**UNIVERSITE ABDELMALEK ESSAADI FACULTE DES SCIENCES ET TECHNIQUES DE TANGER**



**DEPARTEMENT GENIE INFORMATIQUE**

**PROJET DE Module Administration Système ET Réseaux**

**MITBD**



**Sujet:**

**Installation et Configuration des services applicatifs & Configuration des ACLs**

**Réalisé PAR :**

**Sous l’encadrement de : Mr.Abdelhamid ZOUHAIR**

* **STITOU Fatima**
* **HADDAD Alae**
* **HAYOUN Fatima Zohra**
* **OUAAQA Youssef**

#### Année Universitaire : 2023/2024

**Table des matières**

[Partie 1 : Services applicatifs 2](#_bookmark0)

[Introduction 3](#_bookmark1)

1. [Services applicatifs : importance et utilisation 3](#_bookmark2)
   1. [Pourquoi les services d’applicatifs sont-ils importants ? 3](#_bookmark3)
2. [Services Applicatifs Courants 4](#_bookmark4)
3. [Tendances Émergentes et Futur des Services Applicatifs 5](#_bookmark5)
4. [Informatique en Nuage Évoluée 5](#_bookmark6)
5. [Intelligence Artificielle (IA) et Automatisation 6](#_bookmark7)
6. [Edge Computing 6](#_bookmark8)
7. [Interfaces Utilisateurs Évoluées 6](#_bookmark9)
8. [Sécurité Renforcée 6](#_bookmark10)
   1. [Étapes d'Installation d'OpenVPN 6](#_bookmark11)
9. [Connexion au serveur 6](#_bookmark12)
10. [Installation d’OpenVpn 7](#_bookmark13)
    1. [b.1- Mise à jour le système 7](#_bookmark14)
       1. [b.2- Ajouter le référentiel du serveur d'accès OpenVPN 7](#_bookmark15)
    2. [Étapes de configuration 7](#_bookmark16)

[Partie 2 : Configuration des ACL (Access Control List) 12](#_bookmark17)

[Introduction 13](#_bookmark18)

1. [Présentation 13](#_bookmark19)
   1. [Définition des ACL 13](#_bookmark20)
   2. [Les avantages des ACL 13](#_bookmark21)
2. [ACL : entrante ou sortante 13](#_bookmark22)
3. [La correspondance des paquets 14](#_bookmark23)
4. [Les types d’ACLs 17](#_bookmark24)
   1. [Les ACLs standards 17](#_bookmark25)
   2. [Les ACLs étendus 18](#_bookmark26)
   3. [Les ACLs nommées 21](#_bookmark27)
5. [Placement des ACLs 22](#_bookmark28)
6. [Conclusion 22](#_bookmark29)

[Références 22](#_bookmark30)

# Partie 1 : Services applicatifs

**Introduction :**

Les services applicatifs, également appelés services d'application, jouent un rôle essentiel dans le domaine de l'informatique en fournissant des fonctionnalités et des solutions logicielles aux utilisateurs et aux entreprises. Ces services sont conçus pour répondre à des besoins spécifiques liés aux applications informatiques et contribuent à optimiser l'efficacité, la productivité et la convivialité des systèmes informatiques.

Les services applicatifs englobent généralement divers aspects, allant de la conception et du développement à la mise en œuvre, la maintenance et l'amélioration continue des applications logicielles.

1. **Services applicatifs : importance et utilisation**

## Pourquoi les services d’applicatifs sont-ils importants ?

Les services applicatifs sont au cœur de la transformation numérique, facilitant l'innovation, l'efficacité et la collaboration tout en répondant aux besoins changeants des utilisateurs et des entreprises. Leur importance continue de croître à mesure que la technologie évolue et que de nouveaux défis et opportunités émergent.

Voici quelques-unes des raisons qui soulignent l'importance cruciale des services applicatifs :

* + 1. **Répondre aux Besoins Utilisateurs** : Les services applicatifs sont conçus pour répondre aux besoins spécifiques des utilisateurs.
    2. **Automatisation des Processus** : Les services applicatifs automatisent de nombreuses tâches et processus, permettant aux utilisateurs de gagner du temps et d'optimiser leurs opérations.
    3. **Facilitation de la Communication et de la Collaboration** : Les applications collaboratives facilitent la communication et le travail d'équipe.
    4. **Accessibilité et Mobilité** : Les services applicatifs peuvent être accessibles à partir de divers appareils, ce permet de travailler à partir de n'importe quel endroit.
    5. **Gestion des Données** : Il facilite l’utilisation des bases de donner, ce qui permet d'organiser et d'analyser des volumes importants de données.
    6. **Sécurité Informatique** : La sécurité des services applicatifs est une priorité, surtout avec la montée en puissance des cybermenaces.
  1. **Comment fonctionnent les services d’application ?**

Les services applicatifs fonctionnent selon un ensemble de processus qui varient en fonction du type d'application, de son architecture, de la technologie utilisée et de l'environnement d'exécution.

Cependant, vous pouvez voir quelques étapes générales qui expliquent le fonctionnement de ces services.

Voici un exemple d’étape simplifié du processus :

1. **Conception et Développement** : Tout commence par la conception et le développement de l'application.
2. **Déploiement** : Une fois l'application développée, elle est déployée sur une infrastructure informatique.
3. **Accès Utilisateur** : Les utilisateurs accèdent à l'application à l'aide d'interfaces utilisateur telles que des applications mobiles, web ou des logiciels de bureau.
4. **Traitement des Requêtes** : Lorsqu'un utilisateur interagit avec l'application, des requêtes sont générées.
5. **Gestion des Données** : Les services applicatifs traitent souvent des données. Cela peut inclure le stockage, la récupération, la mise à jour et la suppression de données.
6. **Sécurité** : Les services applicatifs intègrent des mesures de sécurité pour protéger les données sensibles, prévenir les attaques et assurer la confidentialité des utilisateurs.
7. **Maintenance et Mise à Jour** : Les services applicatifs nécessitent une maintenance continue pour résoudre les bogues, améliorer les performances et ajouter de nouvelles fonctionnalités.

Les services applicatifs sont des entités complexes qui intègrent conception, développement, déploiement, gestion des données, logique métier et sécurité pour fournir des fonctionnalités spécifiques aux utilisateurs. Leur fonctionnement dépend fortement du contexte de l'application particulière et de ses exigences.

1. **Services Applicatifs Courants**

Il existe une variété de services applicatifs couramment utilisés dans divers domaines, de l'informatique personnelle aux solutions d'entreprise. Voici une liste de certains de ces services avec des exemples :

1. **Services de Développement d'Applications :**

Ces services englobent la conception, le codage, le test et le déploiement d'applications logicielles.

Exemples : AWS Lambda, Microsoft Azure Functions, Google Cloud Functions.

1. **Services de Sécurité des Applications :**

Ces services se concentrent sur la protection des applications contre les menaces, les attaques et les vulnérabilités.

Exemples : Veracode, Checkmarx, Fortify.

1. **Gestion des Performances des Applications (APM) :**

Ces services surveillent et mesurent les performances des applications en temps réel.

Exemples : Dynatrace, New Relic, AppDynamics.

1. **Services d'Applications d'Entreprise :**

Ces services répondent aux besoins spécifiques des entreprises en fournissant des applications personnalisées pour automatiser des processus, gérer des données et améliorer l'efficacité opérationnelle.

Exemples : SAP, Oracle E-Business Suite, Microsoft Dynamics 365.

