TP3 - Windows: Introduction à l'administration de postes clients

1 Accès distant à une machine

Dans le cadre de l'administration d'un parc Windows géré via un domaine, l'approche la plus performante consiste à utiliser les "stratégies de groupe", ou *Group Policy Object* (GPO). Elles permettent la configuration automatique de multiples aspects d'un poste de travail client connecté sur un domaine Windows.

Toutefois, on peut aussi effectuer des tâches plus élementaires via de simples scripts, en utilisant les fonctionnalités réseaux accessibles via le shell. On peut ainsi accéder à une machine distante via son nom d'hôte et des "partages disques", dont certains sont disponibles par défaut. Par exemple, si une machine s'appelle OBIWAN, on pourra accéder au contenu du disque C : via le chemin **`\\OBIWAN\c$`** (**`C$`** étant le partage par défaut pour le disque C :)

On pourra ainsi automatiser des configurations simples de machine à distance, en y accédant via son nom.

Travail à effectuer

Q1.1: En utilisant une boucle "for" numérique (cf. TP2), écrire un script nommé **`T:\M1105\configmach.bat`** qui va copier le fichier **`c:\program files\monapp\config.ini`** sur toutes les 16 machines de la salle "A123" du département "RT"

Pour l'exercice, on admettra que les machines sont nommées **`IUT_XX_YYYY_ZZ`**, avec XX : code du département (2 car.), YYYY : code de la salle, et ZZ numéro de la machine. Vous ferez aussi précéder la commande **`copy`** d'une commande **`echo`**, l'exercice étant virtuel.

Ajouter après la commande de copie un test pour vérifier le succès ou l'échec de la copie, via la variable d'environnement ERRORLEVEL (cf. cours). En cas d'échec, loggez le nom de la machine dans un fichier qui s'appelera **`echecs_copie.txt`**

2 Base de registre Windows

Contrairement à Linux, où la configuration de la machine est stockée dans une multitudes de fichiers textes, sous Windows la configuration est stockée dans une base de données centralisée, appelée la **Base de registre** (BDR). Elle contient à la fois la configuration de l'OS et des applications installées. Elle est organisée de façon arborescente de façon similaire à une arborescence de fichiers. Chaque niveau est appelé une **clé** (*Key*) et peut contenir d'autres clés mais aussi des **valeurs**, qui sont en fait des paires "nom/valeur", chaque valeur ayant un **type**, qui peut être :

- **`REG_SZ`** : chaîne de caractères
- **`REG_EXPAND_SZ`** : chaîne contenant des variables d'environnement destinées à être développées à l'utilisation (par exemple, **`%USERNAME%`**, qui sera remplacé par le login courant)
- **`REG_BINARY`** : valeur binaire arbitraire
- **`REG_DWORD`** : valeur entière

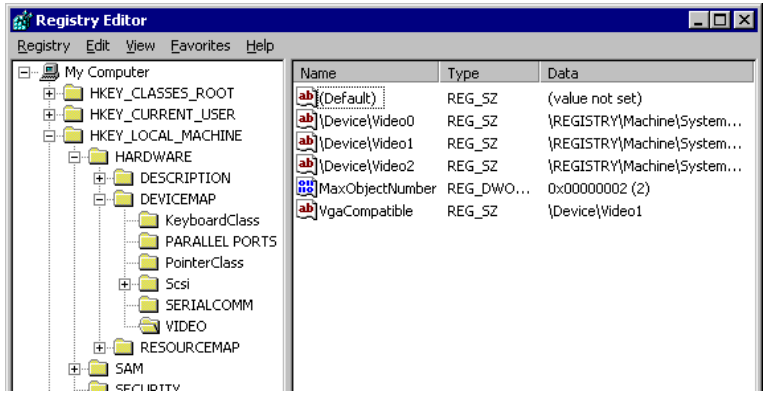

FIGURE 1 – Exemple montrant le contenu de la clé **`HKLM\HARDWARE\DEVICEMAP\VIDEO`**

A la racine se trouve plusieurs "ruches" (*hives*) et dont les principales s'appelent **`HKEY_LOCAL_MACHINE`** (abbrev : HKLM) et **`HKEY_CURRENT_USER`** (abbrev : HKCU)

Cette base de donnée est stockée dans des fichiers systèmes, dont le nombre et l'emplacement dépendent des versions de Windows et auxquels il ne faut surtout pas accéder de façon directe. Mais une partie est aussi générée lors du *boot*, et présente uniquement en RAM à l'exécution.

De façon similaire au système de fichiers, à chaque clé est associée des permissions permettant de spécifier les droits de chaque utilisateur, via le mécanismes des ACL. On peut éditer la BDR soit via un outil GUI (**`regedit.exe`**), soit via des outils CLI dédiés.

Travail à effectuer La BDR stocke en particulier la gestion des associations extension nom de fichier ↔ type de fichier et application associée. A chaque extension est associée un **type de fichier**, qui lui-même sera rattaché à une application par défaut.

Q2.1: Lancer l'éditeur GUI du registre et ouvrir l'arborescence jusqu'à la clé **`HKEY_CLASSES_ROOT\.txt`** et donner le type de fichier associé (valeur par défaut)

Q2.2: Descendre dans cette ruche jusqu'au type de fichier indiqué, et noter l'application indiquée dans la sous clé **`shell/open/command`**

Q2.3: Quel est le type de cette clé :

Q2.4: Donner la commande permettant d'avoir le contenu de la variable d'environnement utilisée dans le chemin de cet exécutable :

Q2.5: Sous Windows 7/8/10, l'outil CLI s'appelle **`REG`**.

Essayer la commande **`REG /?`** puis **`REG QUERY /?`**, et donner la syntaxe de la commande permettant d'avoir le type de fichier associé aux extensions .txt :

3 Monitoring réseau

Une des missions d'un administrateur réseau est de s'assurer que ses différents serveurs "répondent" aux requêtes.

Il existe des outils dédiés (dit de "supervision réseau") mais on va réaliser ici un monitoring basique automatisé sur une plage d'adresse donnée. On peut ainsi utiliser la commande ping pour vérifier qu'une machine répond sur la couche "IP" [1].

Travail à effectuer

Q3.1: Donner la syntaxe de ping pour réaliser un seul test sur la machine 192.168.0.1, avec un délai d'attente de 500ms :

Q3.2: Quelle est la commande permettant de visualiser votre adresse IPv4 :

Q3.3: Demander d'adresse IP de votre voisin, et vérifiez que vous pouvez "pinger" sa machine :

Q3.4: Ecrire un script **`T:\M1105\pingall.bat`** qui va pinger successivement toutes les machines dans la plage d'adresses 192.168.0.1 à 192.168.10, et qui va ensuite afficher à l'écran un fichier texte sous la forme

```
Results:
host 192.168.0.1 : ok
host 192.168.0.2 : non
...
```

Vous utiliserez la commande **`for`** dans sa version "numérique" (cf. TP2) pour générer la plage d'adresses et la commande **`ping`** en spécifiant un seul essai, et récupererez le résultat (succès ou échec) avec la variable d'environnement **`%ERRORLEVEL%`** (cf. cours)

Q3.5: Copier le script en **`pingall2.bat`** et modifier celui-ci pour qu'il s'exécute en boucle, et que l'affichage se fasse dans la console (avec la commande **`cls`** pour vider l'écran à chaque fois).

1. ce qui ne dit rien sur la capacité de cette machine à répondre à une requête applicative, sur une couche supérieure.