CYBERSÉCURITÉ DES SYSTÈMES INDUSTRIELS

Réseaux & Applications

Loïc Rouquette

Licence

Sommaire

La Sécurité du Protocole IP

Préambule

Non prise en compte de la sécurité lors de la création du protocole IP et des protocoles associés : (TCP, UDP, ICMP, routage, etc.).

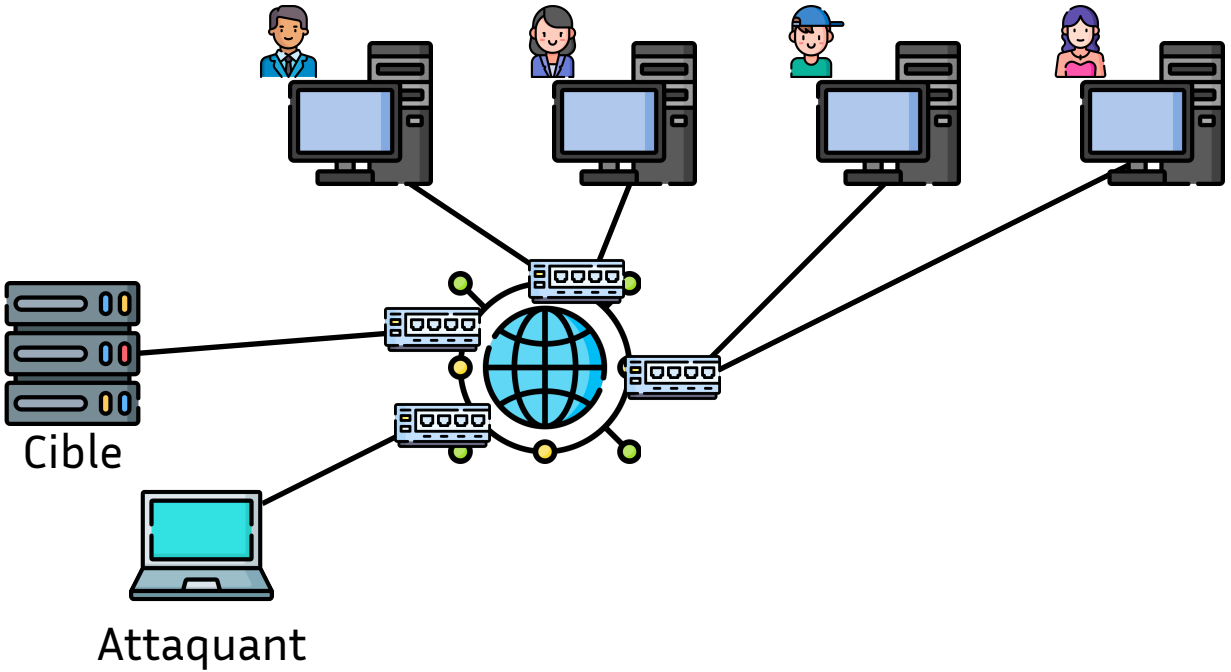
- «Concept sécurité» inconnu à l'époque.
- Conséquence : aucun mécanisme de sécurité n'est implémenté au sein de ces protocoles.

Quleques exemples de faibelesses de ces protocoles :

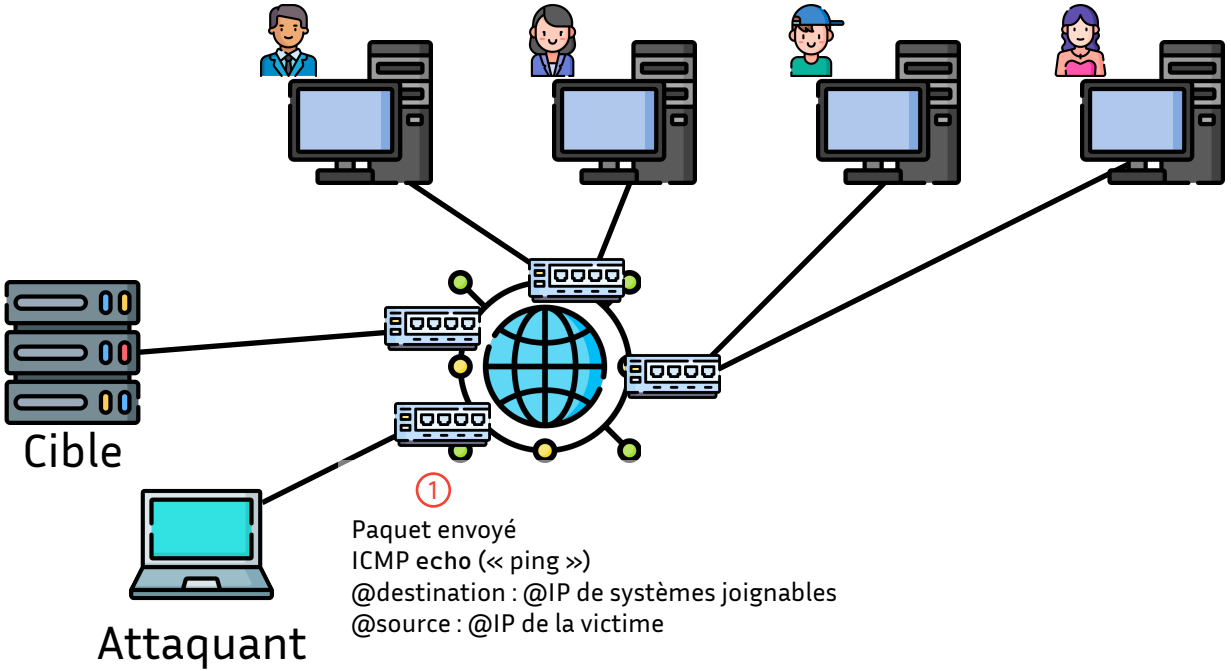
- Absence d'authentefication des émetteurs et des récepteurs d'un datagramme : usurpation d'adresse IP possible ;
- Absence de chiffrement des données : celles-ci sont donc transmises en clair ;
- Le routage des datagrammes peut être modifié de façon à rediriger let datagrammes vers un autre destinataire.

Note : l'exploitation de ces faibelsses nécessite des prérequis techniques et elles ne sont pas exloitables sur tous les réseaux.

Exemple d'attaques par réfexion



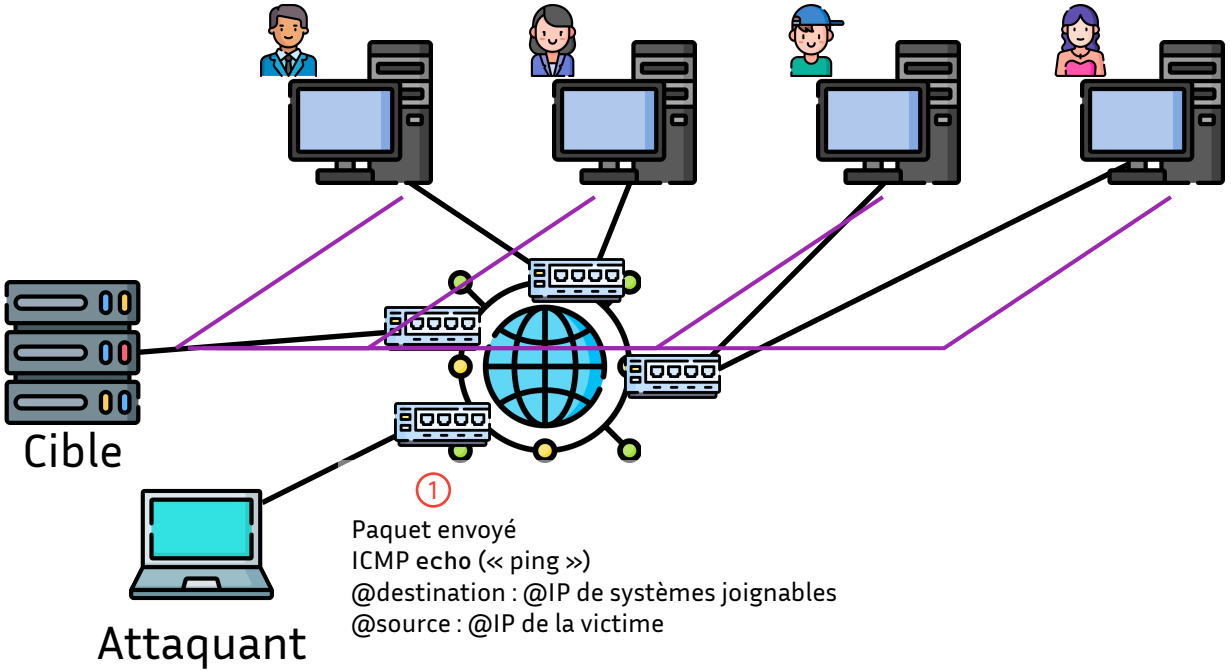
Exemple d'attaques par réfexion



① Paquet envoyé
ICMP **echo** (« ping »)
@destination : @IP de systèmes joignables
@source : @IP de la victime

Exemple d'attaques par réfexion

② Paquets envoyés
ICMP `reply`
@destination : @IP de la victime
@source : @IP de chaque émetteur



① Paquet envoyé
ICMP **echo** (« ping »)
@destination : @IP de systèmes joignables
@source : @IP de la victime

Exemple d'attaques par réfexion

But de l'attaque

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

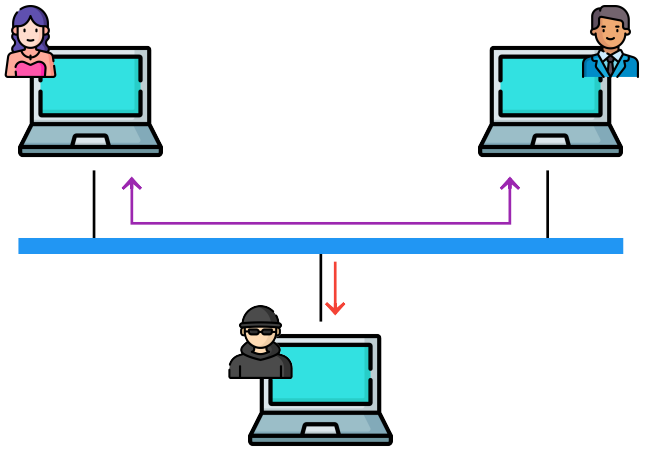
Quelles sont les caractéristiques de l'attaque ?

