Nathan HALLEZ
TG7

# Présentation de Tor (The Onion Routing)

Pdf disponible sur le site:
http://nsitorotg3hhpsk6.onion/

## Sommaire


- Distinction Clear Web, Deep Web, Dark Web
- Histoire
- Routage en oignon
- Chiffrement en oignon
- Les hidden services
- L’adresse d’un .onion
- Utilisation et comparaison avec un VPN
- Limitations et failles
- Sources

# Distinction Clear Web, Deep Web, Dark Web

Il est important de faire la distinction entre les différentes expressions qui sont exprimées pour distinguer différentes catégories du web. Les expressions Deep Web et Dark Web mènent souvent à confusion et sont parfois assimilées à la même chose.
Pour rappel, le web est un système qui fonctionne grâce à internet, il permet de consulter avec un navigateur des pages sur des sites. C'est une partie d'internet.

## Le Clear Web

Le Clear Web correspond à toutes les pages internet qui sont indexées dans les moteurs de recherche. C'est-à-dire toutes les pages qui ont été parcourues par un robot d'un moteur de recherche et qui sont ensuite enregistrées par ce moteur de recherche.
Le robot pour trouver des pages internet va suivre les liens présents sur les pages web qu'il connaît déjà pour en indexer de nouvelles. On peut aussi demander à Google par exemple d'indexer un site via un formulaire de soumission.
Pour résumer plus simplement, tout site qui est trouvable sur un moteur de recherche est un site qui appartient à la catégorie "Clear Web".

## Le Deep Web (ou Web Profond)

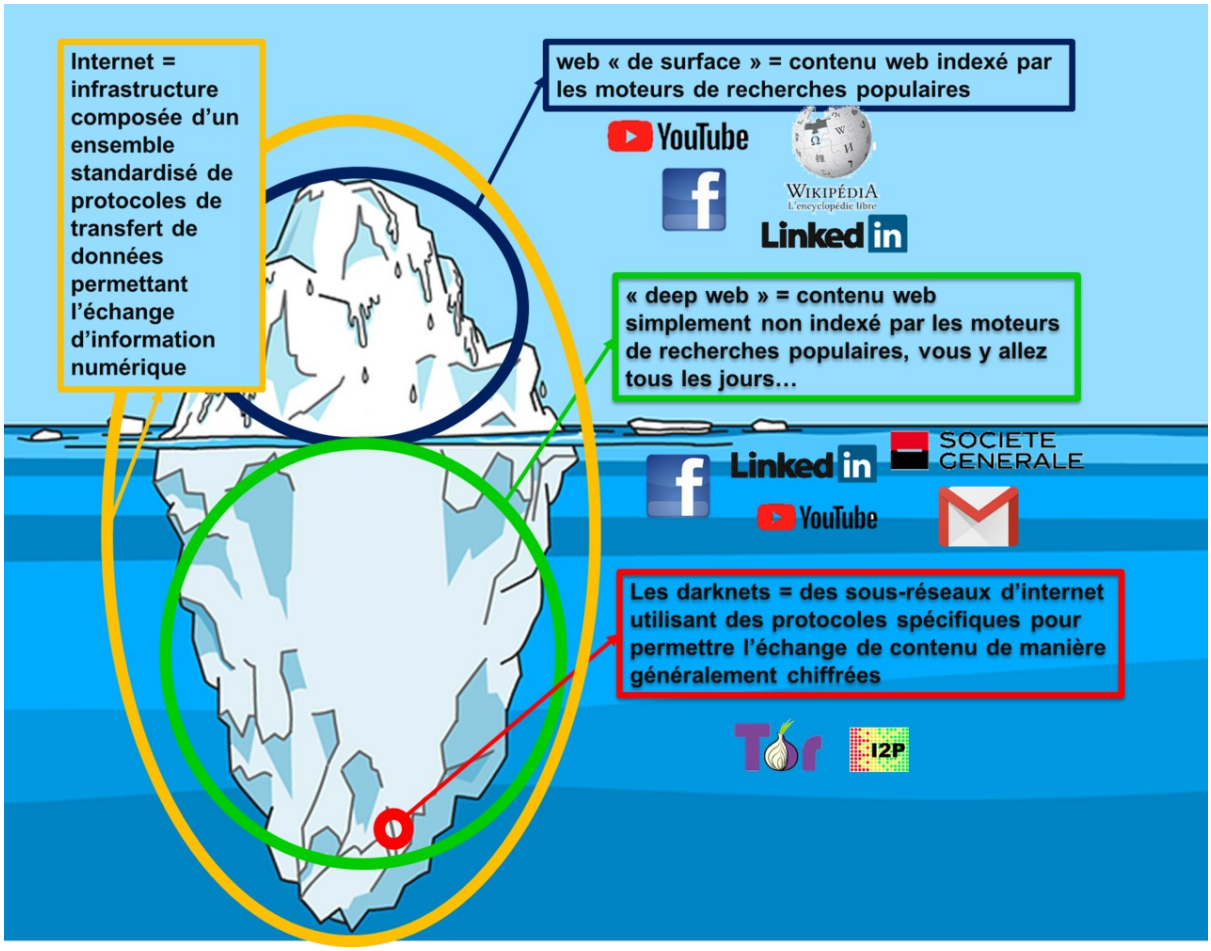
Le Deep Web est une catégorie qui contient beaucoup plus de pages web que le clear web (environ 500 fois plus que le web indexé).
La seule différence avec le clear web, c'est que le deep web n'est pas indexé par les bots avec plusieurs méthodes:
Les sites peuvent décider d'éviter aux robots d'indexer leurs sites en créant un fichier robot.txt, ou les sites en implémentant une page qui est accessible uniquement avec un login et un mot de passe ne seront pas indexés non plus.
Ce qu'il faut retenir du deep web, c'est que les pages dans cette catégorie sont des sites aussi que nous utilisons tous les jours. Cela pourrait être aussi des pages YouTube par exemple qui servent de base de données et qui n'ont pas de raison d'être indexés, ou encore le fil facebook d'un utilisateur qui n'est pas non plus indexé puisqu'il faut des identifiants pour y accéder.

