Table des matières

Avant-propos

1. Conventions typographiques

Type de texte	Description du texte
Arial	Police principale utilisée
`Courrier new`	Chemin des fichiers, Commandes, noms de fichiers
Italique	Mots ou abréviations anglophones
Texte en bleu souligné	Liens http
Arial gras	Titres

2. Structure du document

Ce document est composé de 6 parties. Chaque partie est composée de sections pouvant être elles-mêmes composée jusqu'à 2 niveaux de sous sections.

3. *OS* de PC physiques utilisés

XP Pro Service pack 2 pour toutes les parties,

Windows Server 2003 pour les dossiers Web (partie *EFS*).

4. Remerciements

Gérald Litzistorf, pour ces conseils pendant le travail de diplôme et la qualité de son enseignement.

Eric Jenny, pour ces explications en cas de problème.

Sébastien Contreras, pour son aide régulière.

Mes collègues de travail.

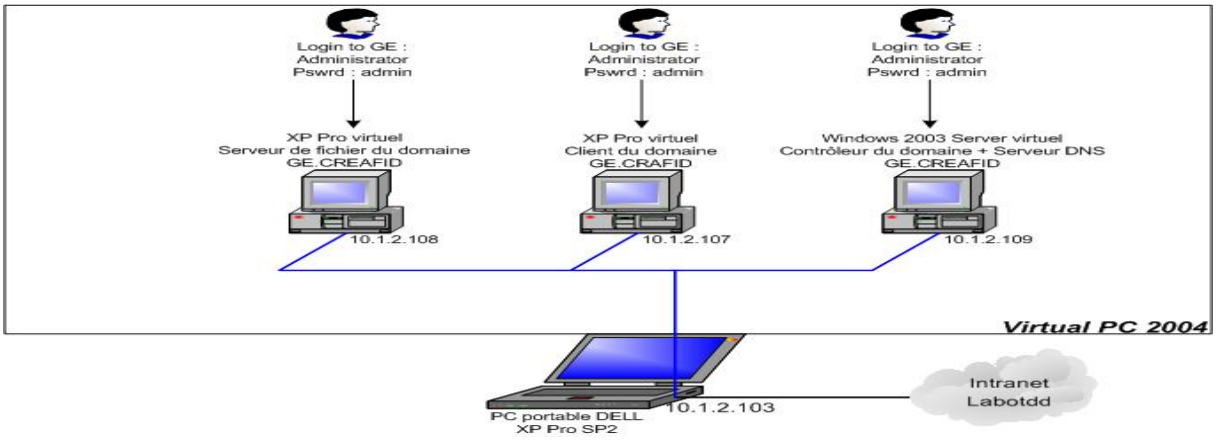
Ma famille et surtout Tahara pour leur soutien.

5. Résumé du travail

Le travail effectué pendant le diplôme peut être résumé comme suit :

- Etude et mise en œuvre des commandes *netsh.*
- Etude et mise en œuvre du chiffrement *EFS* et des dossiers Web.
- Etude et mise en œuvre du mécanisme des fichiers hors connexion.
- Comprendre comment empêcher l'utilisation d'un périphérique de stockage *USB.*
- Etude du mécanisme de restauration système et des points de restauration.
- Mise en œuvre du Labo *XP* 4 avec *Virtual PC*. Etude des mécanisme d'installation des PC virtuels et tests de réseau virtuel.

netsh est un outil en lignes de commandes permettant de configurer les interfaces réseaux d'un PC fonctionnant sous *Windows XP* et qui possède bien d'autres fonctionnalités. J'ai mis en œuvre 5 scénarios permettant d'évaluer ce produit : configuration des interfaces réseau, sauvegarde d'une configuration réseau, diagnostic réseau, réinitialisation de la pile *tcp/ip* puis réparation de *Winsock* et enfin exportation des commandes *netsh* sur une machine distante. (1 semaine)
EFS (Encryption File System) est le mécanisme de chiffrement de *Windows XP*. J'ai étudié le chiffrement, déchiffrement et récupération des données avec agent de recouvrement. J'ai ensuite fait une mise en œuvre à l'aide d'un scénario pratique dans le cadre d'un ordinateur *stand alone* avec partage multi utilisateurs. Puis j'ai étudié le chiffrement *EFS* à distance avec le protocole *webDAV* qui est une extension *http*. (4 semaines)
Les fichiers hors connexion permettent de stocker, sur une machine cliente, des fichiers disponibles sur un serveur dans le cadre d'un partage réseau. J'ai étudié le mécanisme d'utilisation des fichiers hors connexion disponibles dans le cache du client et leur synchronisation avec le serveur pour une mise à jour par le protocole *SMB*. (1 semaine)
L'utilisation des clés USB étant devenue une pratique de tous les jours, une personne mal intentionnée pourrait s'en servir pour voler des données ou installer des virus sur des machines. J'ai étudié comment on empêche l'utilisation des périphériques de stockage *USB* tout en autorisant l'utilisation des autre périphériques *USB*. (1 semaine)
La restauration système permet de revenir à un état antérieur de l'*OS*, avant qu'il ne devienne instable à cause de l'installation d'une application ou d'un *driver* non compatible avec *Windows XP,* sans que les documents personnels ne soient effacés. J'ai donc étudié le mécanisme des points de restauration dans lesquels sont sauvegardés des clichés instantanés de la base de registre et de quelques fichiers système. J'ai aussi étudié le mécanisme de monitorage des fichiers présents dans le point de restauration, le rôle de la base de registre et enfin le mécanisme de restauration.
(2 semaines)
Virtual PC est la réponse de *Microsoft* à l'outil *VMWare*. Il s'agit d'un logiciel permettant d'émuler un ou plusieurs système(s) d'exploitation sur le même ordinateur physique. J'ai donc installer 3 *OS* virtuels sur une machine hôte afin de configurer un domaine dont le but est d'accéder avec le poste client au serveur de fichier par l'intermédiaire d'un annuaire *active directory*. Le but de cette mise en œuvre est de comprendre le fonctionnement de *Virtual PC* ainsi que les aspects liés à la sécurité
(3 semaines)



Partie 1 :

Commandes *netsh*

(1 semaine d'étude)

1. Pourquoi *netsh* ?

1.1. Qu'est-ce que c'est ?

netsh signifie *network shell*. Un *shell* est un programme qui prend les commandes de l'utilisateur et les passe au système d'exploitation pour être exécutées. Son origine vient du monde *UNIX*. *netsh* est apparue avec *Windows2000* et améliorée sous *Windows XP*. C'est un outil utilisé pour configurer et auditer avec l'invite de commande.
L'utilisation de *netsh* ne nécessite pas les droits administrateur.

1.2. Utilité

Netsh peut être utile dans beaucoup de cas. Voici les scénarios les plus courants :

1.2.1. Scénario 1

Je voyage beaucoup avec mon ordinateur portable et il est souvent difficile de me souvenir de toutes les configurations *IP* de mes interfaces pour me connecter aux réseaux que je visite. Il faudrait un outil permettant de sauvegarder chaque configuration IP et de les restituer au moment opportun.

1.2.2. Scénario 2

Je voudrais configurer les paramètres tcp/ip de ma machine en lignes de commandes. Eventuellement faire un fichier .cmd m'évitant de taper la même commande à chaque fois.

1.2.3. Scénario 3

L'administrateur d'un réseau voudrait vérifier que toutes ses machines (postes de travail, serveurs, routeurs) soient atteignables. Il est fastidieux d'effectuer autant de `pings` qu'il y a de machines, il faudrait donc un outil permettant de tout vérifier en une seule fois.

1.2.4. Scénario 4

J'aimerais détecter les *LSP* malveillants et les supprimer totalement. J'aimerais également forcer *Windows* à détruire et reconstruire à neuf son protocole *tcp/ip*.

1.2.5. Scénario 5

Je suis administrateur réseau et je désire configurer ou auditer une machine distante du réseau.

2. Comment netsh ?

2.1. Outil en ligne de commande

```
      Utilisation : netsh [-a Fichier alias] [-c Contexte] [-r Machine
distante] [Commande | -f Fichier Script]

Les commandes suivantes sont disponibles :

Commandes dans ce contexte :

?               - Affiche une liste de commandes.

                     add            - Ajoute une entrée de configuration à
une liste            d'entrées.

      bridge         - Modifications pour le contexte `netsh bridge'.

      delete         - Supprime une entrée de configuration d'une liste
d'entrées.

      diag           - Modifications pour le contexte `netsh diag'.

      dump           - Affiche un script de configuration.

      exec           - Exécute un fichier script.

      firewall       - Modifications pour le contexte `netsh firewall'.

      help           - Affiche une liste de commandes.

      interface      - Modifications pour le contexte `netsh interface'.

      ras            - Modifications pour le contexte `netsh ras'.

      routing        - Modifications pour le contexte `netsh routing'.

      set            - Met à jour les paramètres de configuration.

      show           - Affiche les informations.

      winsock        - Modifications pour le contexte `netsh winsock'.


      Les sous-contextes suivants sont disponibles :

       bridge diag firewall interface ras routing winsock
```

Comme on peut le constater, il existe un bon nombre de commandes. Les 4 scénarios énoncés dans la section1.2 sont les cas où l'usage de `netsh` est très intéressant. Je vais donc montrer quelles commandes sont relatives aux différents scénarios et comment elles fonctionnent.

2.2. Commandes pour le scénario 1

« Je voyage beaucoup avec mon ordinateur portable et il est souvent difficile de me souvenir de toutes les configurations *IP* de mes interfaces pour me connecter aux réseaux que je visite. Il faudrait un outil permettant de sauvegarder chaque configuration IP et de les restituer au moment opportun. »

Ma configuration actuelle est :

IP = 10.1.2.101

Mask = 255.255.0.0

Gateway = 10.1.0.1

DNS = 10.1.1.10

J'aimerais bien la sauvegarder car je dois visiter un autre réseau.

La commande `dump` permet d'afficher et éventuellement de sauvegarder l'ensemble des informations d'un contexte donné. Par défaut, le dump est affiché à l'écran. Il faut donc utiliser une redirection pour l'envoyer dans un fichier.

Commande : `netsh –c interface dump > C:\unrepertoire\location1.txt`

La configuration actuelle de la connexion réseau est alors sauvegardée dans le fichier location1.txt :

```
# ----------------------------------
# Interface IP Configuration
# ----------------------------------
pushd interface ip
# Interface IP Configuration for "Local Area Connection"
set address name="Local Area Connection" source=static addr=10.1.2.101
mask=255.255.0.0
set address name="Local Area Connection" gateway=10.1.0.1 gwmetric=0
set dns name="Local Area Connection" source=static addr=10.1.1.10 register=PRIMARY
set wins name="Local Area Connection" source=static addr=none
popd
# End of interface IP configuration
```

Fig.1: Contenu du fichier location1.txt

A mon retour, je voudrais charger le fichier location1.txt pour éviter d'avoir à rentrer les paramètres à la main.

La commande -f permet de restaurer une configuration sauvegardée au préalable dans un fichier à l'aide de la commande dump.
Voici la commande :

```
netsh –f C:\un_repertoire_au_choix\location1.txt
```

2.3. Commandes pour le scénario 2

« Je voudrais configurer les paramètres tcp/ip de ma machine en lignes de commandes. Eventuellement faire un fichier .cmd m'évitant de taper la même commande à chaque fois. »

La commande `netsh show interface` affiche la liste des interfaces réseau disponibles. Cela permet de retrouver les noms symboliques affectés aux cartes Ethernet tels qu'ils apparaissent dans le dossier **`Connexions réseau et accès à distance.`**

2.3.1. Commandes pour configuration statique

```
netsh int ip set addr name=« Nom_exact_de_la_connexion »
source=static addr=ip_machine mask=valeur_subnet_mask
gateway=ip_gateway gwmetric=1

netsh int ip set dns name=« Nom_exact_de_la_connexion »
source=static addr=ip_dns
```

2.3.2. Commandes pour configuration dhcp

```
netsh int ip set addr name=« Nom_exact_de_la_connexion »
source=dhcp

netsh int ip set dns name=« Nom_exact_de_la_connexion »
source=dhcp
```

2.3.3. Exemple

Le nom de mon interface réseau est : « Connexion au reseau local »

Ma configuration réseau au travail est : ip = 10.1.2.101, mask = 255.255.0.0, gateway = 10.1.0.1, dns = 10.1.1.10. Le contenu du fichier `labo.bat` sera :

```
netsh int ip set addr name="Connexion au reseau local"
source=static addr=10.1.2.101 mask=255.255.0.0 gateway=10.1.0.1
gwmetric=1

netsh int ip set dns name="Connexion au reseau local"
source=static addr=10.1.1.10
```

Fig. 2 : Contenu du fichier `labo.bat`

La configuration réseau chez moi est : ip = 192.168.123.189, mask = 255.255.255.0, gateway = 192.168.123.254, dns = 192.168.1.1. Le contenu du fichier `maison.bat` sera :

```
netsh int ip set addr name="Connexion au reseau local"
source=static addr=192.168.123.189 mask=255.255.255.0
gateway=192.168.123.254 gwmetric=1

netsh int ip set dns name="Connexion au reseau local"
source=static addr=192.168.1.1
```

Fig. 3 : Contenu du fichier `maison.bat`

2.3.4. Remarque

Chaque commande doit être effectuée sur 1 ligne du fichier ➔ 2 lignes.

On peut créer un petit fichier de script que l'on sauvegarde sous l'extension .bat ou .cmd. Le script n'est en fait que les commandes de configuration réseau *netsh*. Quand on clique sur ce fichier, les commandes se lancent automatiquement dans l'invite de commandes *dos*.

2.4. Commandes pour le scénario 3

« L'administrateur d'un réseau voudrait vérifier que toutes ses machines (postes de travail, serveurs, routeurs) soient atteignables. Il est fastidieux d'effectuer autant de *`ping`* qu'il y a de machines, il faudrait donc un outil permettant de tout vérifier en une seule fois. »

La commande `netsh diag gui` permet d'obtenir l'interface *Windows XP* de diagnostic réseau.
On peut définir les options d'analyse :

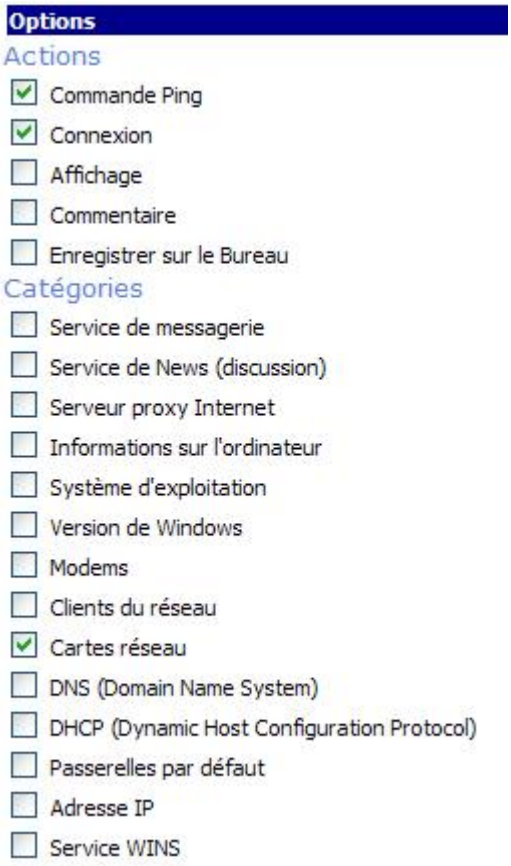


Fig. 4 : Options choisies pour `diag gui`

Ces options permettent de définir les tests suivant leur catégorie
Ici, on demande de faire les tests de connexions avec le serveur dns, le default gateway et notre propre ip qui sont définis comme paramètres de la connexion au réseau local (Carte réseau).

Voici le résultat :

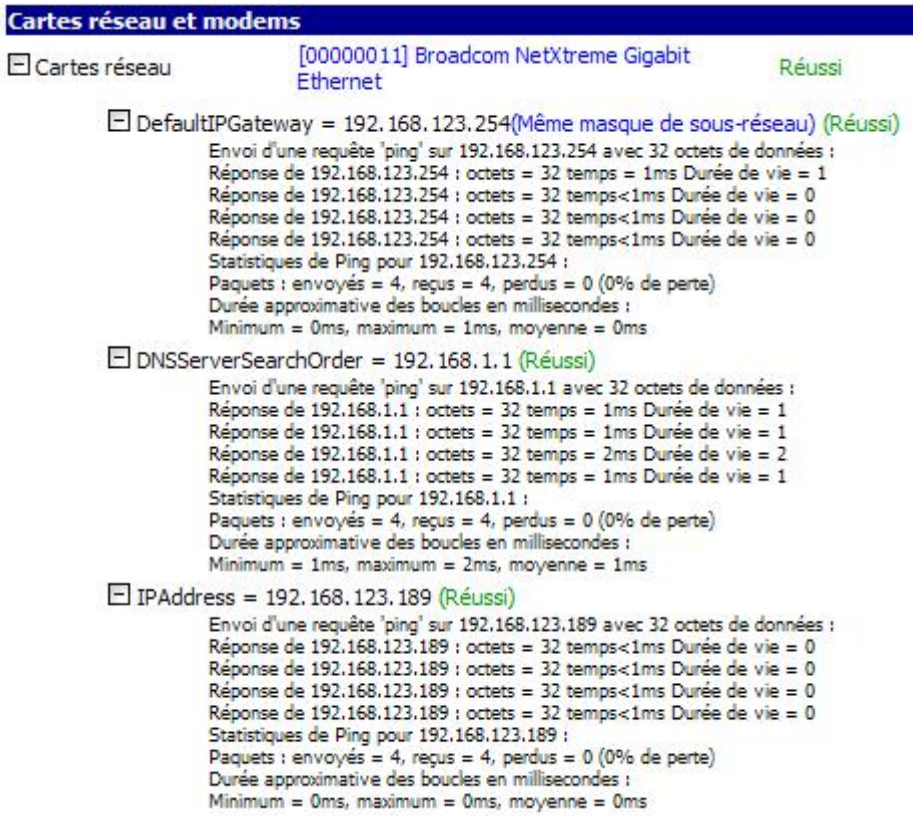


Fig. 5 : Résultat du diagnostic

On peut constater les requêtes « ping » effectuées.

