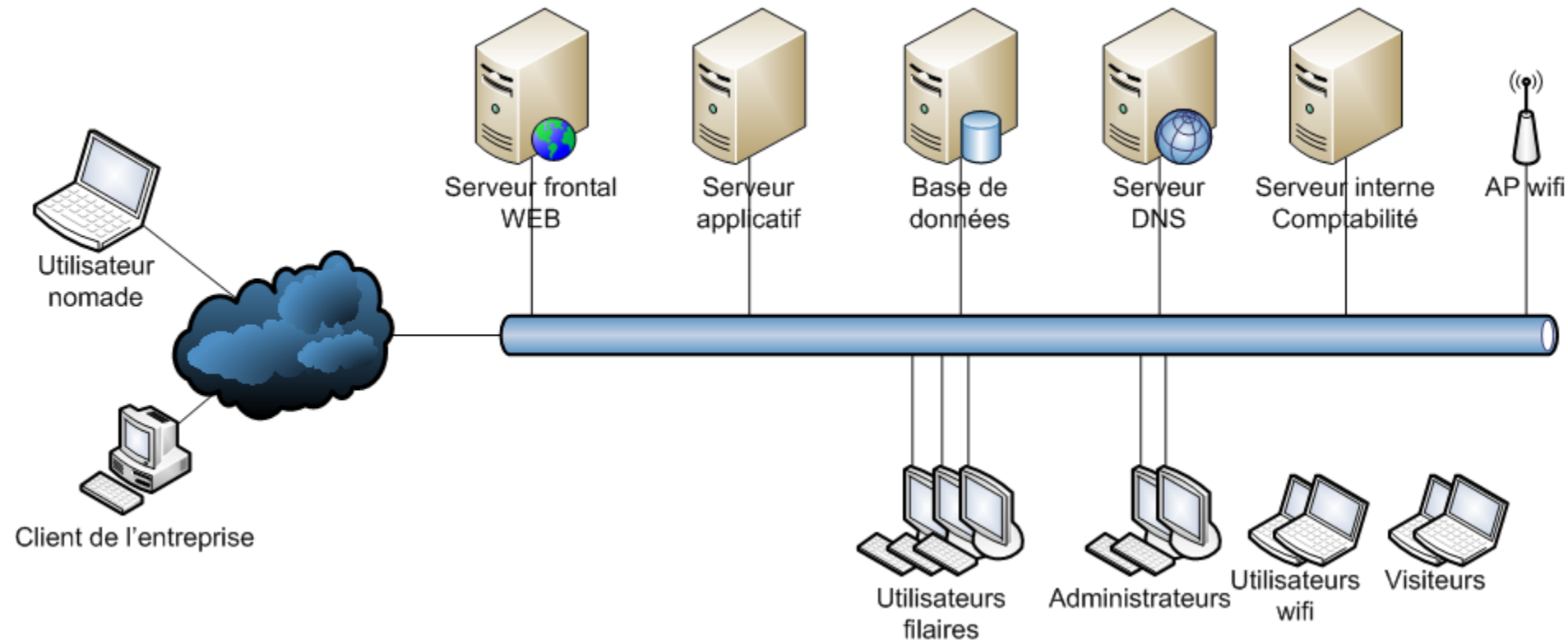
g. Exemple pratique de sécurisation avec un réseau simple

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.

2. Sécurisation d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple



Réseau « à plat », avant sécurisation

2. Sécurisation 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.

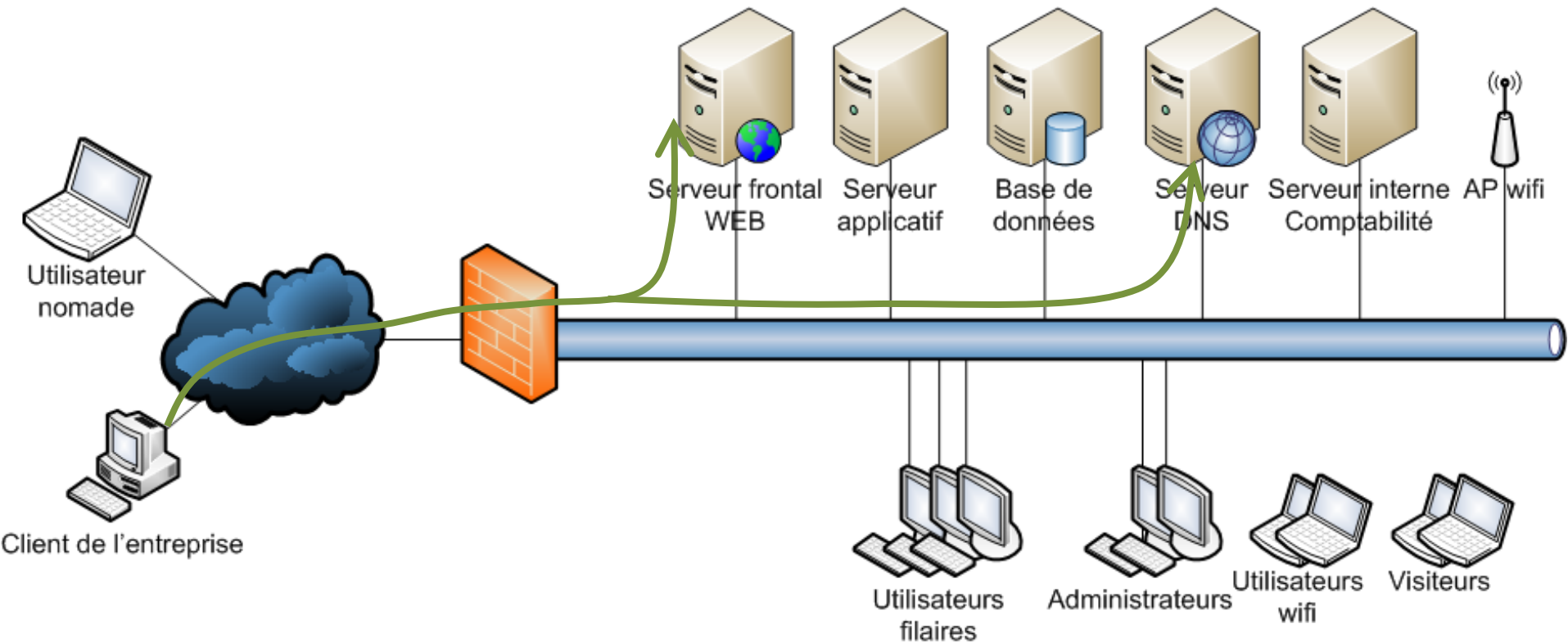
Parmi les nombreuses faiblesses architecturales de ce réseau, nous pouvons identifier au moins le problème suivant :

- Le réseau est **directement connecté à Internet**, i.e. tous les systèmes et utilisateurs et systèmes 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.

2. Sécurisation d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple



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

2. Sécurisation 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 :

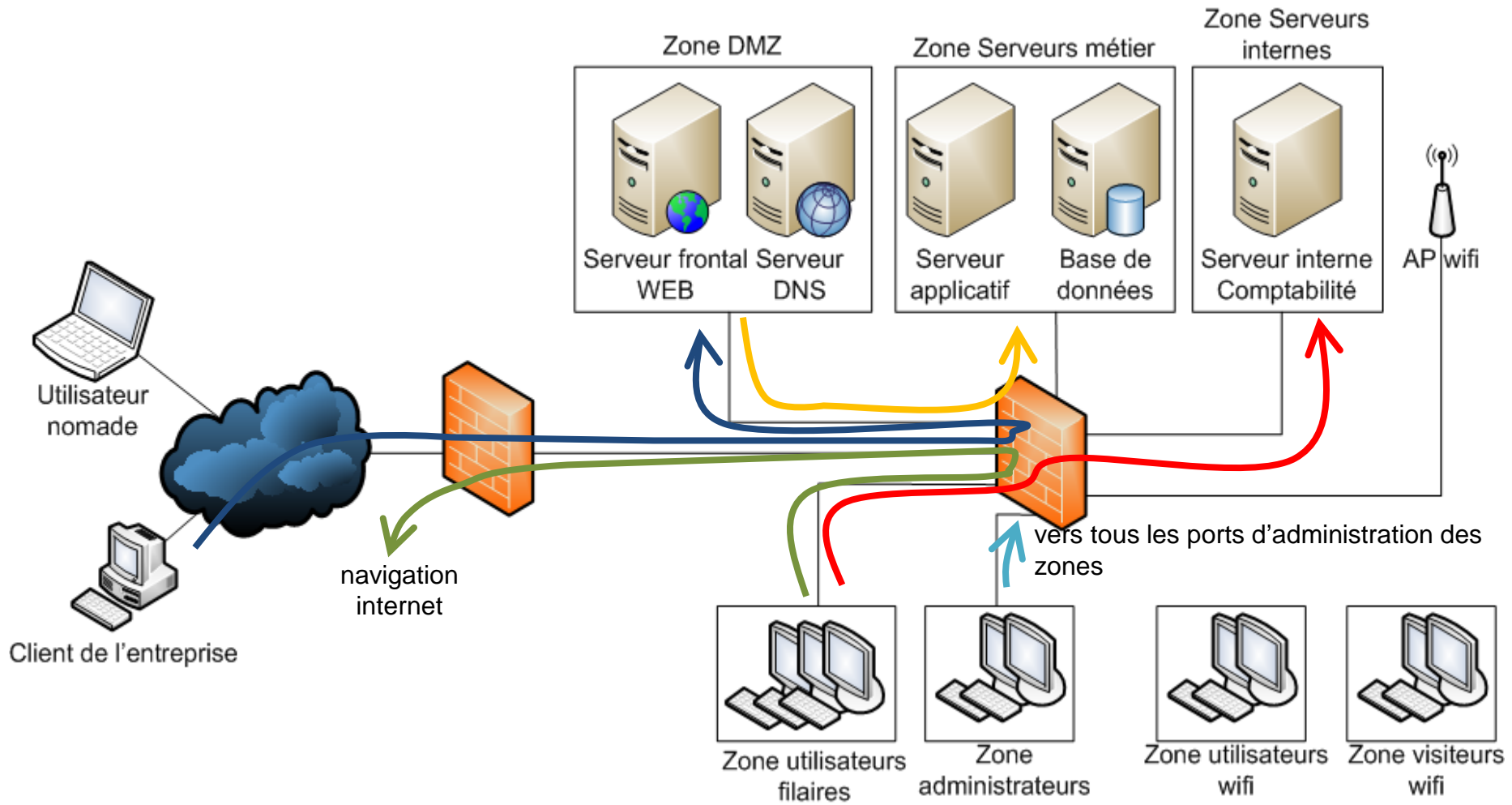
- Une **DMZ (zone démilitarisée)** destinée à héberger tous les serveurs qui doivent être accessibles depuis internet, et uniquement ceux-ci. Ainsi, en cas de faille dans le serveur web, un attaquant aurait plus de difficultés pour rebondir sur le réseau interne ;
- Une zone destinée aux **serveurs internes** 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.