## Le Dark Web

Enfin, le dark web forme une petite partie du deep web, puisqu'elle n'est pas non plus indexée par les moteurs de recherche. Mais le dark web correspond aux sites qui sont uniquement accessibles via des logiciels, configurations ou protocoles spécifiques tels que Tor par exemple. C'est une catégorie radicalement différente au Deep Web puisque le Dark Web permet une anonymisation et peuvent offrir la possibilité à ses utilisateurs de créer ou d'accéder à du contenu illégal dans le pays où ils se trouvent.

## Conclusion et schéma

Pour finir, voici une illustration courante de l'iceberg qui est utilisée pour décrire ces trois catégories du web.



# Histoire

Le concept de Tor a commencé dans les années 1990, le but de cette idée était le même qu'aujourd'hui: ils voulaient que les internautes aient un accès privé à un web non censuré.
Mais dans les années 1990, l'informatique était beaucoup moins sécurisée que maintenant et la surveillance était plus facile, ce qui donnait un intérêt supplémentaire à ce concept.

David Goldschlag, Mike Reed et Paul Syverson sont ceux qui se sont posé la question de savoir s'ils pouvaient créer des connexions internet qui ne révèlent pas qui parle à qui, même à quelqu'un qui surveille le réseau. Ils ont ainsi créé les premiers modèles de recherche et des prototypes de routage en oignon.
Ils faisaient tous les trois partie du Naval Research Laboratory qui est le laboratoire de recherche de la Navy (la marine de guerre des États-Unis).

Le principe qu'on retrouve encore aujourd'hui était déjà donné: chacun des utilisateurs devait acheminer le trafic via plusieurs serveurs tout en chiffrant à chaque étape du processus pour pouvoir utiliser Tor.

Mais la concrétisation réelle de ce projet à eu lieu dans les années 2000, puisque c'est Roger Dingledine et Paul Syverson qui coopèrent pour créer le projet Tor, pour distinguer le travail qui a été fait au laboratoire de recherche naval des ajouts supplémentaires qui sont apparus.
Cet évènement a permis aussi que le logiciel soit libre et gratuit.
A la fin de 2003, on pouvait compter une douzaine de bénévoles qui ont mis à disposition leurs serveurs pour faire fonctionner Tor.

En 2007, Tor fait en sorte de contourner les pare-feu du gouvernement pour que leurs utilisateurs puissent accéder à un web ouvert et qu'ils ne subissent pas la censure.