Ce diagnostic réseau peut s'effectuer en lignes de commandes.
A la suite de `netsh diag`, taper `ping adapter` :

```
C:\Documents and Settings\Alexandre Badan>netsh diag ping adapter

Cartes réseau ([00000011] Broadcom NetXtreme Gigabit Ethernet)

DefaultIPGateway = 192.168.123.254 Même masque de sous-réseau
Envoi d'une requête 'ping' sur 192.168.123.254 avec 32 octets de données :
Réponse de 192.168.123.254 :  octets = 32 temps = 1ms  Durée de vie = 1
Réponse de 192.168.123.254 :  octets = 32 temps<1ms  Durée de vie = 0
Réponse de 192.168.123.254 :  octets = 32 temps<1ms  Durée de vie = 0
Réponse de 192.168.123.254 :  octets = 32 temps<1ms  Durée de vie = 0
Statistiques de Ping pour 192.168.123.254 :
Paquets : envoyés = 4, reçus = 4, perdus = 0 (0% de perte)
Durée approximative des boucles en millisecondes :
Minimum = 0ms, maximum = 1ms, moyenne = 0ms

DNSServerSearchOrder = 192.168.1.1
Envoi d'une requête 'ping' sur 192.168.1.1 avec 32 octets de données :
Réponse de 192.168.1.1 :  octets = 32 temps = 1ms  Durée de vie = 1
Réponse de 192.168.1.1 :  octets = 32 temps = 1ms  Durée de vie = 1
Réponse de 192.168.1.1 :  octets = 32 temps = 1ms  Durée de vie = 1
Réponse de 192.168.1.1 :  octets = 32 temps = 1ms  Durée de vie = 1
Statistiques de Ping pour 192.168.1.1 :
Paquets : envoyés = 4, reçus = 4, perdus = 0 (0% de perte)
Durée approximative des boucles en millisecondes :
Minimum = 1ms, maximum = 1ms, moyenne = 1ms

IPAddress = 192.168.123.189
Envoi d'une requête 'ping' sur 192.168.123.189 avec 32 octets de données :
Réponse de 192.168.123.189 :  octets = 32 temps<1ms  Durée de vie = 0
Réponse de 192.168.123.189 :  octets = 32 temps<1ms  Durée de vie = 0
Réponse de 192.168.123.189 :  octets = 32 temps<1ms  Durée de vie = 0
Réponse de 192.168.123.189 :  octets = 32 temps<1ms  Durée de vie = 0
Statistiques de Ping pour 192.168.123.189 :
Paquets : envoyés = 4, reçus = 4, perdus = 0 (0% de perte)
Durée approximative des boucles en millisecondes :
Minimum = 0ms, maximum = 0ms, moyenne = 0ms
```

Fig. 6 : `netsh diag ping adapter`

N.B : La commande `netsh diag show test` fait pareil mais teste en plus l'interface de rappel 127.0.0.1

2.5. Commandes pour le scénario 4

« J'aimerais détecter les LSP malveillants et les supprimer totalement. J'aimerais également forcer *Windows* à détruire et reconstruire à neuf son protocole *tcp/ip*. »

Un *LSP*, *Layered Service Provider* (couche de services entre les *Winsock* et la couche réseau), est un pilote système ancré dans les services réseau de *Windows*. Il a accès à chaque donnée qui entre ou qui sort de l'ordinateur, et a le pouvoir de les modifier. Quelques uns de ces *LSP*s sont nécessaires pour permettre à *Windows* de me connecter à d'autres ordinateurs, y compris *Internet*. Mais un *Spyware* peut aussi s'installer comme un *LSP*.
Les *spywares* sont des petits programmes indépendants, agissant un peu comme des trojans qui s'installent à notre insu lors de l'installation d'un prétendu *freeware*. Conçus pour se lancer en même temps que le système, pas toujours apparents dans le *Task Manager*, le faux *LSP* a ainsi accès à l'intégralité des données transportées et en profite pour les analyser et rediriger l'utilisateur vers les sites qui lui convient. Désinstaller le programme lié ne supprime pas le spyware, l'éradiquer empêche souvent toute connexion au *Web* car il a modifié *winsock*.

2.5.1. Réinitialisation de la pile tcp/ip

Avec *netsh* on peut remettre le *stack tcp/ip* tel qu'il était lors de l'installation du système.

```
netsh int ip reset lettre_du_lecteur:\nom_du fichier_log.log
```

Il faut spécifier un nom de fichier de log où toutes les actions de *reset* sont notées (environ 10 kB). Toutes les opérations sont effectuées dans les sous-clés de la clé `HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters` de la base de registre.

Voici 3 exemples d'actions typiques effectuées et notée dans le fichier :

- *deleted*
  SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\UseDomainNameDevolution
- *reset*
  SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{26FA1ED9-4E08-49C1-A500-60FAEBC531A7}\DefaultGateway
  old REG_MULTI_SZ = 192.168.123.254
- *added*
  SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\{358A6AE6-5457-4D11-88BD-8BA5C7C33E10}\AddressType

Fig. 7 : Exemple de `C:\reset.log`

On constate les actions *deleted, reset* et *added*. Ainsi, certaines sous-clés de la clé *SYSTEM\CurrentControlSet\Services\Tcpip\Parameters*\ont été supprimées, mises à jour ou ajoutées.

2.5.2. Réparation de Winsock

Une nouvelle commande de *netsh* apparue avec le *SP2* de *Windows XP* permet de réparer *Winsock* sans outil tiers.

Taper `netsh winsock reset catalog`. Tous les *LSP* non-standards du catalogue *Winsock* sont alors supprimés.

La commande `netsh winsock show catalog` permet de lister les *LSP* avant et après la réparation.

```
Entrée de fournisseur du catalogue Winsock

------------------------------------------------------------------

Type d'entrée :               Fournisseur de service debase

Description :                         MSAFD nwlnkspx [SPX II]

ID du fournisseur :                         {11058241-BE47-11CF-95C8-
00805F48A192

}

Chemin d'accès fournisseur :
%SystemRoot%\system32\mswsock.

dll

ID d'entrée de catalogue :                   1016

Version :                            2

Famille d'adresses :                      6

Longueur maximale d'adresse :                 16

Longueur minimale d'adresse :                 14

Type de socket :                         5

Protocole :                           1257

Longueur de chaîne de protocole :       1
```

Fig. 8 : Exemple de résultat de `netsh winsock show catalog`

On constate une entrée de type *MSAFD* qui est standard. Une autre entrée standard est de type *RSVP.* Tout autre type d'entrée dans les fournisseurs du catalogue *Winsock* (*LSP*) serait un *spyware*.

2.5.3. Remarque

La réinitialisation de la pile *tcp/ip*, rétablit une configuration dhcp des paramètres de la `connexion au réseau local`. Il faut donc avoir recours aux commandes de la section 2.3.1. pour bien reconfigurer.

2.6. Commandes pour le scénario 5

« Je suis administrateur réseau et je désire configurer ou auditer une machine distante du réseau. »

La commande `netsh –r addr_ip_machine_distante` permet de réaliser ce scénario. Les commandes *netsh* sont ainsi transportées sur une machine distante spécifiée par son adresse *ip*.

2.6.1. Test 1

Fig. 9 : Configuration pour commande `netsh –r 10.1.2.102`

Voici le résultat :

```
C:\Documents and Settings\Alexandre Badan>netsh -r 10.1.2.103

AVERTISSEMENT : impossible d'obtenir des renseignements sur
l'hôte à partir de l
'ordinateur : [10.1.2.103]. Certaines commandes peuvent ne pas
être disponibles.

Accès refusé.
```

Fig. 10 : Résultat de la commande `netsh –r 10.1.2.102`

On constate que l'accès est refusé.

Analysons pourquoi l'accès est refusé grâce à la capture *Ethereal* du traffic généré par la commande `netsh –r 10.1.2.102`.

```
           11    10.1.2.103            10.1.2.101            TCP      epmap
> 1343 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
           12    10.1.2.101            10.1.2.103            TCP      1343 >
epmap [ACK] Seq=1 Ack=1 Win=65535 Len=0
           13    10.1.2.101            10.1.2.103            DCERPC   Bind:
call_id: 2 [ISystemActivator] UUID: 000001a0-0000-0000-c000-000000000046 ver
0.0, NTLMSSP_NEGOTIATE
           14    10.1.2.103            10.1.2.101            DCERPC
Bind_ack: call_id: 2, NTLMSSP_CHALLENGE accept max_xmit: 5840 max_recv: 5840
           15    10.1.2.101            10.1.2.103            DCERPC   AUTH3:
call_id: 2, NTLMSSP_AUTH
           16 0.007295    10.1.2.101            10.1.2.103            DCERPC
Request: call_id: 2 opnum: 4 ctx_id: 1
           18 0.013376    10.1.2.103            10.1.2.101            DCERPC
Fault: call_id: 2 ctx_id: 1 status: Unknown (0x00000005)
```

Fig. 11 : Capture de la commande netsh –r 10.1.2.102

On constate avec les paquets 13, 14 et 15 qu'un challenge d'authentification est tenté. L'accès refusé vient du fait que l'authentification échoue.
Le problème est que les 2 machines ne sont pas dans un domaine, il n'y a, par conséquent, pas possibilité de s'authentifier de façon centralisée. Il faut, pour que cette commande fonctionne, que l'utilisateur soit le même sur les 2 machines membres du même domaine.

2.6.2. Test 2

Je vais utiliser la configuration *Virtual PC* du labo *XP* 4 mise en œuvre dans ce diplôme (Partie *Virtual PC 2004*, section 4) afin de tester cette commande `netsh –r`. Le fait d'émuler un réseau avec *Virtual PC* permet d'éviter de mobiliser plusieurs machines physiques pour faire des tests. De plus, les fonctionnalités et résultats sont identiques. (Voir partie *Virtual PC 2004* pour plus d'explications)

Le schéma suivant montre la configuration réseau mise en œuvre pour `netsh –r` :

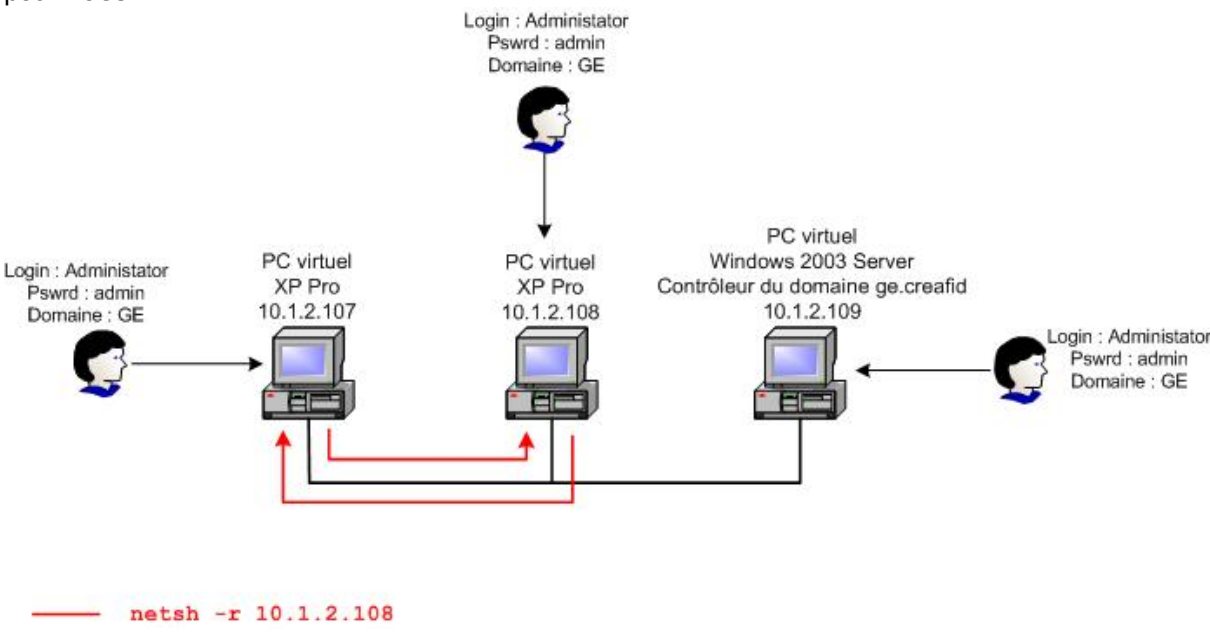


Fig. 12 : Configuration réseau pour commande `netsh –r 10.1.2.108`

Voici les paquets intéressants concernant l'authentification :

```
     13 0.012798    10.1.2.107           10.1.2.108           DCERPC
Bind: call_id: 2 UUID: ISystemActivator, NTLMSSP_NEGOTIATE
     14 0.025869    10.1.2.108           10.1.2.107           DCERPC
Bind_ack: call_id: 2, NTLMSSP_CHALLENGE accept max_xmit: 5840
max_recv: 5840
     15 0.026738    10.1.2.107           10.1.2.108           DCERPC
AUTH3: call_id: 2, NTLMSSP_AUTH
     26 0.087559    10.1.2.107           10.1.2.108           DCERPC
AUTH3: call_id: 1
     33 0.095814    10.1.2.107           10.1.2.108           DCERPC
Bind: call_id: 2 [IWbemLoginClientID] UUID: d4781cd6-e5d3-44df-ad94-
930efe48a887 ver 0.0
     35 0.097373    10.1.2.107           10.1.2.108           DCERPC
AUTH3: call_id: 2
     36 0.097447    10.1.2.107           10.1.2.108           DCERPC
Request: call_id: 2 opnum: 3 ctx_id: 1 [IWbemLoginClientID] UUID:
d4781cd6-e5d3-44df-ad94-930efe48a887 rpcver: 0
     38 0.101504    10.1.2.108           10.1.2.107           DCERPC
Response: call_id: 2 ctx_id: 1 [IWbemLoginClientID] UUID: d4781cd6-
e5d3-44df-ad94-930efe48a887 rpcver: 0
     42 0.103180    10.1.2.107           10.1.2.108           DCERPC
Bind: call_id: 3 [IWbemLevel1Login] UUID: f309ad18-d86a-11d0-a075-
00c04fb68820 ver 0.0
     45 0.105294    10.1.2.107           10.1.2.108           DCERPC
AUTH3: call_id: 3
```

Fig. 13 : Résultat de `netsh –r 10.1.2.108` sur 10.1.2.107

On constate qu'un challenge d'authentification est tenté et cette fois il réussit. Le fait d'être dans un domaine est donc la solution pour faire fonctionner cette commande *netsh*.

Maintenant que les commandes netsh ont été transportées avec succès sur la machine distante 10.1.2.108.

Cependant, on ne peut pas configurer les paramètres tcp/ip à distance (Scénario 2). C'est logique puisqu'on a besoin de l'adresse *ip* de la machine distante pour effectuer les manipulations.

3. Conclusion

Les 5 scénarios ont été réalisés avec succès. *netsh* est donc un outil permettant de configurer et auditer sa propre machine en lignes de commande.

Netsh permet également de réparer *Winsock* et réinitialiser la pile *TCP/IP* de l'*OS*.

On a vu qu'il est aussi possible d'auditer une machine distante avec *netsh* pour peu qu'elle soit membre du même domaine et que la session ouverte sur cette machine le soit par le même utilisateur et le même *login.* On notera qu'un administrateur réseau préfèrera certainement utiliser l'outil *VNC* pour se connecter à une machine.

4. Sources

http://www.bellamyjc.net/fr/windows2000.html#NETSH → Bon lien sur l'utilisation des commandes *netsh*

http://babin.nelly.free.fr/spyware_1.htm → On y trouve des informations très pertinentes sur les fameux *spywares*

http://assiste.free.fr/p/comment/restaurer_winsock_lsp.php → informations sur la réparation de *Winsock* (très complet)

http://support.microsoft.com/?kbid=299357 → Comment réinitialiser la pile *tcp/ip*

http://www.brienposey.com/kb/diagnosing_connectivity_with_netsh.asp → bon article concernant la commande `netsh diag`

PARTIE 2 :

Encryption File System (EFS)

(4 semaines d'étude)

1. Pourquoi chiffrer des données ?

1.1. Besoin d'un service de confidentialité

Authentifier un utilisateur quand il ouvre une session ou donner des autorisations *NTFS* sur les dossiers permet de protéger des données confidentielles. Cependant, une personne mal intentionnée ayant un accès physique à une machine, comme par exemple un ordinateur portable volé, pourrait y installer un nouveau système d'exploitation et passer outre la sécurité du système déjà existant. A ce moment, les données sensibles seraient exposées. Chiffrer ces données avec le système *EFS* permet d'ajouter un degré supplémentaire de sécurité car les données sont protégées même si une personne non autorisée avait un accès total à la machine.

Seuls les utilisateurs autorisés ainsi que les agents désignés de recouvrement des données peuvent déchiffrer les fichiers chiffrés. Tous les autres comptes, y compris des comptes administrateurs, verront l'accès refusé au fichier chiffré s'ils n'ont pas été autorisés par le propriétaire du fichier ou s'ils n'ont pas été désignés comme agents récupérateurs de données.

1.2. Quoi chiffrer ?

Les fichiers individuels et les dossiers (ou sous-dossiers) d'un système formaté *NTFS* peuvent être chiffrés. Quand un dossier est chiffré, tout fichier ou dossier créé ou déplacé à l'intérieur sera automatiquement chiffré. Quand on parle de dossier chiffré, on sous-entend que la totalité de ses fichiers sont chiffrés.

Les fichiers système et tous les fichiers situés dans le répertoire `\WINDOWS` localisé à la racine de la partition d'installation d'*XP* ne peuvent pas être chiffrés avec *EFS*. Un fichier compressé sera automatiquement décompressé s'il est chiffré.

2. Rappels théoriques

2.1. Chiffrement symétrique

Appelé communément « chiffrement à clé partagée », il utilise la même clé pour chiffrer et déchiffrer les données.

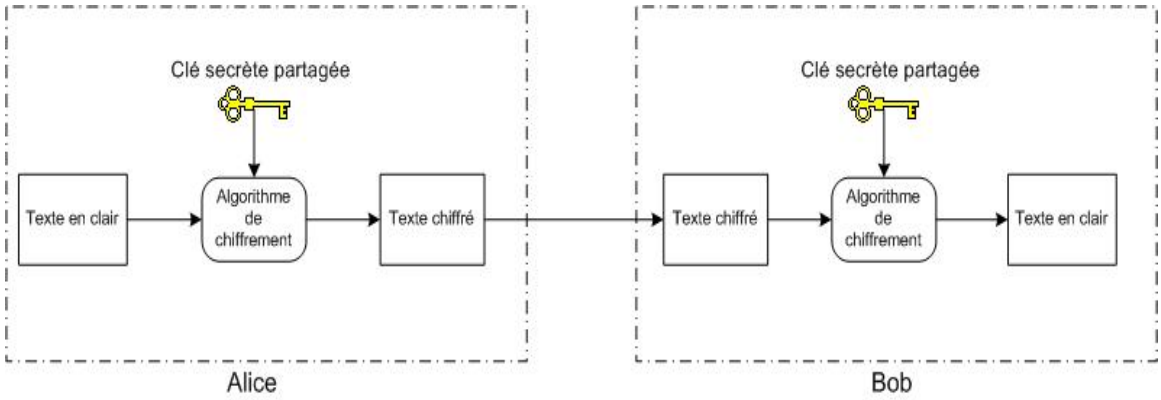


Fig. 14 : Chiffrement symétrique

L'avantage de ce type d'algorithme est sa rapidité à chiffrer de grandes quantités de données. Le désavantage est la distribution de la clé quand plusieurs utilisateurs ont besoin d'accéder aux données chiffrées. Les mécanismes de chiffrement asymétrique apportent une solution à ce problème.

2.2. Chiffrement asymétrique

Les algorithmes de chiffrement asymétrique utilisent la notion de « clé publique / clé privée ». La clé publique d'un utilisateur peut être donnée à n'importe quel autre utilisateur alors que sa clé privée est personnelle et ne doit en aucun cas être transmise. Si quelqu'un l'intercepte, la confidentialité de l'entité chiffrée n'est plus garantie.

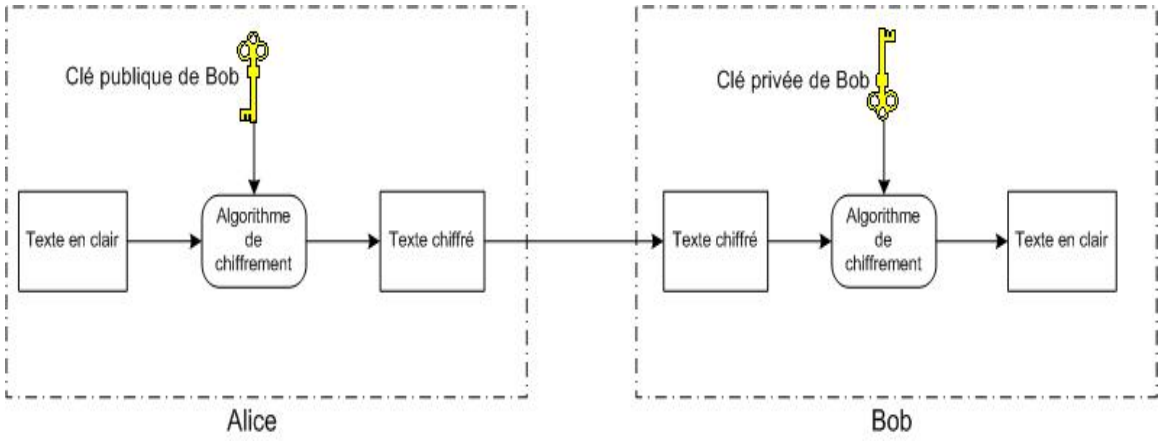


Fig. 15 : Chiffrement asymétrique

Alice a obtenu au préalable la clé publique de Bob. L'intégrité de cette clé a été garantie grâce au mécanisme de certification.