2. Sécurisation d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple

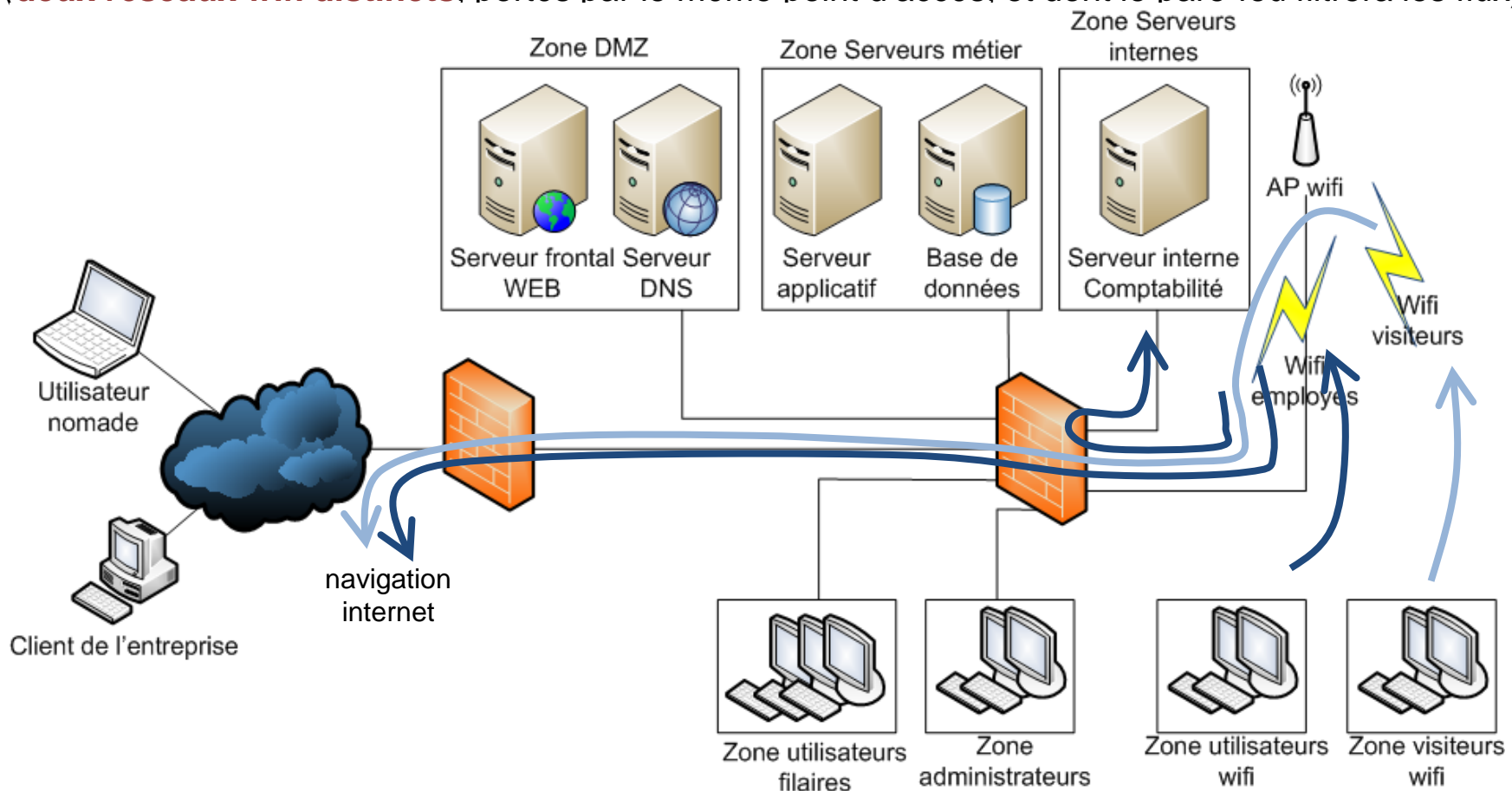


Réseau avec des zones segmentées, et un filtrage systématique via le pare-feu, y compris pour les flux internes.

2. Sécurisation 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 implémenter 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.

2. Sécurisation d'un réseau

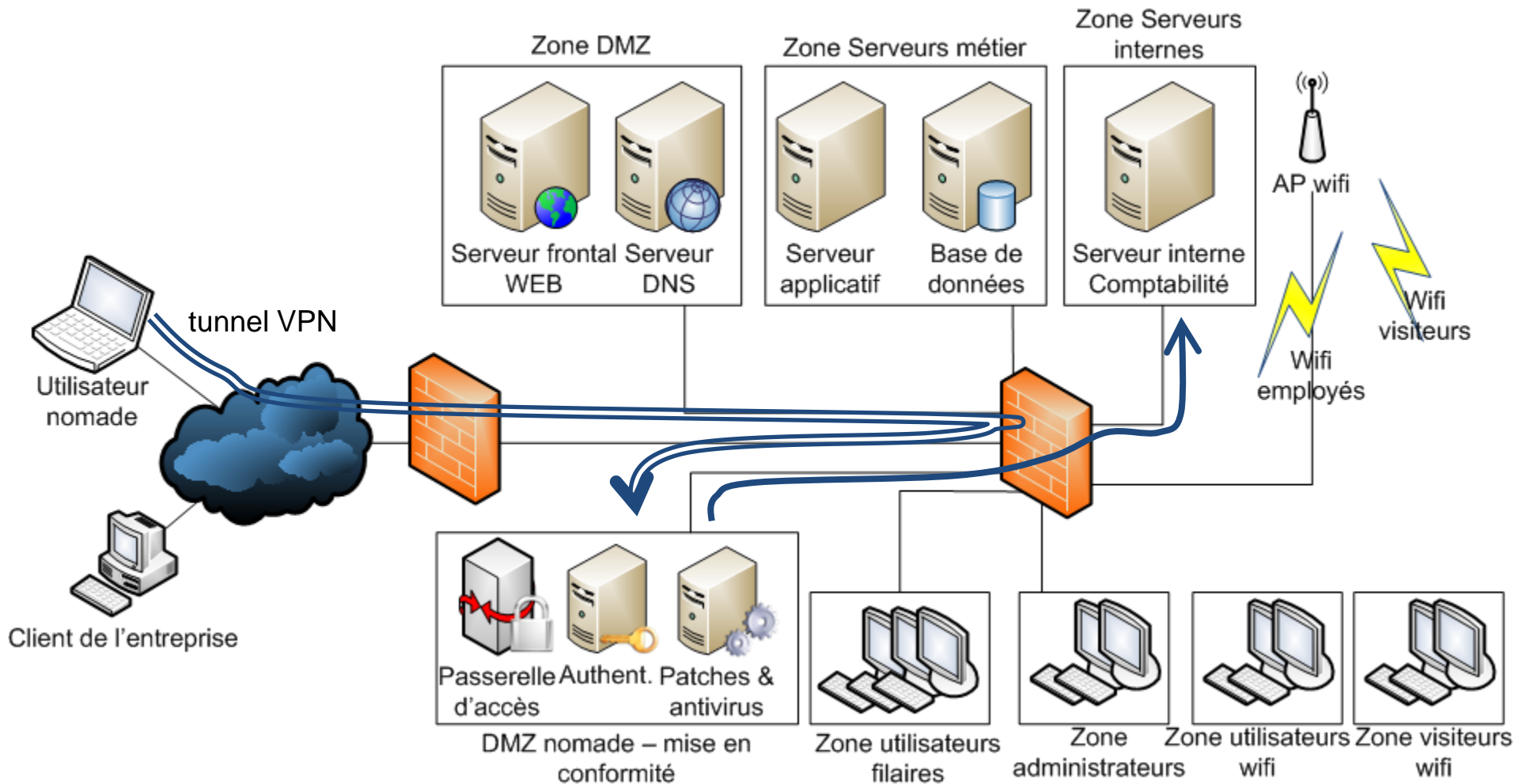
g. Exemple pratique de sécurisation avec un réseau simple

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**.