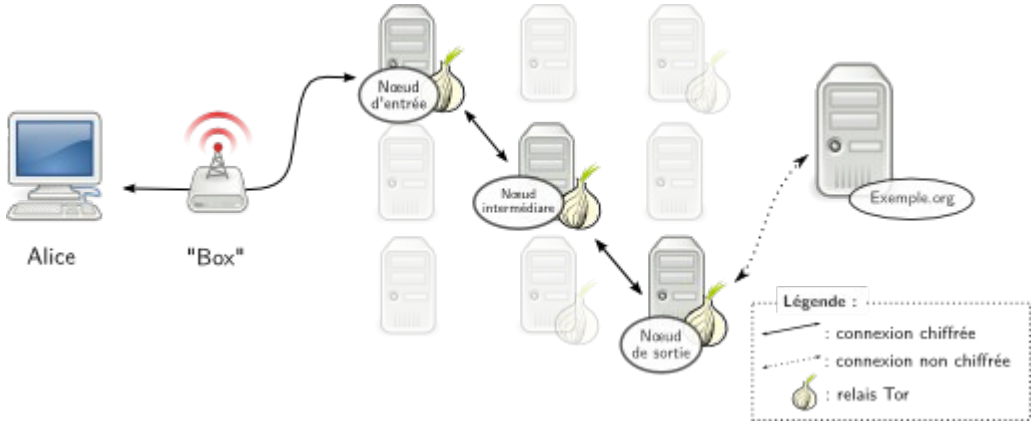
En 2009, la création de Tor Browser, un navigateur web basé sur Firefox, permet de rendre bien plus accessible Tor à tous les utilisateurs d'internet. De nombreuses personnes ont pu accéder à des ressources critiques, aux réseaux sociaux et aux sites web qui sont bloqués dans leurs pays.

Enfin, un exemple connu où Tor a était très utile c'est lors des révélations de Snowden en 2013, qui ont d'ailleurs fait augmenter en visibilité ce réseau.

# Routage en oignon

## Les noeuds de Tor

Pour fonctionner, Tor a besoin de serveurs de personnes bénévoles. Ces serveurs qu'on appelle nœuds du réseau servent de relais entre l'utilisateur et le serveur avec lequel il veut communiquer. Ces nœuds qui s'interposent entre le client et le serveur permet l'anonymat de l'utilisateur qui utilise Tor.
Quand quelqu'un va vouloir communiquer avec un serveur en utilisant Tor, sa requête va passer par plusieurs nœuds pour arriver au serveur.
La liste des nœuds est publique, ce qui veut aussi dire que les sites classiques (Google etc...) peuvent tout à fait bloquer l'accès à leurs sites aux utilisateurs de Tor.




## Celui qui choisit le chemin

Le client est celui qui crée le routage en oignon entre lui et le serveur avec la liste publique des nœuds du réseau, et ce n'est pas une décision prise au hasard.
Si c'est le client qui choisit le chemin et que ça ne serait pas les nœuds eux-même qui le choisirait au fur et à mesure, c'est que grâce à cela, tous les nœuds n'ont pas conscience du chemin en entièreté, ils savent seulement qui leurs a envoyé le message, et à qui ils doivent l'envoyer.

Comme plusieurs noeuds s'interposent entre le client et le serveur, on peut distinguer trois catégories de noeuds:

- Les nœuds en entrée: qui eux connaissent le client puisque c'est lui qui envoie le message.
- Les nœuds intermédiaires: qui eux ne connaissent ni le client ni le serveur.
- Et les nœuds en sortie: qui eux connaissent le serveur.

L’avantage d’avoir plusieurs noeuds dans un circuit Tor et pas un seul par exemple, c’est que si jamais un des noeuds est malveillant et qu’il souhaite connaître à la fois le client et le serveur (puisque les noeuds sont des serveurs de personnes bénévoles et donc on ne peut pas s’assurer qu’ils sont bienveillants) il ne pourra pas puisqu’il connaîtra au maximum que le client ou le serveur.

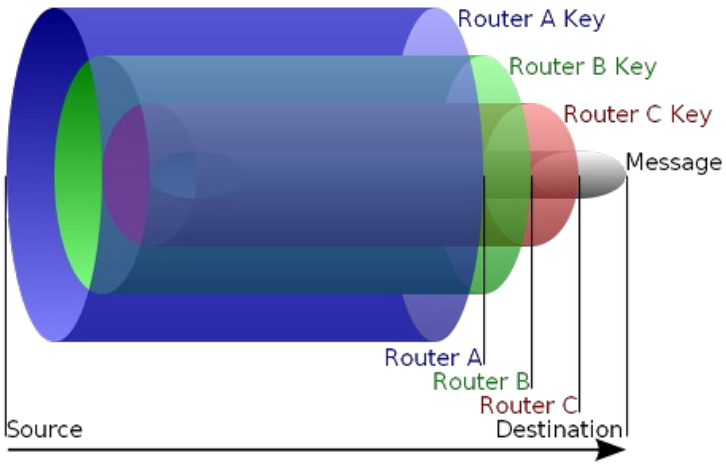
Aussi, comme le nœud de sortie communique avec le serveur, c’est donc l’adresse IP du nœud de sortie que le serveur verra uniquement, il ne pourra pas connaître l’adresse IP de l’utilisateur.