1. **Services de réseau privé (VPN) :**

Un VPN est un service applicatif qui permet de créer une connexion sécurisée et chiffrée entre un utilisateur (ou un réseau local) et un serveur distant.

Exemples: OpenVPN, AnyConnect, IPsec (Internet Protocol Security)

1. **service applicative WAF:**

WAF, ou Web Application Firewall, est un autre type de service applicatif qui se concentre sur la sécurité des applications web.

Exemples : ModSecurity, Cloudflare WAF

1. **Tendances Émergentes et Futur des Services Applicatifs**

Les services applicatifs sont au cœur de l'innovation informatique, et leur évolution continue est façonnée par des tendances émergentes qui redéfinissent la manière dont les applications sont conçues, déployées et utilisées. Voici quelques-unes des tendances émergentes et des perspectives pour l'avenir des services applicatifs :

1. Informatique en Nuage Évoluée :

L'adoption croissante de l'informatique en nuage permet une plus grande flexibilité, une scalabilité accrue et une efficacité opérationnelle optimisée pour les services applicatifs.

L'intégration de technologies de conteneurisation comme Kubernetes facilite la gestion et le déploiement des applications à grande échelle.

1. Intelligence Artificielle (IA) et Automatisation :

L'intégration de l'IA dans les services applicatifs permet une automatisation plus avancée, améliorant la prise de décision, l'efficacité des processus et la personnalisation des expériences utilisateur.

Les chatbots, les assistants virtuels et d'autres solutions basées sur l'IA gagnent en popularité pour enrichir les fonctionnalités des applications.

1. Edge Computing :

La montée de l'informatique en périphérie (Edge Computing) amène le traitement des données plus près de leur source, réduisant la latence et améliorant la réactivité des services applicatifs.

Cela est particulièrement crucial pour les applications nécessitant une faible latence, comme la réalité virtuelle, les jeux en ligne et l'Internet des objets (IoT).

1. Interfaces Utilisateurs Évoluées :

Les interfaces utilisateur évoluent vers des expériences plus immersives, telles que la réalité augmentée (RA) et la réalité virtuelle (RV), intégrant de nouvelles dimensions aux services applicatifs.

1. Sécurité Renforcée :

Avec l'augmentation des cybermenaces, la sécurité des services applicatifs devient une priorité majeure.

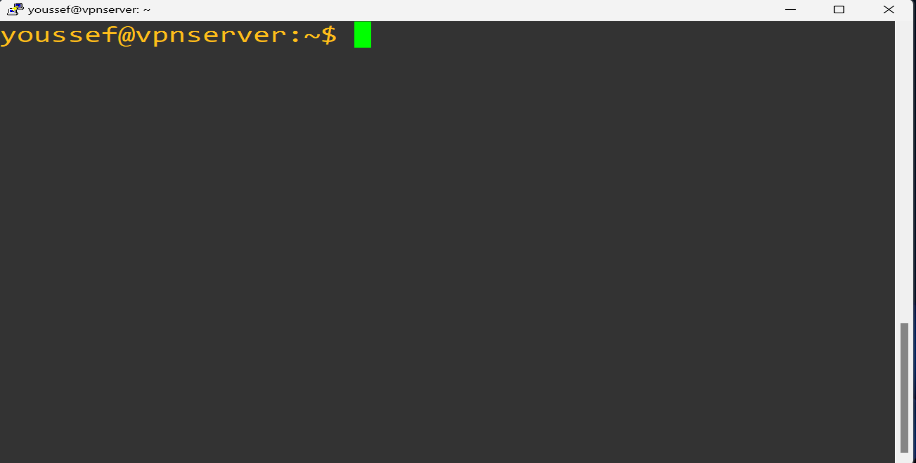
Des technologies telles que la confidentialité homographique, la confidentialité quantique et les nouvelles méthodes de chiffrement émergent pour renforcer la sécurité des données.

1. **Étape d’installation et Configuration**

## Étapes d'Installation d'OpenVPN :

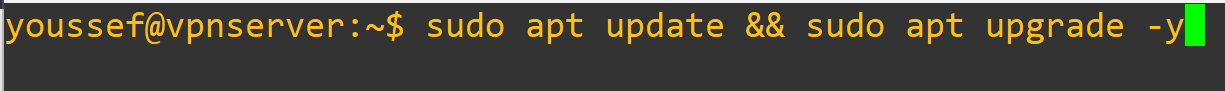
**OpenVPN** est une application logicielle open-source qui met en œuvre des techniques de réseau privé virtuel (VPN) pour créer des connexions sécurisées de point à point ou de site à site dans des configurations routées ou pontées, ainsi que des installations d'accès à distance. Il utilise un protocole de sécurité personnalisé qui exploite SSL/TLS pour l'échange de clés.

* + 1. Connexion au serveur :



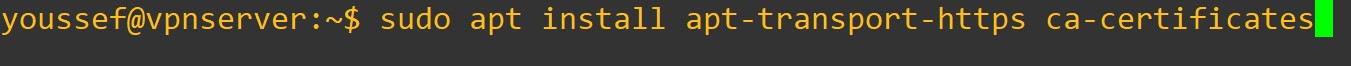
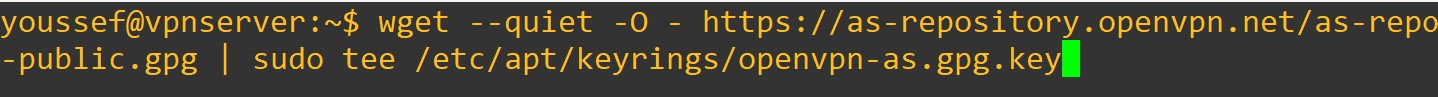
* + 1. Installation d’OpenVpn :
  1. b.1- Mise à jour le système :

Commencez par vous assurer que votre système est à jour



### 1.2.1 b.2- Ajouter le référentiel du serveur d'accès OpenVPN :

A

j o u t e

z ensuite le référentiel OpenVPN Access Server pour Ubuntu qui vous permet de récupérer et d'installer le logiciel.

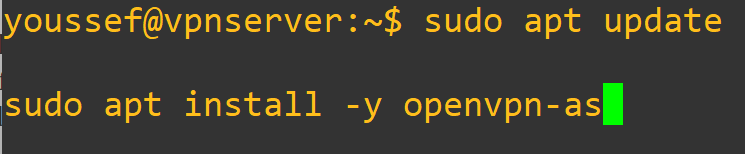
En installant ces packages, vous donnez à votre système la possibilité de récupérer en toute sécurité des packages à partir de référentiels HTTPS. Il s’agit souvent d’une étape nécessaire avant d’ajouter de tels référentiels à la liste des sources de votre système.

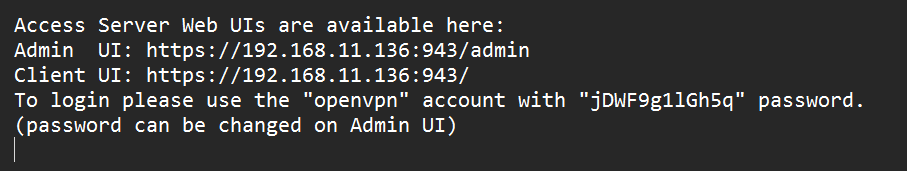
Enfin, actualisez la liste des packages de votre système Ubuntu et procédez à l'installation du serveur d'accès OpenVPN

## Étapes de configuration :

1. Configurer le compte administrateur :

Par défaut, OpenVPN AS crée un utilisateur nommé « openvpn » et génère de manière aléatoire un mot de passe pour le compte administrateur. Les URL du panneau d'administration et du panneau client seront également affichées comme indiqué ci- dessous.