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

Séquences de l'attaque

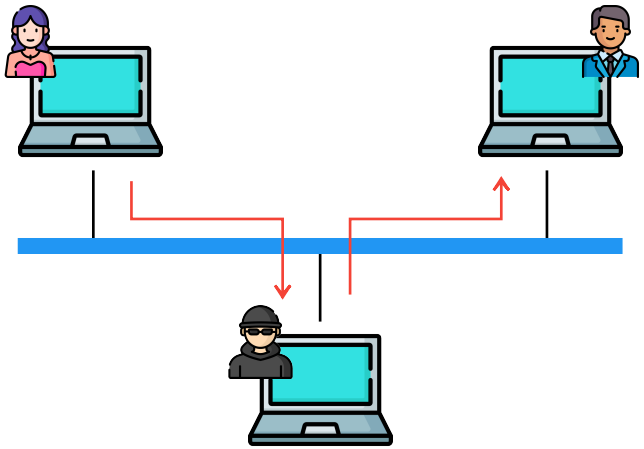
1. Un attaquant envoie des paquets PING à des systèmes tiers joignables en indiquant l'@IP de la future victime comme @IP source ;
2. Chaque système pense ainsi recevoir un PING de la part d'un système distant, et chacun va répondre à ce PING ;
3. Avec suffisamment de ressources, l'attaquant sera en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

Exemples d'écoute de trafic



Écoute passive

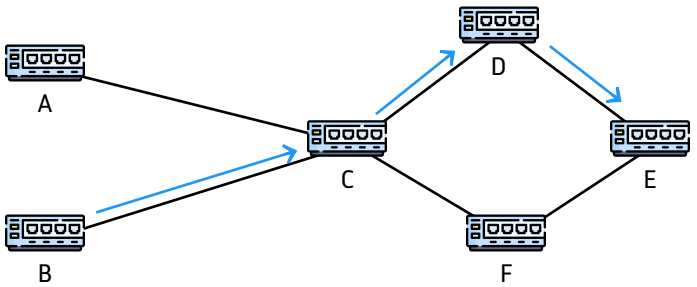
L'attaquant est en mesure d'écouter les conversations entre A et B (atteinte à la **confidentialité** des échanges).



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).

Exemple de modification de 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 maitrisé par l'attaquant.

Méthode :

Sécurisation de protocole IP

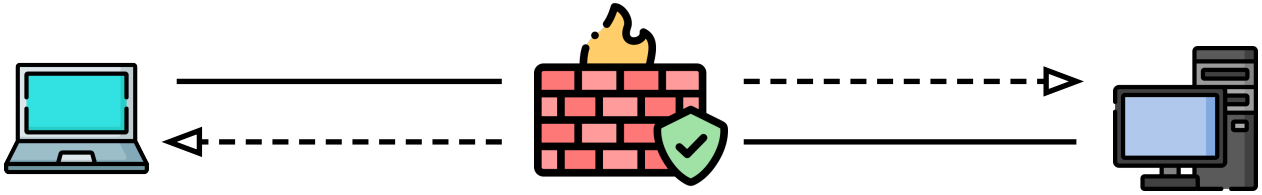
Ainsi, il est nécessaire de **mettre en œuvre des mécanismes de sécurité complémentaires** afin de réduire et maitriser les risques émanant des protocoles historiques régissant les réseaux.

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 ;

Pare-feu

- Équipement en coupure entre 2 ou plusieurs réseaux ;
- Inspecte les paquets réseaux entrantr 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.



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.

Répartiteur de charge (load-balancer)

- 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.

Anti-virus

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

- Virus, vers, keylogger, 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 ;

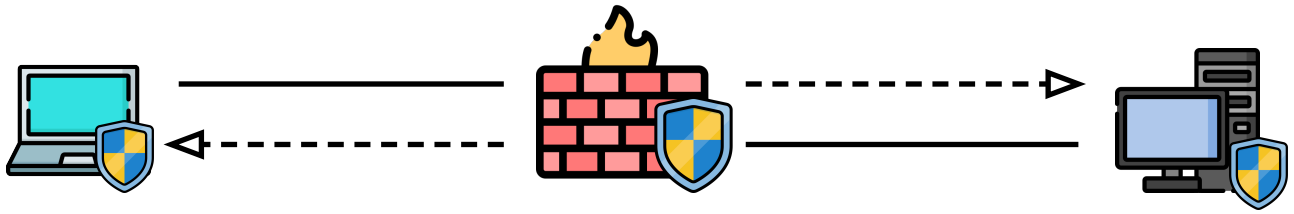
Limite des anti-virus

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.

Anti-virus

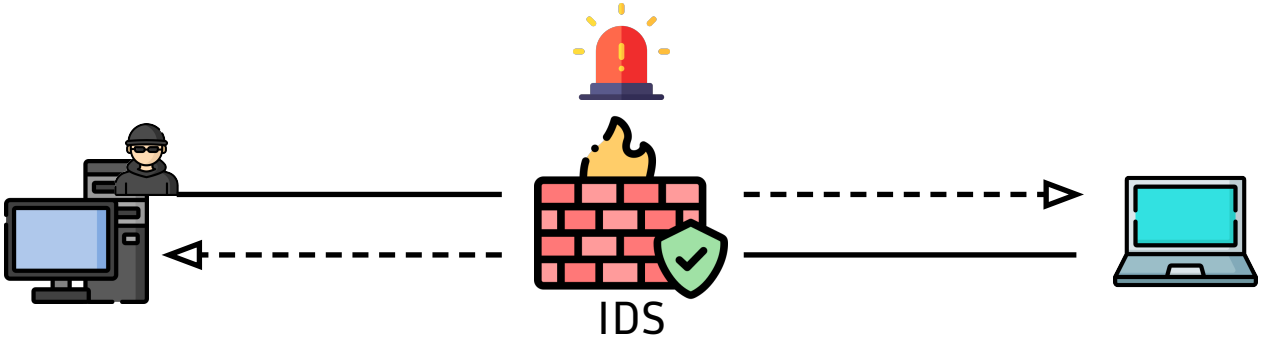
Déploiement

- 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 *Intrusion Detection System* (IDS), mécanisme présenté dans la section suivante.



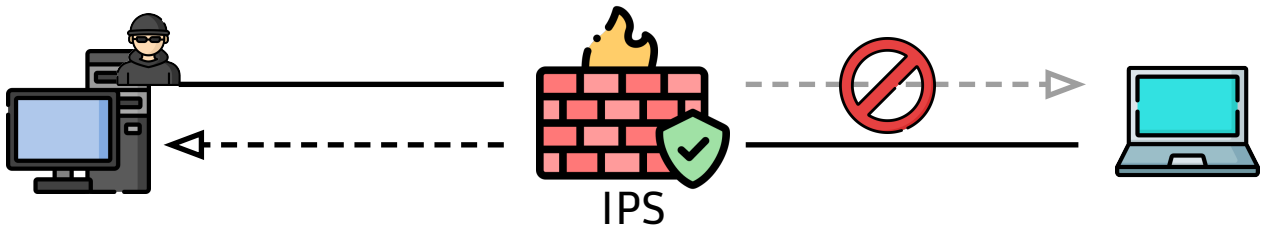
Intrusion Detection System (IDS)

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



Intrusion Protection System (IPS)

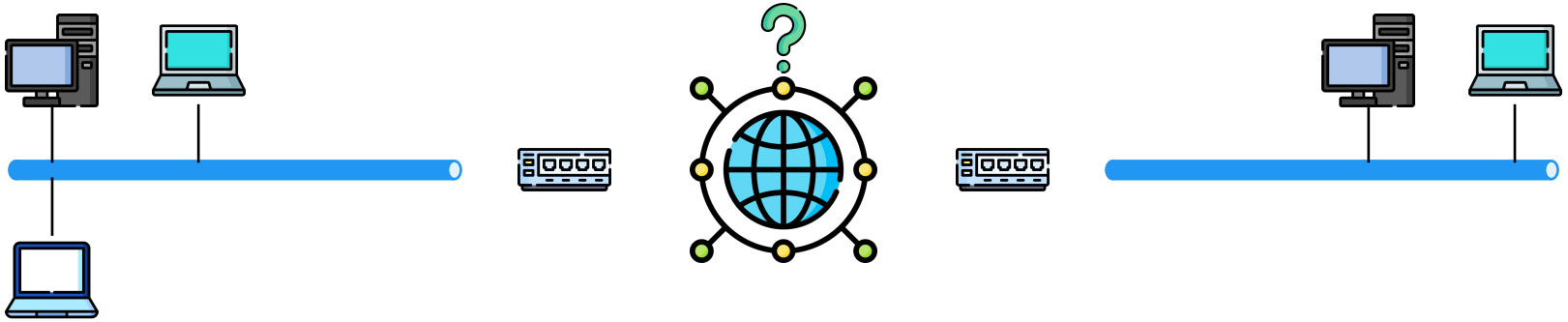
Un IPS **doit forcément** être positionné en coupure de flux de façon à pouvoir bloquer le trafic lorsque cela est nécessaire.