# Chiffrement en oignon

## Chiffrement du message par le client

Pour fonctionner, Tor a aussi besoin d’un système de chiffrement. Sinon tous les nœuds auraient accès au message du client en clair, même le nœud en entrée qui connaît le client.
Pour comprendre son fonctionnement, il faut savoir que chaque nœud possède une clé publique et une clé privée (c’est un chiffrement asymétrique).
La clé publique est disponible dans la liste publique et donc le client (qui fait le routage en oignon) peut la récupérer.

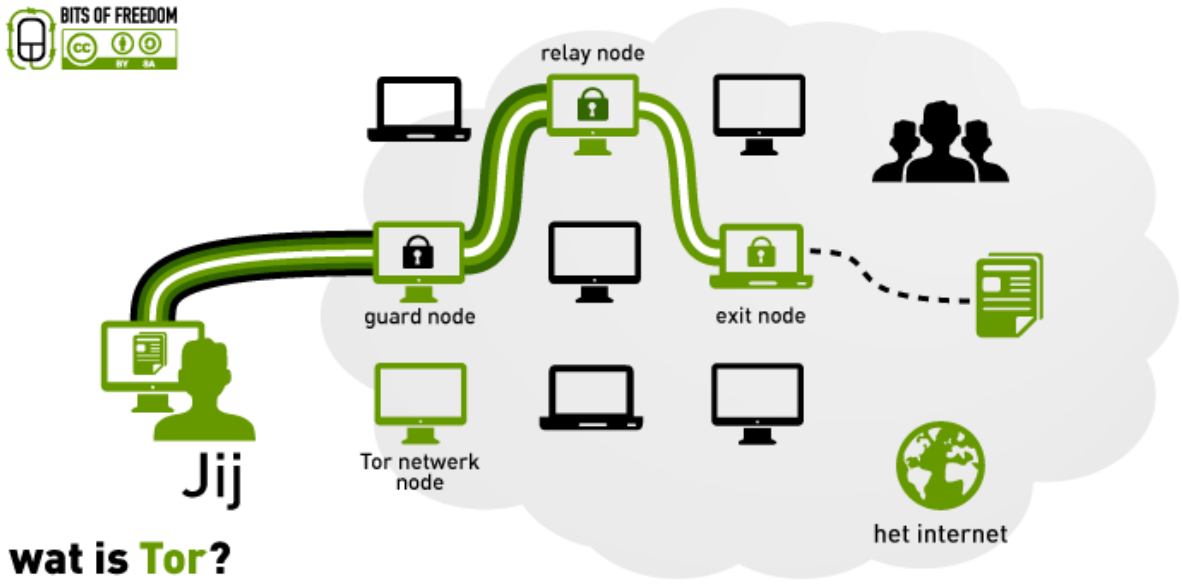
Quand le client va vouloir envoyer un message au serveur, il va devoir chiffrer son message avec chacune des clés publiques des nœuds. Voici un schéma qui illustre ce chiffrement.



Comme on peut le voir, le message est chiffré par trois clés de chiffrement, correspondant aux 3 nœuds du routage. Avec cette illustration, on peut comprendre pourquoi ils ont donné le nom “chiffrement en oignon”, puisque un oignon est constitué de plusieurs couches comme le message chiffré.

## Déchiffrement progressif du message par les noeuds

En envoyant le message au nœud d’entrée, celui-ci va déchiffrer le message avec sa clé privée, le message étant toujours chiffré avec deux autres clés publiques, il lui est impossible d’accéder au contenu du message. Il l’enverra au nœud intermédiaire qui va aussi déchiffrer avec sa clé privée et l’enverra au nœud de sortie qui lui en le déchiffrant aura accès au message. Mais comme il ne connaît pas le client, ça ne pose pas de problème puisque c’est lui qu’on veut protéger (on parlera aussi des hidden services qui ont pour fonctionnalité de protéger les serveurs, pour le moment on se concentre dans le cas où le client utiliserait Tor dans un internet classique). Voici un autre schéma qui explique ce chiffrement en oignon:

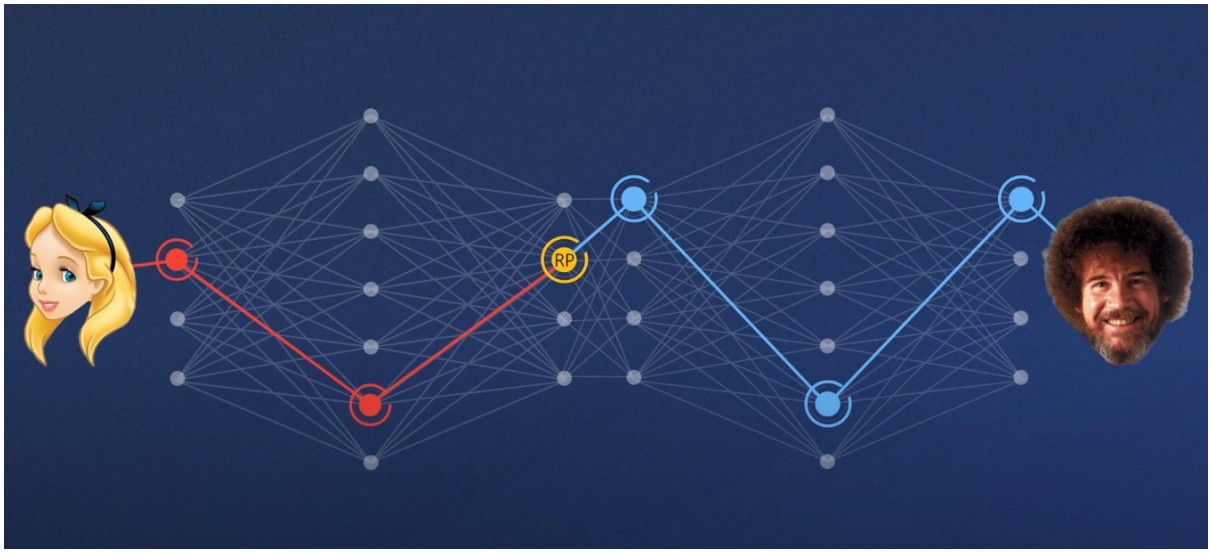


# Les hidden services

Jusque là nous avons appris comment le client faisait pour être anonyme avec Tor, mais ce réseau donne aussi la possibilité aux serveurs d'être anonymes. Le domaine .onion est d'ailleurs réservé aux hidden services et ne peuvent être accessible uniquement avec Tor.

En revanche, le fonctionnement des hidden services (ou service caché en français) est plus complexe que le client puisque normalement pour pouvoir accéder à un site on doit pouvoir connaître son adresse IP. C'est un peu comme si on voulait appeler quelqu'un mais qu'on ne connaît pas son numéro de téléphone.

Dans le principe, quand le client demande la connexion au serveur, le serveur choisit un point de rendez-vous en utilisant le réseau Tor. En plus de ça, le serveur communique avec le point de rendez-vous en utilisant un routage en oignon (donc plusieurs nœuds en intermédiaire). Voici une illustration qui récapitule la communication entre le client et le serveur. Alice représente le client, et Bob le serveur, RP signifie "rendez-vous point" (ou point de rendez-vous en français), et tous les points représentent des nœuds du réseau Tor.




*source: https://youtu.be/zjqw1VnuKLs?t=648*

Et donc le serveur ne connaît pas le client, le client ne connaît pas le serveur. Et en plus de cela, le nœud de sortie du client ne communique plus avec le serveur directement et donc ne le connaît pas, ce qui fait que le nœud de sortie a le message en clair, mais il ne sait pas de qui il provient et à qui il est destiné.

# L’adresse d’un .onion

Si on regarde l’adresse d’un .onion (domaine des hidden services), on remarque que son adresse est très peu compréhensible, on se retrouve avec des caractères qui sembleraient être générés aléatoirement. Ce n’est pas comme une adresse d’un site sur l’internet classique (par exemple *https://google.fr*). Pour comprendre cette adresse, il faut d’abord comprendre ce qu’est la fonction de hachage.