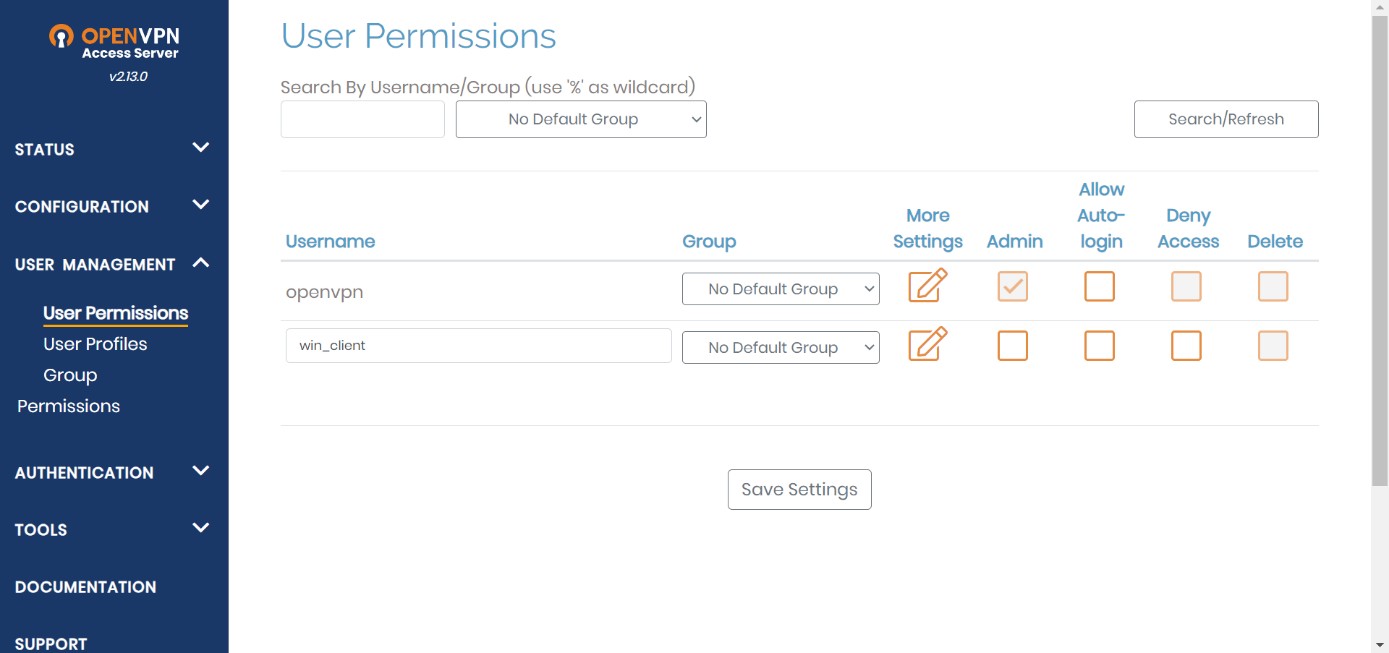


1. Accéder à l'interface Web :

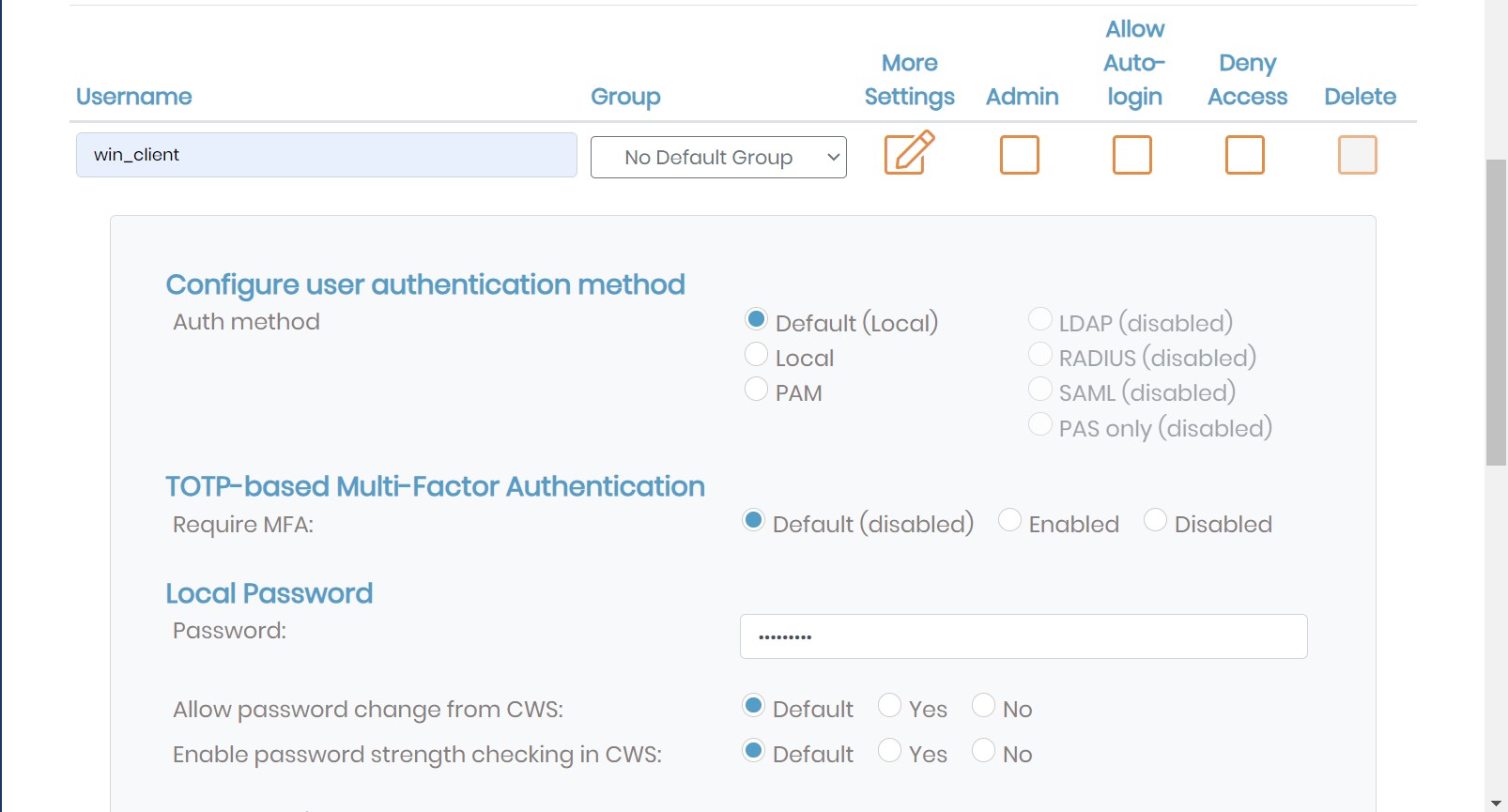
Une fois installé, OpenVPN AS démarrera automatiquement. Vous pouvez accéder à l'interface Web d'administration via votre navigateur Web : https://address\_du\_serveur:943/admin



