Editions ENI

Windows Server 2016

Les bases indispensables pour administrer et configurer votre serveur

(2e édition)

Collection
Ressources Informatiques

Extrait

Chapitre 1
Rôles et fonctionnalités

1. Organisation du livre

Le livre est composé de 18 chapitres présentant les différentes fonctionnalités du système d'exploitation Windows Server 2016.

Afin de pouvoir effectuer la partie pratique dans de bonnes conditions, le chapitre Installation du bac à sable décrit la création d'une maquette. Cette dernière est équipée de 5 machines virtuelles :

– **AD1**, **AD2**, **SV1** et **SRVCore** exécutant Windows Server 2016.

– Une machine cliente **CL10-01** sous Windows 10.

Les chapitres, chacun traitant d'un sujet différent, peuvent être parcourus de façon indépendante. Chaque chapitre est construit afin de vous présenter la théorie mais également la mise en pratique sur une ou plusieurs VM (machine virtuelle). Le système d'exploitation de la machine hôte est Windows Server 2012 R2, les machines virtuelles fonctionnent sous Windows Server 2016 pour les serveurs et sous Windows 10 pour la machine cliente. AD1, AD2 et SV1 exécuteront leur système d'exploitation avec une interface graphique, SRVCore sera uniquement en mode ligne de commande.

Certains scripts ou modèles d'administration au format ADM peuvent être téléchargés au niveau de la page de présentation du livre sur le site des Éditions ENI.

2. Les rôles

Les rôles et fonctionnalités ci-dessous ne sont qu'une petite liste de ceux présents dans Windows Server 2016.

Depuis Windows Server 2008 R2, il est possible d'installer les différents rôles depuis la console **Gestionnaire de serveur**. Ces derniers apportent des fonctions supplémentaires aux serveurs. Ainsi l'équipe IT offre des services supplémentaires (adressage IP automatique des postes et autres équipements raccordés au réseau, serveur d'impression…) à ses utilisateurs. La plupart contiennent des services de rôle, permettant l'activation de certaines fonctionnalités. Ils s'installent généralement lors de l'installation d'un autre rôle ou d'une fonctionnalité. L'ajout peut également s'effectuer après l'installation.

2.1 Accès à distance

Le rôle **Accès à distance** permet de fournir un **service** VPN. La partie routage est également présente et offre la fonctionnalité qui permet le routage de paquets.

Les services de rôle disponibles sont :

- **DirectAccess** et **VPN** : DirectAccess permet la connexion au réseau de l'entreprise sans aucune intervention de l'utilisateur. La connexion est établie uniquement lorsque l'utilisateur est connecté en dehors du réseau local.
- **Routage** : ce service de rôle permet l'installation de plusieurs types de routeurs dont ceux exécutant RIP et les proxys IGMP.

2.2 Hyper-V

Depuis Windows Server 2008, l'hyperviseur de Microsoft, **Hyper-V**, peut être installé. Il permet de mettre en place une plateforme de virtualisation. Il a été enrichi avec Windows Server 2012 et 2012 R2. De nouvelles fonctionnalités ont été intégrées à Windows Server 2016. Il est désormais possible d'ajouter à chaud pour une VM une carte réseau virtuelle ainsi que de la mémoire. Cette fonctionnalité très intéressante permet de réduire le temps d'indisponibilité du service offert par la machine virtuelle concernée. D'autres fonctionnalités comme la distribution du service d'intégration ont été ajoutées à Hyper-V sous Windows Server 2016.

2.3 DHCP (Dynamic Host Configuration Protocol)

Le rôle permet la distribution de baux DHCP aux différents équipements qui en font la demande. Il peut être installé sur un serveur en mode installation complète ou en mode Core (installation sans interface graphique).

2.4 DNS (Domain Name System)

Obligatoire dans un domaine Active Directory, il permet la résolution de noms en adresse IP et inversement. Ce service permet également aux postes clients de trouver leurs contrôleurs de domaine. Il peut être installé sur un serveur ne possédant pas d'interface graphique.

2.5 IIS (Internet Information Services)

Serveur web, il permet l'affichage et le stockage de sites et d'applications web. De nos jours, il est de plus en plus fréquent qu'une application possède une interface web.