2.3. *Certification Authority (CA)*

Une *CA* a pour rôle d'établir et de garantir l'authenticité des clés publiques propres aux utilisateurs.

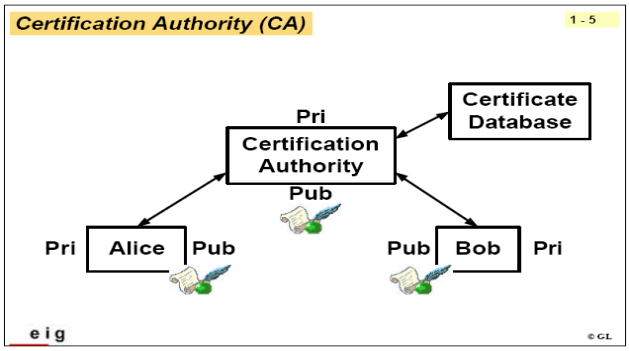


Fig. 16 : *CA* (Cours *PKI* GL)

2.4. Certificat à clé publique

Il s'agit d'un document numérique permettant de certifier que la clé publique correspond à la clé privée appartenant à une entité. Les certificats sont signés numériquement par une *CA* et peuvent être transmis à un utilisateur.

Les certificats sont des petits fichiers divisés en deux parties :

- Informations
- Signature de la *CA*

Les informations contiennent :

- Le nom de la *CA*
- Le nom du propriétaire du certificat
- La date de validité du certificat
- L'algorithme de chiffrement utilisé
- La clé publique du propriétaire

Ces informations sont signées par la *CA*, c'est-à-dire qu'une fonction de hachage crée un haché de ces informations qui est ensuite chiffré avec la clé privée de la *CA*.

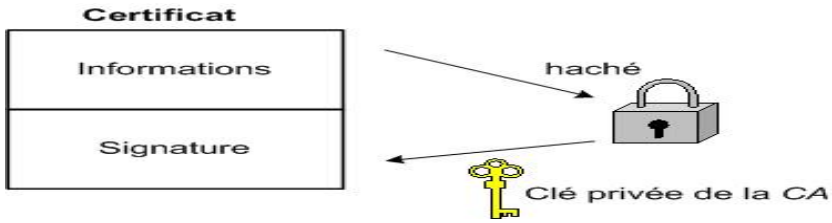


Fig. 17 : Signature des informations par la *CA*

Lorsqu'un utilisateur désire communiquer avec une autre personne, il lui suffit de se procurer le certificat du destinataire. Ce certificat contient le nom du destinataire, ainsi que sa clé publique et est signé par la *CA*. Il est donc possible de vérifier la validité du message en appliquant d'une part la fonction de hachage aux informations contenues dans le certificat, en déchiffrant d'autre part la signature de l'autorité de certification avec la clé publique de cette dernière et en comparant ces deux résultats.

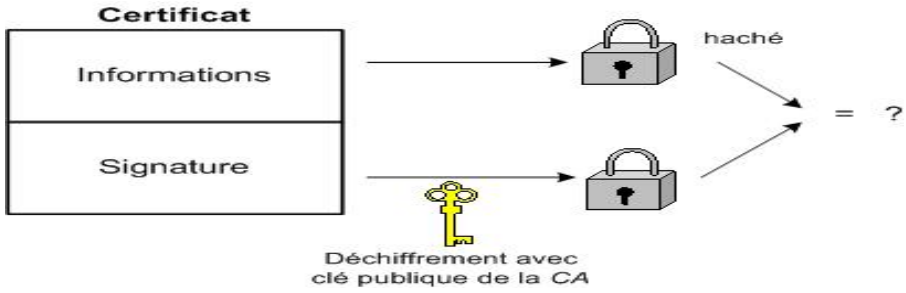


Fig. 18 : Vérification de la validité du message

Un certificat est dit autosigné quand il n'a pas été délivré par une *CA*. Le système EFS, dans le cadre d'une machine stand-alone, génère ce type de certificats (Voir partie 4 pour plus de détails).

2.5. Ordinateur *stand-alone*

Il s'agit d'une machine isolée, ne partageant pas de ressources avec une autre machine, ne faisant pas partie d'un domaine.

3. Système cryptographique Microsoft

Les trois éléments du système de chiffrement Microsoft sont le système d'exploitation, l'application et le *CSP*. Les applications communiquent avec le système d'exploitation par l'intermédiaire de la couche *CryptoAPI*, et le système d'exploitation communique avec les *CSP* via l'interface de fournisseurs de services cryptographiques (*CSPI*)

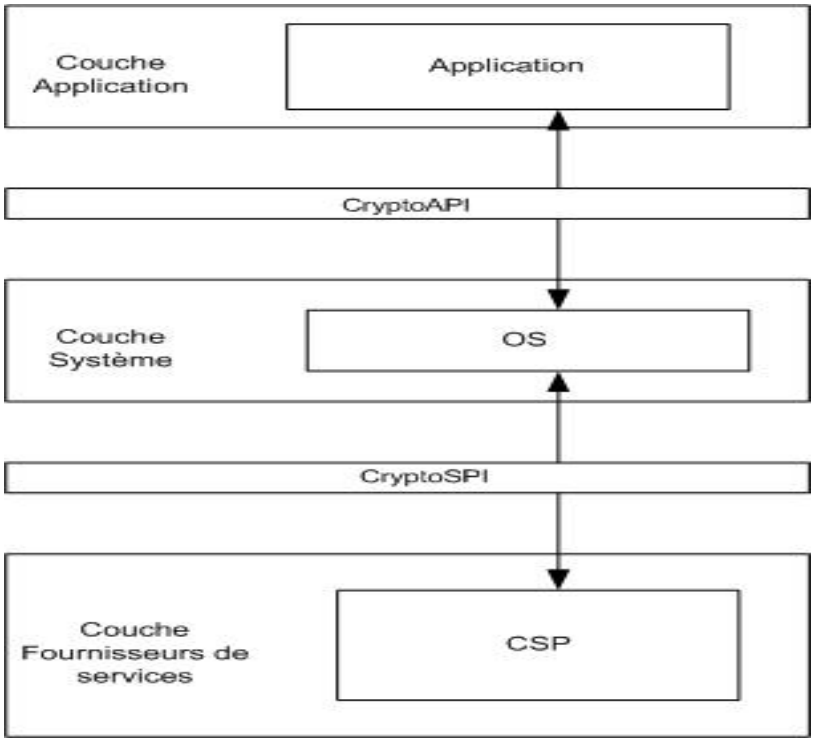


Fig. 19 : système cryptographique Microsoft

Le contexte CryptoAPI contient les fonctions permettant d'utiliser les clés générées par un CSP et donc de faire le lien entre l'*OS* et les applications concernées. Toutes les opérations cryptographiques sont réalisées par des modules indépendants appelés les *CSP* (*cryptographic service providers*). *Windows XP Pro* possède son propre *CSP* appelé le fournisseur de base *RSA (MS_DEF_PROV)*.

Le rôle du *CSP* est de générer les clés symétrique et asymétriques.

4. Fonctionnement *EFS*

4.1. Principe du chiffrement

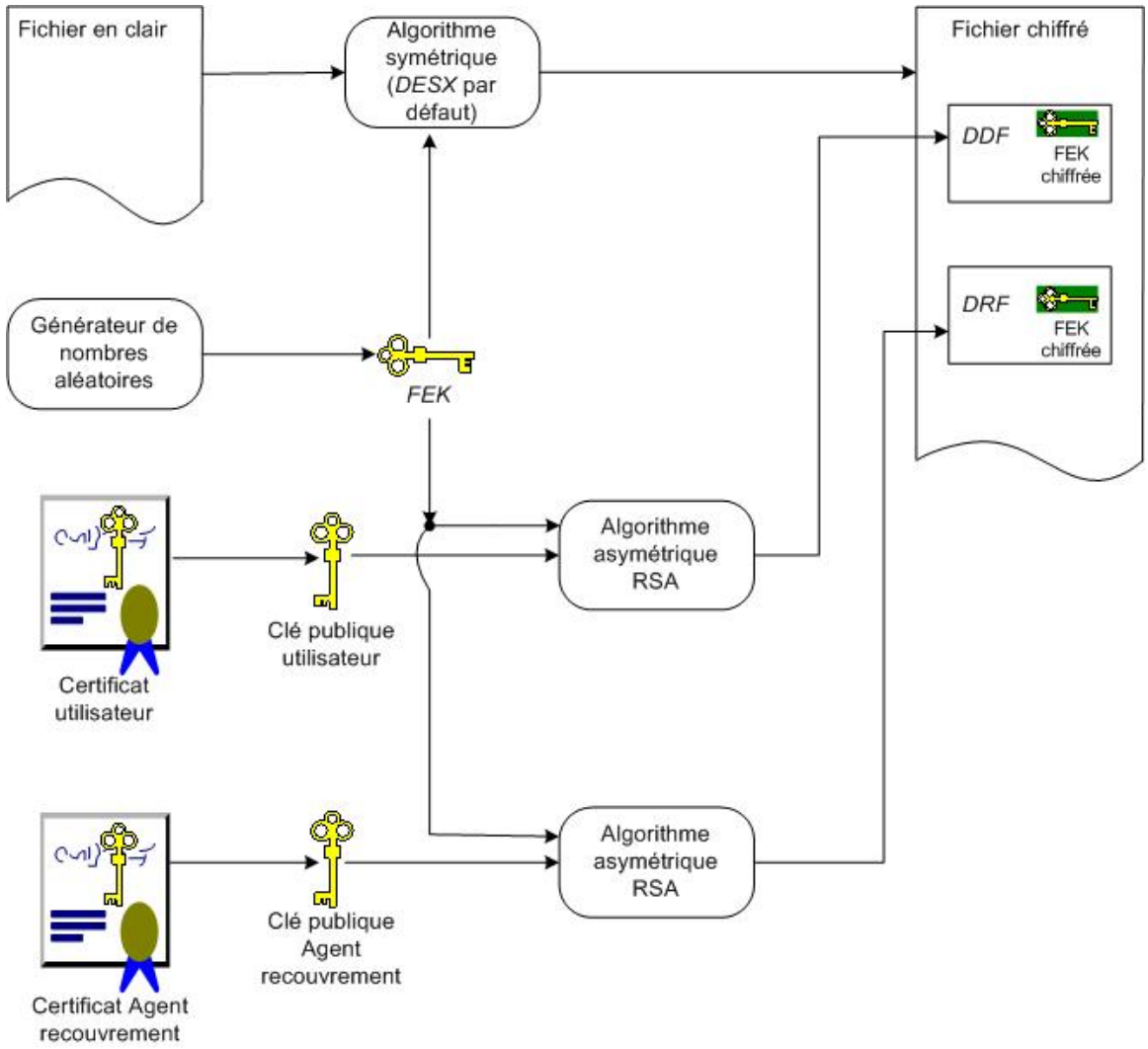


Fig. 20 : Principe du chiffrement

L'en-tête d'un fichier chiffré est constitué des champs de déchiffrement des données et de récupération des données :

4.1.1. Champ de déchiffrement des données (*DDF*)

Un fichier chiffré contient au minimum la *FEK* chiffrée avec la clé publique du propriétaire de ce fichier. Le champ de stockage de cette *FEK* chiffrée s'appelle le champs de déchiffrement des données. Si n utilisateurs partagent ce fichier, il y aura n champs de déchiffrement des données.

4.1.2. Champ de recouvrement des données (*DRF*)

Si l'administrateur d'un ordinateur désigne p agent(s) de recouvrement des données, il y aura p champ(s) de recouvrement des données dans l'en-tête du fichier chiffré.

La littérature, en particulier le « kit de ressource et de documentation XP », utilise abondamment l'abréviation *DDF* pour le champs de déchiffrement des données (*Data Decryption Field*) et *DRF* pour le champs de récupération des données (*Data Recovery Field*).

4.1.3. Agent de recouvrement (*DRA*)

Le problème majeur du chiffrement de données est de ne plus pouvoir les déchiffrer. Si l'administrateur décide de désactiver le compte du propriétaire du fichier, il serait bien que les utilisateurs autorisés à partager ce fichier puissent continuer à le faire. Cela nécessite la présence d'un agent de recouvrement des données. S'il y a perte de la clé privée utilisateur permettant de déchiffrer la clé symétrique de chiffrement, personne ne peut plus ouvrir le fichier ou le répertoire crypté. On va donc avoir recours à la stratégie de récupération qui est définie par l'administrateur. Il désigne des délégués connus sous le nom d'agents de recouvrement ou *DRA* (*Data Recovery Agent*). Un agent de recouvrement est créé de manière classique par l'administrateur de la machine qui va faire en sorte de le mettre dans le groupe des Opérateurs de sauvegarde. Ce groupe est défini comme pouvant passer outre les restrictions de sécurité uniquement dans le but d'effectuer la sauvegarde ou la restauration de fichiers.

Sous Windows2000 Pro il était impossible de chiffrer des fichiers avec *EFS* si un ADR n'avait pas été créé au préalable. Sous *WindowsXP Pro*, un agent de recouvrement n'est plus nécessaire mais il y a danger car le problème énoncé persiste. Il est donc indispensable de posséder au moins un agent de recouvrement de données.

Voir l'annexe paragraphe 4 pour la création pratique d'un agent de recouvrement.

4.2. Génération et stockage des clés symétrique et asymétriques

- La clé symétrique appelée *FEK (File Encryption Key)* est générée par un générateur de nombres aléatoires implémenté dans le *CSP*. Cette clé permet de chiffrer les données. A chaque opération de chiffrement de données par un utilisateur d'*EFS*, une *FEK* est générée, chiffrée avec une clé publique et stockée dans l'en-tête du fichier dans un champ correspondant à cet utilisateur (Voir figure 9). La clé privée permettra de déchiffrer la *FEK*.

- Les clés asymétriques sont générées par le *CSP*. En fait, *EFS* génère une demande de certificat vers *CryptoAPI* qui va passer les informations au *CSP*. Le *CSP* génère alors la paire de clés publique/privée. La clé privée est stockée dans un fichier du répertoire `%userprofile%\Application Data\Microsoft\Crypto\RSA\.`

  La clef **privée** contenue dans ce fichier est chiffrée avec une clé de session, elle même dérivée d'une clé maître.
  Cette clé maître est générée (par hachage *HMAC* et *SHA1*) à partir du mot de passe du compte utilisateur, et est stockée dans un fichier situé dans le dossier : `%userprofile%\Application Data\Microsoft\Protect\<SID du compte>.`

  S'il y a plusieurs fichiers dans ce dossier, le nom de celui à utiliser est dans le contenu du fichier `preferred` (Voir section 6.1. pour plus d'explications sur les niveaux de chiffrement).

  La clé publique est envoyée avec son certificat vers une *CA* disponible afin qu'elle puisse être reconnue digne de confiance. S'il n'y a pas de *CA*, le certificat est autosigné. Enfin la clé privée est stockée avec son certificat dans le répertoire `%userprofile%\Application Data\Microsoft\SystemCertificates\My\Certificates.` Le nom du fichier est l'empreinte numérique (hachage *SHA1*) du certificat.

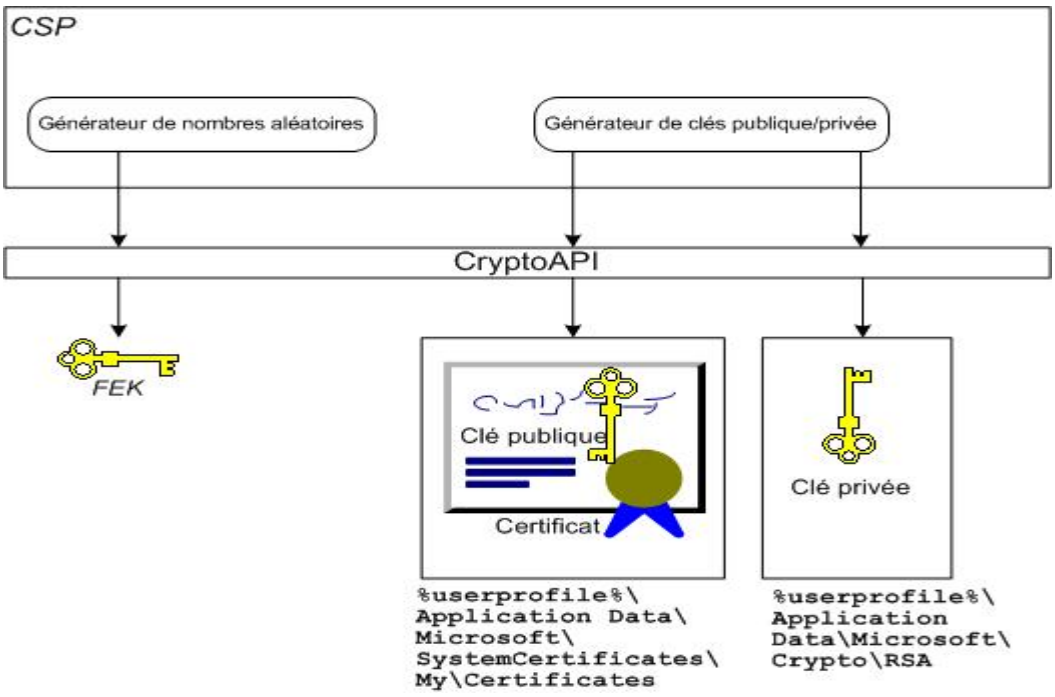


Fig. 21 : Génération et stockage des clés

4.3. Processus de chiffrement

- Ouverture du fichier non-chiffré en accès exclusif ;
- Les données sont copiées dans un fichier temporaire en texte brut dans le dossier *System Volume Information* situé à la racine de la partition ;
- Accès au contexte *CryptoAPI* (fonction *CryptAcquireContext* de l'*API WinCrypt*) ;
- Génération de la clé de chiffrement symétrique *FEK* par le *CSP MS_DEF_PROV* (*fournisseur de base RSA*) ;
- Si *EFS* est appelé la 1ère fois, *le CSP* génère une paire de clés publique/privée.
- Création du champs de déchiffrement des données dans le fichier
- Placement, dans ce champ, de la *FEK* chiffrée avec la clé publique utilisateur;
- Création du champs de recouvrement des données dans le fichier
- Placement, dans ce champ, de la *FEK* chiffrée avec la clé publique de l'agent de recouvrement;
- Création, dans le dossier chiffré, d'un fichier temporaire dans lequel les données en clair sont copiées
- Les données en clair du fichier sont chiffrées avec la *FEK*.