2. Sécurisation d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple



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

2. Sécurisation 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 :

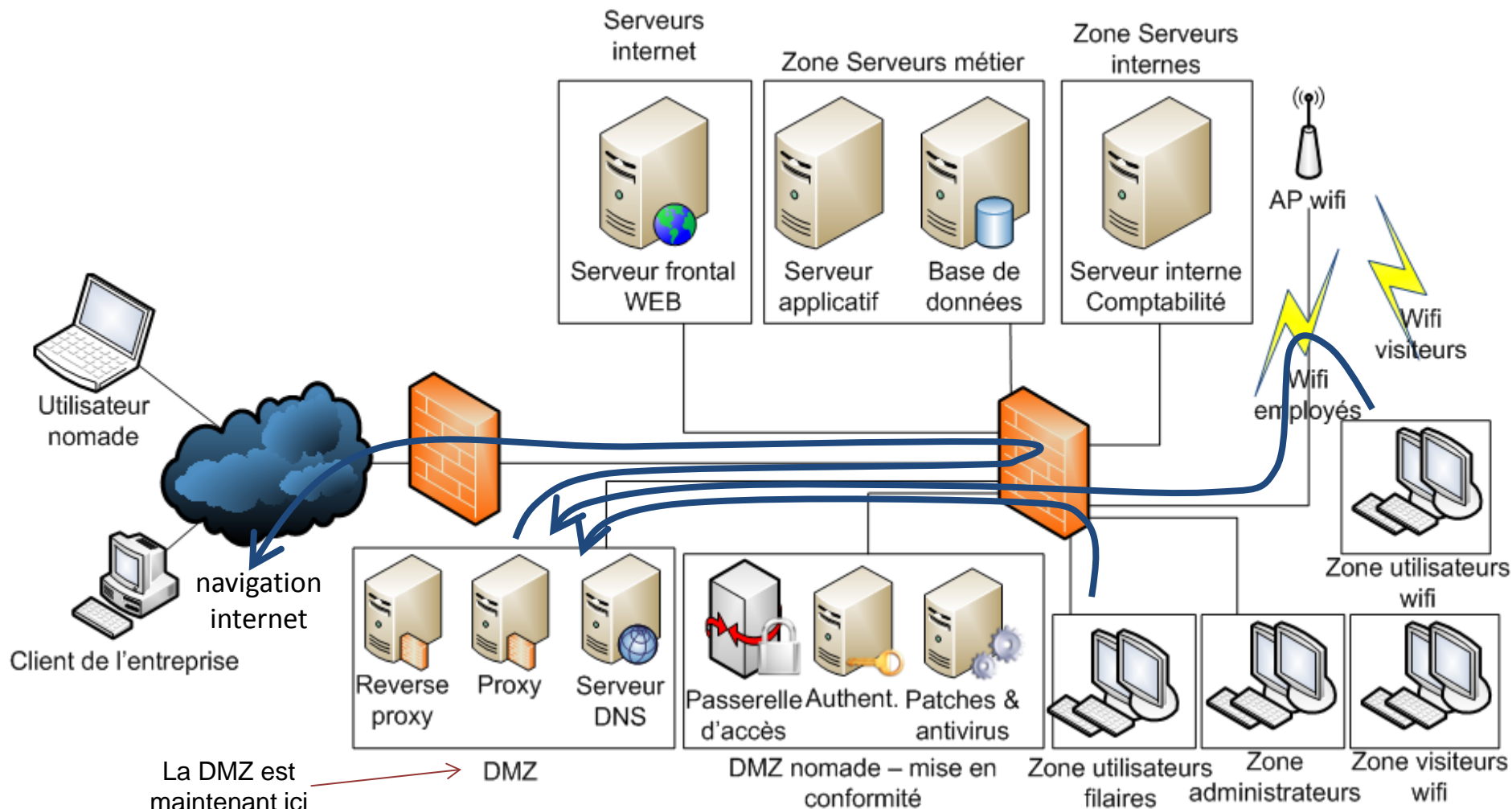
- **Trafic sortant** : définir les catégories de sites WEB que les employés sont autorisés à naviguer, implémenter une liste blanche ou noire de sites autorisés/interdits ;
- **Trafic entrant** : analyser les requêtes WEB d'internet vers le serveur de e-commerce afin d'intercepter les requêtes malveillantes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

2. Sécurisation d'un réseau

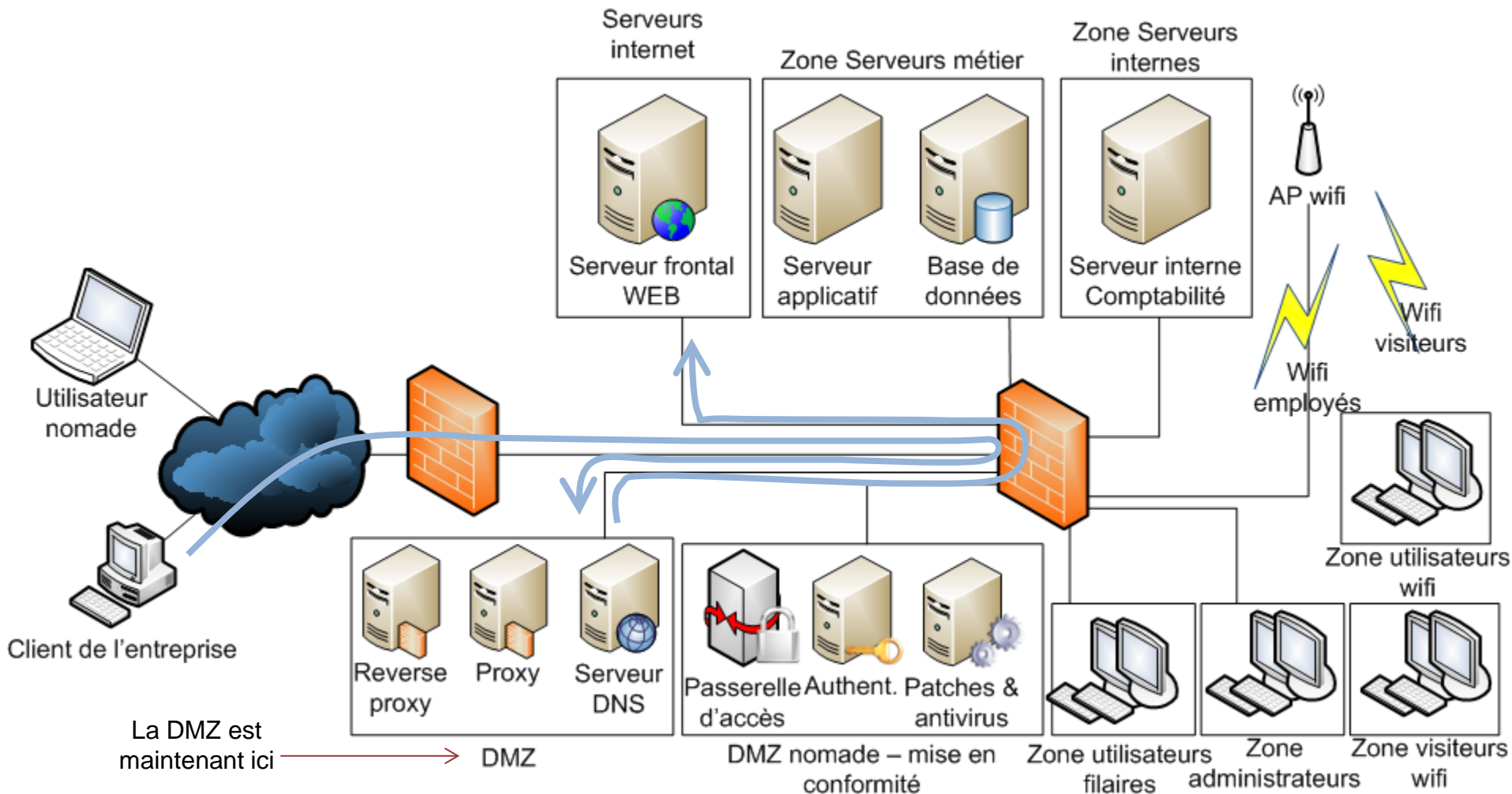
g. Exemple pratique de sécurisation avec un réseau simple



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

2. Sécurisation d'un réseau

g. Exemple pratique de sécurisation avec un réseau simple



La DMZ est maintenant ici →

DMZ

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

3. Les bases de la cryptographie

- a) Vocabulaire
- b) Un peu d'histoire
- c) Chiffrement symétrique
- d) Chiffrement asymétrique
- e) Chiffrement symétrique vs Chiffrement asymétrique
- f) Signature électronique
- g) Certificats électroniques
- h) Jetons cryptographiques

3. Les bases de la cryptographie

a. Vocabulaire

La cryptographie est une discipline consistant à manipuler des données de telle façon que les services suivants puissent être fournis :

Intégrité

Objectif : s'assurer que les données n'ont pas été modifiées sans autorisation.

Remarque : dans les faits, la cryptographie ne s'attache pas vraiment à empêcher une modification de données, mais plutôt à fournir un moyen sûr de détecter une modification malveillante.

Confidentialité

Objectif : ne permettre l'accès aux données qu'aux seules personnes autorisées.

Preuve (authentification et non-répudiation)

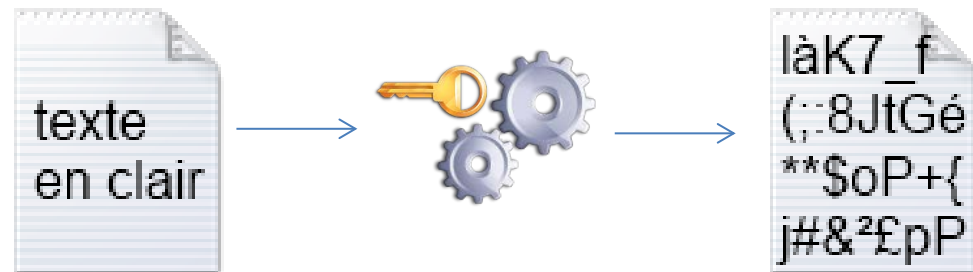
Objectif : fournir un moyen de preuve garantissant la véritable identité des entités ainsi que l'imputation de leurs actions.

3. Les bases de la cryptographie

a. Vocabulaire

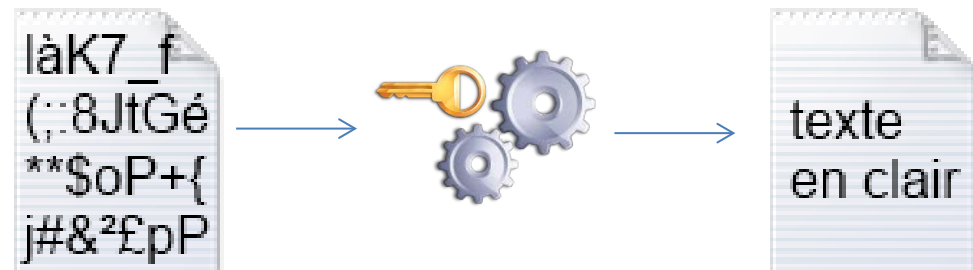
Chiffrer

Transformer une donnée de telle façon qu'elle devienne incompréhensible. Seules les entités autorisées pourront comprendre cette donnée chiffrée.



Déchiffrer

Transformer une donnée précédemment chiffrée pour reconstituer la donnée d'origine. Seules les entités autorisées ont la capacité de procéder à cette action.