Ce rôle est celui qui possède le plus de services de rôle.

- **Fonctionnalités HTTP communes** : installe et gère les fonctionnalités HTTP basiques. Ce service de rôle permet de créer des messages d'erreurs personnalisés afin de gérer les réponses faites par le serveur.
- **Intégrité et diagnostics** : apporte les outils nécessaires à la surveillance et au diagnostic de l'intégrité des serveurs.
- **Performances** : permet d'effectuer de la compression de contenu.
- **Sécurité** : mise en place des outils permettant d'assurer la sécurité du serveur contre les utilisateurs mal intentionnés et les requêtes IIS.
- **Outils de gestion** : fournit les outils de gestion pour les versions précédentes de IIS.
- **Serveur FTP** : permet l'installation et la gestion d'un serveur FTP.

2.6 AD DS (Active Directory Domain Services)

Permet le stockage des informations d'identification des utilisateurs et ordinateurs du domaine. Ce rôle est exécuté par un serveur portant le nom de contrôleur de domaine. Ce dernier a pour fonction d'authentifier les utilisateurs et ordinateurs présents sur le domaine AD.

Ce rôle peut être installé sur un serveur ne possédant pas d'interface graphique.

2.7 AD FS (Active Directory Federation Services)

Le rôle fournit un service fédéré de gestion des identités. Il identifie et authentifie un utilisateur qui souhaite accéder à un extranet.

Ainsi, deux entreprises peuvent partager de manière sécurisée des informations d'identité d'Active Directory pour un utilisateur Office 365 uniquement.

Plusieurs services de rôle le composent :

- **Service de fédération** : l'infrastructure est installée afin de fournir l'accès à des ressources.
- **Agent Web AD FS** : permet de valider les jetons de sécurité délivrés et d'autoriser un accès authentifié à une ressource web.
- **Proxy FSP** (*Federation Service Proxy*) : permet d'effectuer la collecte d'informations d'authentification utilisateur depuis un navigateur ou une application web.

2.8 AD RMS (Active Directory Rights Management Services)

Protège une ressource contre une utilisation non autorisée. Les utilisateurs sont identifiés et une licence leur est attribuée pour les informations protégées.

Il est ainsi plus simple d'interdire à un utilisateur de copier un document sur une clé USB ou d'imprimer un fichier confidentiel.

Lors de l'installation du rôle, deux services peuvent être installés :

- **Active Directory Rights Management Server** : permet de protéger une ressource d'une utilisation non autorisée.
- **Prise en charge de la fédération des identités** : profite des relations fédérées entre deux organisations pour établir l'identité de l'utilisateur et lui fournir un accès à une ressource protégée.

2.9 AD CS (Active Directory Certificate Service)

Installe une autorité de certification afin d'effectuer des opérations d'émission et de gestion de certificats.

Six services de rôle peuvent être ajoutés à l'installation :

- **Autorité de certification** : fournit une infrastructure à clé publique.
- **Inscription de l'autorité de certification via le web** : une interface web est installée afin de permettre à un utilisateur d'effectuer des demandes et renouvellements de certificats. Il est également possible de récupérer des listes de révocation de certificats ou d'effectuer une inscription à des certificats de cartes à puce.

– **Répondeur en ligne** : permet la gestion et la distribution des informations de statut de révocation.
– **Service d'inscription de périphérique réseau** : émet et gère les certificats des routeurs et des autres périphériques réseaux.
– **Service web Inscription de certificats** : ce service de rôle donne la possibilité aux utilisateurs et ordinateurs d'effectuer l'inscription et le renouvellement de certificats.
– **Service web Stratégie d'inscription de certificats** : donne aux utilisateurs et ordinateurs des informations sur la stratégie d'inscription de certificats.

2.10 Service de déploiement Windows (WDS)

Ce rôle fournit un service de déploiement de systèmes d'exploitation à travers le réseau. Le serveur possède deux types d'images : les **images de démarrage** qui permettent l'accès à l'installation de Windows ou à un dossier partagé (MDT) et les **images d'installation** qui contiennent les métadonnées nécessaires à l'installation du système d'exploitation.