4.4. Principe du déchiffrement

4.4.1. Par utilisateur

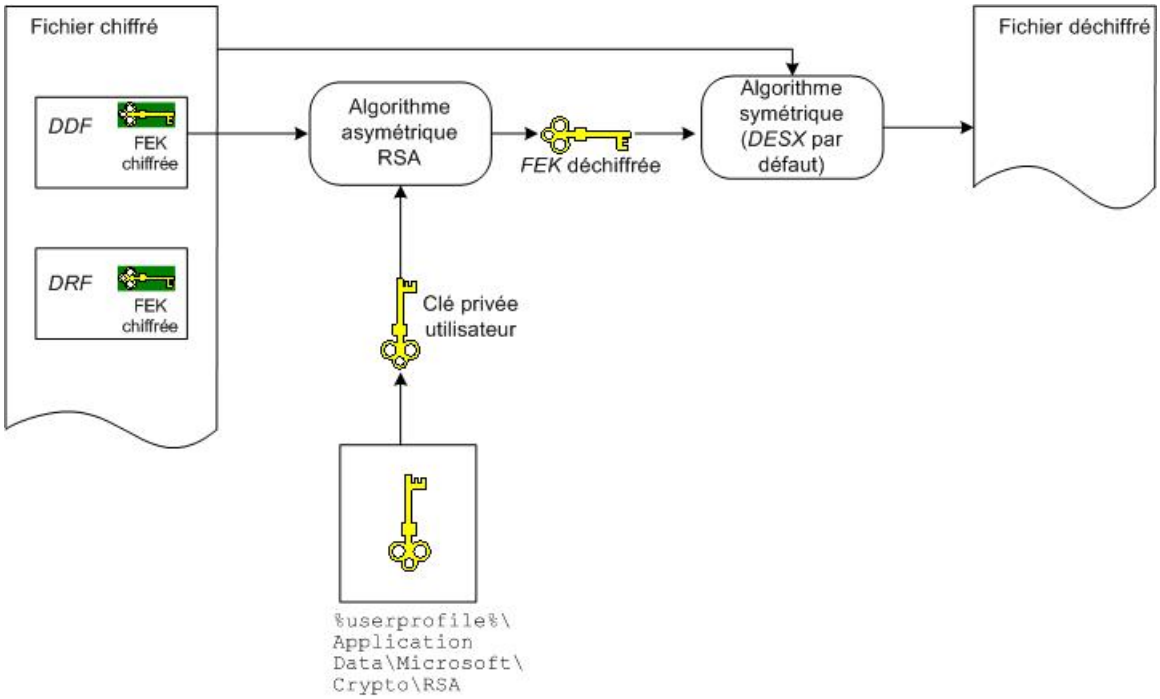


Fig. 22 : Déchiffrement par utilisateur

- La clé privée de l'utilisateur déchiffre la *FEK* stockée dans le champ de déchiffrement (*DDF*) des données.
- *FEK* déchiffre les données.
- Le fichier est déchiffré.

Le principe est le même pour l'agent de recouvrement à part que sa clé privée déchiffre la *FEK* stockée dans le champs de recouvrement des données (*DRF*).

4.4.2. Par Agent de recouvrement

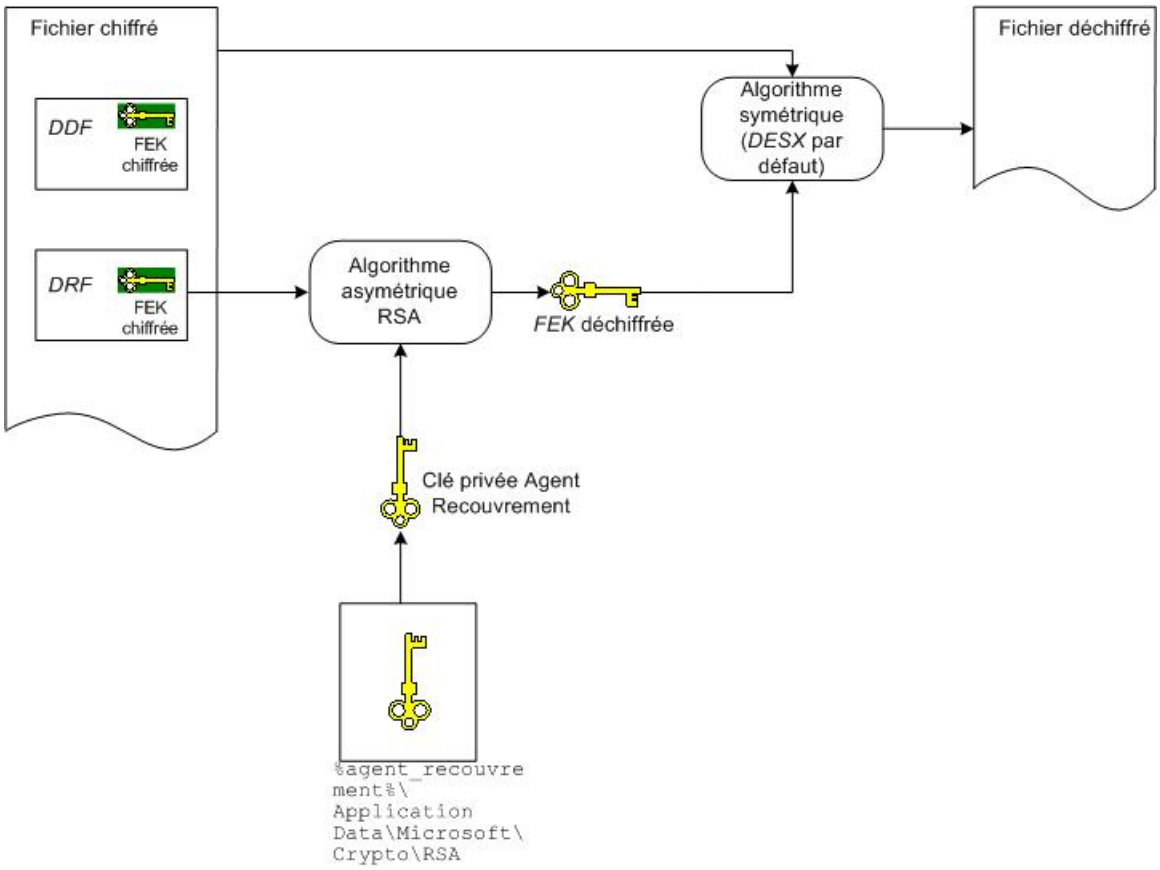


Fig. 23 : Déchiffrement par Agent de recouvrement

4.5. Architecture *EFS*

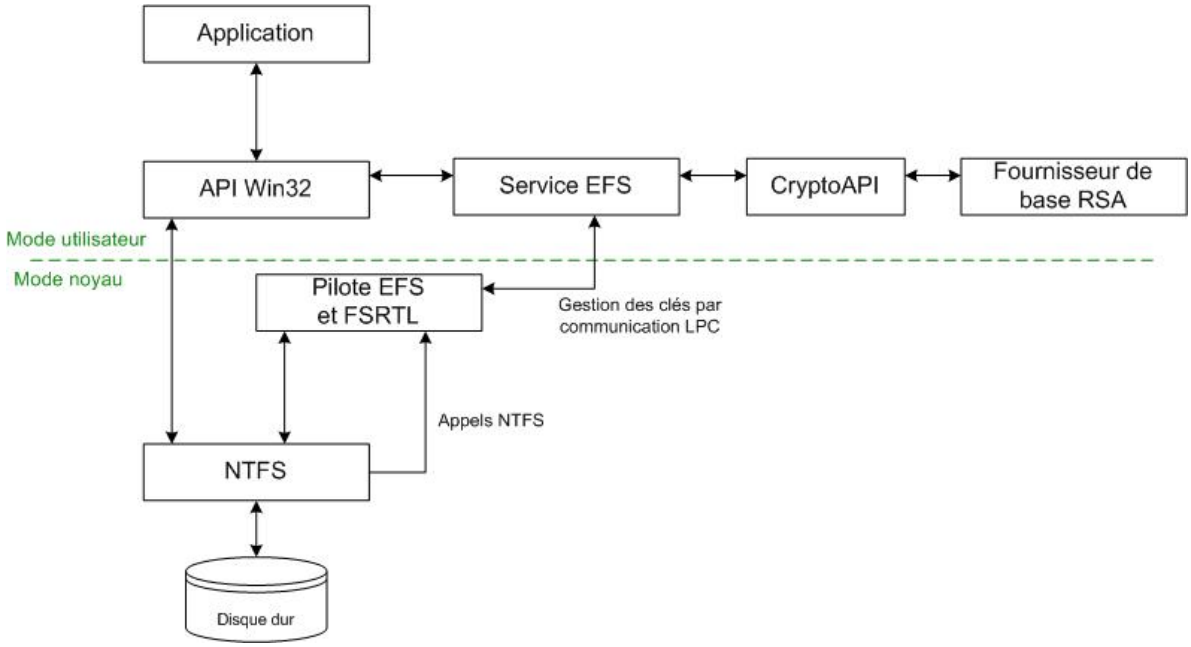


Fig. 24 : Architecture EFS

CryptoAPI

fournit des services qui permettent aux développeurs d'applications d'ajouter des schémas de chiffrement/déchiffrement de données, d'authentifier à l'aide de certificats numériques. Les développeurs peuvent utiliser les fonctions de *CryptoAPI* sans connaître en détail l'implémentation sous-jacente. *CryptoAPI* fonctionne avec plusieurs fournisseurs de services cryptographiques (*CSP*) qui exécutent les fonctions cryptographiques proprement dites, telles que le chiffrement, le déchiffrement, le stockage et la sécurité des clés.

Fournisseur de base *RSA*

Les *CSP* sont des modules indépendants, généralement une DLL, qui exécutent toutes les opérations de chiffrement. Ils sont écrits pour être indépendants d'une application particulière afin que n'importe quelle application puisse s'exécuter avec plusieurs *CSP*. En réalité, certaines applications ont des exigences spécifiques qui nécessitent un CSP personnalisé.

Le *CSP* utilisé pour *EFS* est le **fournisseur de base *RSA***. Il prend en charge la signature numérique et le chiffrement de données.

Applications

Elles peuvent utiliser les fonctions *CryptoAPI* pour générer et échanger des clés, chiffrer et déchiffrer des données, coder et décoder des certificats, gérer et sécuriser des certificats,créer et vérifier des signatures numériques, et calculer du hachage.

service *EFS*

Il appelle *CryptoAPI* pour obtenir la clé symétrique de chiffrement de fichier, puis pour la chiffrer, produisant ainsi le champ de déchiffrement des données dans l'en-tête du fichier chiffré.

FSRTL

(*File System RunTime Library*) est un module du pilote *EFS* qui implémente les appels *NTFS* pour diverses opérations du système de fichiers (lecture, écriture, chiffrement, déchiffrement ou recouvrement des dossiers et fichiers chiffrés).

Pilote *EFS*

Il communique avec le service *EFS* pour requérir les *FEK*, champs de déchiffrement et de recouvrement des données et autres services de gestion des clés.

FSRTL et le pilote *EFS* ne communiquent pas directement, mais par l'intermédiaire des appels de contrôle *NTFS*. Ainsi le système de fichiers prend part à toutes les opérations de chiffrement.

Sources :

- http://www.microsoft.com/france/msdn/technologies/technos/windowsmobile/info/info.asp?mar=/france/msdn/technologies/technos/windowsmobile/info/creationcesecurise.html&xmlpath=/france/msdn/technologies/technos/windowsmobile/mobile_inforef.xml&rang=23

- Kit de ressources et de documentation Microsoft XP Professionnel p.750

4.6. Partage multi utilisateurs

EFS permet aux utilisateurs de chiffrer des fichiers sans droit administratif. Il autorise le partage de fichiers chiffrés entre n utilisateurs par l'utilisation de leurs clés publiques/privées. Le fichier chiffré doit être dans un dossier partagé pour que les autres utilisateurs puissent y accéder. Le propriétaire du fichier peut ajouter des utilisateurs à la liste des autorisés. Le mot « liste » est plus approprié que « groupe » car la notion de « groupe » sous *Windows* est propre à la stratégie établie par l'administrateur en regard des autorisations *NTFS*.

Il est cependant indispensable que l'administrateur établisse une stratégie de groupe intelligente. Il serait impensable, du point de vue stratégie de sécurité, de mettre les utilisateurs dans le même groupe que l'agent de recouvrement des données puisque ce dernier est désigné pour récupérer les données chiffrées après un incident.

Si l'administrateur n'est pas inclus par le propriétaire du fichier dans la liste des autorisés, il ne pourra jamais lire le fichier chiffré. Mais comme il désigne le ou les agent(s) de recouvrement(s), il pourra en ouvrant une session d'agent de recouvrement, lire les données sans problème.

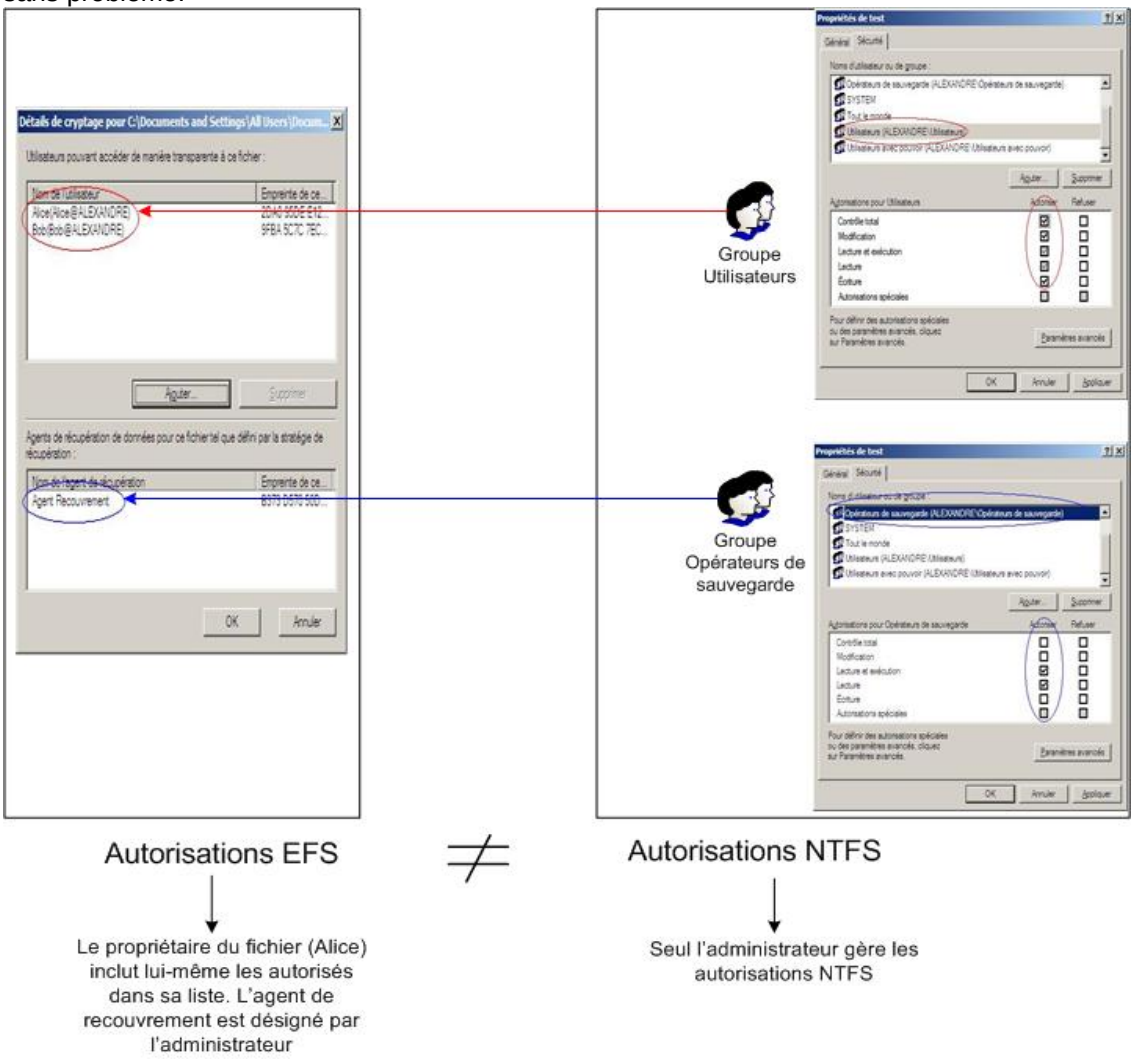


Fig. 25 : Autorisations EFS Vs. Autorisations NTFS

5. Scénario

5.1 Rôles

La configuration multi utilisateurs est effectuée sur un ordinateur portable que l'on appellera *Portable.* Il est intéressant de lui attribuer plusieurs utilisateurs afin d'observer le comportement d' *EFS* en ce qui concerne le partage de fichier.

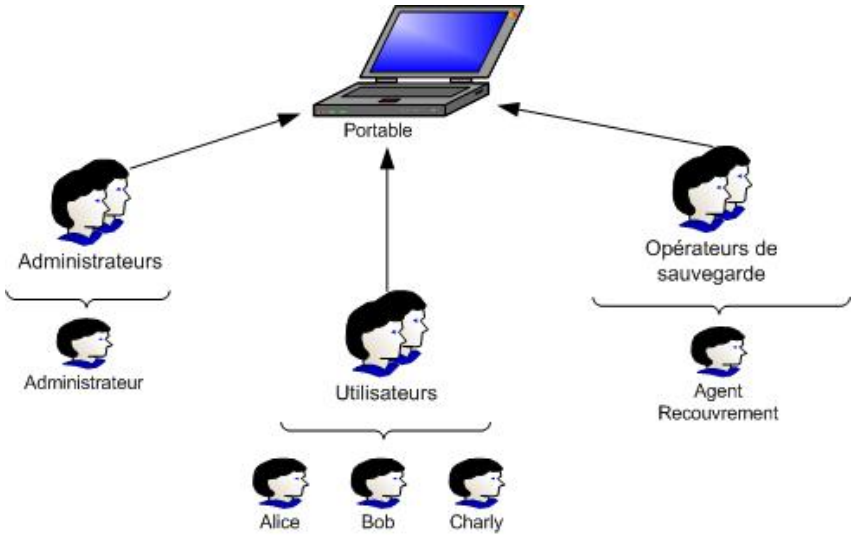


Fig. 26 : Rôles

- Alice : utilisateur, propriétaire du dossier chiffré TEST_EFS
- Bob : utilisateur, ami d'Alice, autorisé par Alice à partager le fichier chiffré.
- Charly : utilisateur, ennemi d'Alice, interdit par Alice de partager le fichier chiffré.
- Agent Recouvrement : agent de recouvrement désigné par l'administrateur.
- Administrateur : l'administrateur de la machine. Il ne sert qu'à créer les comptes et à attribuer des droits de contrôle sur le fichier chiffré. Il n'intervient pas dans le chiffrement *EFS*.

Le fichier chiffré `Test` est disponible dans le dossier partagé `C:\Documents and Settings\All Users\Documents\TEST_EFS`

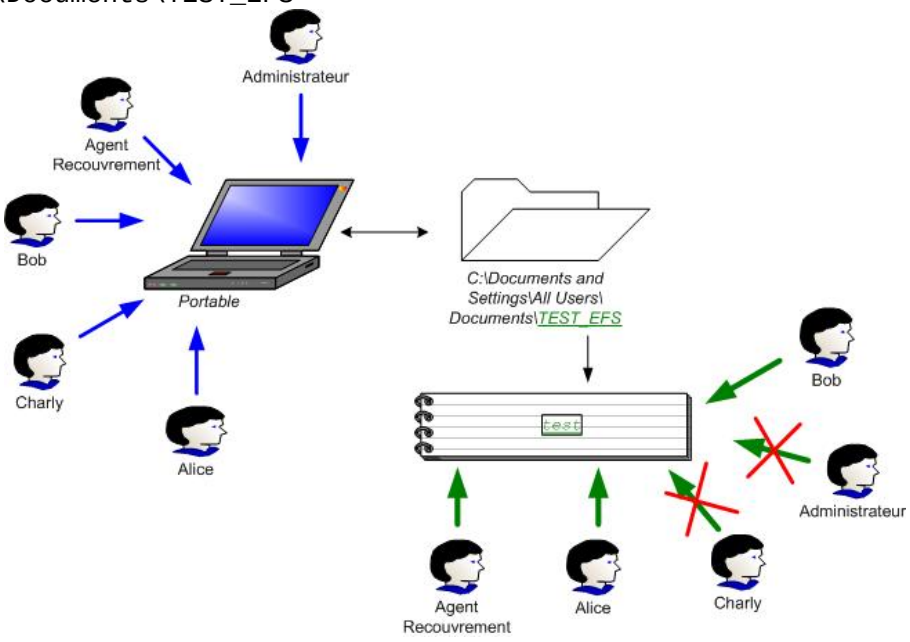


Fig. 27 : Scénario

La couleur verte (par défaut) du dossier *TEST_EFS* et du fichier *Test* signifie « chiffré par *EFS* ».