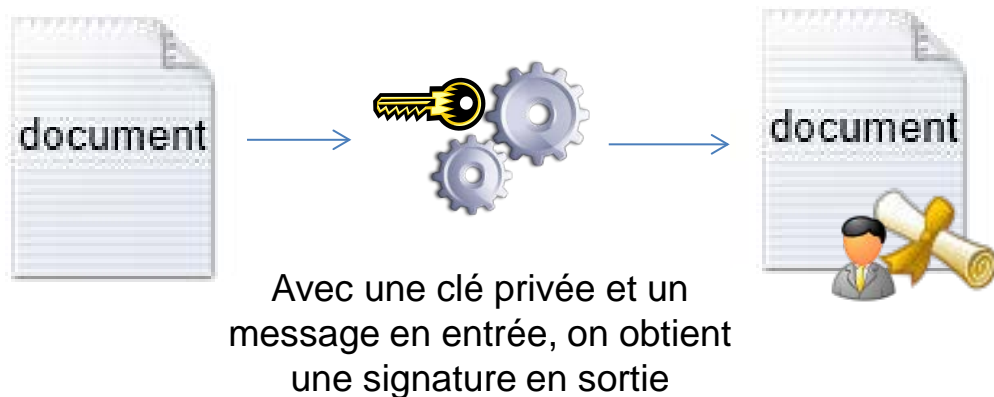
Recours à un algorithme
et à une clé
cryptographique.

3. Les bases de la cryptographie

a. Vocabulaire

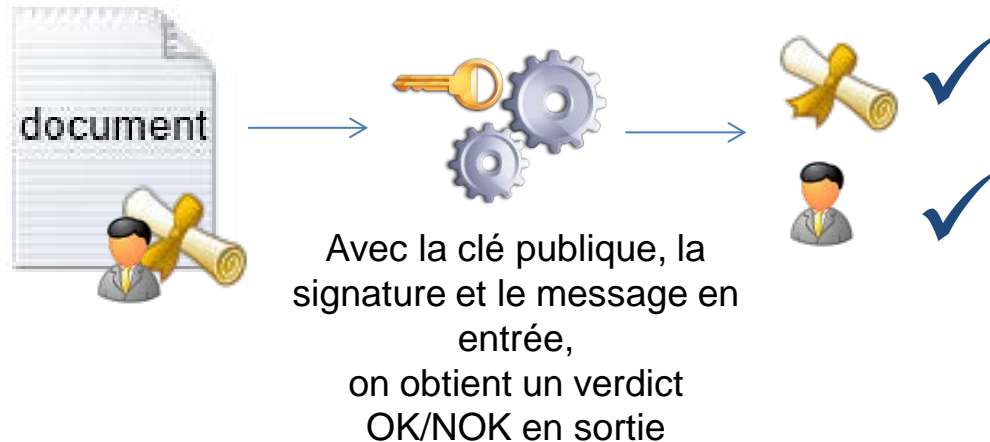
Signer

Créer une signature électronique unique à la donnée et à son auteur. La signature lie donc la donnée d'origine et son auteur.



Vérifier la signature

S'assurer que la donnée d'origine n'a pas été modifiée et que son auteur est authentifié. Si la signature n'est pas valide, alors il ne faut pas faire confiance au document.

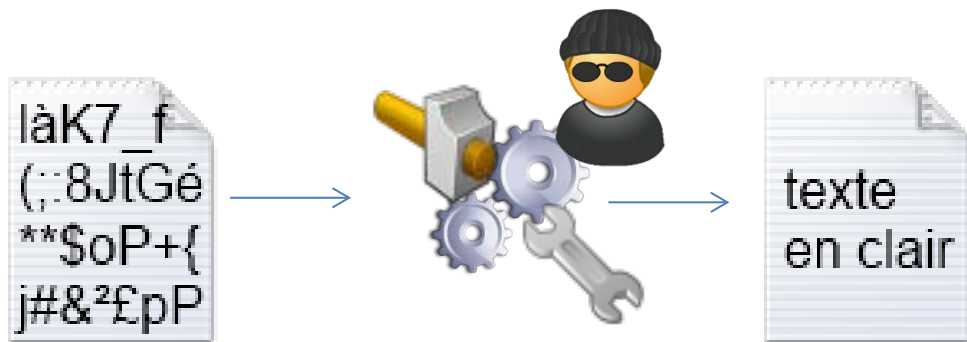


3. Les bases de la cryptographie

a. Vocabulaire

Décrypter

Reconstituer la donnée d'origine en tentant de « casser » la donnée chiffrée ou l'algorithme cryptographique.



Crypter

La notion de crypter n'existe pas. Il s'agit d'un abus de langage.

3. Les bases de la cryptographie

b. Un peu d'histoire : « Chiffrement de César »

Exemple d'algorithme cryptographiques historique. Les algorithmes sont maintenant basés sur des fonctions mathématiques.

Chiffrement de César

Méthode : il s'agit ici de « décaler » chaque caractère par un nombre déterminé.

Exemple : clé = 3

A B C D E F G H U J I K ...
A B C D E F G H U J I K ...

Exercice :

Donnée chiffrée : FBEHUHGX

Quelle est la donnée en clair ?

3. Les bases de la cryptographie

b. Un peu d'histoire : « Machine Enigma »

Présentation des machines Enigma

- Machine initiale conçue au début du XX^e siècle. Elle a bénéficié de plusieurs évolutions et versions, et a été utilisée par les Allemands pendant la seconde guerre mondiale ;
- Les machines Enigma ressemblent à des machines à écrire, avec un clavier destiné à un opérateur, un tableau de sortie (panneau lumineux), plusieurs rotors, un réflecteur et un tableau de connexion ;
- La méthode de chiffrement est basée sur de la substitution :
 - L'opérateur tape le message en clair. Chaque lettre du message en clair est remplacée par une autre lettre dans le message chiffré (les lettres chiffrées s'allument sur le tableau de sortie au fur et à mesure de la frappe en clair de l'opérateur) ;
 - L'utilisation des rotors a pour conséquence qu'une lettre en clair sera être substituée par des lettres différentes tout au long du message chiffré.



source image : <http://museeradiomili.com/cryptographie/>

3. Les bases de la cryptographie

b. Un peu d'histoire : « Machine Enigma »

Les fonctions d'une machine Enigma

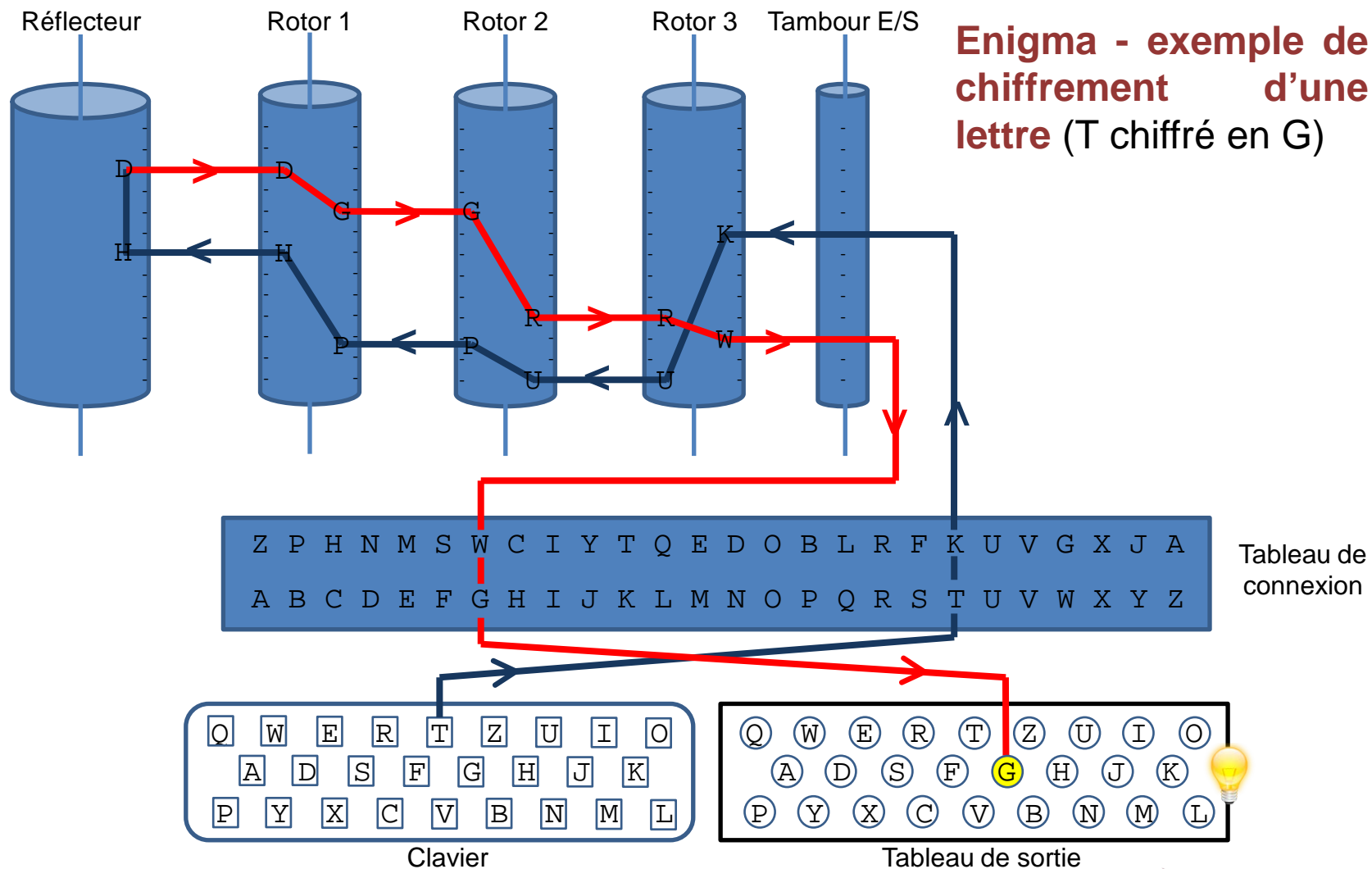
- Tableau de connexion
 - Se situe avant l'entrée sur le brouilleur ;
 - Effectue des permutations simples.
- De 3 à 6 rotors (selon le modèle)
 - Permutations aléatoires des lettres de l'alphabet ;
 - Le rotor tourne à chaque lettre tapée ;
 - Lorsque le premier rotor a fait un tour (26 positions), le second rotor tourne d'un cran, et ainsi de suite.
- Le réflecteur
 - Dernière permutation 2 à 2 des lettres avant de les faire retraverser les rotors et le tableau de connexion.