Avec l'installation de ce service, deux services de rôle peuvent être installés :

– **Serveur de déploiement** : fournit les fonctionnalités nécessaires au déploiement d'un système d'exploitation. Les fonctionnalités de capture sont également prises en compte par ce service.
– **Serveur de transport** : utilisé pour la transmission des données en multidiffusion.

2.11 Service de stratégie et d'accès réseau

Ce rôle permet la gestion des accès au réseau par le biais d'accès sans fil, de serveurs VPN ainsi que de commutateurs d'authentification 802.1x. L'installation de NPS (*Network Policy Server*) permet la mise en place de la protection d'accès réseau (NAP).

Les services de rôle disponibles sont :

– **Serveur NPS** : permet la mise en place des stratégies d'accès réseau pour les demandes de connexion.
– **Autorité HRA** : émission de certificats d'intégrité pour les postes de travail conformes aux exigences d'intégrité.
– **HCAP** (*Host Credential Authorization Protocol*) : la solution NAP est intégrée avec la solution de contrôle d'accès Cisco.

2.12 WSUS

Permet d'approuver les mises à jour avant l'installation sur un poste client, ce dernier étant rangé dans un groupe d'ordinateurs. Cette solution permet d'effectuer une approbation pour un groupe en particulier (exemple : groupe « test » en premier puis, si le correctif ne pose pas de problèmes, il est approuvé pour le deuxième).

Trois services de rôle sont disponibles :

- **WID Database** : installe la base de données utilisée par WSUS dans WID (*Windows Internal Database*). Ce type de base de données est utilisable par d'autres rôles (AD RMS, etc.).
- **WSUS Services** : installe le service WSUS ainsi que tous les composants nécessaires.
- **Base de données** : installe la base de données pour les services WSUS (un serveur SQL est nécessaire, contrairement à WID Database).

2.13 Services de fichiers et iSCSI

Le service de fichiers permet la mise en place de quotas sur le système de fichiers ainsi qu'un système de filtrage par extension afin d'interdire le stockage de certains fichiers. Un espace de noms DFS peut être installé par l'intermédiaire d'un service de rôle.

Les services suivants offrent la possibilité d'être installés en tant que service de rôle :

- **Serveur de fichiers** : gestion des dossiers partagés.
- **BranchCache pour fichier réseau** : prise en compte de BranchCache sur le serveur. Ce service permet la mise en cache de documents afin de réduire l'utilisation de la ligne reliant deux sites distants. L'utilisateur n'a par exemple plus besoin de venir chercher les documents à son siège social, ces derniers sont mis en cache sur un serveur ou un poste local.
- **Déduplication des données** : permet de libérer de l'espace disque en supprimant les données dupliquées, une copie unique des données identiques est stockée sur le volume.
- **Espace de noms DFS** : installe les outils nécessaires pour la création et la gestion de l'espace de noms.
- **Gestionnaire de ressources du serveur de fichiers** : outil permettant la gestion d'un système de fichiers en effectuant la création de quotas et le filtrage de fichiers.
- **Réplication DFS** : synchronise des dossiers sur plusieurs serveurs locaux ou sur un site distant.

Editions ENI

PowerShell Core et Windows PowerShell

Les fondamentaux du langage

(2e édition)

Collection
Expert IT

Extrait

Chapitre 12
Sécurité

1. La sécurité : pour qui ? Pourquoi ?

L'arrivée des réseaux locaux et d'Internet a changé beaucoup de choses dans la manière de protéger son PC. Il ne suffit plus d'attacher son disque dur au radiateur et de fermer la porte du bureau le soir pour ne pas se faire voler ou pirater des données. Maintenant, protéger son poste de travail est devenu essentiel pour ne pas faire les frais d'intrusions ou de malveillances.

Mais alors contre qui se prémunir ? Eh bien, contre tout ce qui bouge… et même ce qui ne bouge pas. En effet, que ce soient des programmes malveillants, des utilisateurs mal intentionnés, voire des utilisateurs inexpérimentés, tous peuvent être considérés comme une menace. C'est pour cela que vous devez verrouiller votre système en établissant des règles de sécurité, en les appliquant et en vous assurant que les autres en font tout autant.