## Qu’est-ce que la fonction de hachage?

La fonction de hachage permet de calculer une empreinte numérique: À partir d’un message, elle va pouvoir générer une chaîne de caractères.
Le processus ne fonctionne que dans un sens, c'est-à-dire qu’on peut créer le hash d’un message, mais on ne peut pas avoir le message à partir du hash.
Une de ces utilisations courantes se trouve dans le stockage des mots de passe. L’ordinateur dans ce cas là calcule le hash du mot de passe entré par l’utilisateur et le compare avec le hash stocké.
Il y a plusieurs fonctions de hachage qui existent comme le md5 par exemple.

## L’adresse .onion utilisé comme vérification

L’adresse d’un .onion est créée à partir de la clé publique du hidden service.
Ce que va faire le client quand il sera au point de rendez-vous, c’est qu’il va calculer à son tour le hash de la clé publique du hidden service, si il trouve le même hash qu'une adresse en .onion, il pourra s’assurer qu’il s’agit bien du bon serveur.

Cette étape est nécessaire puisque quand quelqu’un veut créer un hidden service, il peut très bien changer à tout moment la clé privée et l’adresse du site (calculée à partir de la clé privée). Et donc, une personne peut très bien usurper l’identité d’un autre serveur en mettant la même adresse en .onion que le serveur et utiliser sa propre clé privée. Mais comme il y a cette vérification, le client pourra reconnaître que ce n’est pas lui puisqu’en calculant le hash de la clé publique, il se rendra compte que le hash et l’adresse ne correspondent pas.

## Comment avoir une adresse personnalisée?

Mais au final, il y a quand même une méthode pour avoir une adresse en .onion qui soit un peu plus lisible. Par exemple, le site sur lequel se trouve le pdf commence par “nsitor”. Mais le problème, c’est que on ne peut pas avoir une clé publique à partir d’un hash, ça ne fonctionne que dans un sens.

La seule méthode pour y arriver c’est de tester des millions de clés publiques pour trouver des adresses avec des mots clés.
Par exemple, les logiciels Scallion (qui utilise la carte graphique pour fonctionner) et Eschalot (qui semble un peu plus complet) permettent de créer des adresses à partir de mots clés.
Pour Eschalot, on peut rentrer un mot, et au bout d’un certain nombres d’essais, il renverra une adresse en .onion et sa clé privée qu’on pourra mettre directement dans la configuration de l’hidden service.

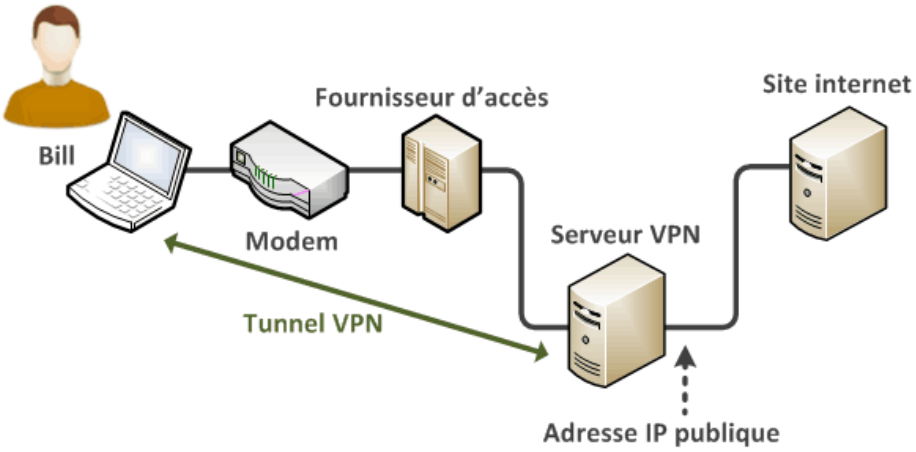
# Utilisation et comparaison avec un VPN

Les VPN sont devenus aujourd’hui plus populaires qu’auparavant dûe à des campagnes marketing organisées par NordVPN par exemple qui ont envahi YouTube. Le problème, c’est que les pubs qui sont présentées vendent des fonctionnalités qui ne sont pas le but des VPN. Ils vendent que les VPN par exemple rendent anonyme.

## Qu’est-ce qu’un VPN?

Un VPN est un réseau privé virtuel. Il permet de lier un ordinateur d’un réseau local à un autre réseau local pour communiquer avec les appareils de la même manière qu’un réseau local. Et c’est une méthode sécurisée, ainsi, si quelqu’un d’extérieur veut accéder à ce réseau sans avoir les permissions, il ne pourra pas.
C’est un système qui est surtout utilisé par les entreprises, puisque grâce au VPN, les salariés peuvent tout de même se connecter au réseau local de l’entreprise tout en étant en télétravail par exemple.