source images : https://interstices.info/jcms/jalios_5127/accueil

3. Les bases de la cryptographie

b. Un peu d'histoire : « Machine Enigma » »



3. Les bases de la cryptographie

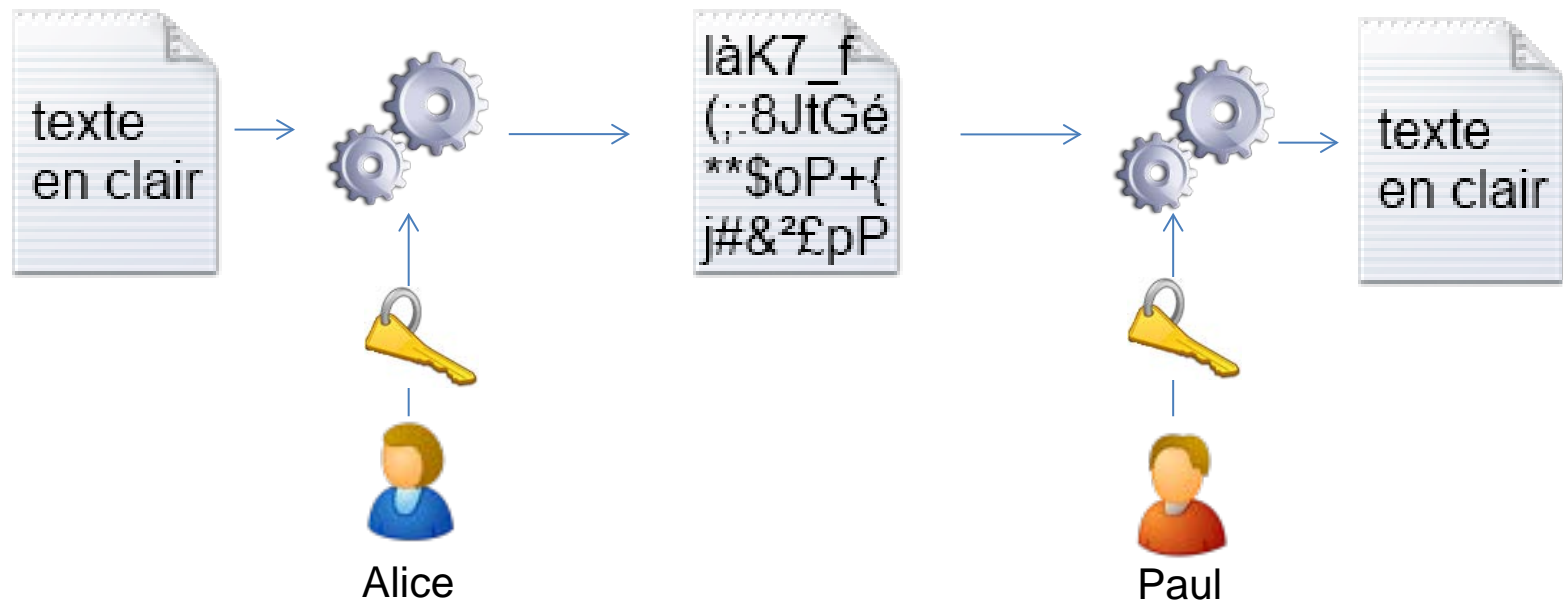
c. Chiffrement symétrique

- La clé utilisée pour le chiffrement est la **même** que celle utilisée pour le déchiffrement ;
- Cette clé doit être **secrète** : seules les personnes habilitées doivent posséder cette clé, sinon la confidentialité du message n'est plus assurée !

3. Les bases de la cryptographie

c. Chiffrement symétrique

- Exemple : Alice souhaite envoyer un message confidentiel à Paul



Clé secrète partagée entre Alice et Paul

3. Les bases de la cryptographie

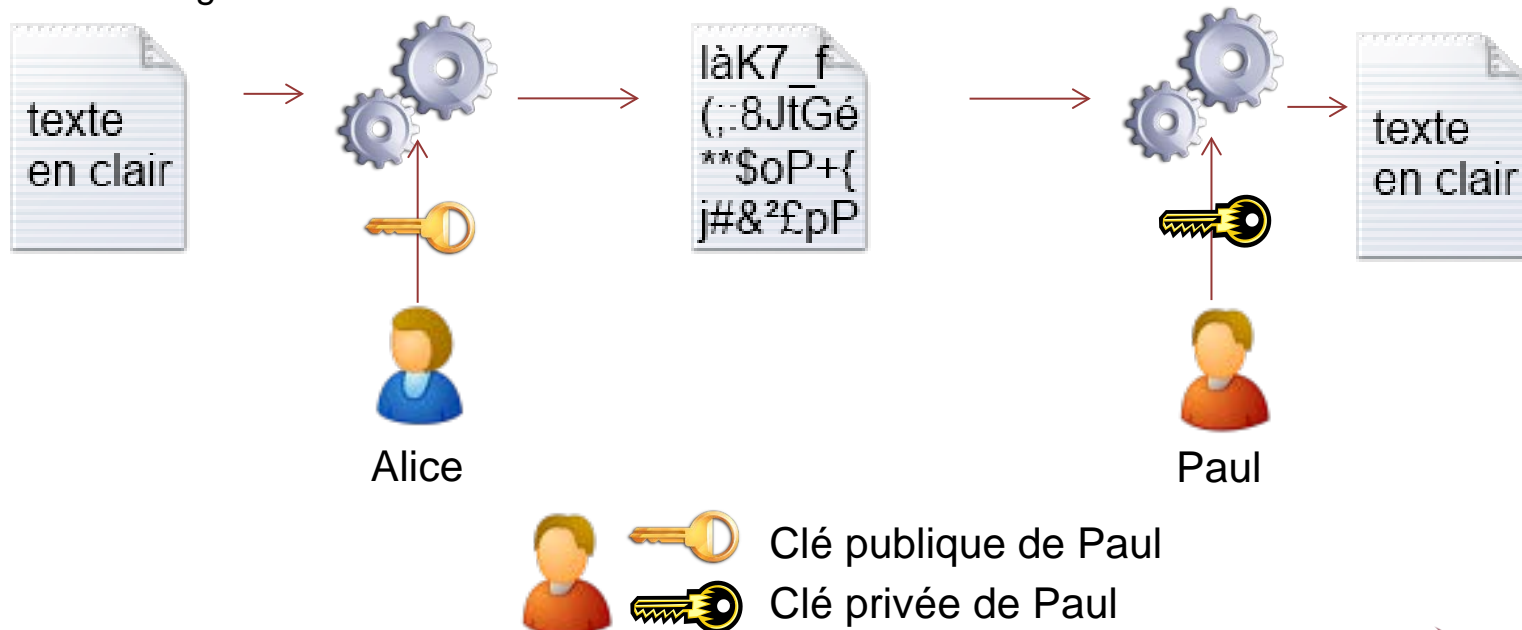
d. Chiffrement asymétrique

- La clé utilisée pour le chiffrement est **différente** de celle utilisée pour le déchiffrement. Il est nécessaire d'utiliser 2 clés :
 - Clé publique : comme son nom l'indique, cette clé est publique et peut être donnée à tout le monde ;
 - Clé privée : cette clé doit être personnelle et connue de son seul propriétaire. Elle ne doit jamais être divulguée !
- Ces deux clés sont mathématiquement liées
 - La connaissance de la clé publique ne permet pas de calculer de manière efficace la clé privée (attention à la taille de la clé, qui doit être suffisamment longue) ;
 - Chaque personne doit donc posséder 2 clés : une clé privée (confidentielle) et une clé publique qu'il peut divulguer à tout le monde.

3. Les bases de la cryptographie

d. Chiffrement asymétrique

- Exemple : Alice souhaite envoyer un message confidentiel à Paul
 - Alice chiffre le message avec la clé publique de Paul ;
 - Paul déchiffre le message grâce à sa privée ;
 - Notes :
 - Alice ne pourra jamais (et n'aura jamais besoin de) utiliser la clé privée de Paul puisque celle-ci est confidentielle à Paul !
 - Alice n'a pas besoin d'utiliser ses clés personnelles dans cet exemple de chiffrement sans signature.



3. Les bases de la cryptographie

e. Chiffrement symétrique vs Chiffrement asymétrique

Chiffrement symétrique

Avantages

- Rapidité des opérations (adapté à du trafic en temps réel) ;
- Clés courtes (256 bits suffisent actuellement) ;

Inconvénients

- Difficulté d'échange sécurisé des clés secrètes : comment le faire en protégeant ce secret ?

Chiffrement asymétrique

- Facilité d'échange des clés : les seules clés qui ont besoin d'être échangées sont des clés publiques (dont il faut assurer la protection en intégrité) ;

- Lenteur des opérations (peu adapté à du trafic en temps réel) ;
- Grande taille des clés (2048 bits minimum actuellement) ;

Exemples d'algorithmes sûrs (janvier 2015)

- AES.

- RSA.

3. Les bases de la cryptographie

f. Signature électronique

Rappel de l'objectif : **s'assurer de la non-modification d'une donnée**, et **s'assurer de l'identité de son auteur**. Si la signature n'est pas valide, cela indique que l'auteur « n'est pas le bon » ou que le donnée reçue n'est pas celle que son auteur avait signé.

Notes :

- **La signature électronique n'assure pas la confidentialité des données**, mais leur intégrité et la notion de preuve ;
- **Lorsque l'on chiffre un message, il est fortement recommandé de le signer également** afin d'assurer l'intégrité du message.