Alice est le propriétaire du fichier, elle le chiffre et n'autorise que Bob à le lire. Même si elle ne veut pas que l'agent de recouvrement puisse le lire, elle ne peut rien y faire car il a été désigné par l'administrateur. Il sera également l'agent de recouvrement de n'importe quel fichier chiffré par n'importe quel utilisateur.

La mise en œuvre pratique est disponible dans l'annexe de ce rapport

Il est également possible de chiffrer en lignes de commandes. Voir annexe section 5.

5.2. Récupération des données après incident

Alice, en explorant les fichiers cachés de son profil, arrive dans le répertoire suivant :

C:\Documents and Settings\Alice\Application Data\Microsoft\Crypto\RSA\S-1-5-21-220523388-2139871995-725345543-1005

Il s'agit du répertoire contenant sa clé privée de déchiffrement mais elle ne le sait pas. Pensant que ce fichier est inutile, elle décide de le supprimer. Elle a besoin ensuite d'accéder à un fichier chiffré pour une raison quelconque mais elle ne peut plus le lire.
Elle contacte alors l'administrateur pour lui demander de régler ce problème.
L'administrateur ouvre alors une session Agent Recouvrement et déchiffre le fichier sans problème. Cependant, Alice n'aura plus jamais accès à ce fichier dans sa propre session.

5.3. Synthèse

Ce scénario peut se résumer en 3 étapes :

- Chiffrement du fichier par Alice → Création du 1er champs de déchiffrement (*DDF*) dans le fichier chiffré et création du champs de recouvrement. Il s'agit typiquement du schéma de la section 4.1. :

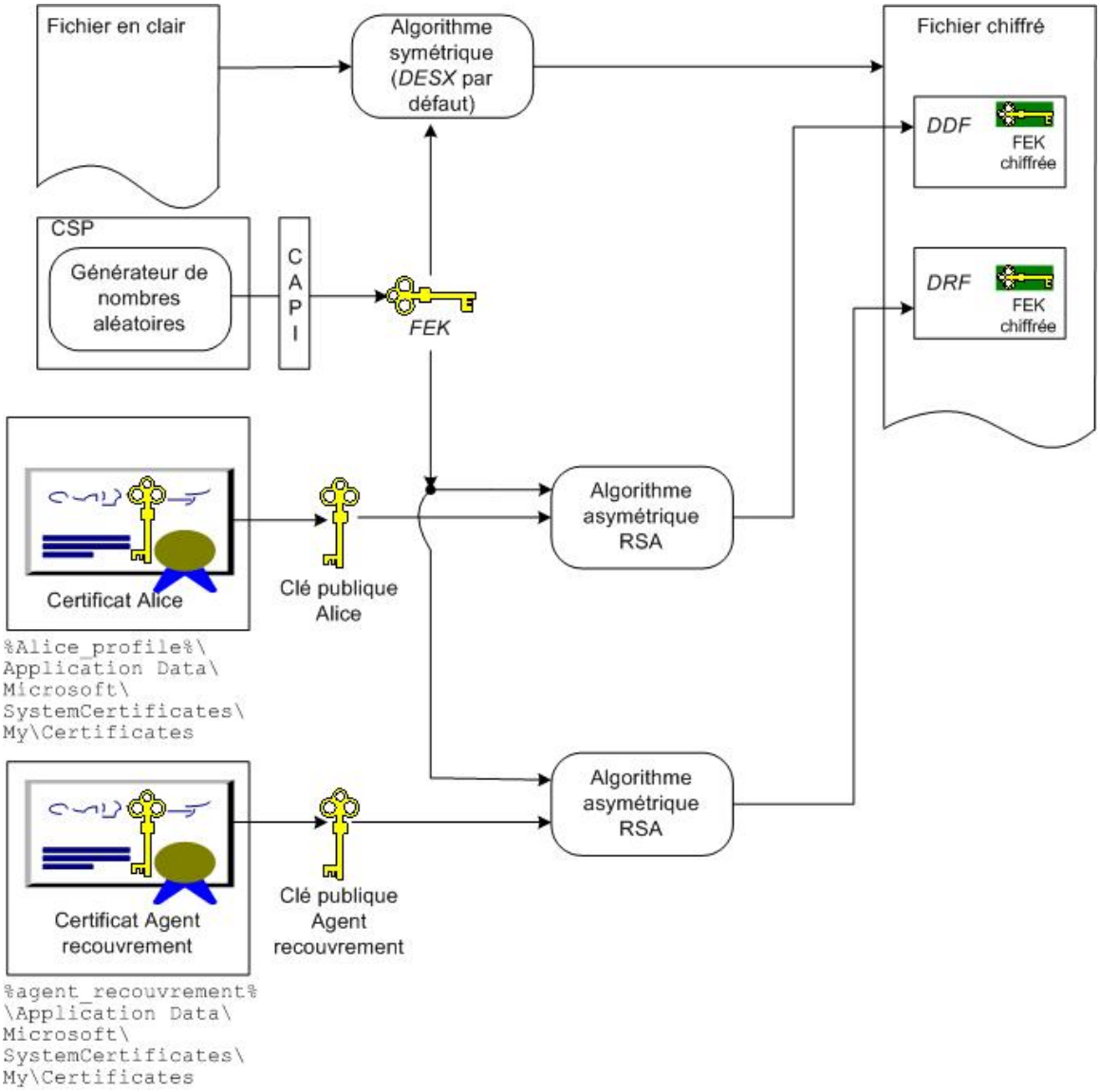


Fig. 28 : 1ère étape (chiffrement par Alice)

- Ajout de Bob à la liste des utilisateurs autorisés par Alice à lire le fichier chiffré :

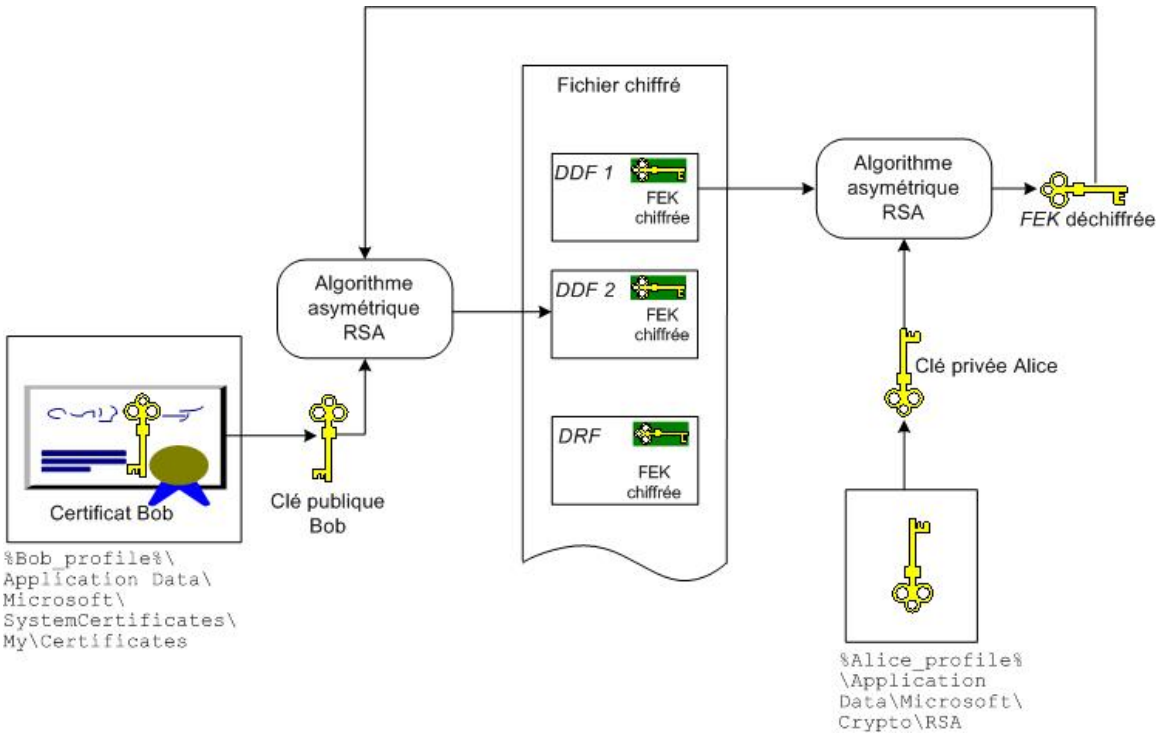


Fig. 29 : Alice authorise Bob à déchiffrer le fichier

La FEK, déchiffrée par la clé privée d'Alice, est à nouveau chiffrée par la clé publique de Bob et stockée dans un 2ème champ de recouvrement des données (*DDF*).

- Le fichier peut ainsi être déchiffré par Alice, Bob ou Agent de recouvrement :

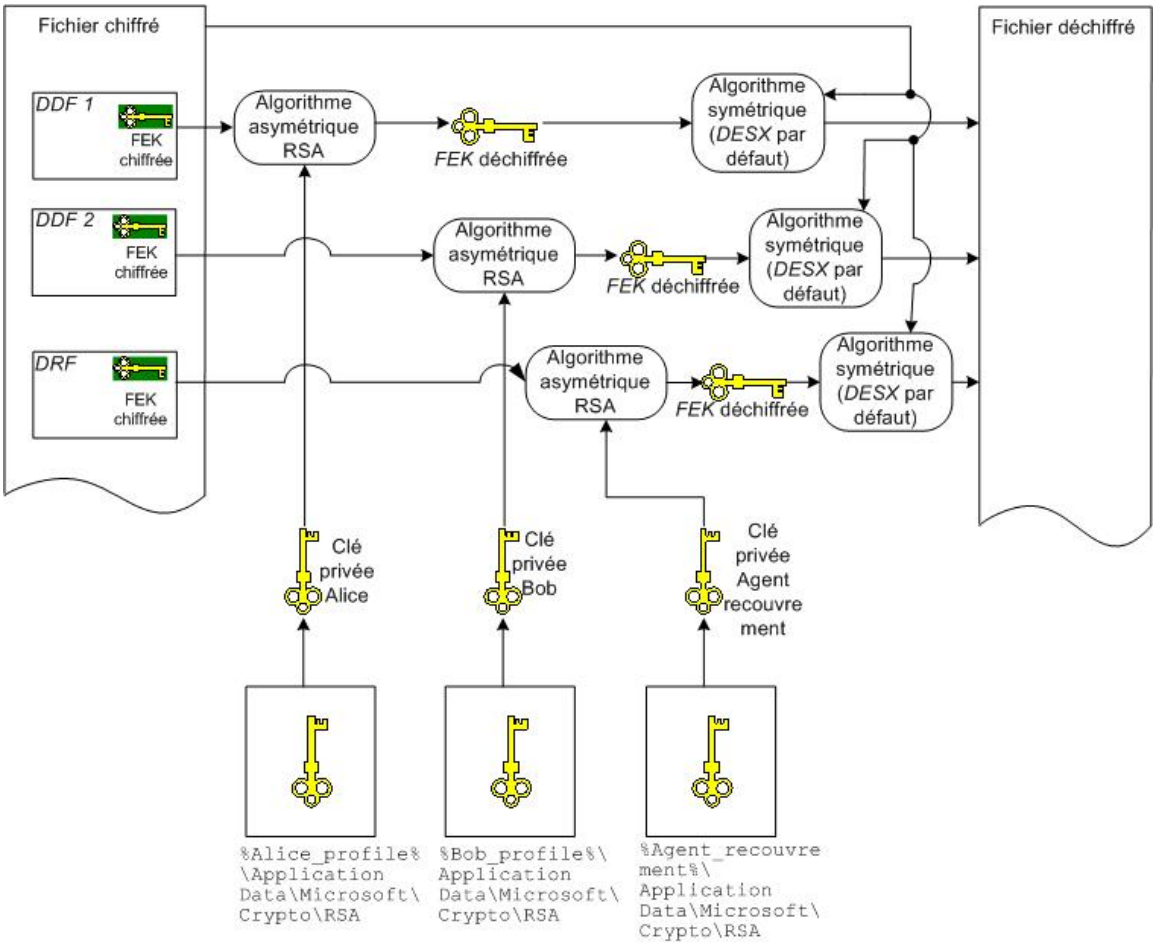


Fig. 30 : Alice, Bob ou Agent Recouvrement peuvent déchiffrer

6. *EFS* est-il sûr ?

6.1. Niveaux de chiffrement des clés

Quand un fichier chiffré est sauvegardé, Windows XP Professionnel fournit automatiquement 4 niveaux de chiffrement et un 5ème : une clé de démarrage configurable.

1. Le CSP fournit la FEK, qui chiffre les données du fichier.
2. EFS utilise la clé publique contenue dans le certificat EFS de l'utilisateur pour chiffrer la FEK. Ce certificat est stocké dans le magasin de certificats de l'ordinateur. La clé privée correspondante est stockée chiffrée dans le répertoire RSA du compte utilisateur.
3. L'API de protection des données génère la clé maître de l'utilisateur afin de chiffrer les clés privées.
4. L'API de protection des données génère une clé symétrique de chiffrement dérivée du hash du mot de passe afin de chiffrer la clé maître.
5. Une clé de démarrage (*syskey*) peut être utilisée pour protéger toutes les clés maître. Au démarrage, *syskey* chiffre toutes les clés privées de l'utilisateur, dont la clé privée *EFS*.

Voici les étapes du processus de déchiffrement de la clé privée :

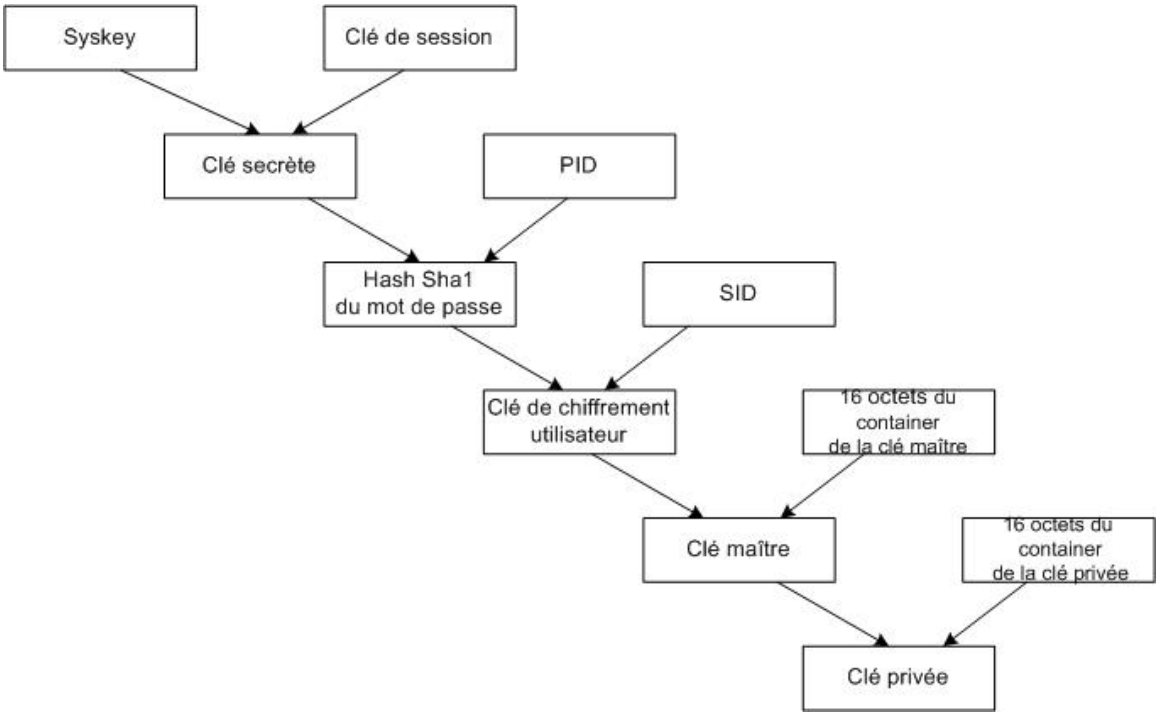


Fig. 31 : Déchiffrement de la clé privée

Source : http://www.blackhat.com/presentations/bh-europe-03/bh-europe-03-malyshev.pdf

La clé de démarrage *SYSKEY* est utilisée pour protéger la *SAM* (*System Account Manager)*. Voir section 6.3 pour explications sur la *SAM*.

6.2. Renforcement du chiffrement

Il est possible de changer l'algorithme de chiffrement de la *FEK* *DESX* en un plus robuste comme *3DES* ou *AES*. Il suffit pour cela d'aller dans la base de registre :

- Autoriser le chiffrement *3DES* ou *AES* :
  `HKLM\SYSTEM\CurrentControlSet\Control\LSA\` puis créer l'entrée `FipsAlgorithmPolicy` avec la valeur 0x1

- Choix de 3DES ou AES :

  `HKLM\Software\Microsoft\Windows NT\CurrentVersion \EFS\` puis créer l'entrée `AlgorithmID` avec la valeur

  0x6603 pour *3DES* ou 0x6610 pour *AES_256*

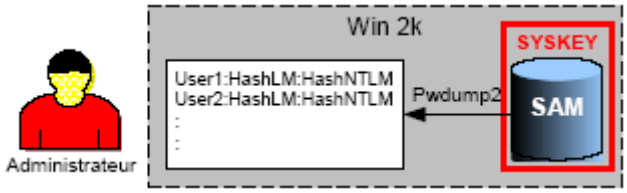
La puissance de ces 2 algorithmes nécessitent une description qui sort du sujet étudié. Néanmoins, l'excellent lien http://www.securiteinfo.com/crypto/aes.shtml fournit des explications très détaillées.

6.3. Attaque sur EFS

Etant donnés les différents niveaux de chiffrement et la possibilité d'utiliser des algorithmes symétriques plus robustes, il est probable que la seule réelle attaque possible sur *EFS* est de cracker le mot de passe utilisateur.

Les clés privées *EFS* étant stockées dans le profil utilisateur, n'importe qui ayant cracké le mot de passe utilisateur pourrait ouvrir une session et ainsi avoir accès aux données sensibles.

La *SAM* (*Security Accounts Manager*) est l'élément qui contient les noms des utilisateurs et les *hashs* des *passwords* de tous les utilisateurs d'un système local (comptes locaux). Ces informations sont stockées dans le fichier `%systemroot%\system32\config\SAM`. Ce fichier représente l'entrepôt de stockage physique des données spécifiées dans la base de registre `HKEY_LOCAL_MACHINE\SAM`. Il est verrouillé par le système d'exploitation, donc n'est pas lisible ni par les utilisateurs et ni par l'administrateur. Il est toutefois récupérable en *bootant* le machine à l'aide d'un autre OS compatible avec le système de fichier (Disquette Dos + NTFSDOS).
SYSKEY ajoute un niveau supplémentaire d'encryptions aux *hashs* des *passwords* stockés dans la *SAM*.
L'outil *pwdump2* est capable de contourner *SYSKEY*



Pwdump2 exploite l'injection de DLL pour charger son propre code dans l'espace de traitement d'un processus doté de droits d'accès élevés. Le processus visé est lsass.exe, c'est à dire le sous-système local de sécurité. Une fois chargé dans le processus, le code pirate peut effectuer un appel API interne accédant aux *hashs* des *passwords* cryptés par SYSKEY, sans avoir à les décrypter.
Pwdump2 doit être lancé dans l'espace de traitement du système cible (localement). Des droits d'accès Administrateur sont nécessaires.

