



# Contrôle d'accès : TP - ACL

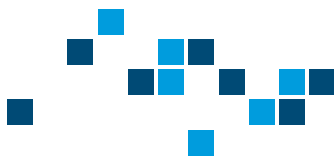
*BLUEM Juliette - SEZNEC Lucas - MAIMBOURG Gaston*

9 juin 2022



**UNIVERSITÉ  
DE LORRAINE**

**LORRAINE INP**  
les talents se lèvent à l'Est



## 1 Introduction

Les ACL (Access Control List) sont un mécanisme de contrôle d'accès à une ressource, désignant une liste d'adresses ou de ports autorisés ou interdits.

Les Access Control List sont divisés en deux grandes catégories, l'ACL standard et l'ACL étendue. Basiquement, l'ACL standard ne contrôle que l'adresse source, alors que l'ACL étendue peut également contrôler l'adresse de destination, le type de protocole (TCP, UDP, ICMP, IGRP, IGMP, etc.), le port source et destination, les flux TCP, IP TOS (Type of service) ainsi que les priorités IP.

Les ACL conviennent bien à des protocoles dont les ports sont statiques et connus à l'avance, comme pour des protocoles classiques (FTP, HTTP, SMTP...), mais ne suffisent pas toujours, comme pour des logiciels où les ports peuvent varier (P2P par exemple).

Une ACL est constituée d'Access Control Entries (ACE) qui définissent chacune une autorisation ou une interdiction pour un critère donné. L'ordre des ACE au sein de l'ACL est primordial, car l'équipement filtrant appliquera ces règles dans l'ordre à chaque paquet jusqu'à ce qu'un critère corresponde à la règle. L'autorisation ou l'interdiction liée à cette ACE est alors appliquée, sans consulter le reste de l'ACL.

Afin de prendre en main ce mécanisme de contrôle d'accès, nous mettons en place une topologie très simple composée d'un routeur et de deux hôtes.

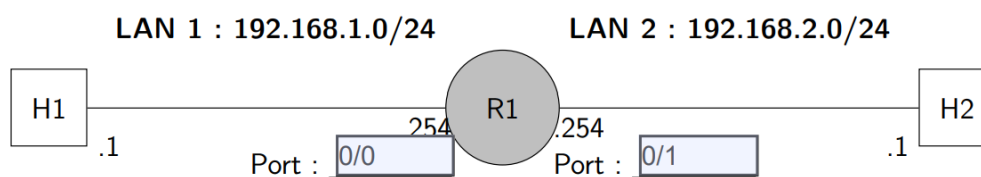


FIGURE 1 – Topologie préliminaire

Nous configurons un accès telnet sur R1. Cet accès nous servira plus tard lors de ce TP.

## 2 ACL Standard

Nous allons mettre en place des ACL standard sur notre routeur R1.

Tout d'abord, une ACL 10 bloquant uniquement l'adresse de l'hôte H1, en entrée sur le port LAN 1. Toutes les autres adresses sources doivent-être autorisées. Pour cela, nous utilisons les ACE suivantes (dans l'ordre) : autoriser tout le monde (par défaut), interdire l'IP de l'hôte H1.