1. Crée un nouveau client



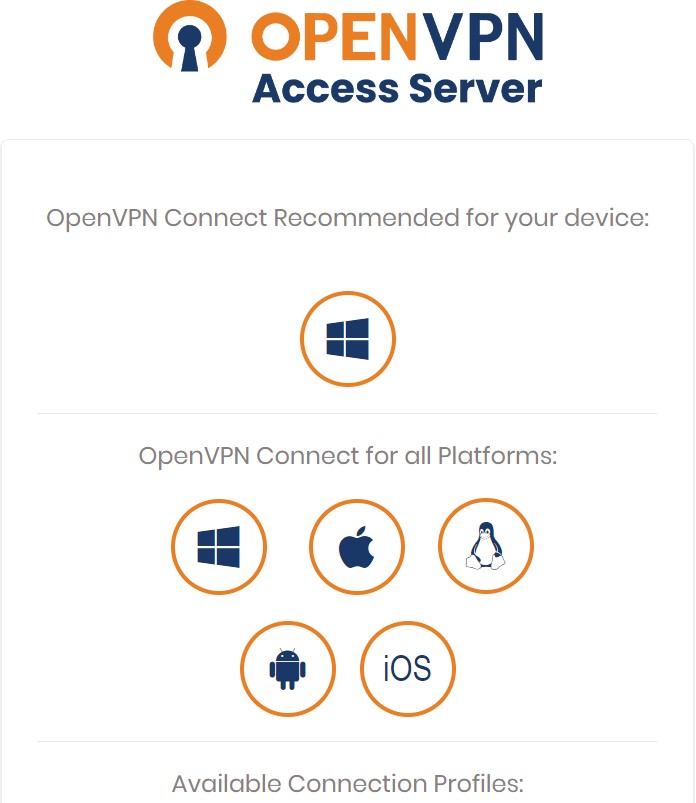
1. Configurer le mot de pass



1. Télécharger le logiciel client

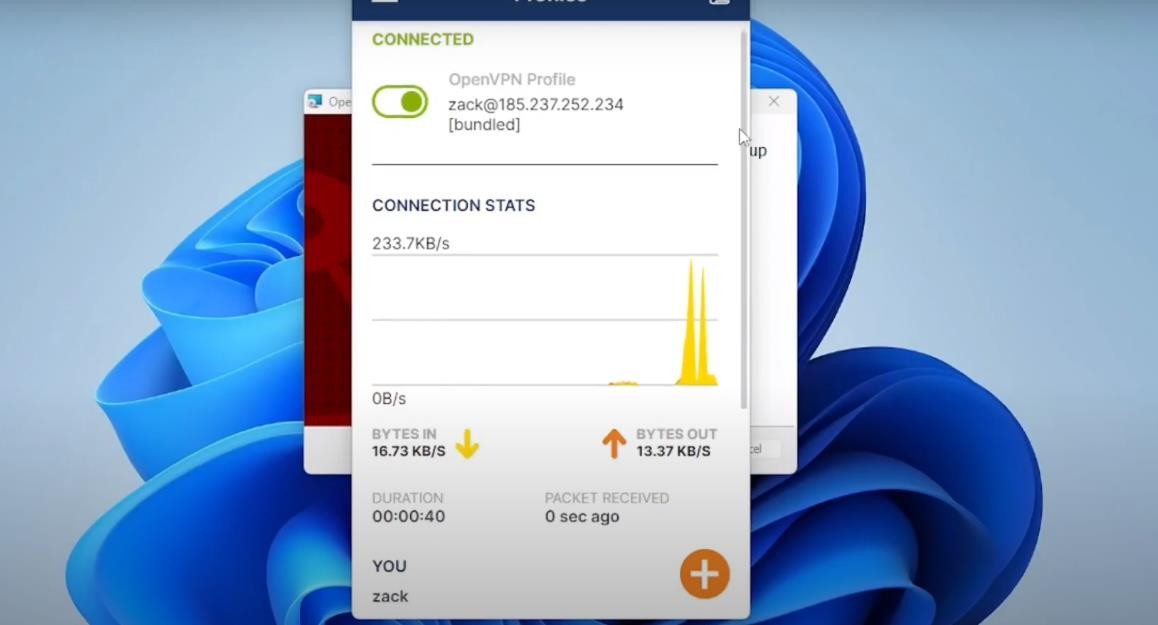
Pour se connecter à votre VPN, les utilisateurs auront besoin du client OpenVPN Connect. Il peut être téléchargé depuis : https://ip\_serveur:943

Une fois sur place, les utilisateurs peuvent se connecter et télécharger le client spécifique à leur système d'exploitation



1. Connectez-vous au VPN

Après avoir téléchargé et installé le client OpenVPN Connect, les utilisateurs peuvent simplement se connecter automatiquement au serveur. Ils utiliseront les informations d'identification que vous avez configurées dans la section Gestion des utilisateurs de l'interface d'administration.



# Partie 2 : Configuration des ACL (Access Control List)

**Introduction :**

Les ACL sont largement utilisées dans le domaine de la sécurité informatique pour réguler l'accès aux fichiers, aux dossiers, aux réseaux, aux périphériques, et plus encore. Elles permettent aux administrateurs de contrôler de manière précise qui peut accéder à quelles ressources et quelles opérations peuvent être effectuées sur ces ressources.

1. **Présentation**

## Définition des ACL :

Les **ACL**, pour ***A****ccess* ***C****ontrol* ***L****ist*, sont des règles appliquées aux trafics transitant via les interfaces du routeur que ce soit en entrée (*in*) ou en sortie (*out*). Les ACL filtrent le trafic en demandant aux interfaces d’acheminer ou non les paquets qui y transitent. Pour ce faire, le routeur lit l'en-tête de chaque paquet afin de déterminer s'il doit être acheminé ou non en fonction des conditions définies dans la liste de contrôle d’accès ACLs.

## Les avantages des ACL :

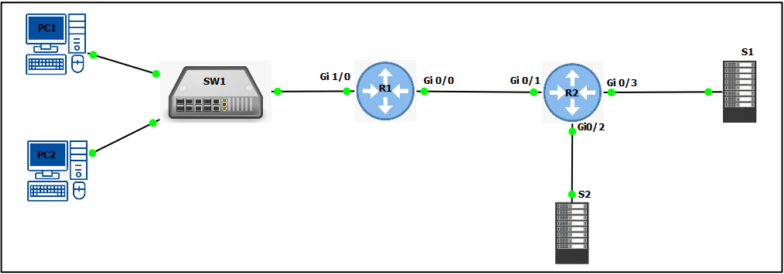
* Maitriser le réseau en déterminant quel type de trafic sera acheminé ou bloqué.
* Augmenter le niveau de sécurité d’accès réseau de manière basique en accordant ou non l’accès à un segment du réseau.
* Optimiser le réseau à d'autres fins que la sécurité, par exemple pour contrôler la bande- passante, restreindre le contenu des mises à jour de routage ou identifier et classer le trafic par fonctionnalités de qualité de service (QoS).

**Les ACL s'appliquent selon un ordre séquentiel en évaluant les paquets au début de la liste d’instruction**. Si le paquet répond aux critères de la première instruction (appelées ACE[2] pour *Access Control Entry*), il ignore le reste des règles. Pour reproduire ce tutoriel, vous pouvez utiliser du matériel virtuel [Cisco[](https://www.it-connect.fr/cisco-configurer-la-sauvegarde-automatique-avec-une-tache-kron/)1] sous Packet Tracer.