Fig. 32 : « Authentification Windows 2000 » de Philippe Logean 17 Avril 2003

7. Les dossiers *Web*

7.1. Introduction

Un dossier *Web* fournit une interface pour organiser les fichiers sur plusieurs types de serveurs http distants et notamment sur les serveurs *web* 2003 (*IIS 6.0*) de *Microsoft*.
Le but des dossiers Web est de pouvoir organiser les répertoires et fichiers présents sur le serveur *web* à distance. Egalement de transmettre un fichier chiffré *EFS* entre client et serveur, par le réseau, sans qu'il passe en clair. Le problème du partage de fichier classique est que les fichiers chiffrés sont déchiffrés automatiquement avant d'être transmis (*SMB*).

7.2. Services fournis

Un client, depuis son *Windows Explorer*, peut créer un fichier ou un dossier dans un répertoire du serveur dont l'option Partage Web est activée. C'est comparable au partage de fichiers entre un client et un serveur de fichiers. La différence entre ces 2 procédés est que le partage de fichiers s'effectue avec le protocole *SMB*, alors que le partage de dossiers Web s'effectue avec le protocole *http*.

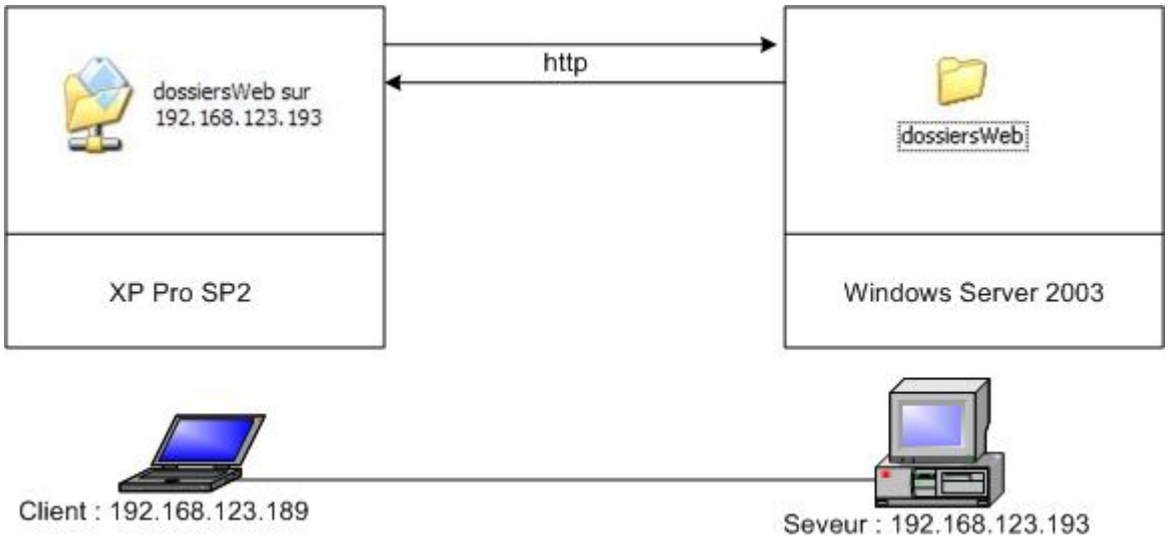


Fig. 33 : Dossier Web partagé Client-Serveur Web

L'administrateur du serveur Web attribue les permissions de lecture, écriture, exécution dans les propriétés de partage du dossier Web.

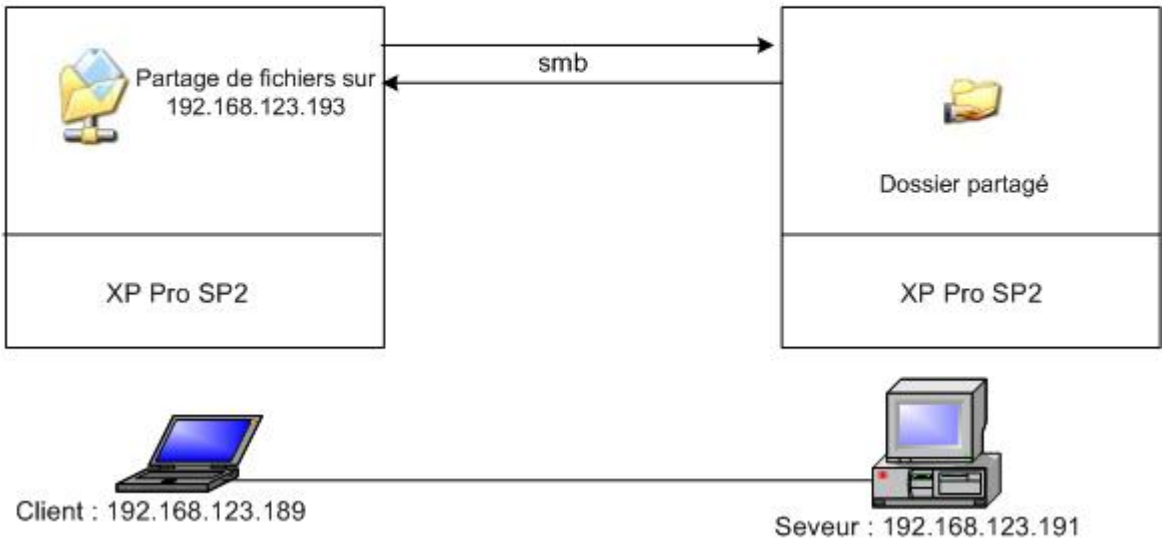


Fig. 34 : Dossier partagé Client-Serveur de fichiers

La machine de droite (fig. 34) a un rôle de serveur de fichiers et non de serveur Web, elle ne peut donc pas effectuer un partage de dossiers Web. La grande différence est le protocole d'échange : SMB pour partage de fichiers, http pour partage de dossiers Web.

Un dossier Web est ainsi accessible en lecture avec Internet Explorer. C'est une page Web normale :

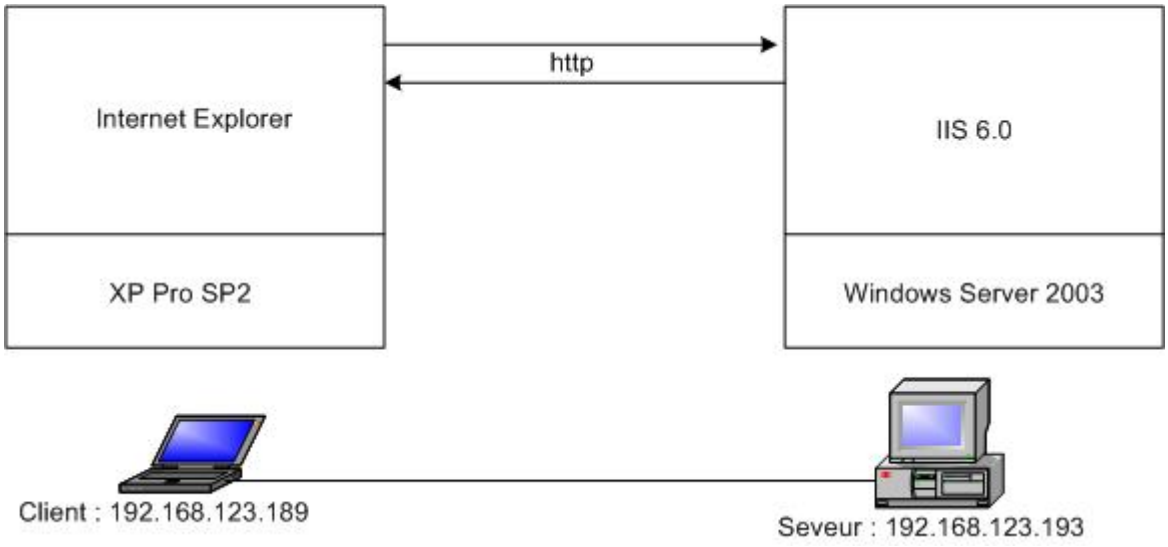


FIg. 35 : Lecture d'un dossier Web depuis IE

7.3. *webDAV*

7.3.1. Rappel du protocole *http*

Le protocole *http* est un protocole de niveau applicatif. Il est utilisé pour la navigation sur Internet avec le port TCP 80.
Il consiste en un message « *request* » envoyé par un client vers un serveur et d'un message « *reply* » qui est la réponse du serveur. Il y a 3 éléments importants dans un *http request* : la méthode, l'*URI* et l'en-tête. La méthode décrit le type de requête (p.ex. *GET*). L'*URI* permet d'identifier la ressource sur laquelle la méthode doit opérer. L'en-tête fournit des informations supplémentaires sur la requête.

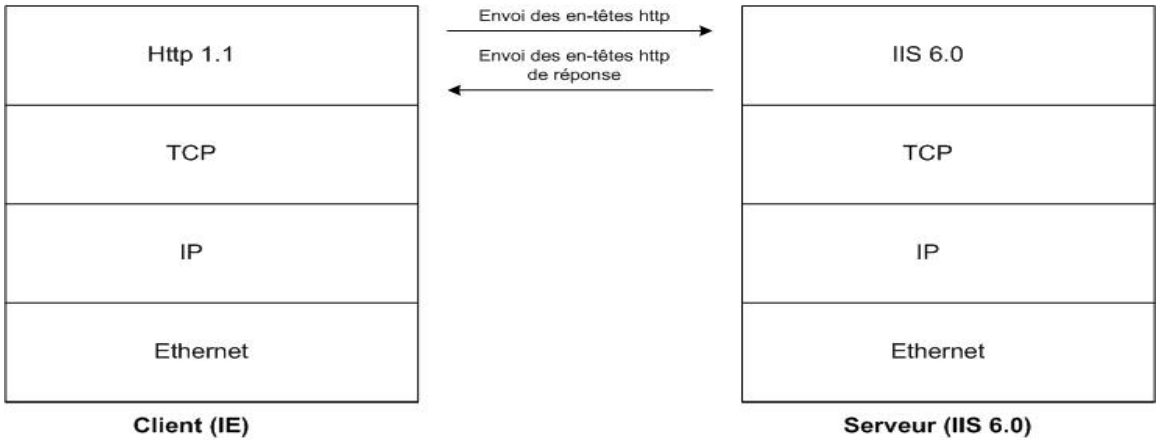


Fig. 36 : http, protocole de niveau applicatif

7.3.2. Nouvelles méthodes *http*

Pendant une navigation sur le *Web*, les seules méthodes *http* utilisées sont les *GET,* pour obtenir un document, et les *POST*, pour soumettre des formulaires vers un serveur. La spécification *webDAV* (*RFC 2518*) décrit un jeu de nouvelles méthodes autorisant un client à publier des documents, et à effectuer toutes sortes de manipulations sur certains répertoires du serveur distant.
Il y a 3 groupes de méthodes :

1. *PROPFIND* et *PROPPATCH*; pour demander et manipuler des propriétés.
2. *LOCK* et *UNLOCK*; pour les verrouillages et déverrouillages de données.
3. *MOVE, COPY* et *MKCOL*; pour les manipulations de base de répertoires.

Les dossiers Web utilisent une interface de programmation appelée *OLE DB* qui permet de transformer une action sur un dossier Web en une requête *http* vers un serveur *Web*. Quand un dossier Web contacte un serveur la première fois, il passe par un processus de découverte pour savoir quel protocole utiliser.

Il envoie une requête *HTTP OPTIONS* standard dans le mécanisme de découverte *http*. Si la réponse contient la valeur *DAV* dans l'en-tête, le protocole *webDAV* (*Web Distributed Authoring and Versioning*) est utilisé.

WebDAV introduit le concept de collection de ressources http. Une collection est l'équivalent d'un répertoire dans un système de fichiers traditionnel. Les collections sont créées avec la méthode *MKCOL*, qui est similaire à la commande *DOS* `mkdir`.

WebDAV permet de définir les propriétés d'une ressource.
La méthode *PROPFIND* est utilisée pour l'ouverture d'un document. Voir section 7.2.2. pour l'observation pratique de ces méthodes.

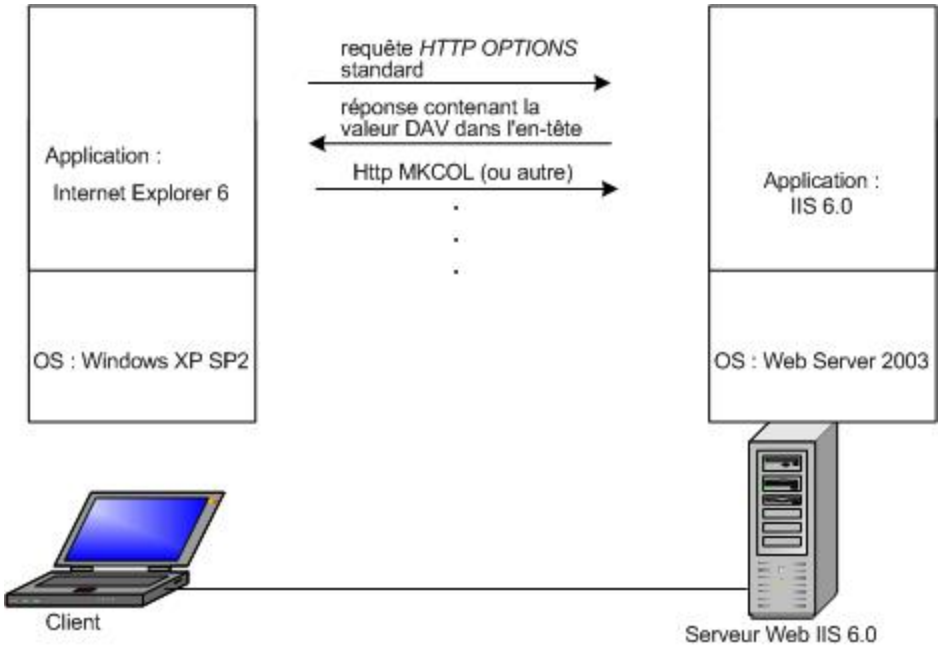


Fig. 37 : Découverte de *webDAV*

.

7.4. Mise en œuvre des dossiers Web

La configuration est la suivante :

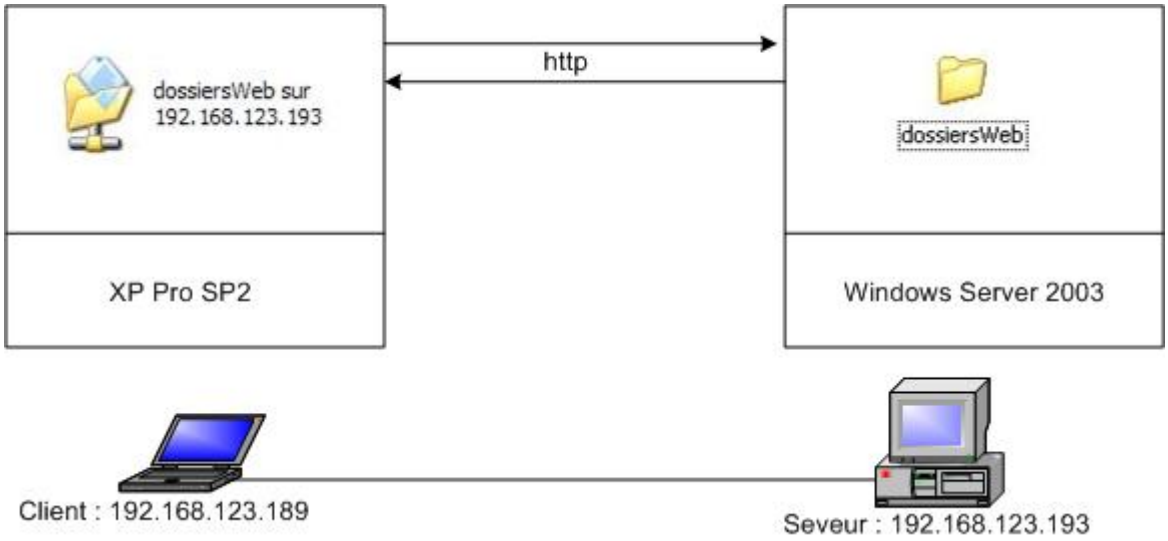


Fig. 38 : Configuration réseau lors de la mise en œuvre des dossiers Web

Sur le serveur, le dossier `dossiersWeb` se trouve dans `C:\Inetpub\wwwroot`
Sur le client, connexion au serveur dans `Favoris Réseau.`

Voir annexe section 6 pour la création pratique de dossiers Web.

7.4.1. Test 1

Le client accède au dossier Web par son explorateur *Windows*

```
     16 0.057358    192.168.123.189       192.168.123.193       HTTP
PROPFIND /dossiersWeb HTTP/1.1
     17 0.064848    192.168.123.193       192.168.123.189       HTTP
HTTP/1.1 207 Multi-Status
     18 0.065210    192.168.123.193       192.168.123.189       HTTP
Continuation
```

Fig. 39 : Observation Ethereal des paquets 16, 17, 18 du test 1 (par le client)

On constate que la requête http utilise la méthode PROPFIND de webDAV décrite à la section 7.3.2.

7.4.2. Test 2

Le client voudrait maintenant créer un document .txt dans le dossier Web.
Voici ce qu'on observe côté client :

```
     21 0.024860    192.168.123.189       192.168.123.193       HTTP
PROPFIND /dossiersWeb HTTP/1.1, NTLMSSP_AUTH
     22 0.035315    192.168.123.193       192.168.123.189       HTTP
HTTP/1.1 207 Multi-Status
     23 0.035682    192.168.123.193       192.168.123.189       HTTP
Continuation
     24 0.035819    192.168.123.189       192.168.123.193       TCP
1161 > http [ACK] Seq=908 Ack=5241 Win=64547 [CHECKSUM INCORRECT]
Len=0
     25 0.166491    192.168.123.189       192.168.123.193       HTTP
PUT /dossiersWeb/Nouveau%20Document%20texte.txt HTTP/1.1
     26 0.170530    192.168.123.193       192.168.123.189       HTTP
HTTP/1.1 201 Created
     31 0.176938    192.168.123.189       192.168.123.193       HTTP
PROPPATCH /dossiersWeb/Nouveau%20Document%20texte.txt HTTP/1.1,
NTLMSSP_AUTH (text/xml)
     32 0.238872    192.168.123.193       192.168.123.189       HTTP
HTTP/1.1 207 Multi-Status
     33 0.239235    192.168.123.193       192.168.123.189       HTTP
Continuation
```

Fig. 40 : Observation des paquets 21 à 25 et 31 à 33 du test 2 (Création dans dossiersWeb) par le client

On peut voir la méthode *PROPFIND* paquet 21 et la méthode *PROPPATCH* paquet 31 demandant la création d'un document .txt.

7.5. Chiffrement de fichier dans dossier Web côté serveur

Les utilisateurs peuvent chiffrer et déchiffrer des fichiers stockés sur un serveur de fichiers ou sur un dossier Web de serveur IIS 6.0. Les dossiers Web, comparés aux serveurs de fichiers, offrent l'avantage d'une transmission de données chiffrées entre le client et le serveur, ce qui n'est pas le cas pour les serveurs de fichiers.