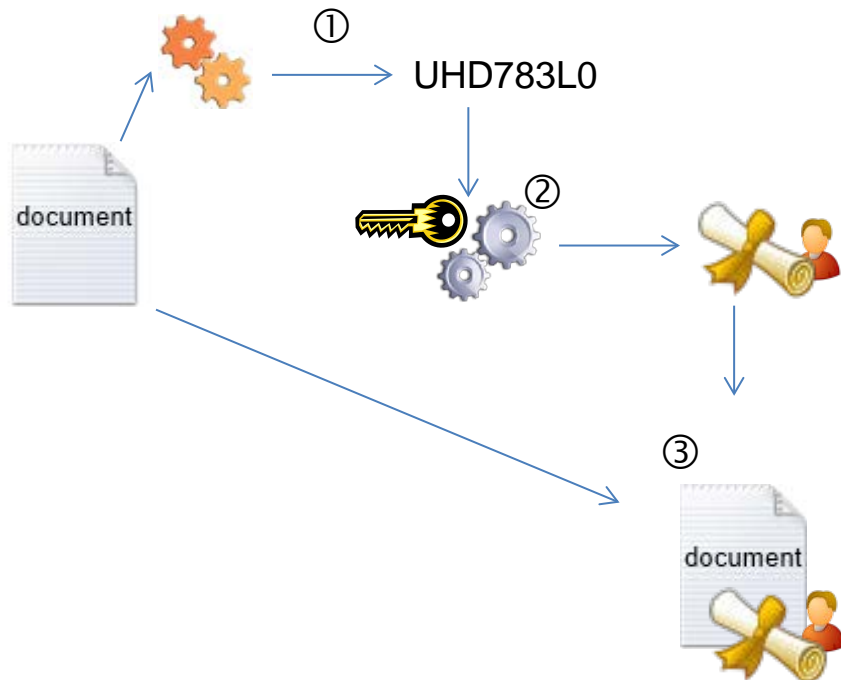
3. Les bases de la cryptographie

f. Signature électronique : principe

1. Le signataire d'un message génère – grâce à un algorithme cryptographique spécifique – une valeur unique calculée à partir du message que l'on souhaite signer : un condensat (un haché) ;
 - Les algorithmes de calcul de condensat sont publics et ne gèrent pas de secret, donc tout le monde peut les utiliser et calculer les mêmes condensats à partir d'un même message ;
 - Deux messages différents ne peuvent pas donner lieu au même condensat.
2. Le signataire utilise l'algorithme de signature, qui prend en entrée sa clé privée et le condensat précédent, pour produire une signature électronique ;
3. Le signataire envoie (ou stocke) le message et la signature électronique, permettant ainsi à un lecteur d'en prendre connaissance ;
4. Le lecteur calcule lui-même le condensat du message en clair ;
5. Le lecteur utilise l'algorithme de vérification de signature, qui prend en entrée la clé publique du signataire, le condensat et la signature, pour rendre un verdict. Si le verdict est négatif, alors il ne faut pas faire confiance au message reçu (celui-ci ne correspond pas — pour une raison que l'on ignore — au message du signataire).

3. Les bases de la cryptographie

f. Signature électronique : illustration



Etapes de la signature :

- ① Le signataire génère le condensat unique associé au message ;
- ② Le signataire utilise l'algorithme de signature, qui prend en entrée sa clé privée et le condensat précédent, pour produire une signature électronique ;
- ③ Le signataire envoie (ou stocke) le message et la signature électronique, permettant ainsi à un lecteur d'en prendre connaissance ;

La vérification par le destinataire/lecteur est décrite sur la diapositive suivante.



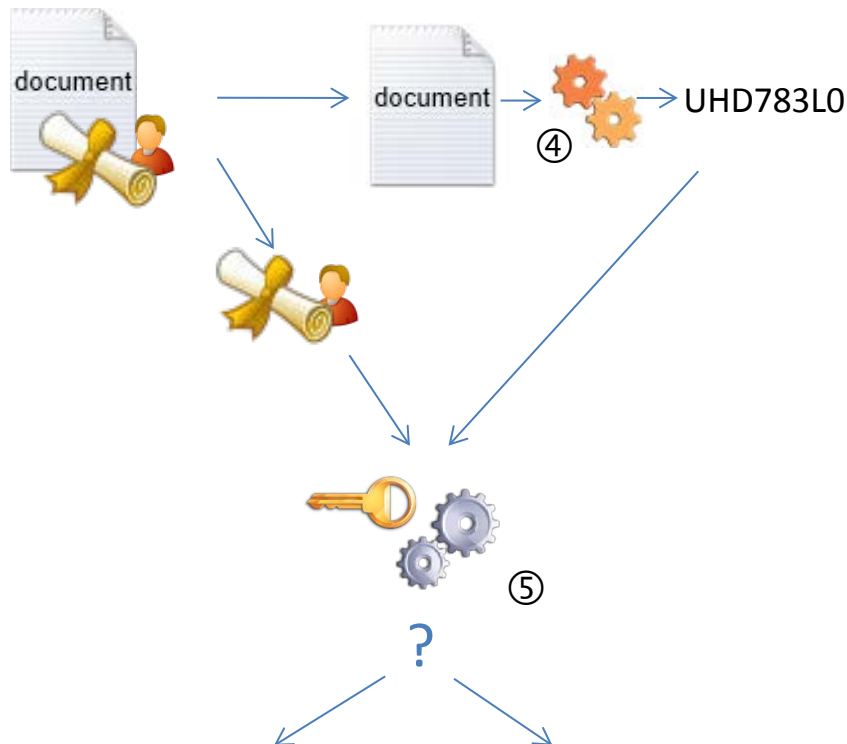
Clé publique du signataire



Clé privée du signataire

3. Les bases de la cryptographie

f. Signature électronique : illustration



✓ La signature est valide. Le message est intègre.

✗ La signature est invalide. Le message n'est pas intègre.

Etapes de la vérification de la signature par un lecteur/destinataire :

④ Le lecteur calcule le condensat du message en clair ;

⑤ Le lecteur utilise l'algorithme de vérification de signature, qui prend en entrée la clé publique du signataire, le condensat et la signature, pour rendre un verdict. Si le verdict est négatif, alors il ne faut pas faire confiance au message reçu (celui-ci ne correspond pas — pour une raison que l'on ignore — au message du signataire).



Clé publique du signataire



Clé privée du signataire

3. Les bases de la cryptographie

g. Certificats électroniques

Un aspect important n'a pas été traité jusqu'à maintenant :



Clé publique de Paul

Clé privée de Paul

Les interlocuteurs de Paul ont besoin d'utiliser sa clé publique. Comment peuvent-ils **être certains que la « clé publique de Paul » appartient effectivement à Paul** et qu'elle n'a pas été générée frauduleusement en son nom ? Autre exemple, comment les visiteurs d'un site web bancaire peuvent **être certains que le site web est légitime** et qu'il ne s'agit pas d'un site frauduleux imitant celui d'une banque ?

- Solution : utilisation de certificats électroniques.

3. Les bases de la cryptographie

g. Certificats électroniques

Un certificat est un **fichier électronique** qui comprend notamment :

- La **clé publique** d'un individu (ou d'une entité ou d'un nom de domaine) ;
- Les détails de cet individu (ou de cette entité) : nom, prénom, nom de domaine, etc. ;
- La **signature par un tiers de confiance**, chargé de garantir que le propriétaire de la clé publique a été vérifié et – par conséquent – l'authenticité de la clé publique vis-à-vis de son propriétaire. La signature porte sur l'identité du détenteur et la clé publique afin d'assurer l'intégrité de l'ensemble ;
- D'autres informations telles que l'usage de la clé, les dates de validité, des informations concernant la révocation, etc.

Le tiers de confiance, une autorité de certification, en charge de :

- **Vérifier l'identité** de la personne demandant à créer le certificat ;
- **Créer le certificat** après vérification, **puis le signer** (avec la clé privée de l'autorité de certification) ;
- **Tenir à jour une liste des certificats qui ont été révoqués** (par exemple si la clé a été compromise).

3. Les bases de la cryptographie

g. Certificats électroniques

Comment connaître les autorités de certification ?

- Elles sont directement intégrées par les éditeurs dans les systèmes d'exploitation et/ou les navigateurs ;
- L'utilisateur est également libre de rajouter l'autorité de certification de son choix si il choisit de faire confiance à des certificats signés par une autorité non-intégrée dans son navigateur.

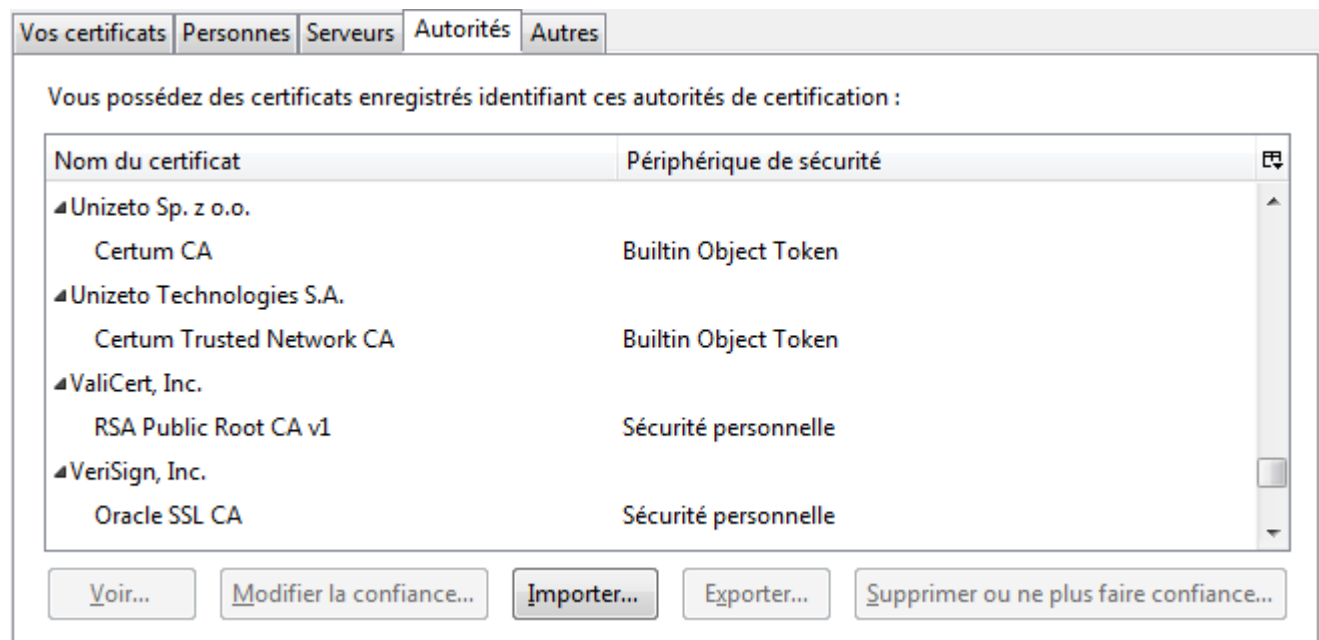


Image : magasin de certificats de Firefox

3. Les bases de la cryptographie

g. Certificats électroniques

Exemple d'un certificat pour le site web www.france-universite-numerique-mooc.fr

Général

Détails

Ce certificat a été vérifié pour les utilisations suivantes :