1. **ACL : entrante ou sortante**

Une question très importante que l'on peut se poser lorsque l’on veut mettre en place l’ACLs, c'est savoir sur quelle interface faut-il appliquer les ACLs ? L’interface entrante ou sortante du routeur ? Répondons à cette question dans cette première partie de l'article.

Les ACLs peuvent être associés à une interface particulière et pour une direction du flux (en entrée ou en sortie). En outre, ces règles de filtrages peuvent être appliquées avant que le routeur ne prenne sa décision de routage (interface en entrée), ce qui est un bon moyen d'économiser les ressources matérielles du routeur, ou après que le routeur ait pris sa décision de transfert et déterminé l’interface de sortie à utiliser pour acheminer le paquet.



Dans la topologie ci-dessus, l’interface entrante de **R1** est **Gi 1/0,** et l’interface sortante est **Gi 0/0**.

L’interface entrante de **R2** est **Gi 0/1** et l’interface sortante est **Gi 0/3**. Si par exemple, nous avons activé une ACL sur l’interface **Gi 0/2** de **R2**, cette ACL ne pourrait pas filtrer les paquets envoyés de PC2 au serveur S1, car les paquets ne passent pas l’interface **Gi 0/2**.

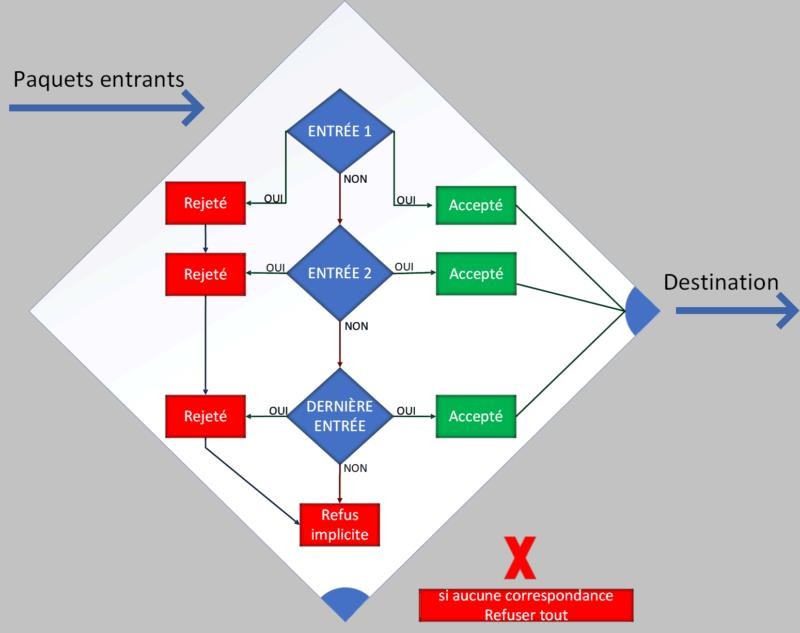
**En somme, pour filtrer un paquet, on doit activer une ACL sur l’interface qui traite le paquet.**

Si on active l’ACL sur **R1** pour les paquets entrants sur l’interface Gi1/0, R1 va **comparer chaque paquet entrant sur cette interface aux entrées de l’ACL afin de décider le sort de ce paquet** : continuer sans changement et acheminer le paquet ou le rejeter.

1. **La correspondance des paquets**

C’est la manière dont on configure les ACL, on indique les informations qui seront utilisées par le processus de filtrage, cela peut être **une adresse IP ou bien un protocole réseau**. En se basant sur ces informations, le routeur déterminera s’il doit autoriser on refuser les paquets.

Le diagramme ci-dessous illustre mes propos :



Lors du traitement de l’ACL, le routeur traite le paquet comme suit :

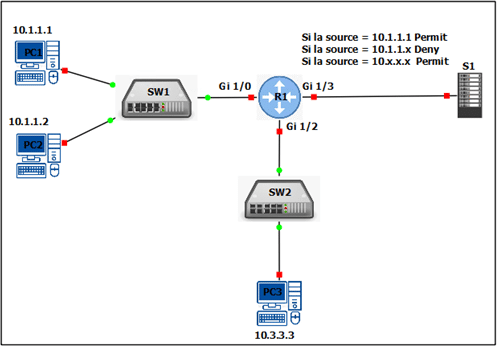
**Les ACLs utilisent la logique de première correspondance.** Une fois qu’un paquet correspond à une ligne dans l’ACL, le routeur exécute cette règle puis le processus s’arrête.

L’exemple ci-dessous nous permettra de voir ce que cela signifie précisément :

Sur l'exemple ci-dessous, l'ACL est appliquée sur l’interface **Gi1/3** et elle contient trois règles

:

* Si la source = 10.1.1.1, on autorise le paquet (*permit*)
* Si la source = 10.1.1.X, on refuse le paquet (*deny*)
* Si la source = 10.X.X.X, on autorise le paquet (*permit*



On considère qu’un paquet est envoyé par le **PC1 (10.1.1.1)** au serveur **S1**, le routeur **R1** compare ce paquet à l’ACL correspondant à la première ligne de l'ACL, le paquet sera autorisé, car l'adresse IP du PC correspond à celle définie dans l'ACL.

Ensuite, considérons qu’un paquet est envoyé par le **PC2 (10.1.1.2)** au serveur **S1.** À l’arrivée du paquet, le routeur va poursuivre la même logique de recherche en comparant le paquet à la première ligne de notre ACL. Il ne fera pas une correspondance, car 10.1.1.2 (adresse IP de PC2) n'est pas égal à l'adresse IP définie dans la première règle (10.1.1.1). Donc, R1 passe ensuite à la deuxième instruction à savoir "*10.1.1.x deny*", où **x** correspond à n’importe quelle valeur qui peut exister dans le dernier octet. En ne comparant que les trois premiers octets, R1 réalise que notre paquet émis depuis PC2 a une adresse IP source qui correspond à "10.1.1". De ce fait, R1 considère qu’il s’agit bien d’une correspondance à la deuxième règle de notre ACL et applique l’action comme indiqué dans la règle, à savoir refuser le paquet (*deny*).

**Le routeur R1 arrête également le traitement ACL sur ce paquet, en ignorant la troisième règle de l’ACL.**

Finalement, admettons que le PC3 (10.3.3.3) veut envoyer un paquet au serveur S1. R1 examine les deux premières règles dans l’ACL, et il voit qu’elles ne correspondent pas à l'adresse IP 10.3.3.3 (adresse IP source du paquet), donc il poursuit l’exécution de l’ACL. De ce fait, il regarde la troisième règle qui s'applique à toutes les adresses IP sous la forme "10.X.X.X", ce qui correspond à l'adresse IP de PC3. La troisième règle de l'ACL s'applique, donc le paquet est autorisé et poursuit son chemin vers le serveur S1.