2. Les risques liés au scripting

Vous allez vite deviner que ce qui fait la force du scripting en fait aussi sa faiblesse. La facilité avec laquelle vous pouvez tout faire, soit en cliquant sur un script, soit en l'exécutant depuis la fenêtre de commande, peut vous mettre dans l'embarras si vous ne faites pas attention.

Imaginez un script de logon qui dès l'ouverture de la session la verrouille aussitôt ! Alors oui, c'est sympa entre copains, mais en entreprise, nous doutons que cela soit de bon ton. Plus grave encore, un script provenant d'une personne mal intentionnée ou vraiment peu expérimentée en PowerShell (dans ce cas, nous vous conseillons de lui acheter un exemplaire de ce livre…) peut parfaitement vous bloquer des comptes utilisateurs dans Active Directory, vous formater un disque, vous faire rebooter sans cesse. Vous l'avez compris, un script peut tout faire. Car même si aujourd'hui des alertes sont remontées jusqu'à l'utilisateur pour le prévenir de l'exécution d'un script, elles ne sont pas capables de déterminer à l'avance si un script est nuisible au bon fonctionnement du système.

Les risques liés au scripting se résument à une histoire de compromis : soit vous empêchez toute exécution de scripts, c'est-à-dire encourir le risque de vous « pourrir la vie » à faire et à refaire des tâches basiques et souvent ingrates, soit vous choisissez d'ouvrir votre système à PowerShell, en prenant soin de prendre les précautions qui s'imposent.

Mais ne vous laissez pas démoraliser car même si l'exécution de scripts vous expose à certains problèmes de sécurité, PowerShell se dote de certains concepts qui font de lui l'un des langages de script les plus sûrs. Il ne faut pas non plus oublier qu'en cas de problème de sécurité, c'est l'image tout entière de Microsoft qui en pâtit…

3. Optimiser la sécurité PowerShell

3.1 La sécurité PowerShell par défaut

Vous l'avez compris, la sécurité est une chose très importante, surtout dans le domaine du scripting. C'est pour cela que les créateurs de PowerShell ont inclus deux règles de sécurité par défaut.

Des fichiers ps1 associés au bloc-notes

L'extension « **`.ps1`** » des scripts PowerShell, est par défaut associée à l'éditeur de texte bloc-notes (ou Notepad). Ce procédé permet d'éviter de lancer des scripts potentiellement dangereux sur une mauvaise manipulation. Le bloc-notes est certes un éditeur un peu classique, mais il a le double avantage d'être inoffensif et de ne pas bloquer l'exécution d'un script lorsque celui-ci est ouvert avec l'éditeur. Vous remarquerez cependant que l'édition (clic droit + **Modifier**) des fichiers **`.ps1`** est associée à l'éditeur ISE.

Remarque

Ce type de sécurité n'était pas mis en place avec les scripts VBS dont l'ouverture était directement associée au Windows Script Host.

Une stratégie d'exécution restreinte

La seconde barrière de sécurité est l'application de la stratégie d'exécution **`Restricted`** par défaut pour les postes clients et **`RemoteSigned`** celle par défaut pour les serveurs (depuis Windows Server 2012 R2).

La stratégie **`Restricted`** est la plus restrictive. C'est-à-dire qu'elle bloque systématiquement l'exécution de tout script. Seules les commandes tapées dans le shell seront exécutées. Pour remédier à l'inexécution des scripts, PowerShell requiert que l'utilisateur change le mode d'exécution avec la commande **`Set-ExecutionPolicy <mode d'exécution>`**. Mais pour ce faire il faut être administrateur de la machine.

Remarque

Peut-être comprenez-vous mieux pourquoi l'utilisation de PowerShell sur vos machines ne constitue pas un accroissement des risques, dans la mesure où certaines règles sont bien respectées.

3.2 Les stratégies d'exécution

PowerShell intègre un concept de sécurité que l'on appelle les stratégies d'exécution (execution policies) pour qu'un script non autorisé ne puisse pas s'exécuter à l'insu de l'utilisateur. Il existe sept configurations possibles : **`Restricted`**, **`RemoteSigned`**, **`AllSigned`**, **`UnRestricted`**, **`Bypass`**, **`Default`** et **`Undefined`**. À chacune d'elles correspond un niveau d'autorisation d'exécution de scripts particulier. Vous pourrez être amené à en changer en fonction de la stratégie que vous souhaitez appliquer.