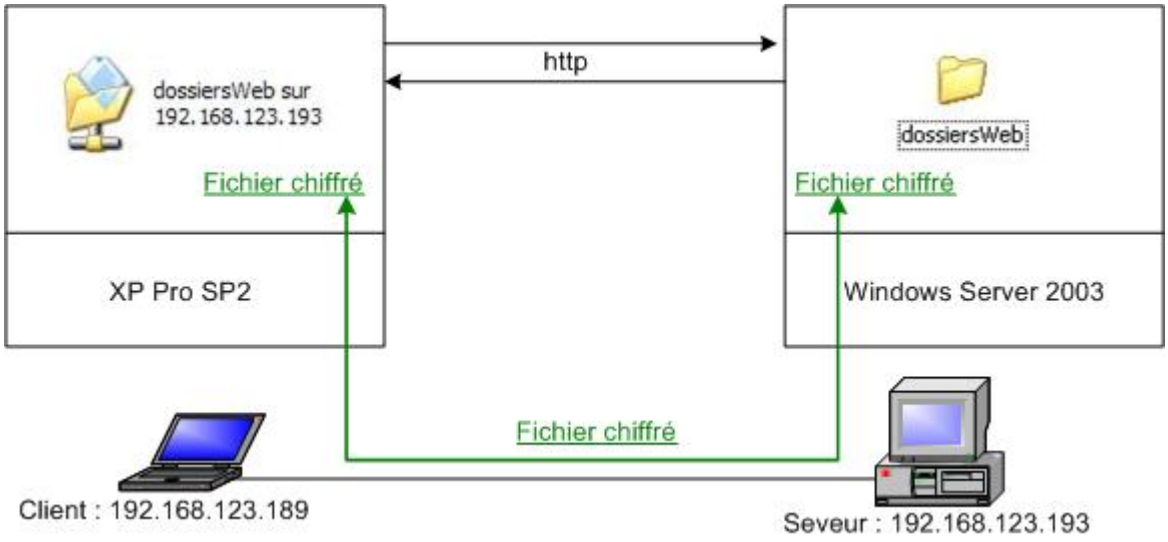


Fig. 41 : *EFS* et dossiers web

La transmission « client - serveur de fichiers » se fait par le protocole *smb*. Le fichier chiffré est déchiffré avant d'être transmis puis rechiffré lors de son arrivée sur le serveur de fichier. La confidentialité n'est pas garantie car possibilité d'attaque *Man-In-The-Middle*.

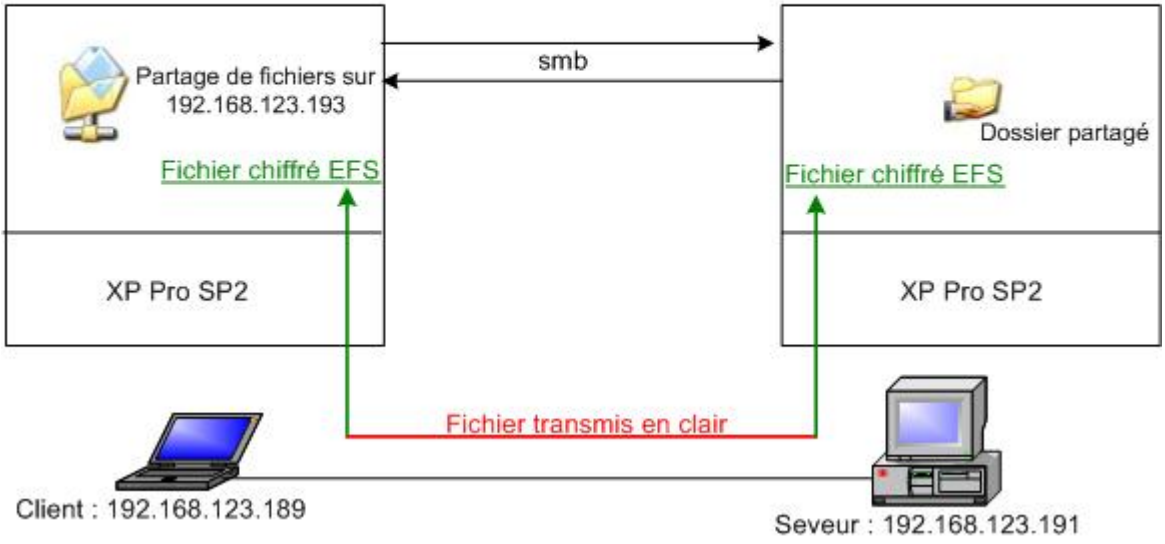


Fig. 42 : EFS et partage de fichiers

Le partage de fichiers chiffrés *EFS* distant exige que client-serveur soient dans un domaine *Windows 2000* minimum. En effet, *EFS* doit simuler l'utilisateur par le biais d'une délégation *Kerberos* afin de chiffrer et déchiffrer les fichiers.

Mon étude est portée sur une machine *stand-alone* faisant partie d'un groupe de travail et non d'un domaine. On verra lors de la mise en œuvre que le chiffrement des dossiers Web ne sera pas possible. Il est tout de même important d'aborder ce sujet car ce type de partage de fichiers chiffrés est une avancée incontestable dans le domaine de la sécurité.

La transmission « Client – Server IIS 6.0 » se fait par le protocole http port 80 avec *webDAV* enclenché. Le fichier chiffré reste chiffré lors de sa transmission. La confidentialité est garantie. Quand un fichier est chiffré côté serveur, il est impossible de l'ouvrir côté client. Après avoir chercher dans beaucoup de directions, j'ai décidé de regarder le journal d'évènements côté serveur et ai constaté le *log* suivant :

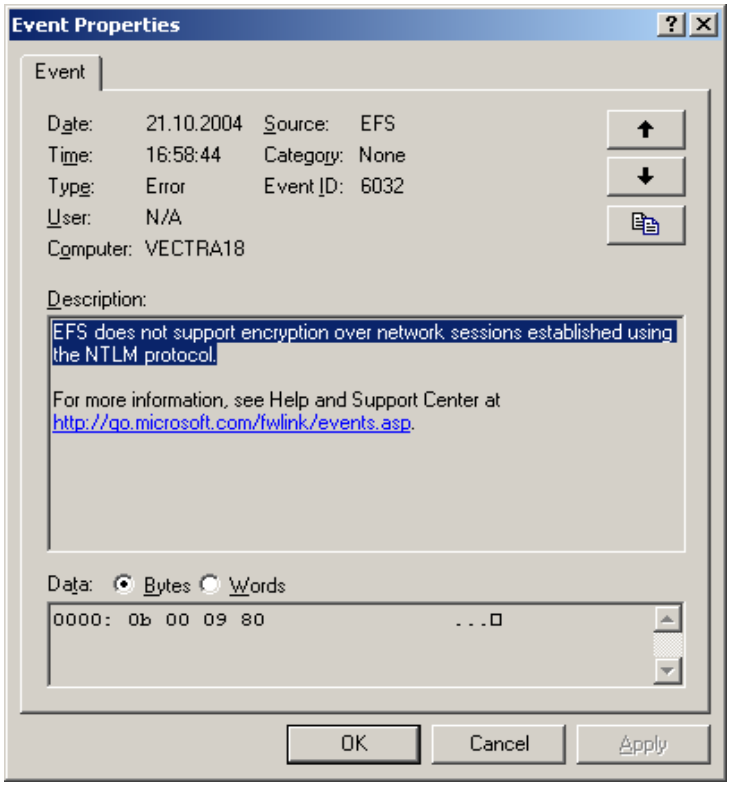


Fig. 43 : Log d'erreur côté serveur

EFS ne supporte pas le chiffrement sur une session établie par le protocole *NTLM*.

Client et serveur doivent faire partie d'un domaine et utiliser le protocole *Kerberos* pour l'authentification.

Voici un extrait du travail de diplôme de Yann Souchon sur l'architecture réseau basée sur *Windows 2000*. Ayant trouvé ses explications sur *Kerberos* et *NTLM* très pertinentes, je me permets de les mettre telles quelles :

> Toute personne qui désire accéder au serveur WEB doit s'**authentifier**.
> L'authentification choisie est ***Integrated Windows authentication***. C'est une méthode d'authentification sécurisée car le nom d'utilisateur et le mot de passe ne sont pas transmis en clair via le réseau. Le mot de passe est transmis en utilisant **une fonction de hachage**.
> Ce type d'authentification peut utiliser **deux protocoles d'authentification** :
>
> - **Kerberos v5** : protocole décrit dans le chapitre § 4.
> - **NTLM** : ancien protocole propriétaire de Microsoft utilisé par les pre-Windows 2000 (→ Annexe 1).

Fig. 44 : Exrait du rapport de Yann Souchon (décembre 2001) sur l'authentification Windows dans un domaine

N.B : Le rapport de Philippe Logean sur l'authentification *Windows 2000* datant du 17 Avril 2003 fournit des informations très détaillées sur *NTLM* et *Kerberos*.

7.6. Conclusion

Les dossiers Web sont une bonne alternative au partage de fichier classique car ils utilisent le protocole *http* avec l'extension *webDAV* plutôt que smb. Non seulement la fonctionnalité de partage de fichiers est conservée, mais il y a possibilité de publier les ressources du dossier Web sur des pages web et surtout de transmettre des fichiers chiffrés *EFS* de façon chiffrée.

La contrainte du chiffrement est de travailler sur une machine membre d'un domaine et non sur une machine *stand alone.*

7.7. Sources

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/e2k3/e2k3/_webdav_web_store_http_webdav_protocol_reference.asp → tout sur le protocole webDAV

http://asg.web.cmu.edu/rfc/rfc2518.html#sec-1 → RFC webDAV

http://www.univ-st-etienne.fr/criter/guide/webdav/ → bien pour la mise en œuvre

Kit de ressources XP p. 763

8. Conclusion

EFS est un système de chiffrement permettant uniquement de chiffrer des données et pas des fichiers système. C'est donc un système conçu pour chiffrer des documents confidentiels critiques en cas de vol de machine.

L'utilisation du protocole *webDAV* pour transporter des fichiers chiffrés entre client et serveur web est une fonctionnalité supplémentaire. Le problème est que client et serveur doivent faire partie d'un domaine et mes tests se sont portés sur une machine *stand alone*, il n'a donc pas été possible de réellement testé le mécanisme de chiffrement à distance.

Avantages :

- Simple d'utilisation
- Chiffrement rapide par clé symétrique
- Clé symétrique sécurisée car chiffrée par clé publique
- Difficile à casser car plusieurs niveaux de chiffrement
- Recouvrement des données en cas de perte de clé privée
- Possibilité d'autoriser plusieurs utilisateurs à lire le fichier chiffré
- Possibilité de chiffrement à distance sans passer par *SMB*

Inconvénients :

- Ne permet pas le chiffrement des fichiers systèmes
- Chiffrement à distance nécessite de faire partie d'un domaine

9. Références *EFS*

Kit de ressources XP : p. 748 pour vue d'ensemble, p. 750 pour les composants, p. 752 pour les mécanismes de chiffrement et déchiffrement, p. 771 pour les certificats EFS, p. 776 pour le déchiffrement multi utilisateurs.

http ://www.laboratoire-microsoft.org/whitepapers/6715/ → Accès au téléchargement du *whitepaper EFS* qui est bien détaillé (document Microsoft)

http ://search.microsoft.com/search/results.aspx?st=b&na=88&View=fr-fr&qu=efs → lien sur les mécanismes de chiffrement, déchiffrement et récupération des données (documentation technique *Windows 2000 advanced server*)

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dncapi/html/msdn_cryptapi.asp → Documentation *Microsoft* sur *CryptoAPI* et *CSP*

http://www.microsoft.com/france/technet/securite/protect_data_efs.mspx → Documentation Microsoft pour la mise en œuvre EFS (Très bien détaillé)

Partie 3 :

Les fichiers hors connexion

(1 semaine d'étude)

1. Introduction

Le partage de fichier entre un client et un serveur de fichiers du même réseau s'effectue par le protocole *SMB*. Ce type de partage est utile pour le client afin qu'il puisse stocker des fichiers sur le serveur. Il peut ainsi travailler à distance sur ces fichiers et les sauvegarder de façon transparente :

Fig. 45 : Partage de fichier entre Client XP Pro et un serveur de fichiers XP Pro

Si le client déconnecte son pc du réseau, il ne peut plus accéder aux fichiers stockés sur le serveur. La fonctionnalité hors connexion de *Windows XP* va permettre au client de travailler sur les fichiers partagés même quand il n'est pas connecté au serveur de fichiers :

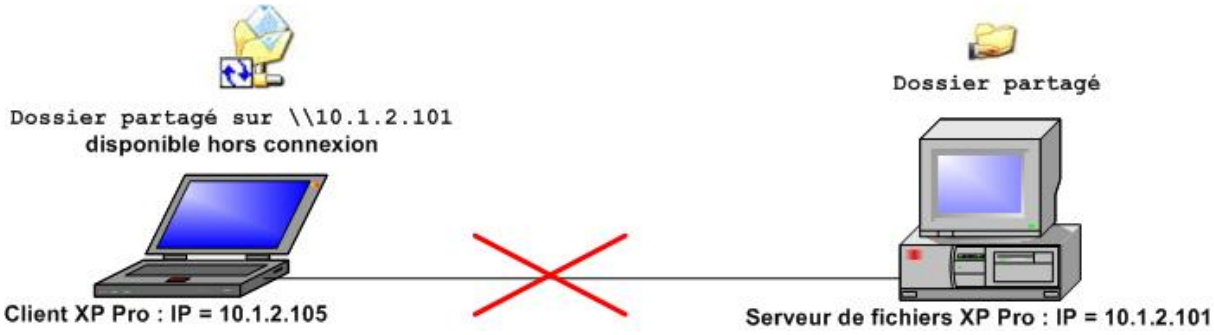


Fig. 46 : Dossier partagé disponible hors connexion par client

Les modifications que le client aura effectuées seront mises à jour automatiquement sur le serveur au moment de la reconnexion au réseau. C'est le mécanisme de synchronisation des fichiers hors connexion :

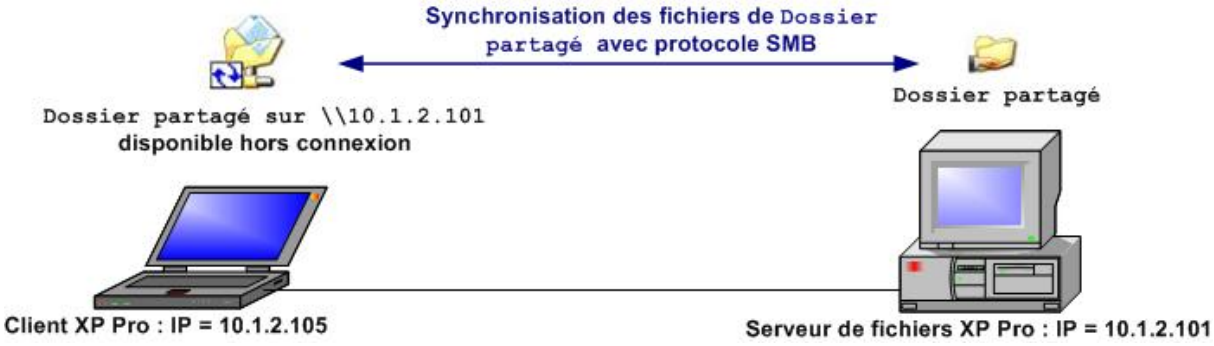


Fig. 47 : Mise à jour automatique des fichiers hors connexion sur le serveur

2. Mécanisme

Le mécanisme de fichiers hors connexion crée un répertoire caché appelé `CSC` *(Client Side Cache)*, localisé dans le répertoire `\WINDOWS\` de la partition d'installation de l'OS. Le répertoire `CSC`,ou « cache client », garde les copies des fichiers ayant été lus sur le serveur de fichier distant.

Quand on veut, depuis la machine cliente, accéder à un fichier présent sur le serveur de fichiers, le mécanisme de fichiers hors connexion regarde d'abord dans le CSC si ce fichier n'est pas déjà présent. Si c'est le cas, le mécanisme demande au serveur l'heure, la date et la taille du fichier. Si ces 3 valeurs correspondent à celles caractérisant la copie du fichier dans le cache client alors le fichier est délivré depuis le cache, ce qui économise de la bande passante réseau. Si la connexion réseau ne fonctionne pas, il est toujours possible de travailler sur les fichiers disponibles hors connexion puisqu'ils sont dans le cache.

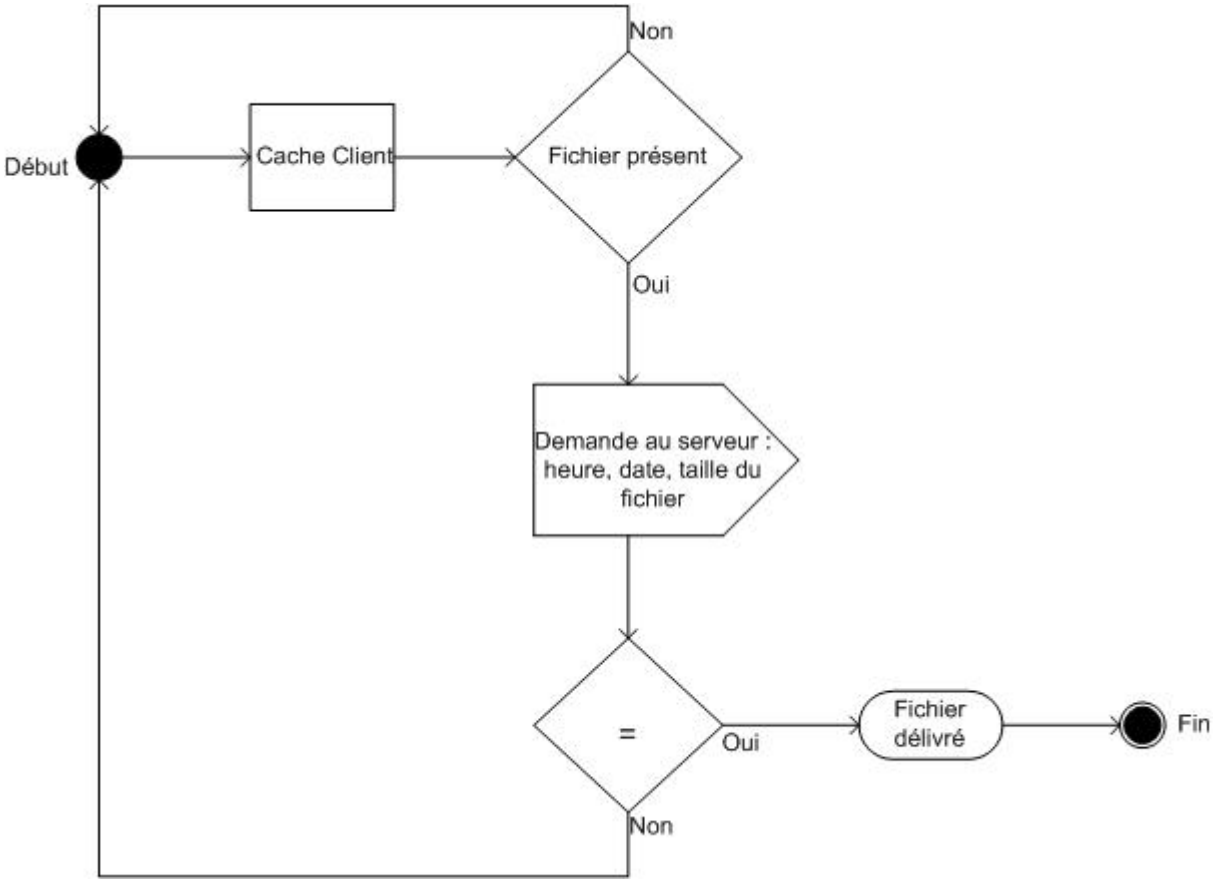


Fig. 48 : Mécanisme hors connexion

3. Mise en œuvre

Les manipulations qui vont être décrites supposent que le modèle client – Serveur de fichiers est déjà mis en place, c'est-à-dire que l'on a déjà la configuration décrite à la figure 45.