**Si jamais un paquet ne correspond à aucune des règles de la liste ACLs, le paquet est rejeté. On parle d'un refus implicite.**

1. **Les types d’ACLs**

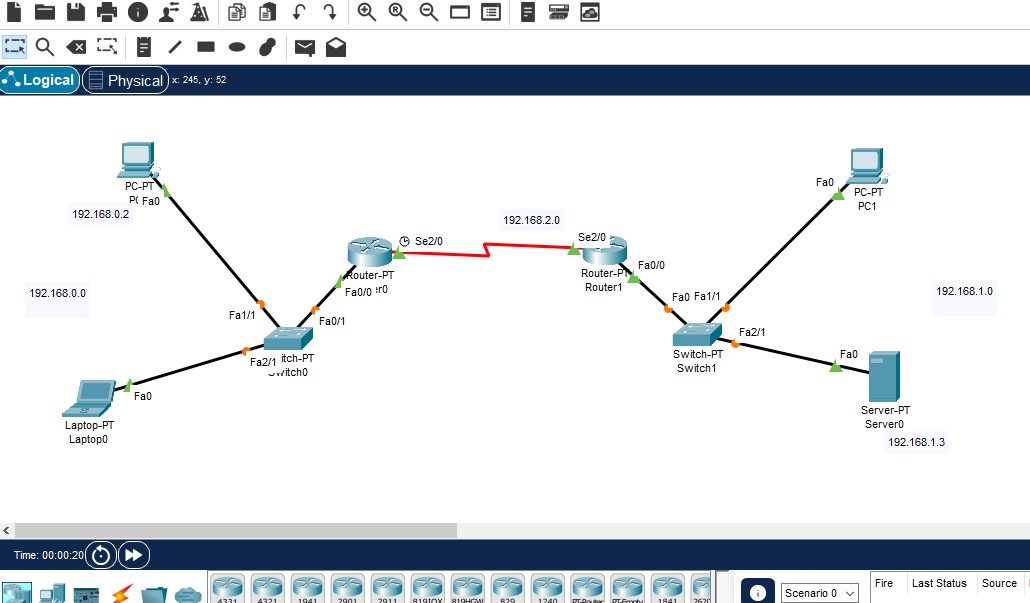
Il existe plusieurs types d'ACL, que nous allons découvrir ensemble sans plus attendre.

## Les ACLs standards

Dans ce type, l’ACL ne peut être liée qu'à l’adresse IP source du paquet. Ces ACLs sont identifiables par identifiant correspondant à un nombre allant de **1 à 99** et de **1300 à 1999**.

Nous pourrons utiliser ce type d'ACL pour autoriser ou interdire un segment du réseau ou l'adresse IP d’une machine à communiquer avec un autre segment de réseau ou une autre machine.

Afin de concrétiser la notion d’ACL standard, nous allons les mettre en place sur un routeur [Cisco.](https://www.it-connect.fr/cours-tutoriels/administration-reseau/cisco/)



Dans la configuration ci-dessus il y a deux reseaux different communiquent entre

eux « 192.168.1.0 **» et** « 192.168.0.0**» le reseau 1 contient le pc 1 le serveur 0 et le reseau 0 contient le pc 0 le laptop 0 ,** Le but est d’interdire au PC1« 192.168.1.2**»** de communiquer avec le Laptop 0 « **192.168.0.3»** pour ce faire, nous allons, sur l’interface **S2/0** du routeur 0 Pour parvenir aux résultats souhaités, nous appliquons trois étapes :

1 - Après s’être connecté sur le routeur en mode de configuration globale en tapant les commandes "*enable"* puis "*configuration terminal",* on commence par la création de la règle :

Router(config)#access-list 1 deny 192.168.0.3

En précisant "access-list 1" on attribue **un ID à notre ACL**, puis ensuite on précise que l'on veut refuser avec "deny", et enfin on précise l'adresse IP de destination

Deuxièmement, il faut autoriser explicitement les autre périphériques de réseau que l’on veut laisser passer,

Router(config)#access-list 1 permit any

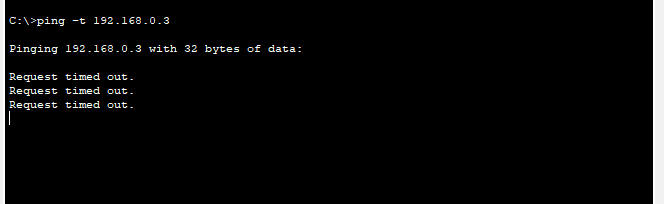
Troisièmement, on sélection l’interface concernée par la règle :

Router(config)#interface S2/0

Enfin, on applique la règle en sortie

Router(config-if)# ip access-group 1 out

On test la connectivité par ping –t pour l’enregistrement ne s’arrête pas on constate qu’on a bloqué la connectivité



## Les ACLs étendus

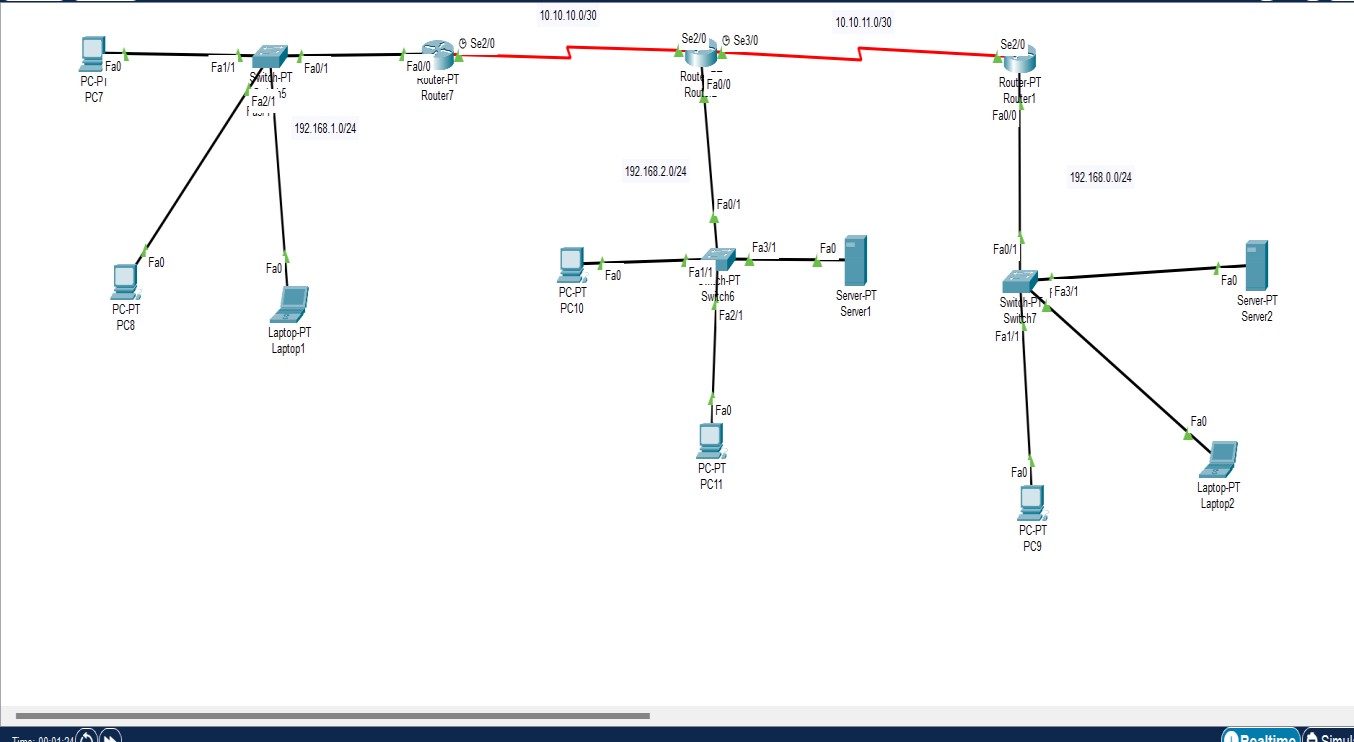
Les ACL étendues présentent plusieurs similitudes par rapport aux ACL Standards décrites dans la section précédente. Tout comme une ACL standard, **on active les ACL étendues sur les interfaces pour les paquets entrants ou sortants**, puis le routeur cherche dans la liste de manière séquentielle.