3.2.1 Les différentes stratégies d'exécution

`Restricted` : c'est la stratégie la plus restrictive, et c'est aussi la stratégie par défaut sur les postes clients (de Windows 7 à Windows 10). Elle ne permet pas l'exécution de scripts mais autorise uniquement les instructions en ligne de commande tapées dans la console (mode interactif). Cette stratégie peut être considérée comme la plus radicale étant donné qu'elle protège contre l'exécution des fichiers **`.ps1`**.

Lors d'une tentative d'exécution de script avec cette stratégie, un message de ce type est affiché dans la console :

```
.\lanceur.ps1 : File C:\temp\lanceur.ps1 cannot be loaded because run-
ning scripts is disabled on this system. For more information, see
about_Execution_Policies at https://go.microsoft.com/fwlink/
?LinkID=135170.
At line:1 char:1
+ .\lanceur.ps1
+ ~~~~~~~~~~~~~
```

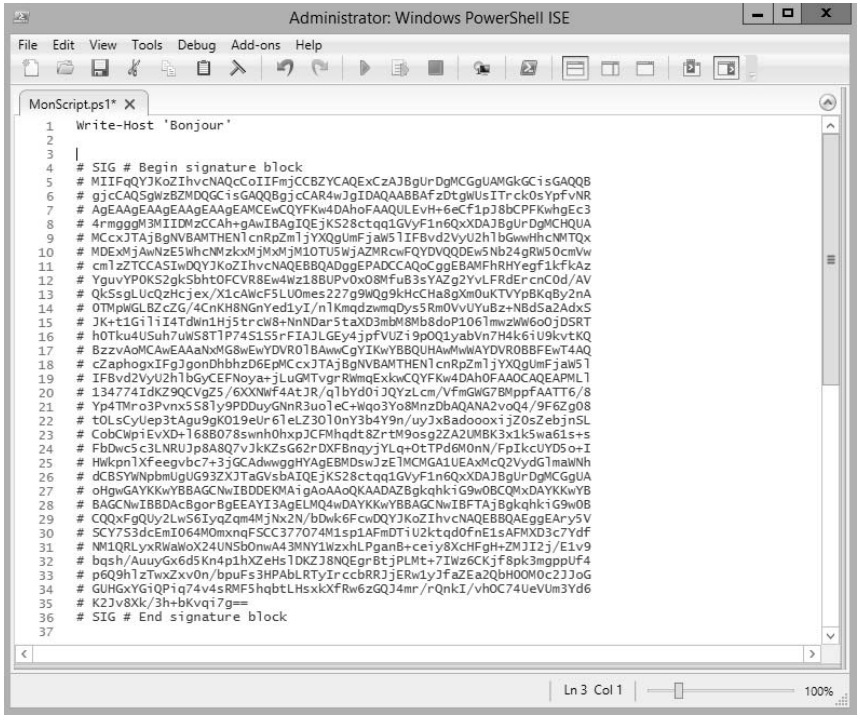
```
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Si cette stratégie est celle définie par défaut lors de l'installation de PowerShell, il vous faudra la changer pour l'exécution de votre premier script.

Remarque

Pour pouvoir modifier la stratégie d'exécution PowerShell, il vous faudra être administrateur local de la machine.

`AllSigned` : c'est la stratégie permettant l'exécution de scripts la plus « sûre ». Elle autorise uniquement l'exécution des scripts signés. Un script signé est un script comportant une signature numérique comme celle présentée sur la figure ci-dessous.



Exemple de script signé

Avec la stratégie **`AllSigned`**, l'exécution de scripts signés nécessite que vous soyez en possession des certificats correspondants (cf. section Signature des scripts).

RemoteSigned : cette stratégie se rapporte à **AllSigned** à la différence près que seuls les scripts ayant une origine autre que locale nécessitent une signature. Par conséquent, tous vos scripts créés localement peuvent être exécutés sans être signés.

À partir de Windows Server 2012 R2, PowerShell s'exécute désormais avec cette stratégie d'exécution par défaut, ce qui n'était pas le cas dans les versions antérieures de Windows Server.