Certificat client SSL

Certificat serveur SSL

Émis pour	
Nom commun (CN)	www.france-universite-numerique-mooc.fr
Organisation (O)	<Ne fait pas partie du certificat>
Unité d'organisation (OU)	Domain Control Validated
Numéro de série	00:EE:CE:37:A0:F9:50:16:57:BC:0A:C2:4B:A8:9F:0E:41
Émis par	
Nom commun (CN)	TERENA SSL CA
Organisation (O)	TERENA
Unité d'organisation (OU)	<Ne fait pas partie du certificat>
Période de validité	
Début le	08/10/2013
Expire le	08/10/2016
Empreintes numériques	
Empreinte numérique SHA-256	6E:D0:7E:51:A4:2A:86:97:A0:A8:C0:70:9C:32:E8:8B:16:B3:89:22:A2:C5:AE:5A:FE:35:99:0E:B3:79:10:EB
Empreinte numérique SHA1	86:22:B9:4F:FB:7B:9F:45:DF:B0:89:C0:A6:C0:83:DF:F6:2E:0B:9A

Les détails techniques du certificat, la clé et la signature se trouvent dans **Détails**

Détenteur de la clé publique

Autorité de certification

Dates de validité du certificat

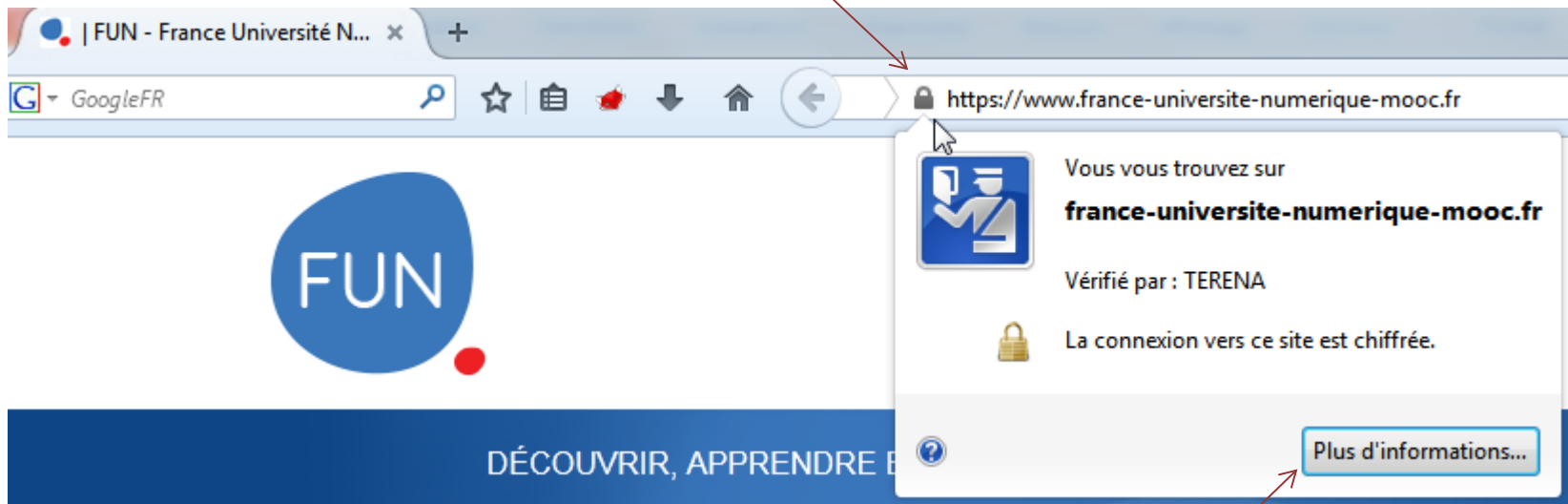
3. Les bases de la cryptographie

g. Certificats électroniques

Où trouver les certificats dans un navigateur ?

Exemple avec Firefox pour ouvrir le certificat d'un site WEB

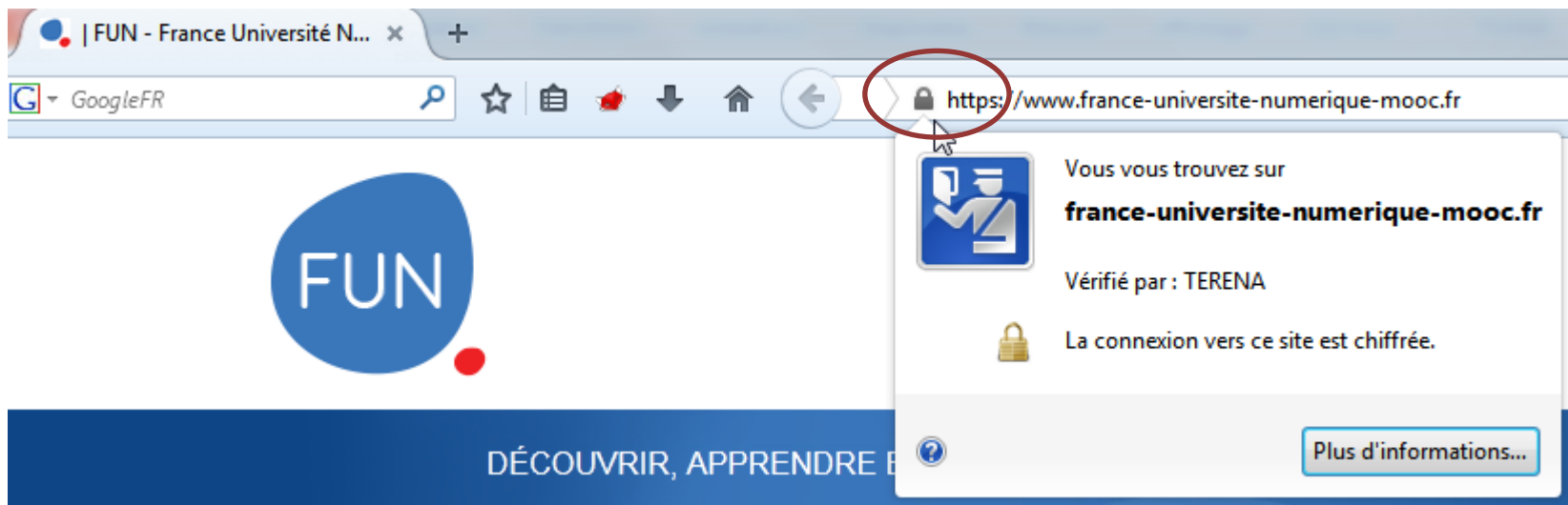
Cliquer sur le cadenas à côté de l'URL



Cliquer ici pour afficher le certificat

3. Les bases de la cryptographie

g. Certificats électroniques



Puisque le certificat du site WEB est disponible et valide, cela amène donc deux avantages à l'utilisateur, caractéristiques du **HTTPS**

- Nous sommes confiants que **le site WEB est légitime** (i.e. le certificat a été vérifié et signé par une autorité de certification de confiance) ;
- Puisque le certificat contient la clé publique du site WEB, nous pouvons donc **chiffrer nos connexions vers ce site** (méthode : chiffrement avec la clé publique du destinataire comme nous l'avons vu au préalable dans ce cours).

3. Les bases de la cryptographie

h. Jetons cryptographiques (tokens)

- Les jetons sont utilisés pour **stocker des clés privées** (cryptographie asymétrique) ou **secrètes** (cryptographie symétrique) ;
- Puisqu'un jeton contient une information sensible (une clé privée ou secrète), il faut donc **protéger ce jeton** pour que seules les personnes habilitées puissent l'utiliser ;
- Exemples de jetons et leurs moyens de protection (ainsi que leur niveau de sécurité) :



- **Fichier sur disque**, associé à un mot de passe connu de l'utilisateur seulement (exemple avec l'application libre GPG) ;



- **Jeton USB**, associé à un mot de passe (exemple de nombreux produits commerciaux qui utilisent un jeton physique pour authentifier un utilisateur sur un poste de travail) ;



- **Carte à puce**, associée à un mot de passe simple (exemple des cartes bancaires avec un code PIN permettant d'authentifier le propriétaire de la carte avant d'autoriser la transaction).
 - Afin d'éviter qu'une personne malveillante ne découvre facilement le mot de passe simple, on impose un verrouillage de la carte à puce après 3 tentatives infructueuses.