Un VPN est composé d’un client et d’un serveur, quand le client voudra envoyer une requête, il passera par ce serveur. Le serveur lui renverra les réponses. Les données envoyées par le client sont chiffrées.
L’avantage qu’on voit ici, c’est que comme le serveur est celui qui envoie les requêtes à la place du client, les services auquel le client veut accéder ne connaîtront pas son adresse IP mais celle du VPN.

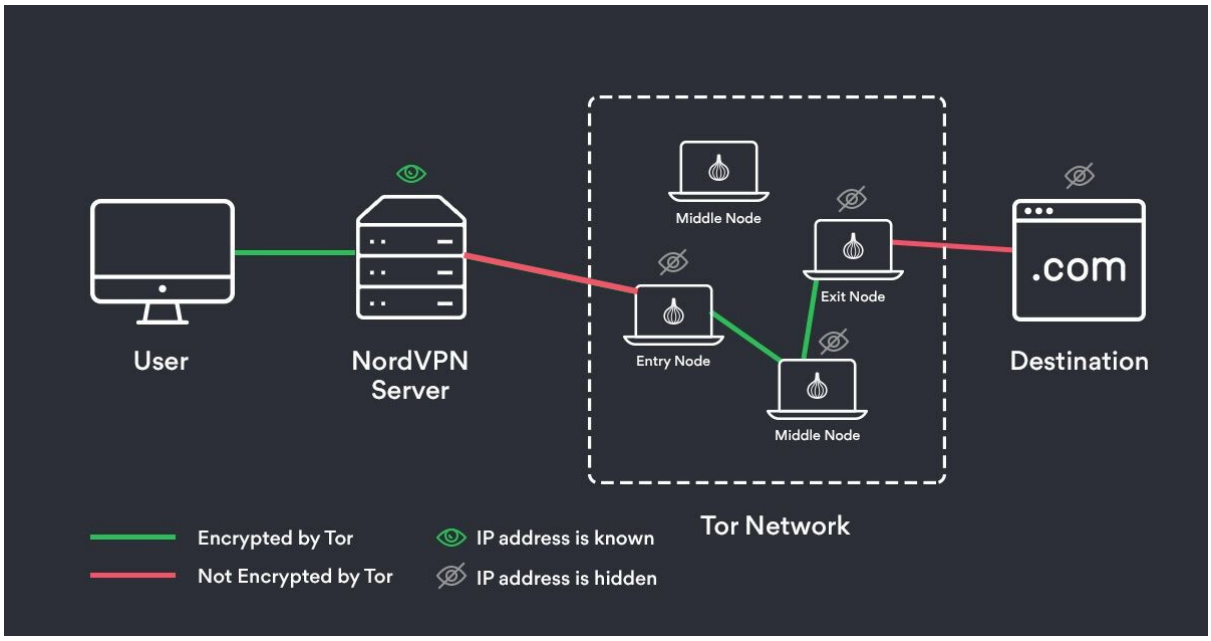



## Comparaison avec Tor

Si on compare le système du VPN avec Tor, on remarque qu'il y a des ressemblances. On pourrait dire qu'en fait un VPN est un routage en oignon qui n'a qu'un seul nœud. L'utilisation d'un VPN pour être complètement anonyme sur internet est donc erronée puisque le fournisseur du VPN connaît le client, les informations qu'il envoie, et le serveur. On ne pourra jamais vraiment savoir si un service de VPN enregistre les informations puisque ça ne dépend que de lui.
Tor en revanche, même si un des nœuds est malveillant, il ne pourra rien faire pour récupérer des informations intéressantes comme nous l'avions expliqué précédemment.
Néanmoins, le VPN fonctionne dans quelques cas spécifiques. Par exemple, si un utilisateur utilise un VPN et qu'il est connecté dans un réseau public et qu'il navigue uniquement sur des liens en HTTP donc non chiffré (très rare aujourd'hui), si il y a un hacker qui veut l'observer, il ne pourra pas. Un autre exemple un peu plus parlant, c'est que si quelqu'un joue à un jeu vidéo en utilisant un VPN, et que quelqu'un de malveillant souhaite faire une attaque DDOS sur cet utilisateur, il ne pourra pas le faire puisqu'il ne connaît pas l'adresse IP du joueur.

## Utilisation de Tor avec un VPN

En revanche, là où il y a un réel intérêt d'utiliser un VPN c'est pour l'utiliser avec Tor.