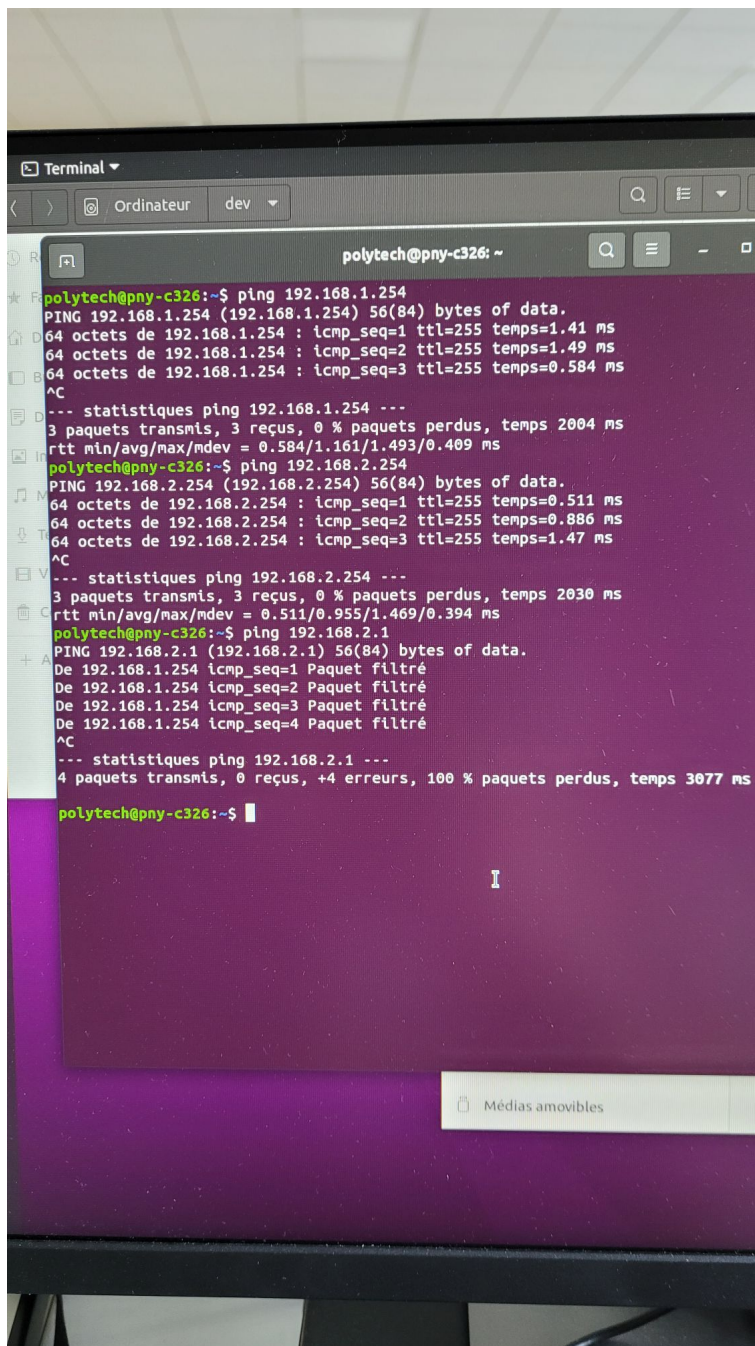
Grâce à cette ACL, H1 ne peut pas communiquer avec H2, ni même avec R1.

En retirant l'ACL et en appliquant l'ACL 10 en sortie sur le port LAN 2 de notre routeur R1, H1 accède aux interfaces du routeur.

Nous supprimons maintenant notre ACL et en créons une nouvelle. Cette ACL 20 doit permettre uniquement à l'adresse 192.168.2.1 d'accéder au LAN 1.

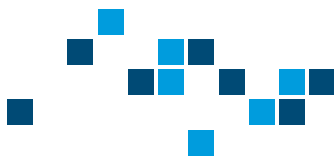
Sur l'interface 0/1, en sortie, elle utilise les ACE suivantes : refuser tout, permettre 192.168.2.1.

Cette fois encore, l'hôte H1 a accès au routeur, mais pas à H2. Seulement, le message de retour de ping est différent : "Paquet filtré" au lieu d'une simple perte de données.



```
polytech@pny-c326:~$ ping 192.168.1.254
PING 192.168.1.254 (192.168.1.254) 56(84) bytes of data:
64 octets de 192.168.1.254 : icmp_seq=1 ttl=255 temps=1.41 ms
64 octets de 192.168.1.254 : icmp_seq=2 ttl=255 temps=1.49 ms
64 octets de 192.168.1.254 : icmp_seq=3 ttl=255 temps=0.584 ms
^C
--- statistiques ping 192.168.1.254 ---
3 paquets transmis, 3 reçus, 0 % paquets perdus, temps 2004 ms
rtt min/avg/max/mdev = 0.584/1.161/1.493/0.409 ms
polytech@pny-c326:~$ ping 192.168.2.254
PING 192.168.2.254 (192.168.2.254) 56(84) bytes of data:
64 octets de 192.168.2.254 : icmp_seq=1 ttl=255 temps=0.511 ms
64 octets de 192.168.2.254 : icmp_seq=2 ttl=255 temps=0.886 ms
64 octets de 192.168.2.254 : icmp_seq=3 ttl=255 temps=1.47 ms
^C
--- statistiques ping 192.168.2.254 ---
3 paquets transmis, 3 reçus, 0 % paquets perdus, temps 2030 ms
rtt min/avg/max/mdev = 0.511/0.955/1.469/0.394 ms
polytech@pny-c326:~$ ping 192.168.2.1
PING 192.168.2.1 (192.168.2.1) 56(84) bytes of data:
De 192.168.1.254 icmp_seq=1 Paquet filtré
De 192.168.1.254 icmp_seq=2 Paquet filtré
De 192.168.1.254 icmp_seq=3 Paquet filtré
De 192.168.1.254 icmp_seq=4 Paquet filtré
^C
--- statistiques ping 192.168.2.1 ---
4 paquets transmis, 0 reçus, +4 erreurs, 100 % paquets perdus, temps 3077 ms
polytech@pny-c326:~$
```

