Le protocole SMB

I. Présentation

.

Le **protocole SMB**, pour ***S****erver****M****essage****B****lock*, est un **protocole client-serveur** qui permet d'**accéder à des ressources via le réseau**, et particulièrement l'**accès à des fichiers et des dossiers**. Par exemple, si vous utilisez partagez des fichiers avec une machine Windows et que vous accédez à ses fichiers depuis une autre machine, la communication sera effectuée via le protocole SMB. C'est également le cas si vous montez un lecteur réseau vers ce partage. On retrouve essentiellement **le protocole SMB sur les réseaux locaux**, car il a été pensé pour cet usage.

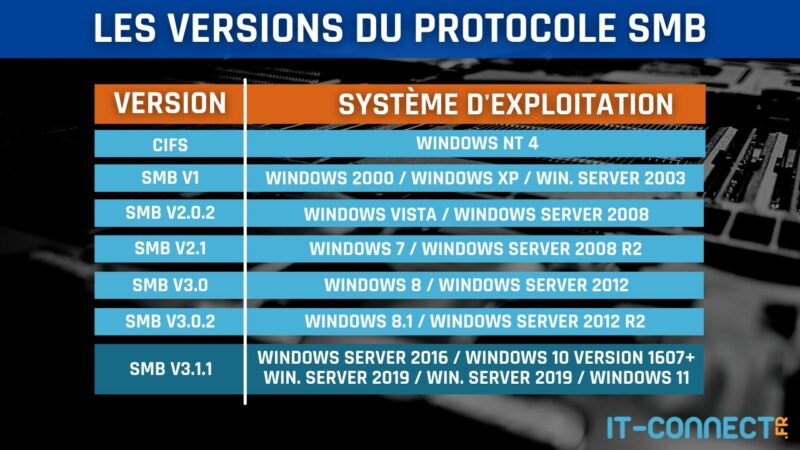
Le protocole SMB est implémenté dans Windows, mais il est également disponible sous**Linux et son implémentation se nomme SAMBA**.

II. Les différentes versions de SMB

Le protocole SMB n'est pas nouveau, je dirais même qu'il n'est pas récent puisqu'**il existe sous Windows depuis Windows NT 4.0**. Une version de Windows sortie le 31 juillet 1996. Néanmoins, c'est bien IBM qui l'a créé en 1985 : tout cela ne nous rajeunit pas. Au moment de son intégration au sein de Windows, il a été renommé **CIFS** pour ***C****ommon****I****nternet****F****ile****S****ystem* et ce terme est aujourd'hui encore utilisé dans certains logiciels, il est donc à connaître également. Aujourd'hui, le protocole se nomme officiellement SMB.

Ensuite, Microsoft a fait évoluer ce protocole en même temps que Windows. **La première version nommée "SMB" a vu le jour avec Windows 2000.** Ensuite, SMB 2 (ou 2.0.2) a vu le jour lorsque Windows Vista est arrivé, puis cette version était toujours là sur Windows 7. En 2012, Microsoft a sorti SMB 3 en même temps que Windows 8 et [Windows Server](https://www.it-connect.fr/cours-tutoriels/administration-systemes/windows-server/) 2012. Depuis, le protocole a légèrement évolué sur les dernières versions de Windows.

**La version actuelle du protocole SMB est la version 3.1.1, disponible depuis Windows Server 2016 et Windows 10 version 1607.**

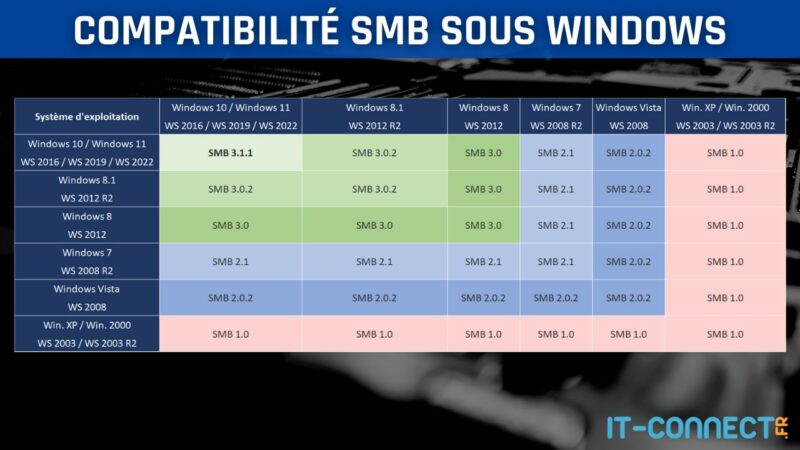


Lorsque Microsoft fait évoluer le protocole SMB, c'est pour renforcer la sécurité, notamment la sécurité des échanges avec des algorithmes de chiffrement plus robustes (AES-128-GCM avec SMB V3.1.1). Par ailleurs, la gestion de la mise en cache a évoluée et Microsoft a introduit des fonctions liées aux performances comme le SMB Direct et le SMB Multichannel (utile pour les stockages à hautes performances).