4. La sécurité des applications web

- a) Usurpation d'identité via les cookies
- b) Injection SQL

4. La 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.

4. La 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 nécessitent d'identifier ou d'authentifier l'utilisateur, en permettant notamment la mise en œuvre de sessions :

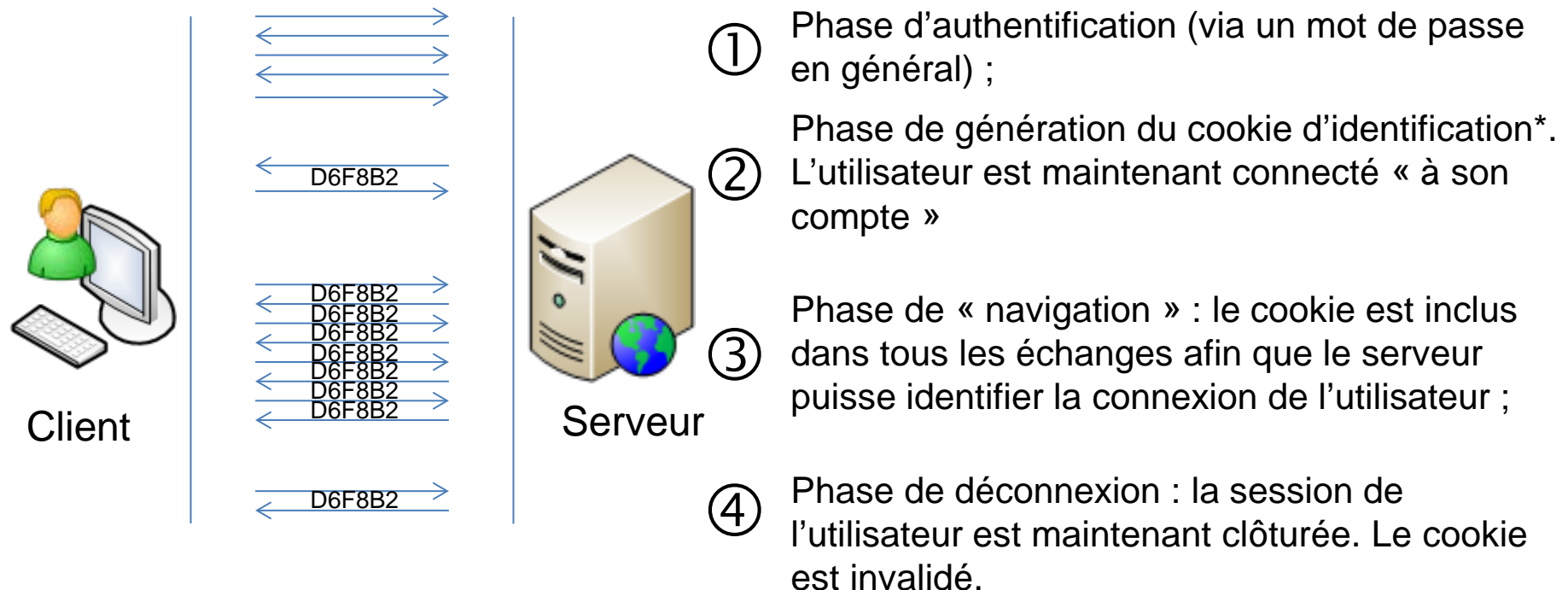
- les sites marchand (afin d'afficher le panier de l'utilisateur connecté) ;
- les sites bancaires (afin d'afficher le solde du compte de l'utilisateur connecté et non pas celui d'un autre client) ;
- les sites « en général » (afin d'afficher des publicités ciblées sur notre navigation).

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'identification.

4. La sécurité des applications web

a. Usurpation d'identité via les cookies

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

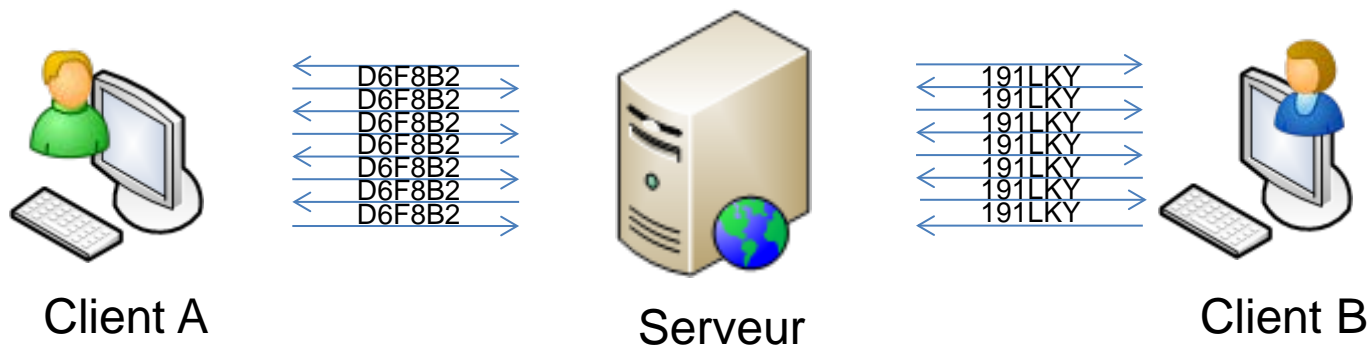


- Un cookie d'identification est en fait une chaîne de caractères aléatoire et **unique**, suffisamment longue pour qu'elle ne puisse pas être générée deux fois par erreur.
Exemple d'un cookie d'identification : D6F8B2BE3ED3040D9A3C10-D6F8B2A305D048B9

4. La 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.

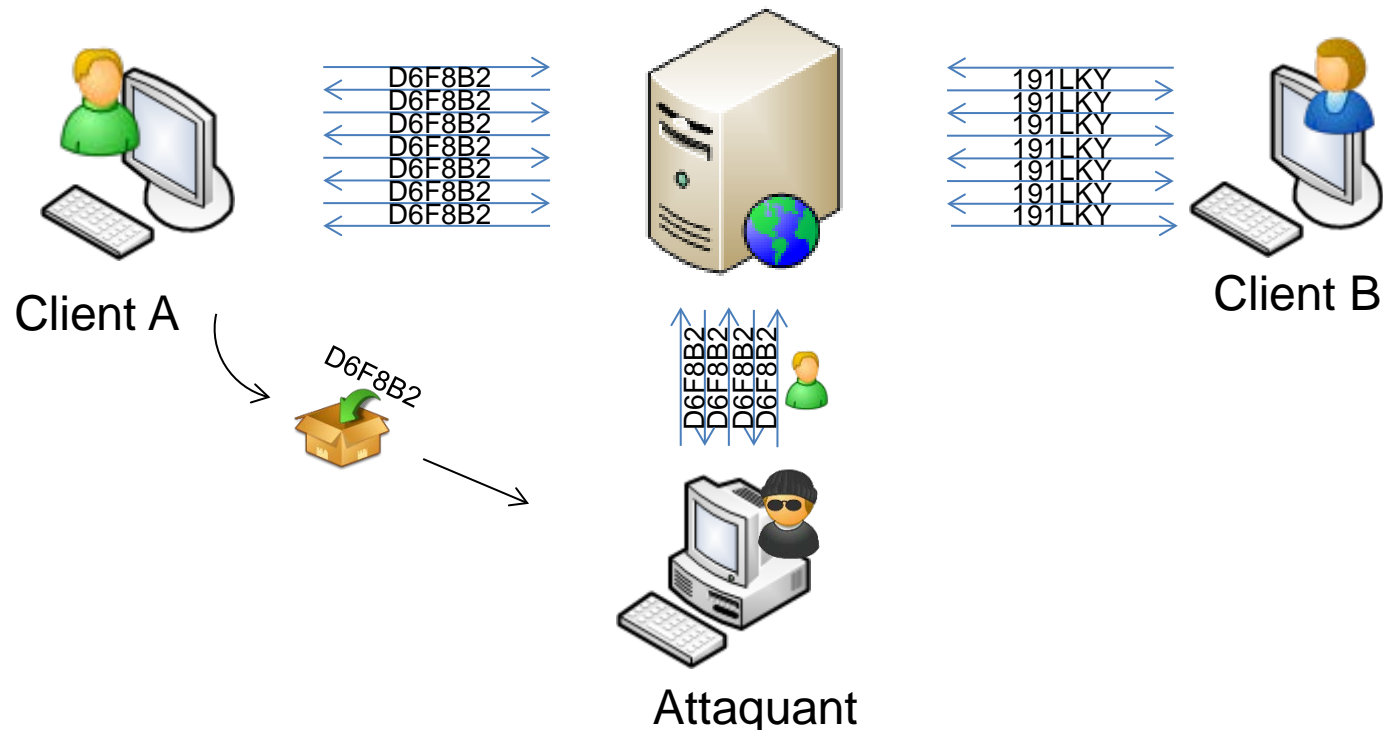


4. La sécurité des applications web

a. Usurpation d'identité via les cookies

Mais que se passe-t-il si un attaquant arrive à dérober le cookie d'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.



4. La sécurité des applications web

a. Usurpation d'identité via les cookies

L'attaquant peut dérober un cookie d'identification 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 module 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**.



4. La sécurité des applications web

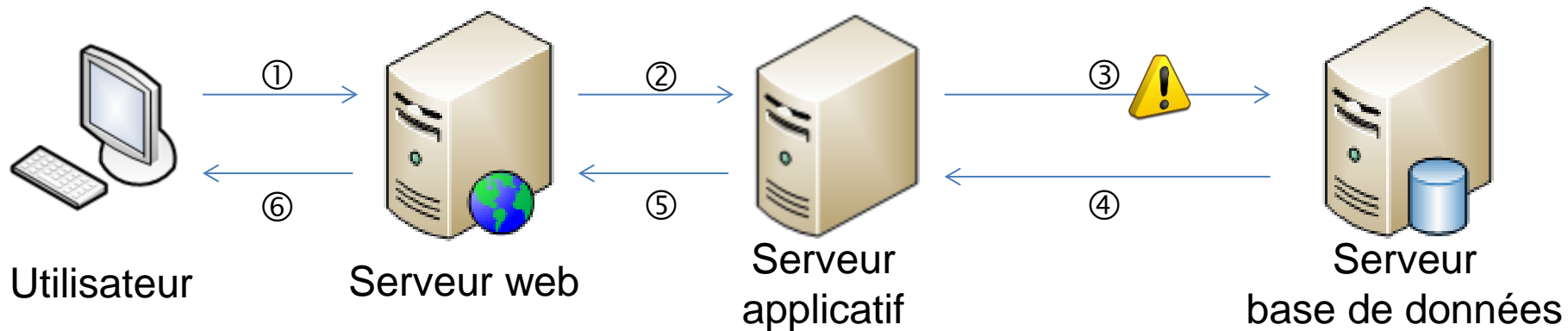
b. Injection SQL

- 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 interdite) ;
- 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.

4. La sécurité des applications web

b. Injection SQL

Architecture standard logicielle 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.

4. La 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.

4. La sécurité des applications web

b. Injection SQL

Formulaire WEB :

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

Login	Mot de passe
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 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'
```

4. La sécurité des applications web

b. Injection SQL

Formulaire WEB :

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

Login	Mot de passe
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 !

4. La 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.



CyberEdu

La sécurité par l'enseignement supérieur des NTIC

Merci de votre attention

Ce document pédagogique a été rédigé par un consortium regroupant des enseignants-chercheurs et des professionnels du secteur de la cybersécurité.



Il est mis à disposition par l'ANSSI sous licence Creative Commons Attribution 3.0 France.