Virtual Private Network (VPN)

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 maitrise pas ?

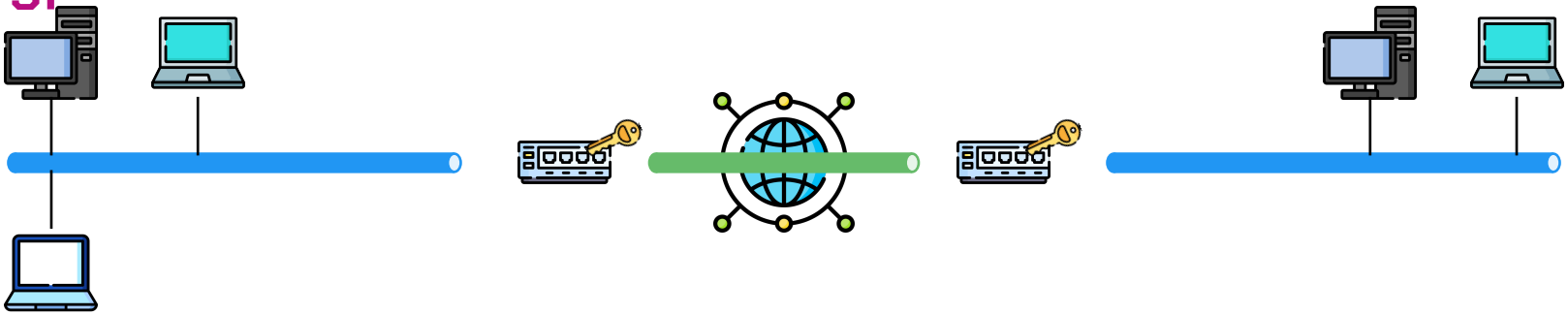


VPN

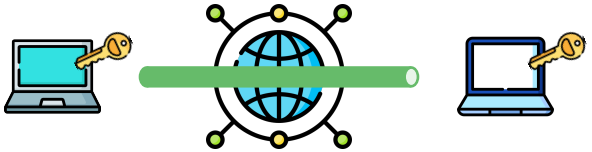
Solution grâce à des mécanismes cryptographiques, appliquer un **chiffrement des données**, ainsi qu'un **motif d'intégrité**, à **tous les flux** entre les 2 sites.

- 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.

Types de VPN

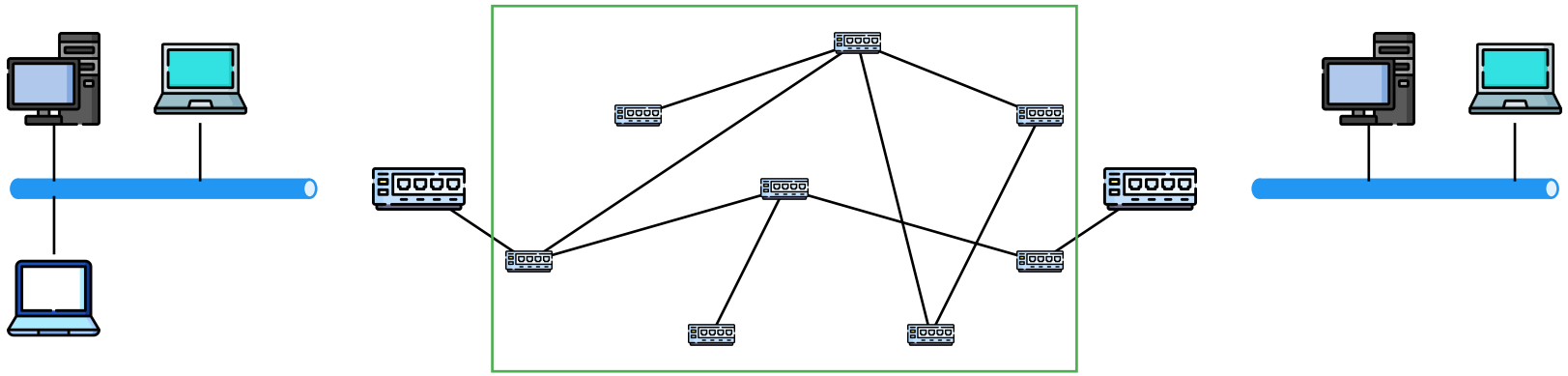


VPN de site à site dont le tunnel est géré par les routeurs.
IPsec - au niveau de la couche de transport.



VPN entre systèmes.
TLS - au niveau de la couche Internet.

Types de VPN



Réseau opérateur *Multiprotocol Label Switching* (MPLS)
dont le cœur est inaccessible aux clients se connectant sur ce réseau.

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.

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 physigues différents.

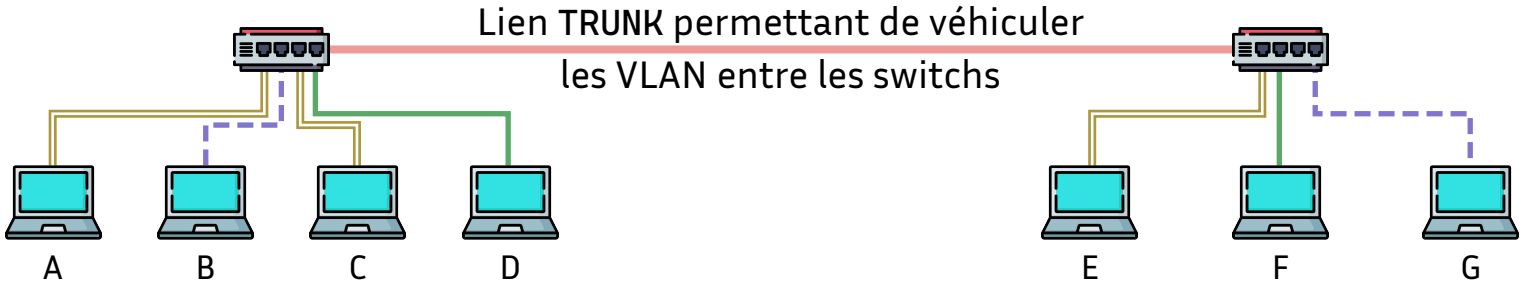
Avantages : étanchéité réseau parfaite ;
Inconvénients : peu adapté aux réseaux d'entreprise quie ont besoin de communiquer.

Segmentation : *Virtual Local Area Network* (VLAN)

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 avec précaution car le niveau de sécurité effectif est limité.

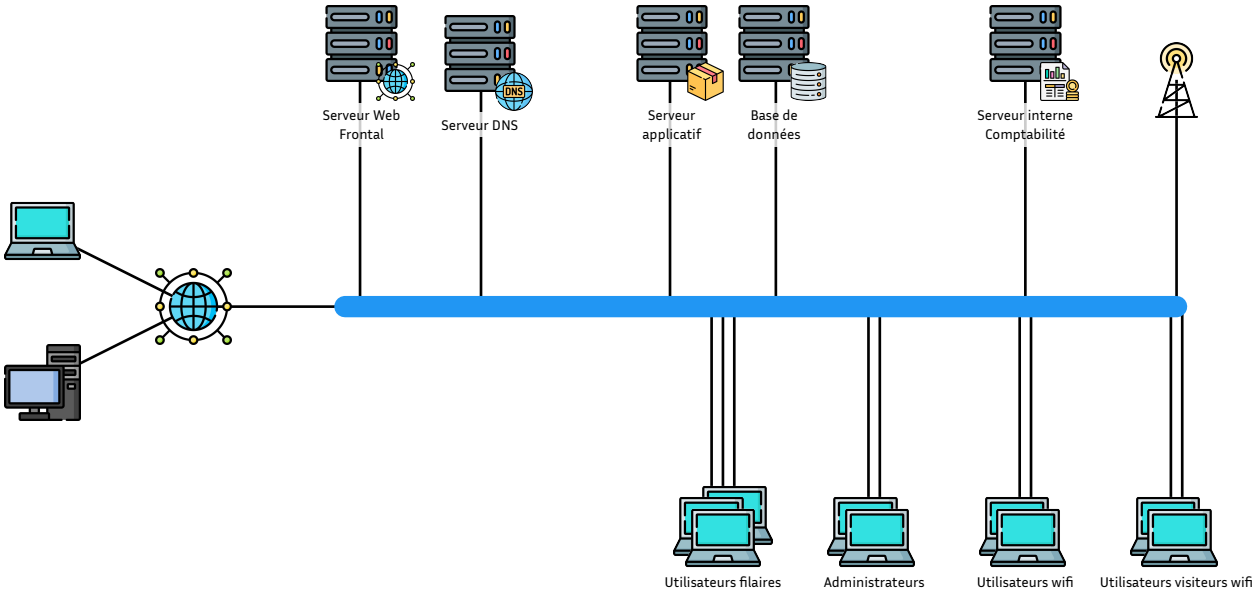
Segmentation : VLAN



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.