III. Compatibilité SMB pour la négociation client/serveur

En regardant la synthèse ci-dessus, on comprend vite que la version du protocole SMB implémentée dans Windows n'est pas la même d'une version à l'autre. Se pose alors la question de la compatibilité du protocole SMB entre toutes ces versions : **est-ce qu'une machine avec le SMB V2 peut communiquer avec une machine équipée du SMB V3 ?**La réponse est oui, lorsque la connexion sera négociée entre les deux hôtes, **un dialecte SMB commun sera sélectionné**.

**La version la plus faible sera retenue pour la communication SMB**. C'est ce qu'il faut retenir et pour que ce soit plus clair, voici un récapitulatif. Néanmoins, dans certains vous pouvez rencontrer des problèmes, comme nous allons le voir juste après.



IV. Le SMB v1 : une version vulnérable et à éviter

En production, le **SMB version 1 ne doit pas être utilisé, car il contient des failles de sécurité**. En 2017, une faille dans SMB v1 a fait beaucoup de bruit : **EternalBlue - CVE-2017-0144**. Elle a été exploitée pour compromettre des millions de machines dans le monde et elle a été utilisée par des ransomwares ravageurs comme WannaCry et Petya. Depuis, Microsoft a corrigé cette faille, mais cette version n'est plus maintenue.

Cette version est obsolète et dépréciée depuis 2014 ! **Microsoft a désactivé la prise en charge du SMB V1 au sein de Windows depuis Windows 10 version 1709 et Windows Server version 1709**. Néanmoins, il est possible de réactiver le SMB v1 dans les options de Windows ou à l'aide de PowerShell.

Cette réactivation du protocole SMB V1 sur un serveur de fichiers est nécessaire si vous avez des clients SMB qui ne supportent que le SMB V1. Par exemple, c'est le cas si vous utilisez des machines Windows XP ou si vous avez des copieurs qui ont de l'âge et que vous souhaitez effectuer du "scan vers partage".

**Si vous pouvez éviter le SMB V1, c'est mieux pour réduire la surface d'attaque de vos postes de travail et de vos serveurs.** Lorsque le SMB V1 est actif, c'est une porte d'entrée pour la propagation de malwares, notamment les ransomwares.

Sur son site, Microsoft explique comment désactiver SMB v1 sur les différentes versions de Windows.

V. Quels sont les ports utilisés par le protocole SMB ?

Pour fonctionner, le protocole SMB utilise le**port 139** ou le **port 445**. Il n'utilise pas les deux en même temps, voici plus d'informations :

* **SMB - Port 139**

Historiquement, sur Windows NT 4.0, pour transférer des fichiers via le protocole SMB il était nécessaire d'établir une connexion sur le **port 139**. Pour contacter l'hôte, le protocole SMB s'appuyait sur NetBIOS.

* **SMB - Port 445**

Le protocole SMB s'appuie sur une **connexion TCP** et le **port 445** pour établir une connexion sécurisée entre le client et le serveur. La résolution du nom pour établir la connexion s'appuie sur le DNS.

**Le port 445 est utilisé pour les connexions SMB sur tous les systèmes depuis Windows 2000**, et c'est encore le cas aujourd'hui avec Windows 10, Windows Server 2019, mais aussi les nouveaux systèmes Windows 11 et Windows Server 2022.

Je mentionne les deux ports, car en fonction de l'implémentation du protocole SMB, le port 139 peut être encore utilisé via NetBIOS même si le port 445 est présent dans la majorité des cas. Malgré tout, c'est bien de le savoir.

VI. Créer son premier partage et tester l'accès SMB

Pour finir, nous allons manipuler le protocole SMB. L'objectif est simple : **créer un partage sur une machine Windows et accéder à ce partage à partir d'une autre machine Windows**. Bien sûr, vous pouvez utiliser un NAS comme serveur et un PC sous Windows comme client, voire même une machine Linux.

Nous avons besoin de deux machines :

* **Une machine avec un partage, qui jouera le rôle de serveur au sein de la connexion SMB**
  + Système d'exploitation : Windows Server 2019
* **Une machine pour accéder aux données, qui jouera le rôle de client dans la connexion SMB**
  + Système d'exploitation : Windows 10
* **Les deux machines sont membres d'un même domaine Active Directory**

A. Créer un partage de fichiers sur le serveur

Sur le serveur Windows, commençons par créer le dossier à partager. Par exemple : "**C:\Partage**". Ensuite, il faut accéder aux propriétés du dossier pour créer le partage : clic droit puis "**Propriétés**".