Si vous essayez d'exécuter un script provenant d'Internet sans que celui-ci ne soit signé, vous obtiendrez le message d'erreur suivant :

```
.\Script.ps1 : File C:\Temp\Script.ps1 cannot be loaded.
The file C:\Temp\Script.ps1 is not digitally signed.
```

Remarque

Vous vous demandez sûrement comment PowerShell fait pour savoir que notre script provient d'Internet ? Réponse : Grâce aux « Alternate Data Streams » qui sont implémentés sous forme de flux cachés depuis des applications de communication telles que Microsoft Outlook, Internet Explorer, Outlook Express et Windows Messenger (voir partie traitant des Alternate Data Streams). En gros, lorsqu'un script est téléchargé à partir d'un client Microsoft, la provenance de celui-ci lui est attachée.

Unrestricted : avec cette stratégie, tout script, peu importe son origine, peut être exécuté sans demande de signature.

Cette stratégie affiche tout de même un avertissement lorsqu'un script téléchargé d'Internet tente d'être exécuté.

```
PS > .\script.ps1

Security warning

Run only scripts that you trust. While scripts from the internet can
be useful, this script can potentially harm your computer. If you trust
this script, use the Unblock-File cmdlet to allow the script to run
without this warning message. Do you want to run C:\Temp\script.ps1?
[D] Do not run  [R] Run once  [S] Suspend  [?] Help (default is "D"):
```

Bypass : c'est la stratégie la moins contraignante, et par conséquent la moins sûre. Rien n'est bloqué et aucun message d'avertissement ne s'affiche. C'est donc la stratégie où le risque d'exécuter des scripts malveillants est le plus élevé.

Undefined : pas de stratégie d'exécution définie dans l'étendue courante. Si toutes les stratégies d'exécution de toutes les étendues sont non définies alors la stratégie effective appliquée sera la stratégie **Restricted**.

Default : positionne la stratégie par défaut, à savoir **Restricted**.

Remarque

Microsoft a mis en place ces mécanismes afin de tenter de limiter les risques liés à l'exécution de scripts provenant de l'extérieur de l'entreprise et donc potentiellement malveillants. La configuration par défaut permet d'atteindre cet objectif mais elle ne garantit en aucun cas une sécurité parfaite.

3.2.2 Les étendues des stratégies d'exécution

PowerShell permet de gérer l'étendue des stratégies. L'ordre d'application est le suivant :

- **Étendue `Process`** : la stratégie d'exécution n'affecte que la session courante (processus Windows PowerShell). La valeur affectée à l'étendue **`Process`** est stockée en mémoire uniquement ; elle n'est donc pas conservée lors de la fermeture de la session PowerShell.
- **Étendue `CurrentUser`** : la stratégie d'exécution appliquée à l'étendue **`CurrentUser`** n'affecte que l'utilisateur courant. Le type de stratégie est stocké de façon permanente dans la partie du registre **`HKEY_CURRENT_USER`**.
- **Étendue `LocalMachine`** : la stratégie d'exécution appliquée à l'étendue **`LocalMachine`** affecte tous les utilisateurs de la machine. Le type de stratégie est stocké de façon permanente dans la partie du registre **`HKEY_LOCAL_MACHINE`**.

La stratégie ayant une priorité 1 est plus propriétaire que celle ayant une priorité 3. Par conséquent, si l'étendue **`LocalMachine`** est plus restrictive que l'étendue **`Process`**, la stratégie qui s'appliquera sera quand même la stratégie de l'étendue **`Process`**. À moins que cette dernière soit de type **`Undefined`** auquel cas PowerShell appliquera la stratégie de l'étendue **`CurrentUser`** puis tentera d'appliquer la stratégie **`LocalMachine`**.

À noter que l'étendue **`LocalMachiPne`** est celle par défaut lorsque l'on applique une stratégie d'exécution sans préciser d'étendue particulière.

3.2.3 Identifier la stratégie d'exécution courante

La stratégie d'exécution courante s'obtient avec la commandelette **`Get-Execution-Policy`**.

Exemple

```
PS > Get-ExecutionPolicy
Restricted
```

Avec cette commande, nous bénéficions du commutateur **`-List`**. Grâce à lui, nous allons savoir quelles stratégies s'appliquent à nos étendues.