Les ACL étendues utilisent également la logique de première correspondance, car dès que la première instruction est mise en correspondance, le routeur arrête la recherche dans la liste d'ACL, en effectuant l’action définie.

En comparaison des ACL standards, **les ACLs étendues vont permettre d'analyser une plus grande variété de champs au sein de l'en-tête d'un paquet**. Cela rend les ACL étendues plus puissantes, plus précises, mais aussi un peu plus complexes.

Les ACL étendues suivent la même logique que les ACL standards, elles sont identifiables par un numéro, allant de **100 à 199** et de **2000 à 2699**.

Un exemple sera plus parlant, nous allons créer une règle qui aura pour but d’**interdire le** [**Ping[**](https://www.it-connect.fr/le-ping-pour-les-debutants/)**3] de PC2 vers le PC3, tout en l'autorisant vers le PC1,** en posant les règles sur les sous-réseaux (tous en /24). Mettons ça en place en reprenant la même topologie :



Voici la configuration à appliquer sur notre routeur2 Cisco :

Router2(config)# access-list 100 deny icmp 192.168.1.0 0.0.0.255 host 192.168.2.4 eq 80

Router2(config)# access-list 100 permit icmp 192.168.1.0 0.0.0.255 host 192.168.2.4 eq 80

Router2config)#interface se 2/0 Router(config-if)#ip access-group 100 in

Dans l'exemple ci-dessus, "**icmp[4]**" correspond au protocole, en l'occurrence ici c'est un moyen de bloquer le ping. Ensuite, "192.168.1.0" c'est l'adresse IP d'origine suivie de son masque générique, et "192.168.2.4" l'adresse IP de destination.

Il est à noter que les ACLs étendues peuvent aussi examiner des parties d'en-têtes TCP ou UDP, en particulier les champs qui contiennent **le numéro de port source et port de destination**. Les numéros de port identifient le service qui envoie ou reçoit les données. Quand le mot "***tcp[5]***" ou "***udp[6]***" est inclus dans la règle de l'ACL, cela permet de préciser le port source et le port de destination afin d'avoir une règle plus précise.

Voici un exemple :

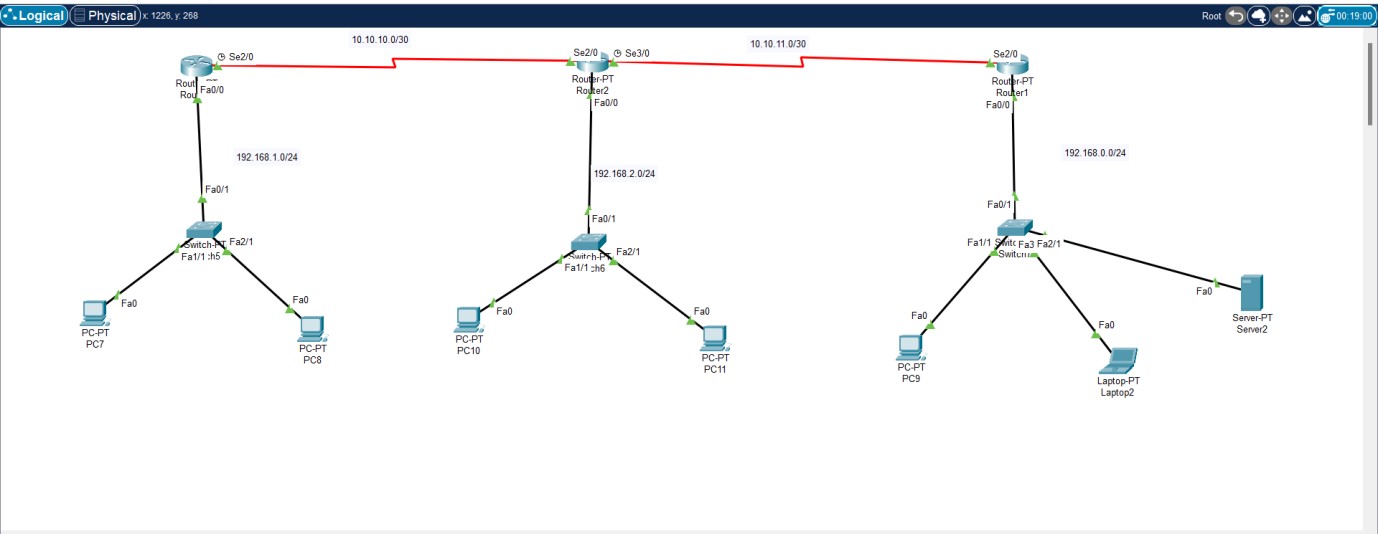
Router2(config)# access-list 100 permit tcp 192.168.1.0 0.0.0.255 host 192.168.2.2 any eq 23

Concernant « **eq »** il s’agit d’un opérateur qui veut dire égal, et le numéro de port 23 utilisé par le serveur Telnet. On autorise le réseau 192.168.1.0/24 à communiquer sur le port 23 (pour le Telnet) avec la machine 192.168.2.2.

Prenons un autre exemple...

L’exemple suivant se concentre sur la compréhension de la syntaxe de base. Ici, l'objectif est de créer une ACL pour empêcher l'utilisateur David d’utiliser le protocole **Telnet** pour se connecter sur les serveurs. Pour cela, on va le bloquer en entrée sur l’interface **Fa0/0** de **R3**, car c’est l’interface la plus proche de la source, donc l’ACL sera associée à cette interface. Quant à l'utilisateur Florian, on va lui interdire l’accès au **serveur Web de serveur1**. Dans cet exemple, on active l’ACL sur **R2** en entrée de l'interface **Fa0/0**.

Voici la topologie :



Je vous propose que l'on se connecte sur **R2** afin de créer la première règle.

La première chose à faire, c’est interdire David d’accéder à tous les serveurs Telnet :

access-list 101 deny tcp host 192.168.2.2 192.168.0.0 0.0.0.255 eq 23

Au sein de la commande ci-dessus, on précise bien l'hôte de David avec "**host 192.168.2.2**, car cela correspond à la source à bloquer selon notre schéma initial.

La deuxième règle va permettre d'autoriser tout le reste du trafic :

access-list 101 permit ip any any

On applique l'ACL à l'interface concernée :

R2(config)# interface fa 0/0

R2(config-if)#ip access-group 101 in

Interdisons maintenant à Florian d’accéder au serveur Web :

access-list 101 deny tcp host 172.16.2.10 host 172.16.1.100 eq www

Encore une fois, on autorise tout le reste du trafic ensuite :

access-list 101 permit ip any any

Après avoir créé les règles d'ACL, il nous reste plus qu’à les appliquer toujours à partir du routeur R3 :

R3(config)# interface fa 0/0 R3(config-if)#ip access-group 101 in

Voilà, mission accomplie !