Comme le montre cette image, avec un VPN, il est encore plus difficile de tracer le client puisque même le nœud en entrée dans le réseau Tor ne connaît pas l'adresse IP du client, il ne connaît que le VPN. Encore une fois, ce n'est pas infaillible, puisque le VPN peut tout de même enregistrer le trafic de ces clients, mais ça rend le traçage plus difficile.

# Limitations et failles

## La lenteur

Si on essaye Tor, on remarque que pour charger une page, c'est très lent. Cela s'explique simplement par le fait que la requête passe par plusieurs nœuds, ce qui fait donc plusieurs détours. Si en plus de ça le serveur avec lequel on veut communiquer est un hidden service, alors la requête va devoir aussi passer par les nœuds du serveur.

## Les paquets UDP

Tor ne recommande pas d'utiliser les paquets UDP dans une navigation, surtout si on navigue avec Tor sur des sites classiques. Comme la communication des paquets UDP sont unidirectionnel, et qu'il n'y a pas un système d'accusé de

réception comme pour le TCP,  ça ne fonctionne pas bien avec Tor puisqu'il y a plusieurs nœuds entre le client et le serveur.

## Certaines fonctionnalités web limités

Il a fallu faire quelques sacrifices au niveau du javascript et d'autres extensions comme Flash par exemple pour garder un anonymat, puisqu'ils peuvent réussir à traquer l'utilisateur en repérant une empreinte de l'ordinateur. C'est pourquoi sur Tor Browser est présente l'extension Noscript qui permet de bloquer les scripts. On retrouve d'ailleurs d'autres extensions comme HTTPS everywhere qui est la deuxième extension majeure utilisée pour le Tor Browser et qui sert à chiffrer la communication du nœud de sortie au serveur pour les sites qui n'ont pas encore le protocole HTTPS.

## Attaque time pattern

Comme nous l'avions vu précédemment, le nœud en entrée connaît l'adresse IP du client et le nœud en sortie connaît l'adresse IP du serveur. Théoriquement, si un utilisateur lors du routage en oignon attribue le noeud en entrée et le noeud de sortie à noeuds appartenant à la même personne, alors celui-ci pour savoir qui parle avec qui (sans connaître pour autant les informations échangés dans le cas où la communication est chiffré avec le serveur).
Cela reste une attaque théorique puisque pour arriver à faire cette attaque il faudrait que la personne qui veut connaître ces informations possède énormément de serveurs pour augmenter la probabilité d'être en amont et en aval du routage en oignon de l'utilisateur.

## Faille possible

Une faille de Tor aurait été exploitée lors d'une attaque appelée Operation Onymous qui aurait permis un traçage physique des serveurs, ce qui retirerait l'anonymat des hidden services. Cependant, cette faille reste secrète puisque les attaquants ne souhaitent pas expliquer cette faille.

# Sources

## Histoire

*https://www.torproject.org/about/history/*
*https://fr.wikipedia.org/wiki/Naval_Research_Laboratory*
*https://fr.wikipedia.org/wiki/Tor_Browser*

## Routage en oignon

*https://fr.wikipedia.org/wiki/Fonction_de_hachage*

## Les hidden services

*https://fr.wikipedia.org/wiki/.onion*

## L'adresse  d'un .onion

*https://fr.wikipedia.org/wiki/Fonction_de_hachage*
*https://youtu.be/zjqw1VnuKLs*
*https://youtu.be/VCltQ_G-KA0?t=318*
*https://security.stackexchange.com/questions/29772/how-do-you-get-a-specific-onion-address-for-your-hidden-service*

## Distinction Clear Web, Deep Web, Dark Web

*https://www.ledieu-avocats.fr/clear-web-deep-web/*
*https://optimiz.me/indexation-comprendre-le-fonctionnement-de-lindex-google/*
*https://fr.wikipedia.org/wiki/Dark_web*
*https://fr.wikipedia.org/wiki/World_Wide_Web*

## Utilisation et comparaison avec un VPN

*https://youtu.be/ckZGQ5cLIfs*
*https://www.parlonsgeek.com/vpn-vs-tor-avantages-et-inconvenients/* (il y a certains points qui sont faux mais ça m'a permis de relever certains points pertinents)

## Limitations et failles

https://www.reddit.com/r/TOR/comments/7oizra/why_tor_doesnt_support_udp/
https://fr.wikipedia.org/wiki/Tor_(r%C3%A9seau)#Limitations
https://fr.wikipedia.org/wiki/Tor_(r%C3%A9seau)#Failles_possibles