Exemple pratique



Exemple

Comment améliorer le réseau ?

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.

Exemple

Comment améliorer le réseau ?

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.

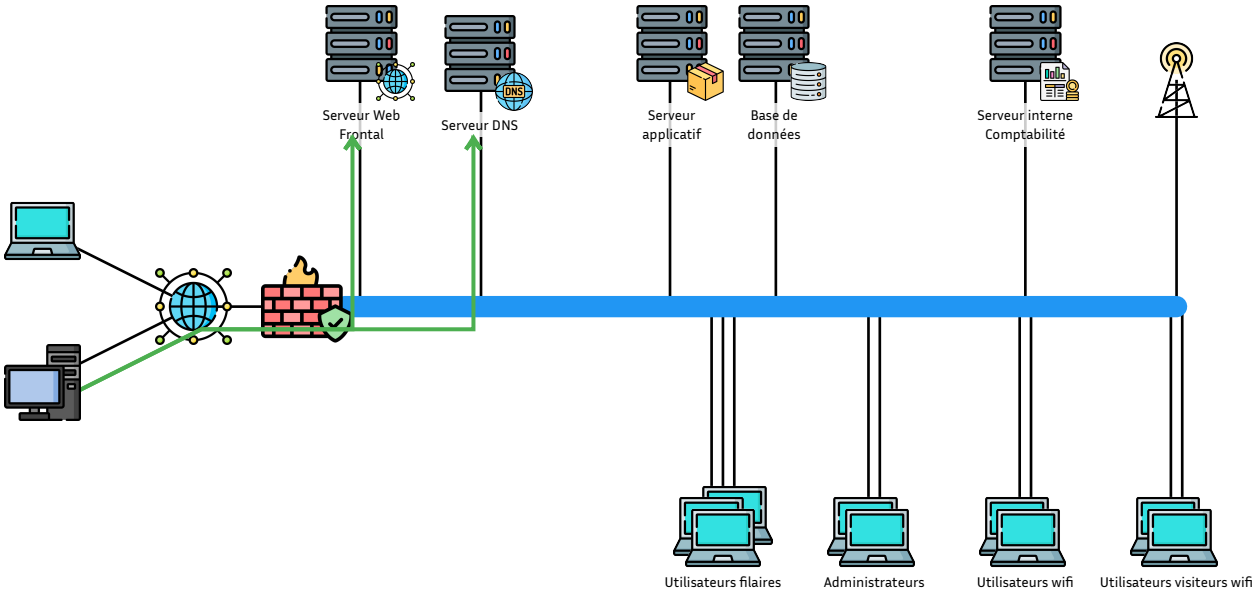
Correction

Mise en place d'un pare-feu en frontal qui va autoriser uniquement les flux entrants :

- vers le serveur `WEB` (`TCP/80` et `TCP/443`) ;
- vers le serveur `DNS` (`UDP/53` et `TCP/53`).

Ainsi, Internet ne pourra plus accéder au reste du réseau interne.

Exemple


Serveur Web Frontal
Serveur DNS
DNS
Serveur applicatif
Base de données
Serveur interne Comptabilité
Utilisateurs filaires
Administrateurs
Utilisateurs wifi
Utilisateurs visiteurs wifi

Exemple

Le parefeu permet d'empêcher la connexion directe entre internet et le réseau, mais si le serveur WEB présente une vulnérabilité, un hacker peut potentiellement prendre la main sur ce serveur et rebondir sur le réseau interne.

Exemple : Segmentation

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

Exemple : Segmentation

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

- Une *Zone Démilitarisée* (DMZ) 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 ;

Exemple : Segmentation

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

- Une *Zone Démilitarisée* (DMZ) 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 ;

Exemple : Segmentation

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

- Une *Zone Démilitarisée* (DMZ) 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 wifi des utilisateurs ;

Exemple : Segmentation

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

- Une *Zone Démilitarisée* (DMZ) 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 wifi des utilisateurs ;
- Une zone pour les postes wifi des visiteurs ;

Exemple : Segmentation

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

- Une *Zone Démilitarisée* (DMZ) 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 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...).

Exemple : Segmentation

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

- Une *Zone Démilitarisée* (DMZ) 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 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.

Exemple : Segmentation

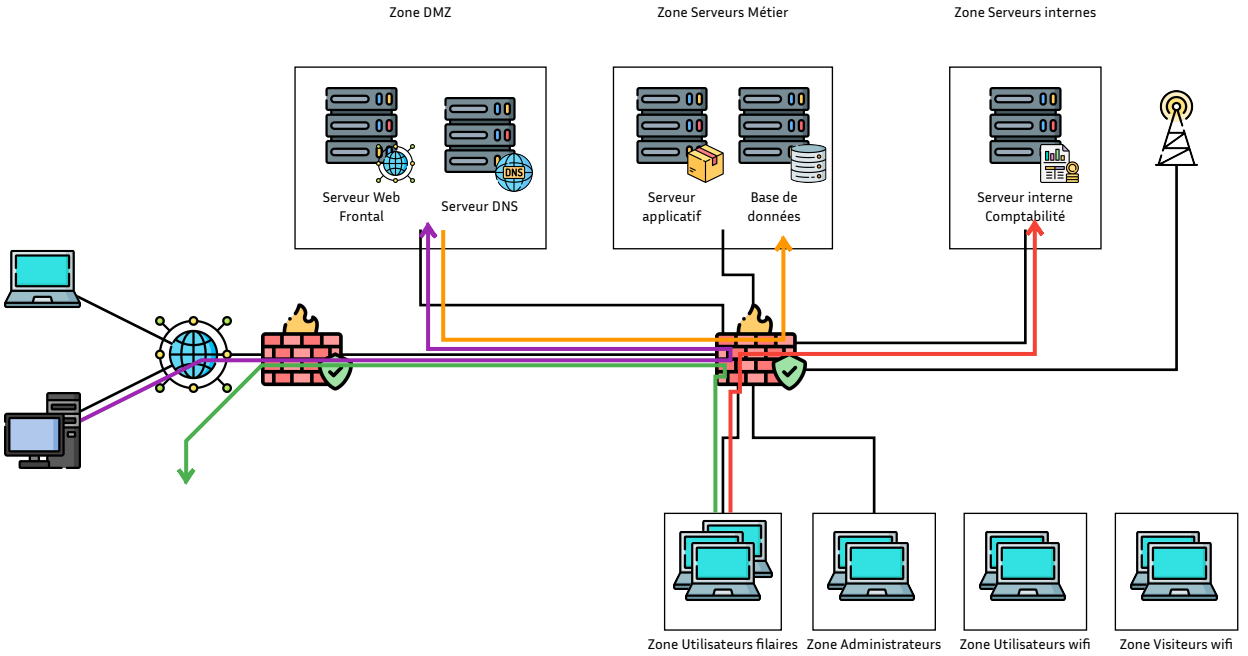
Nous allons donc segmenter notre réseau en différentes zones de criticité, notamment :

- Une *Zone Démilitarisée* (DMZ) 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 ;
-
-
-

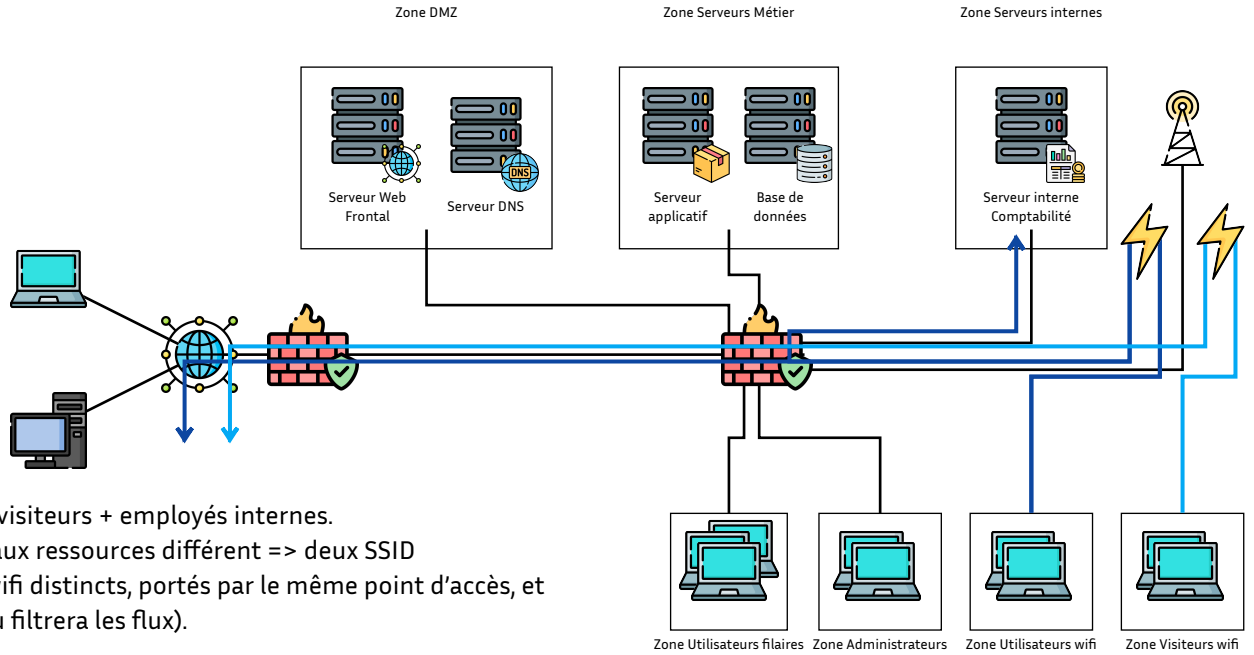
On observe malheureusement souvent des réseaux segmentés mais non filtrés. Cela ne sert à rien en terme de sécurité, car toutes les zones peuvent communiquer entre-elles.