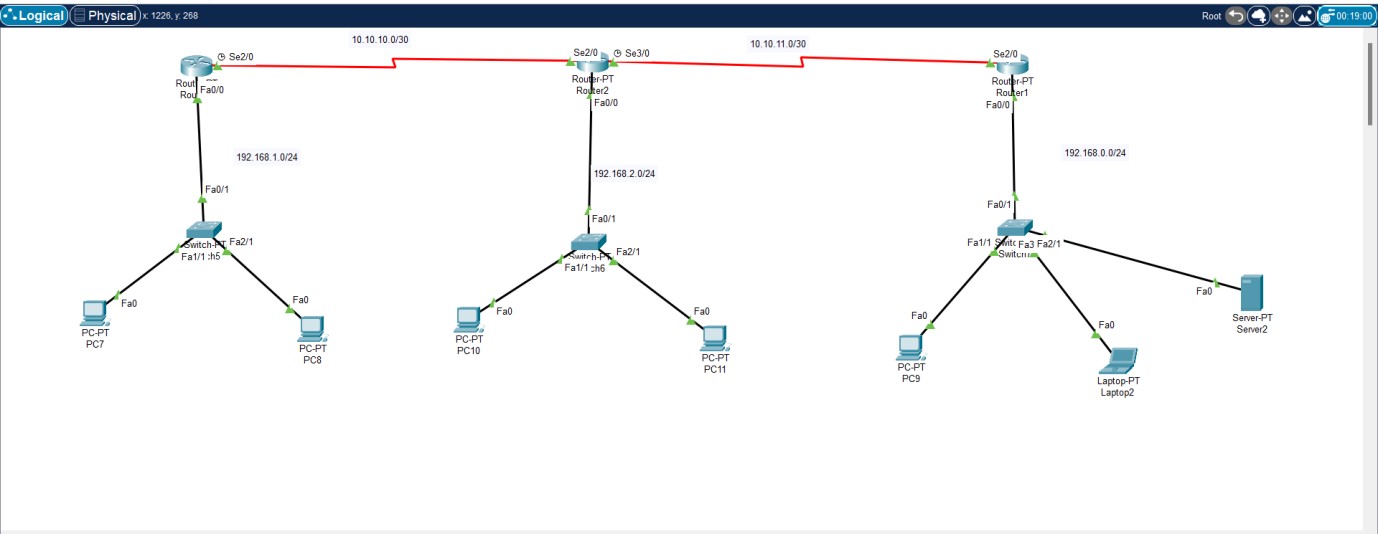
## Les ACLs nommées

Comme vous le savez, toutes les listes d’accès doivent être identifiées par un nom ou un numéro. Ce type de liste d’accès est plus pratique, car on peut spécifier un nom significatif qui est facile à retenir et associer à une règle.

Les ACL nommées peuvent correspondre aux mêmes champs qu’une ACL standard et étendue, cependant, elles présentent trois grandes différences par rapport aux listes de contrôles d’accès numérotées, voyons cela ensemble :

* L’utilisation de noms au lieu de chiffres pour identifier la liste ACL facilite la mémorisation et l'identification de l'ACL.
* Utilisation de sous-commandes ACL, et non de commandes globales, pour définir l'action et les paramètres correspondants.
* Utilisation des fonctionnalités d'édition de l'ACL qui permettent de supprimer des lignes individuelles de l'ACL et d'insérer de nouvelles instructions à une liste d'accès nommée.

On va reproduire le même schéma :



Cette fois-ci, on va changer un peu les règles du jeu : le but sera d’interdire David d’accéder au serveur Web (Serveur1) via le protocole http. On aura besoin donc d’une ACL qui contiendra plusieurs informations : IP source, IP destination et un protocole bien précis. Pas le choix, il faudrait utiliser une ACL étendue.

Profitons de cet exemple pour utiliser une ACL nommée et poursuivre notre découverte des ACLs.

Sur le R1, tout d’abord, on crée l’ACLs avec le nom "etd" : Router(config)#ip access-list extended etd

Puis, on se retrouve dans cette ACL, donc on ajoute notre règle :

Router(config-ext-nacl)#deny tcp 192.168.2.0 0.0.0.255 host 172.16.0.4 eq 80

Vous constatez bien que le prompt a changé, car on se trouve dans l’invite de configuration de l’ACL nommée. On pose une interdiction via la directive "**deny**" depuis l’adresse source "**192.168.2.10**" vers le serveur web avec l'adresse IP "**172.16.0.4**". **« 80»** pour **<<http>>** c’est pour faire référence au service web, mais on peut bien évidement mettre **www** au lieu de port 80 .

En dernier lieu, on autorise toutes les autres connexions, car souvenez-vous, la directive implicite (deny any ) se trouve à la fin de chaque ACL !

Router(config-ext-nacl)# permit ip any any

Il ne reste plus qu’à associer l’ACL à l’interface G0/0 de R2 comme vous savez le faire maintenant. Vous pouvez retrouver d'autres exemples sur le [site de Cisco.](https://www.cisco.com/c/en/us/support/docs/ip/access-lists/26448-ACLsamples.html)

1. **Placement des ACLs**

Bon, nous savons maintenant qu'il y a deux types d'ACL : standards et étendues. On sait aussi qu'il y a deux possibilités pour les placer : en entrée ou en sortie d'une interface. Mais alors, où placer tel type d'ACL? Sur quel routeur? Quelle interface?

Dans tous les cas, il existe une règle simple et immuable : **une ACL standard se placera toujours au plus proche de la destination, une ACL étendue se placera toujours au plus proche de la source.**

**Pourquoi?** Tout simplement, car une ACL standard, comme vous avez pu le constater, est très restrictive (blocage de tout le trafic d'un réseau ou d'un hôte), un placement au plus proche de la source risque de limiter fortement les possibilités de celles-ci. En revanche, l'ACL étendue filtrant au niveau de la couche 4 (ports TCP/UDP), son placement au plus proche de la source permet d'éviter de faire transiter des paquets qu'on ne souhaite pas voir sur le réseau, ainsi, on économise du temps de calcul.

1. **Conclusion :**

Les ACL permettent un filtrage au niveau du réseau et peuvent être complétées avec le filtrage applicatif (Proxy) qui va venir **au niveau de la couche Application du modèle OSI[7].** Le filtrage de paquet via **ACLs opère niveaux 3 et 4 du modèle OSI**, ce qui permet de faire déjà beaucoup de choses, mais peut ne pas suffire dans certains cas.

**Références**

1. [Cisco : configurer la sauvegarde automatique avec une tâche Kron | IT-Connect](https://www.it-connect.fr/cisco-configurer-la-sauvegarde-automatique-avec-une-tache-kron/)
2. [Access Control Entries - Win32 apps | Microsoft Learn](https://learn.microsoft.com/en-us/windows/win32/secauthz/access-control-entries)
3. [Ping : définition (journaldunet.fr)](https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203401-ping-definition/#%3A~%3Atext%3DLe%20PING%20d%C3%A9signe%20%C3%A9galement%20le%20nom%20d%27une%20commande%2Cround-trip%20time%20pour%20%C3%A9voquer%20ce%20temps%20de%20requ%C3%AAte-r%C3%A9ponse%29)
4. [What is ICMP? | Internet Control Message Protocol | Cloudflare](https://www.cloudflare.com/en-gb/learning/ddos/glossary/internet-control-message-protocol-icmp/)

**[5][6]**[UDP (User Datagram Protocol) : qu'est-ce que c'est, UDP vs TCP (vasexperts.com)](https://vasexperts.com/fr/resources/glossary/udp/#%3A~%3Atext%3DProtocole%20de%20datagramme%20utilisateur%20%28UDP%29%20est%20un%20protocole%2C%C3%A0%20faible%20latence%20entre%20les%20applications%20en%20ligne)

**[7]**[Modèle OSI — Wikipédia (wikipedia.org)](https://fr.wikipedia.org/wiki/Mod%C3%A8le_OSI)