**Note** : nous allons utiliser la méthode classique sous Windows, sans passer par le Gestionnaire de serveur.

A screenshot of a computer

Description automatically generated

Cliquez sur l'onglet "**Partage**" où nous pouvons voir que pour le moment le dossier n'est pas partagé. Cliquez sur "**Partage avancé**" : c'est ma méthode favorite pour créer un partage simple sur Windows, tout en ayant le contrôle.

Cochez l'option "**Partager ce dossier**" et donnez un nom au partage. Pour ma part, je vais le nommer "**Partage**", tout simplement. Il faut savoir que**ce partage sera visible sur le réseau**, mais il y a une petite astuce pour masquer un partage : il suffit d'**ajouter un "$" à la fin du nom pour rendre le partage invisible**.

A screenshot of a computer screen

Description automatically generated

Cliquez sur le bouton "**Autorisations**" : une autre fenêtre va s'ouvrir. Elle permet de définir **les autorisations sur le partage**. Sous Windows, lorsqu'un partage est créé il y a **deux niveaux d'autorisations** : **les autorisations sur le partage**et **les autorisations au niveau du système de fichiers**.

Si le groupe "**Tout le monde**" a les droits de lecture et d'écriture sur le partage, cela ne signifie pas que tout le monde peut accéder à votre partage et lire/modifier des fichiers. Cela dépend des autorisations définies sur le système de fichiers, que nous allons voir après.

A screenshot of a computer screen

Description automatically generated

Validez... Désormais, c'est bien spécifié "**Partagé**". Le chemin réseau est également précisé : \\SRV-ADDS-01\Partage, ce qui correspond à : **\\Nom-du-serveur\Nom-du-partage.**

Ce type de chemin est appelé un chemin UNC pour ***U****niversal****N****aming****C****onvention*. Si votre collègue vous demande le chemin UNC pour accéder au partage, il faudra lui donner "*\\SRV-ADDS-01\Partage*". Tout en sachant qu'un chemin UNC peut pointer vers un fichier ou un dossier directement, par exemple : "*\\SRV-ADDS-01\Partage\Dossier1\MonFichier.txt*".

A screenshot of a computer

Description automatically generated

Basculez sur l'onglet "**Sécurité**" : c'est ici que s'affichent **les autorisations du système de fichiers**, autrement dit cela correspond aux **droits NTFS** (si vous utilisez le NTFS sur ce volume). Par défaut, on peut voir qu'il y a déjà des droits, et on voit aussi que tout le monde n'a pas les droits ! En l'occurrence ici, ce sont des groupes du domaine Active Directory "*IT-CONNECT*".

Laissons comme cela, car à partir du poste client et d'un compte Administrateur du domaine, nous allons pouvoir accéder à notre partage.

A screenshot of a computer screen

Description automatically generated

Il n'y a plus qu'à tester...

B. Se connecter au partage SMB depuis le poste client

Voilà connecté sur le poste client Windows 10. Je vais établir une connexion avec mon partage. Il y a plusieurs façons de faire, commençons par la plus courante : on saisit le chemin UNC du partage directement dans la barre d'adresse de l'explorateur.... Bingo ! J'accède bien à mon partage, même s'il est vide.

A screenshot of a computer

Description automatically generated

**Je vous assure que la connexion au partage est effectuée grâce au protocole SMB.** Par contre, nous ne savons pas quelle version est utilisée...

Ce que je vous propose, c'est d'ouvrir une console PowerShell, puis exécutez la commande suivante :

Get-SmbConnection

Cette commande permet d'obtenir plusieurs informations telles que le nom du serveur distant, le nom du partage, le nom du compte utilisé pour accéder au partage et le dialecte SMB utilisé ! Ici, on peut voir que **la version "3.1.1" du protocole SMB est utilisée pour cette connexion.**

A screenshot of a computer

Description automatically generated

Une autre façon d'ouvrir une connexion SMB sans passer par l'interface graphique de Windows, c'est d'utiliser PowerShell et le cmdlet **Get-ChildItem**. Il permet de lister le contenu d'un répertoire.

Essayez ceci :

Get-ChildItem "\\SRV-ADDS-01.it-connect.local\Partage"

Ensuite, si vous relancez la commande "**Get-SmbConnection**", vous allez voir la connexion SMB apparaître !

Je souhaitais partager avec vous une astuce ! Si vous désirez**connaître le dialecte SMB supporté par votre machine**, vous pouvez interroger votre propre machine en ciblant le [partage administratif](https://www.it-connect.fr/les-partages-administratifs-sous-windows-10/) C$. Exécutez la commande ci-dessous (*dir* est un alias de *Get-ChildItem*)

dir \\localhost\c$

Relancez une nouvelle fois la commande "**Get-SmbConnection**" : la connexion va apparaître et comme vous interrogez la machine locale, cela vous donne la version de SMB de cette machine en elle-même.

A screenshot of a computer

Description automatically generated