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.

Exemple


Zone DMZ
Zone Serveurs Métier
Zone Serveurs internes
Serveur Web Frontal
Serveur DNS
DNS
Serveur applicatif
Base de données
Serveur interne Comptabilité
Zone Utilisateurs filaires
Zone Administrateurs
Zone Utilisateurs wifi
Zone Visiteurs wifi

Exemple



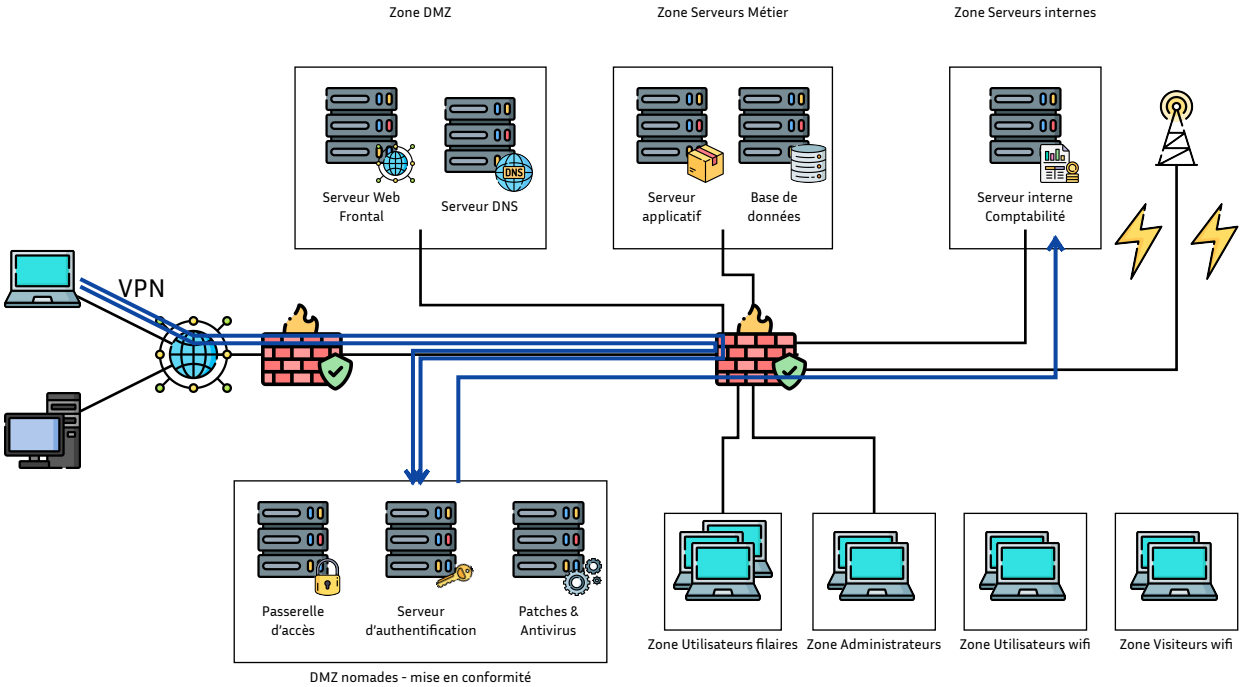
Accès wifi pour visiteurs + employés internes.
Besoin d'accès aux ressources différent => deux SSID
(deux réseaux wifi distincts, portés par le même point d'accès, et dont le pare-feu filtrera les flux).

Exemple

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**.

Exemple


Zone DMZ
Zone Serveurs Métier
Zone Serveurs internes
Serveur Web Frontal
Serveur DNS
Serveur applicatif
Base de données
Serveur interne Comptabilité
VPN
Passerelle d'accès
Serveur d'authentification
Patches & Antivirus
DMZ nomades - mise en conformité
Zone Utilisateurs filaires
Zone Administrateurs
Zone Utilisateurs wifi
Zone Visiteurs wifi

Exemple

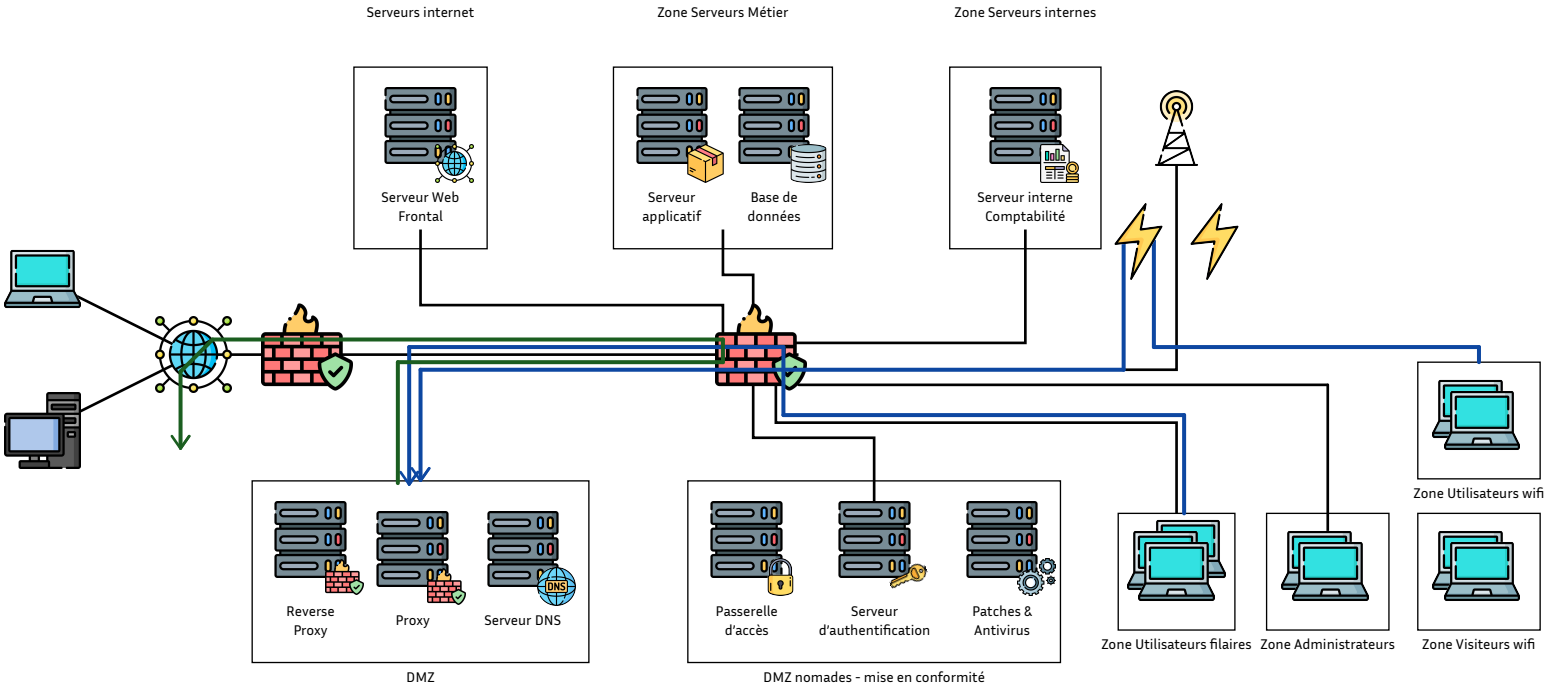
Enfin, il serait souhaitable de mieux filtrer le trafic WEB entrant et sortant :

- **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.).