FIGURE 2 – Résultat ping depuis H1



Nous créons une nouvelle ACL 30 sur R1. Cette ACL doit permettre à tout le LAN 2 d'accéder au LAN 1. Cette ACL doit donc faire en sorte que tout autre réseau, par exemple le réseau 192.168.3.0/24, ne doit pas pouvoir contacter le LAN 1. Mais on ne veut pas d'une ACL à rallonge qui bloque tous les réseaux existants. L'ACL 30 utilise les ACE suivant : refuser tout le monde, autoriser le réseau 192.168.2.0 255.255.255.0.

Imaginons que nous voulons restreindre l'accès FTP vers votre hôte H2 pour des raisons de sécurité. Par exemple s'il s'agissait d'un serveur de dépôt de fichier pour les employés d'une entreprise. Mais nous voulons autoriser d'autres types de trafic vers votre réseau interne.

Dans ce cas, nous serions limités par les ACL standard. En effet, nous ne pouvons pas faire de distinctions dans les données échangées. Autrement dit, soit nous bloquons tous les échanges entre deux hôtes, soit nous les autorisons. C'est justement en cela que de les ACL étendues sont utiles ! Grâce à cette technologie, nous pouvons spécifier quels protocoles sont permis ou refusés.

### 3 ACL étendues

Nous avons pu remarquer précédemment les limitations des ACL standards, nous allons donc étudier ici plus en détail les ACL étendues.

Après avoir supprimé toutes nos ACL standards, nous mettons en place une ACL 110 sur R1. Cette ACL doit bloquer toute communication à destination de H1. Pour cela, nous n'avons qu'une ACE : bloquer ip 0.0.0.0 (toutes les ip) vers h1. Nous l'appliquons à l'interface 0/0 en sortie.

Nous créons une nouvelle ACL 120 sur R1. Cette ACL doit bloquer tout trafic ICMP à destination de H1. C'est donc maintenant que nous allons voir l'intérêt des ACL étendues.

L'ACE est une adaptation de l'ACE utilisée pour l'ACL 110, nous lui précisons simplement le protocole ICMP. Nous l'appliquons encore une fois sur l'interface 0/0 en sortie.

Cette méthodologie peut s'appliquer également sur les protocoles TCP, UDP par exemple.

### 4 ACL pour connexion distante

Les lignes TTY (TeleTYpe) sont les connexions aux terminaux physiques, tels que le port console. En configuration, il apparaît sous "line con 0". Les VTY (Virtual Terminal Lines) sont une sorte de TTY implémentées en logiciel, et donc virtuelles. Elles servent pour les connexions à distance de type Telnet ou encore SSH. En configuration, elles apparaissent sous la forme "line vty 0 4", ce qui signifie que les lignes 0 à 4 sont définies. On peut changer le nombre de sessions virtuelles, contrairement aux connexions physiques qui sont limitées par les ports de l'équipement.

Après avoir supprimé toutes nos ACL étendues, nous créons une ACL 50 qui bloque tout trafic, et nous l'appliquons dans les VTY 2 à 4. Puis, via une ACL 51, nous faisons en sorte que la VTY 1 soit toujours réservée à notre hôte H1. Enfin, nous n'appliquons aucune ACL à la VTY 0.

Les ACE utilisées sont les suivantes : refuser tout pour l'ACL 50 et autoriser l'IP de H1 pour l'ACL 51.

Pour nos tests, nous établissons une connexion telnet de H2 vers R1. Celle-ci est autorisée : c'est la connexion numéro 1, elle est rattachée au TTY0. Sur TTY0, tout le monde peut se connecter. Ensuite, tout en maintenant la première connexion ouverte, nous essayons d'en ouvrir une seconde entre H2 et R1. Cette connexion nous est refusée, en effet cette connexion est rattachée à TTY1. Sur TTY1, seul H1 peut se connecter. C'est pourquoi, si l'échange telnet entre H2 et R1 reste ouvert et que nous essayons d'en ouvrir une entre H1 et R1, cette dernière est autorisée. Nous ne pouvons pas en ouvrir d'autres car les TTY2 à TTY4 sont complètement bloquées.