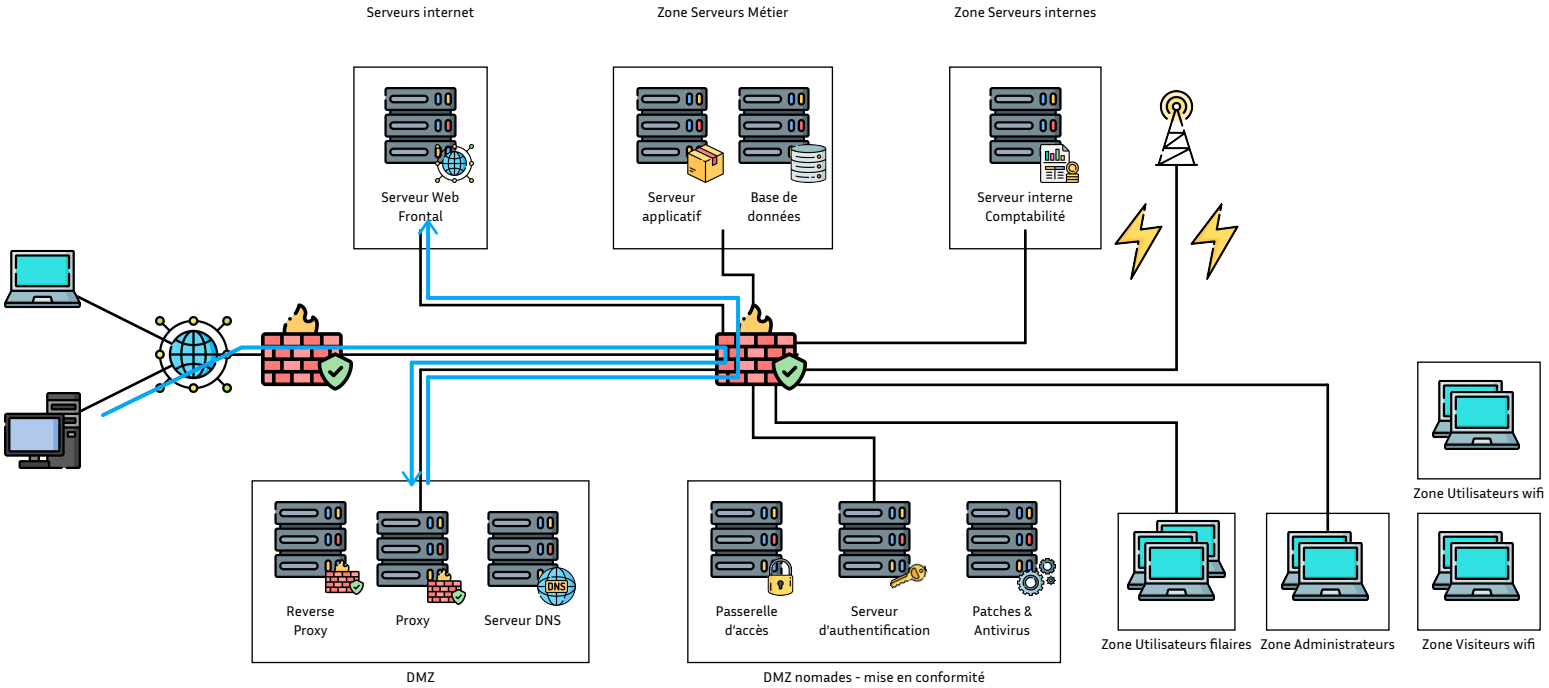
Nous allons donc recourir à un **proxy pour analyser les flux sortants**, et un **reverse-proxy pour analyser les flux entrants**. Ces équipements étant en coupure, ils empêchent donc les postes de travail des utilisateurs d'être connectés directement à Internet tout en leur permettant de naviguer sur 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.

Exemple


Serveurs internet
Zone Serveurs Métier
Zone Serveurs internes
Serveur Web Frontal
Serveur applicatif
Base de données
Serveur interne Comptabilité
Reverse Proxy
Proxy
Serveur DNS
DMZ
Passerelle d'accès
Serveur d'authentification
Patches & Antivirus
DMZ nomades - mise en conformité
Zone Utilisateurs wifi
Zone Utilisateurs filaires
Zone Administrateurs
Zone Visiteurs wifi

Exemple


Serveurs internet
Zone Serveurs Métier
Zone Serveurs internes
Serveur Web Frontal
Serveur applicatif
Base de données
Serveur interne Comptabilité
Reverse Proxy
Proxy
Serveur DNS
DMZ
Passerelle d'accès
Serveur d'authentification
Patches & Antivirus
DMZ nomades - mise en conformité
Zone Utilisateurs wifi
Zone Utilisateurs filaires
Zone Administrateurs
Zone Visiteurs wifi

Les bases de la cryptographie

Vocabulaire

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

Intégrité :

Vocabulaire

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

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

Vocabulaire

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

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

Confidentialité :

Vocabulaire

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

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

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

Vocabulaire

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

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

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

Preuve (authentification et non-répudiation) :

Vocabulaire

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

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

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

Preuve (authentification et non-répudiation) : fournir un moyen de preuve garantissant la véritable identité des entités ainsi que l'imputation de leurs actions.

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.

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.

Vocabulaire

Décrypter :

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

Crypter :

Ça n'existe pas !

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 !

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.

Chiffrement Symétrique vs Asymétrique

Chiffrement Symétrique		Chiffrement Asymétrique
	Avantages	
· Rapidité des opérations ; · Clés courtes (256bits suffisent actuellement).		Facilité d'échange des clés (les seules clés qui sont échangées sont les clés publiques - attention à l'intégrité).
	Inconvénients	
Difficulté d'échange des clés.		· Lenteur des opérations ; · Grande taille des clés (2048 bits **minimum** actuellement)
	Standards	
AES		RSA

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.

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.

La signature électronique n'assure pas la confidentialité des données, mais leur intégrité et la notion de preuve ;

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.

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.

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.

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 ;

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 ;

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 ;

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).

Signature électronique : Illustration

Étapes de signature


: clé publique


: clé privée

Signature électronique : Illustration

Étapes de signature


: clé publique


: clé privée

Signature électronique : Illustration

Étapes de signature


: clé publique


: clé privée

Signature électronique : Illustration

Étapes de signature


: clé publique


: clé privée

Signature électronique : Illustration

Étapes de signature


: clé publique


: clé privée

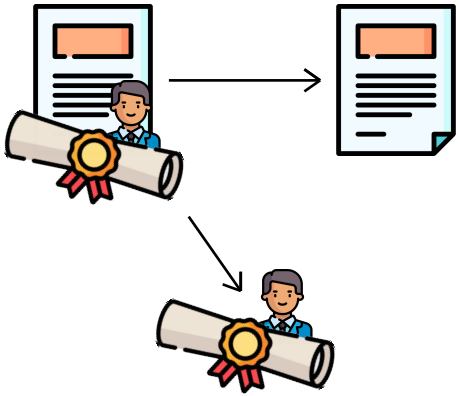
Signature électronique : Illustration

Étapes de signature


: clé publique


: clé privée

Signature électronique : Illustration

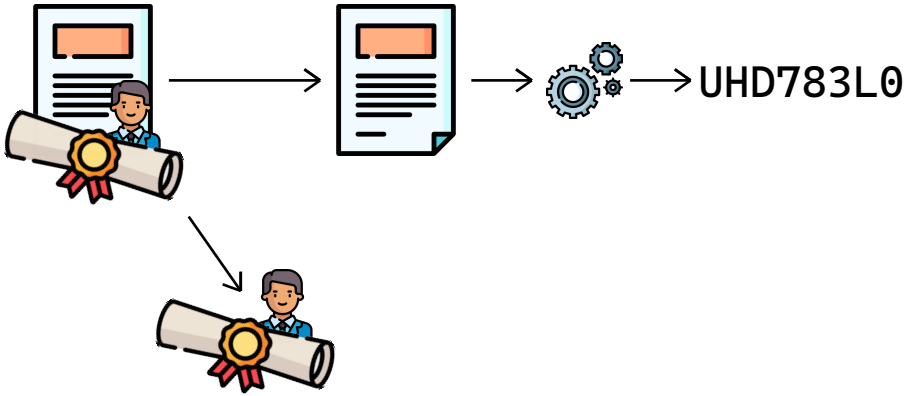


Étapes de vérification

 : clé publique

 : clé privée

Signature électronique : Illustration

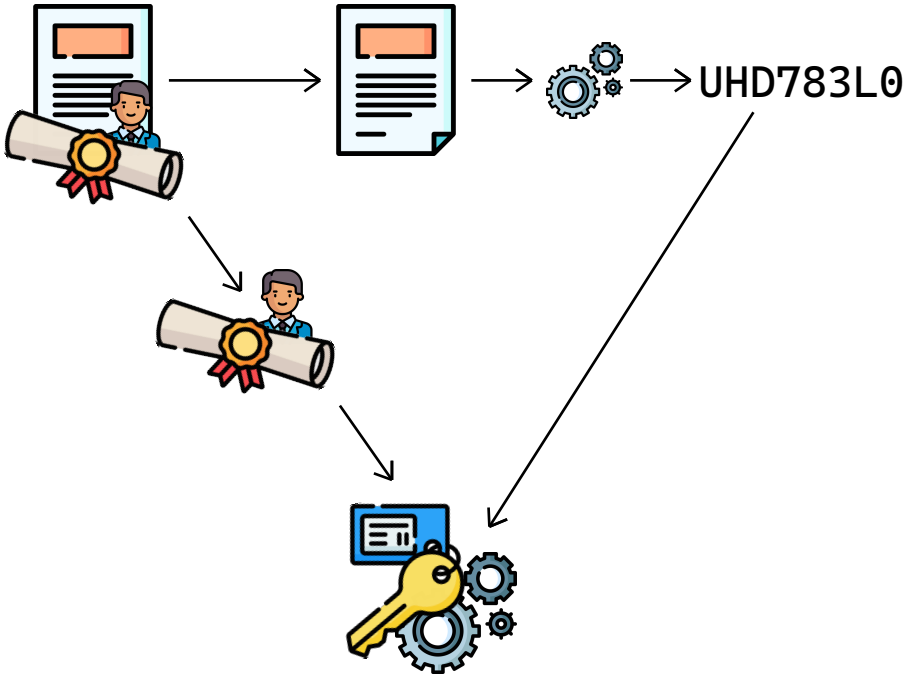


Étapes de vérification

: clé publique

: clé privée

Signature électronique : Illustration

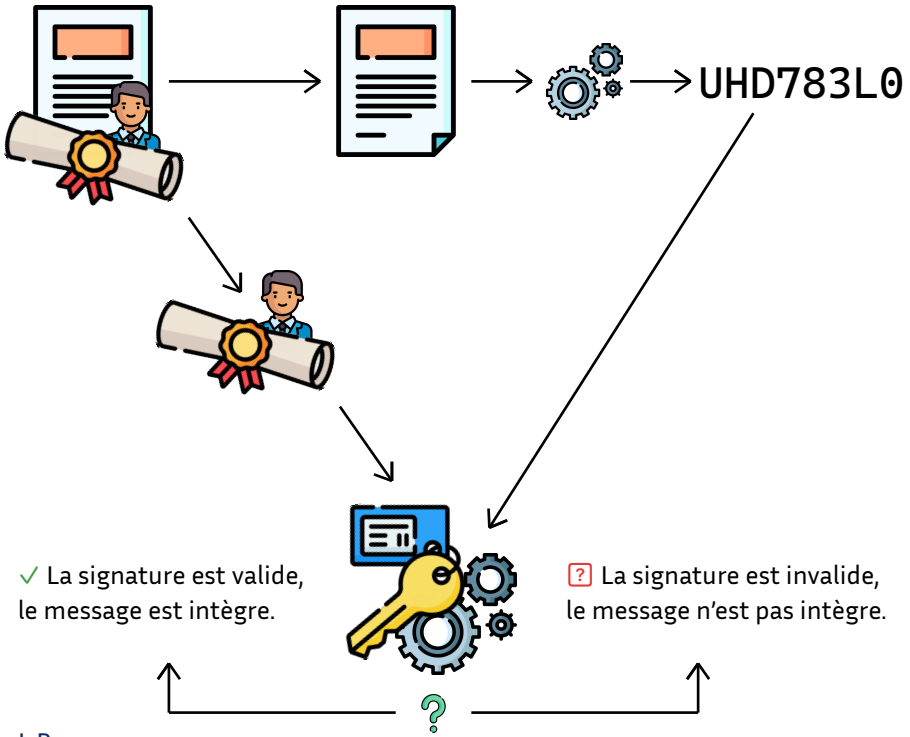


Étapes de vérification

: clé publique

: clé privée

Signature électronique : Illustration



Étapes de vérification

: clé publique

: clé privée

Certificats électronique


: clé publique


: clé privée

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 ?

Certificats électronique

 : clé publique

 : clé privée

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 ?

Solution : utilisation de certificats électroniques.

Certificats électronique

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

Certificats électronique

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) ;

Certificats électronique

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. ;

Certificats électronique

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 ;

Certificats électronique

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.

Certificats électronique

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 :

Certificats électronique

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 ;

Certificats électronique

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) ;

Certificats électronique

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).

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[1] permettant d'authentifier le propriétaire de la carte avant d'autoriser la transaction).

[1] 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.

La sécurité des applications web

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 :

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.

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.

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.

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).

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.

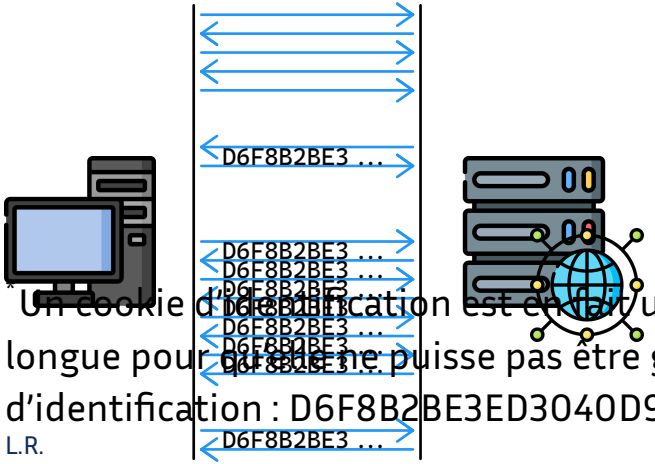
Les cookies sont nécessaires pour toutes les pages web dynamiques qui nécessitent d'identifier ou d'authentifier l'utilisateur, en permettant notamment la mise en œuvre de sessions :

- 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.

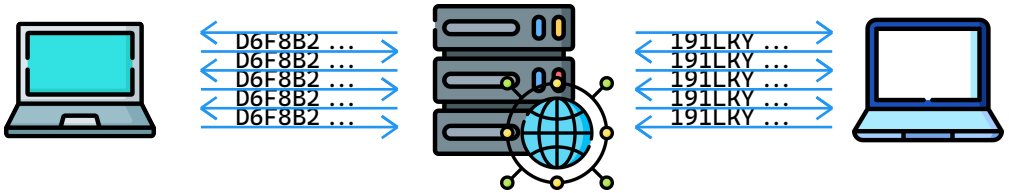
Usurpation d'identité via les cookies

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 chaine 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

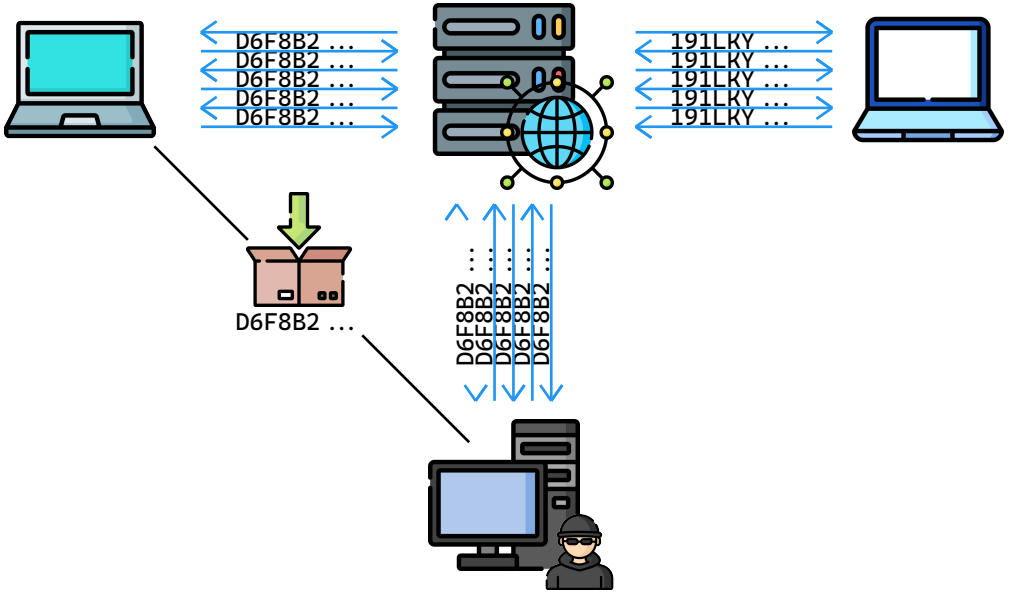
Usurpation d'identité via les cookies



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.

Usurpation d'identité via les cookies

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



Usurpation d'identité via les cookies

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

Vulnérabilité	Contremesure
Écoute du trafic réseau HTTP et en interception des données applicatives, dont le cookie	L'utilisateur doit s'assurer que le site auquel il est connecté utilise du HTTPS (le cookie est donc chiffré pendant le transport)
En utilisant une vulnérabilité du système	L'utilisateur doit **sécuriser son système d'exploitation et ses logiciels correctement**
Via des méthodes d'ingénierie sociale ciblées sur l'utilisateur.	L'utilisateur doit **être sensibilisé aux méthodes d'ingénierie sociale** (phishing, spam, etc.) afin de « ne pas tomber dans le panneau »
Via une faille sur le serveur	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**.

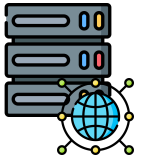
Injection SQL

- 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.

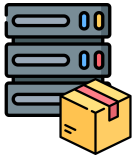
Injection SQL

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

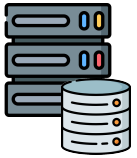
Utilisateur



Serveur Web



Serveur applicatif



Serveur
base de données

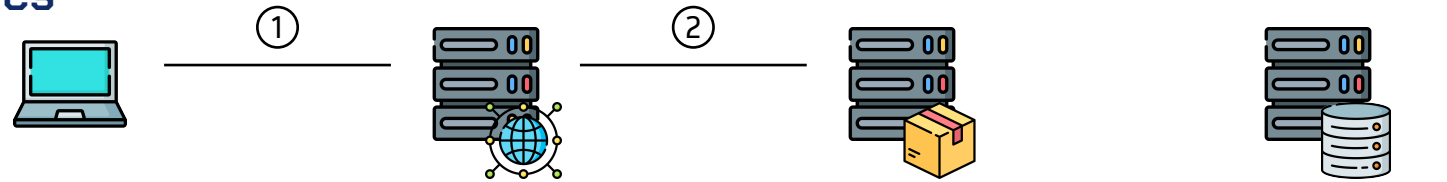
Injection SQL

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

>

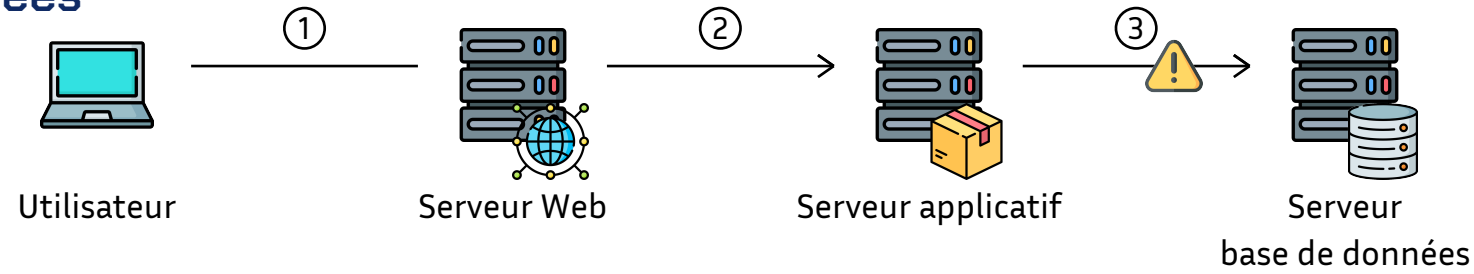
Injection SQL

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



Injection SQL

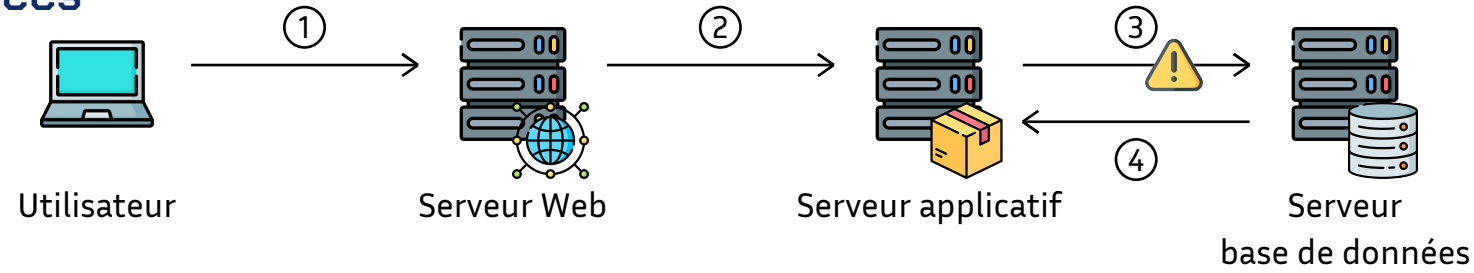
Architecture logicielle standard d'un site faisant appel à une base de données



<

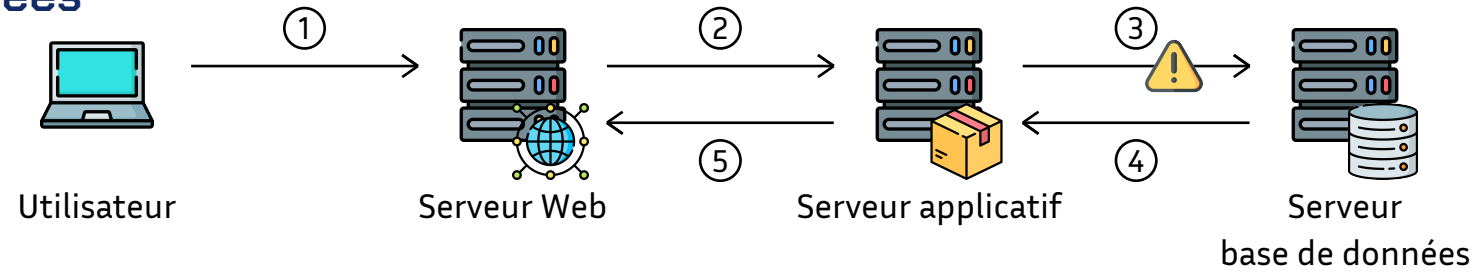
Injection SQL

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



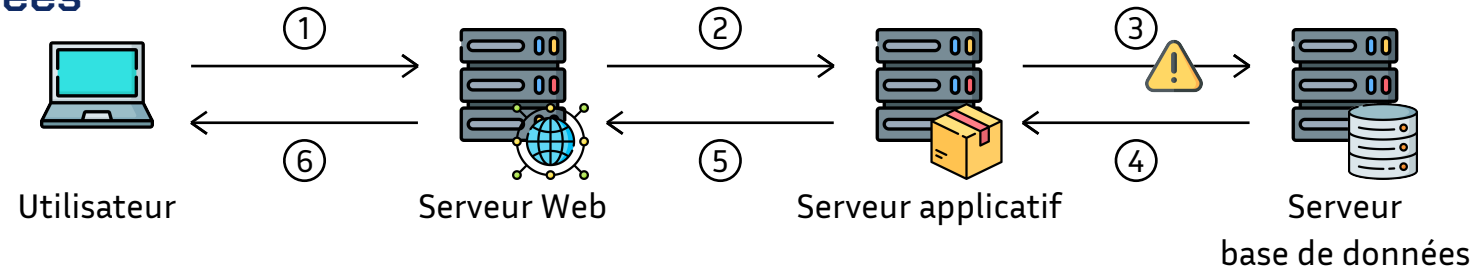
Injection SQL

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



Injection SQL

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

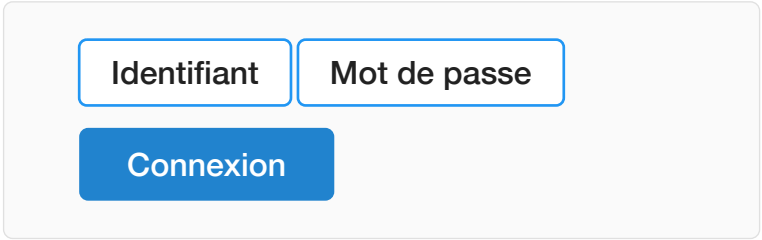


Injection SQL

- 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.

Injection SQL

Formulaire Web



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

La requête SQL permettant de vérifier l'identifiant et le mot de passe est la suivante :

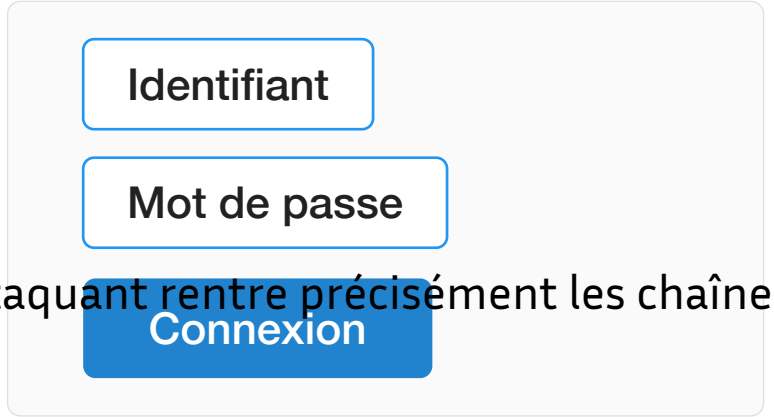
```
select count(*) from user where user=`$user` and password=`$pwd`
```

Ainsi, une requête légitime serait la suivante :

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

Injection SQL

Formulaire Web

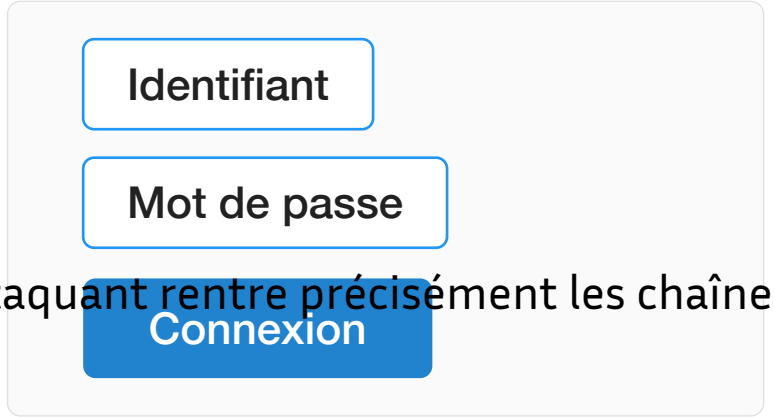


Que ce passe-t-il si un attaquant rentre précisément les chaînes de caractère suivantes :

```
$user = azerty
$pwd = abcd' or 1=1/*
```

Injection SQL

Formulaire Web



Que ce passe-t-il si un attaquant rentre précisément les chaînes de caractère suivantes :

```
$user = azerty
$pwd = abcd' or 1=1/*
```

La requête SQL :

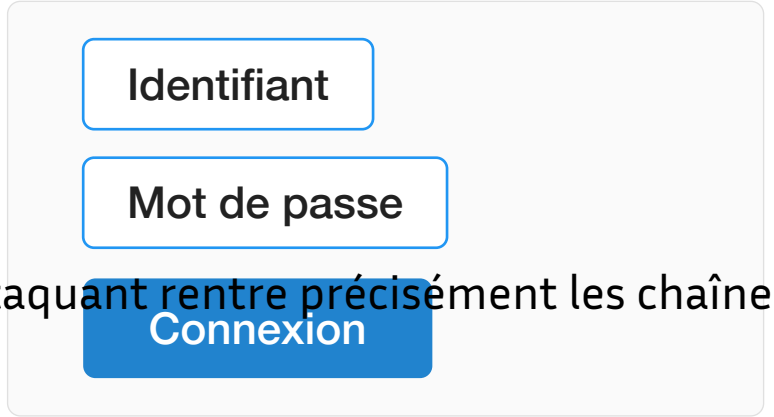
```
select count(*) from user where user=`$user` and password=`$pwd`
```

devient donc :

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

Injection SQL

Formulaire Web



Que ce passe-t-il si un attaquant rentre précisément les chaînes de caractère suivantes :

```
$user = azerty
$pwd = abcd' or 1=1/*
```

La requête SQL :

```
select count(*) from user where user=`$user` and password=`$pwd`
```

devient donc :

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

Condition toujours vraie

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.

Merci de votre attention

Merci de votre attention