3.1. Côté serveur

Pour activer la fonction de mise `hors connexion` coté serveur, il suffit d'activer le cache du dossier partagé afin de le rendre disponible hors connexion (Propriétés→ Partage → Mise en cache). 3 modes de mise en cache sont alors disponibles :

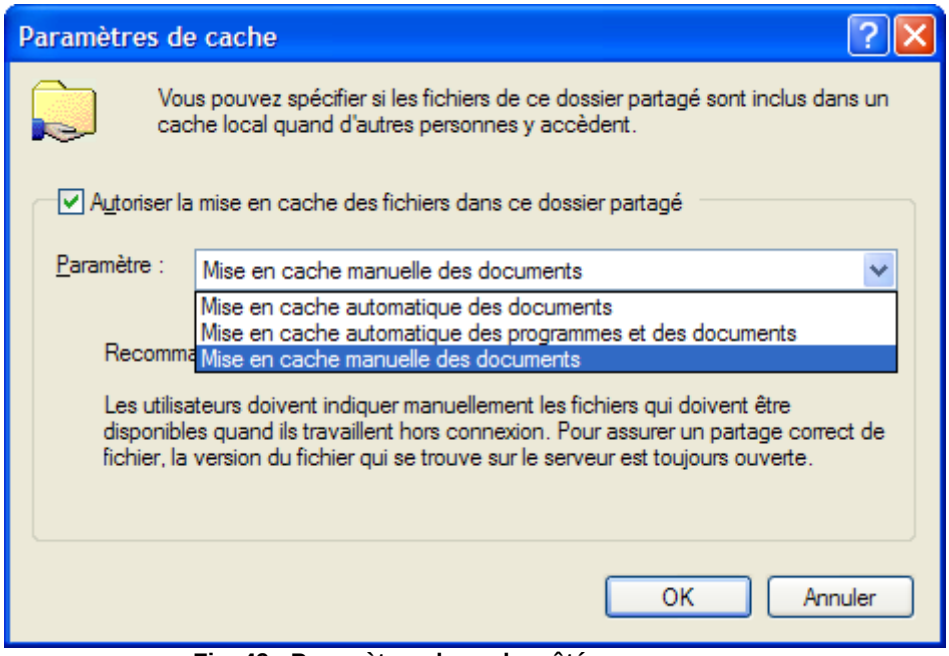


Fig. 49 : Paramètres de cache côté serveur

- `Mise en cache manuelle des documents` : réglage par défaut. Les utilisateurs doivent spécifier quels documents ils souhaitent rendre disponibles hors connexion.
- `Mise en cache automatique des documents` : tous les fichiers ouverts par un utilisateur sont mis en cache sur son disque dur pour une utilisation hors connexion – les versions anciennes du document sur le disque sont automatiquement remplacées par des versions plus récentes du partage quand elles existent.
- `Mise en cache automatique des programmes et des documents`: Ce paramétrage est recommandé pour les dossiers hors connexion ne contenant que des fichiers en lecture seule, ou pour des applications configures pour n'être lancées que depuis le réseau. Cette option n'est pas désignée pour le partage de fichiers. En effet, les anciennes copies de fichiers sont automatiquement supprimées pour laisser la place aux copies de fichiers ayant été accédés plus récemment par le client.

3.2. Côté client

L'activation de la fonction hors connexion, chez le client, s'effectue dans les `options des dossiers` sous l'onglet `fichiers hors connexion` :

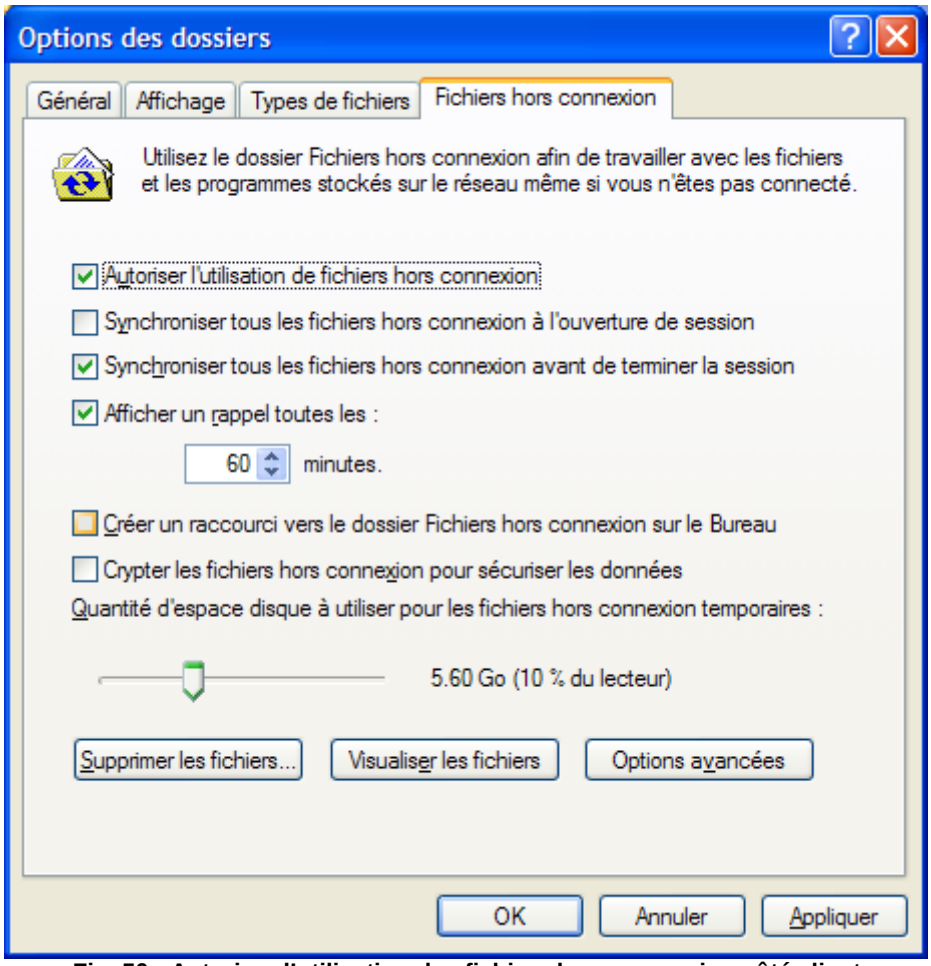


Fig. 50 : Autoriser l'utilisation des fichiers hors connexion côté client

On constate sur la figure 5 plusieurs options à cocher :

- Il faut avant tout cocher la case `Autoriser l'utilisation des fichiers hors connexion`
- Viennent ensuite les options pour synchroniser : Par exemple,si on a travaillé chez soit avec son laptop et que l'on revient le lendemain au labo, il est recommandé de cocher `synchroniser à l'ouverture de session` afin de tout de suite mettre à jour les fichiers sur le serveur. Le choix dépend donc du type d'utilisation des fichiers hors connexion.
- Possibilité de chiffrer les `fichiers hors connexion` avec *EFS*. Voir la partie *EFS* de ce travail de diplôme pour l'explication détaillée de ce type de chiffrement. On notera que lorsque le client et le serveur sont connectés, les fichiers chiffrés ne le sont plus lors du partage avec le protocole *SMB*. En effet, ils passent en clair sur le réseau. Il y a donc danger pour la confidentialité car une personne mal intentionnée peut sniffer le réseau et récupérer les informations. L'intérêt est uniquement de protéger les données hors connexion en cas de vol de la machine.

- Possibilité également de régler la quantité d'espace disque allouée au cache des `fichiers hors connexion`. 10% du lecteur par défaut.
- Les options avancées permettent de configurer la manière dont l'ordinateur fonctionne en cas de perte de connexion avec le serveur. On peut soit démarrer le travail hors connexion, soit l'interdire.

La dernière étape est de cliquer droit sur l'icône de connexion à un lecteur réseau correspondant au serveur et activer `synchroniser tous les fichiers hors connexion`.

4. Synchronisation

L'utilitaire de Synchronisation permet de spécifier les fichiers qui seront synchronisés et le moment où cette synchronisation est effectuée (lors d'une connexion, d'une déconnexion, lorsque l'ordinateur est en veille,…).

La synchronisation s'effectue par le protocole *SMB* (Voir fig. 47) :

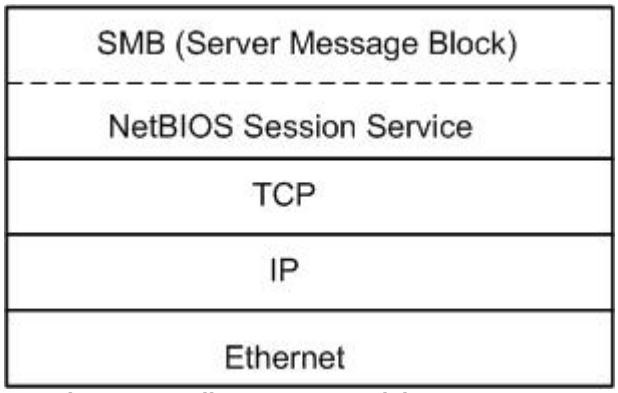


Fig. 51 : Empilement protocolaire pour *SMB*

La capture *Ethereal* de la synchronisation (effectuée figure 47) montre 357 paquets échangés, ce qui n'est pas très parlant. L'analyse détaillée de cette capture m'a permis d'en extraire l'essentiel dans la figure 51.

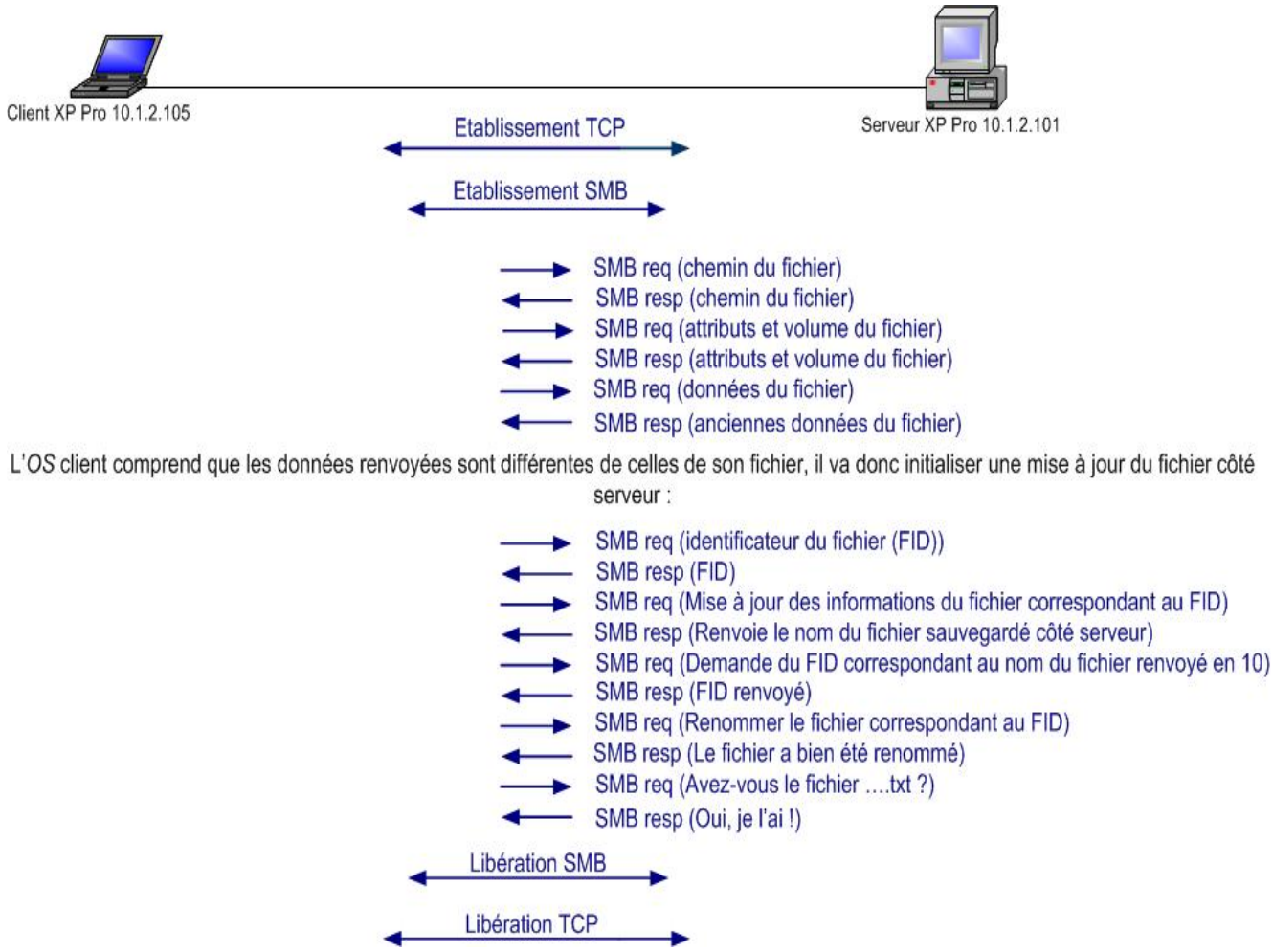


Fig. 52 : Synchronisation des fichiers hors connexion avec détail de la mise à jour du fichier (échanges SMB)

On notera que le protocole *SMB (Server Message Block)* renommé *CIFS* (*Common Internet File System*), est un protocole de niveau applicatif orienté connexion. Il est donc situé au dessus de TCP et comporte 3 phases : Etablissement, Transfert, Libération.

La procédure de mise à jour du fichier côté serveur est effectuée pendant le transfert, la synchronisation est l'ensemble des 3 phases.

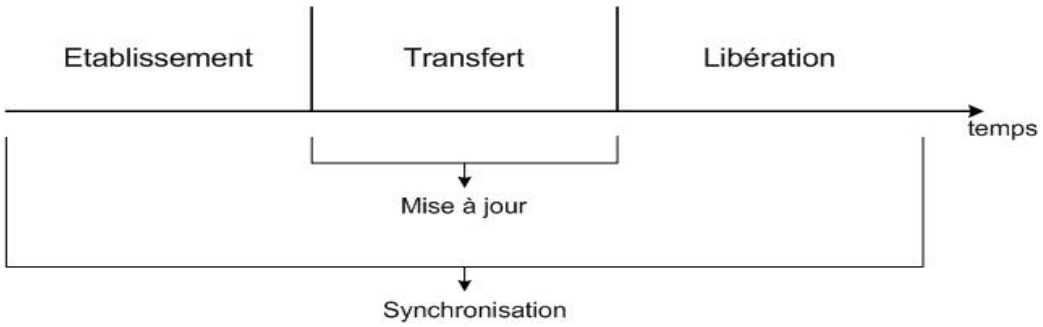


Fig. 53 : Relation entre phases SMB et synchronisation des fichiers hors connexion

La conversation *SMB* commence après l'établissement de session *TCP*. Le client, premier à parler, envoie des requêtes auxquelles le serveur répond.

Au lieu d'ouvrir une connexion *TCP* par processus, l'*OS* du client multiplexe tout le traffic *SMB* vers le serveur sur une seule connexion (Dans le schéma ci-dessous, T0, T1, … sont les différentes tâches d'un processus). Ceci explique pourquoi une simple mise à jour d'un fichier de seulement 10 octets génère un trafic d'environ 300 segments *TCP*. Parmi les différents processus, il y a celui de synchronisation.

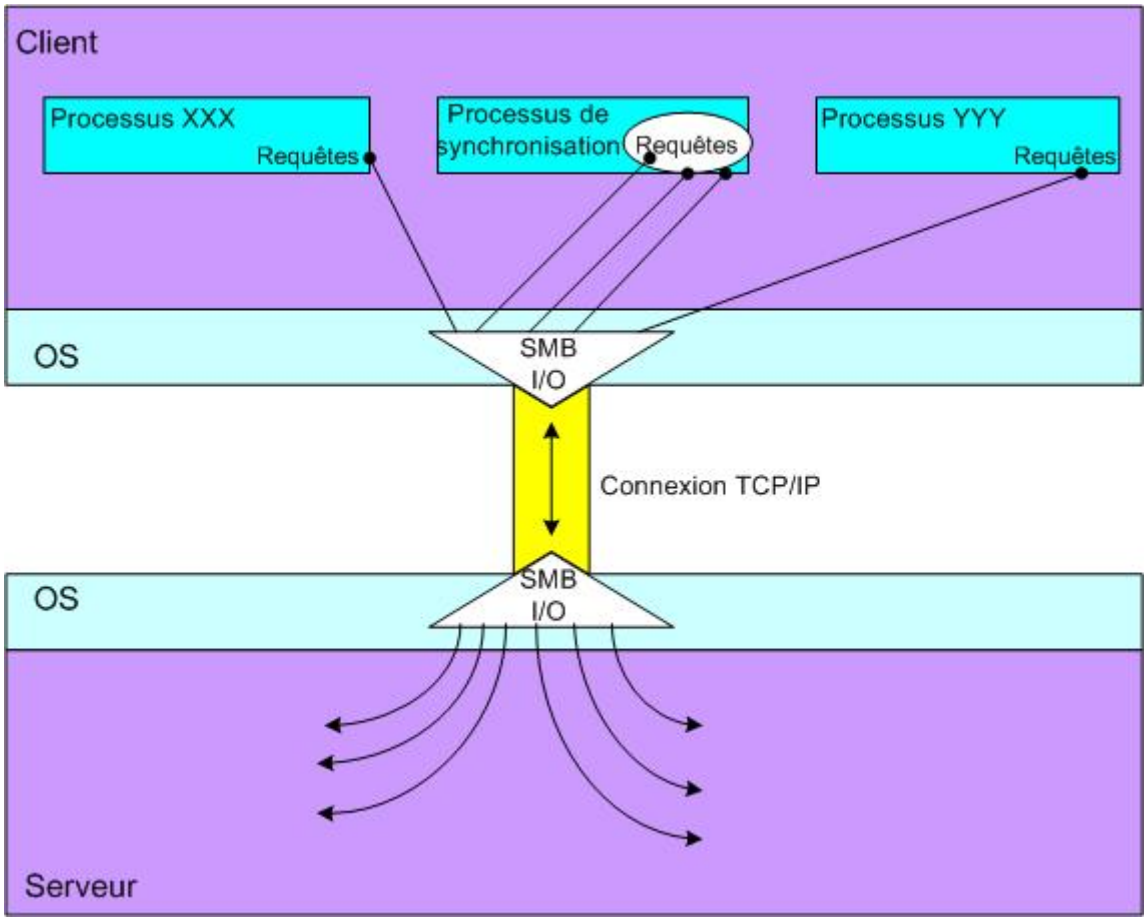


Fig. 54 : Multiplexage du traffic SMB

5. Conclusion

Les fichiers hors connexion sont une option du partage de fichiers avec un serveur de fichiers. Comme ils sont stockés localement chez le client, ce dernier peut travailler dessus comme s'ils étaient sur le serveur.

Avantages :

- Evitent de surcharger le réseau
- Possibilité de travailler sur des fichiers du serveur même quand les réseau n'est pas en service
- Facile à mettre en œuvre
- Possibilité de chiffrement *EFS*

Inconvénients :

- Synchronisation générant beaucoup de trafic (300 segments TCP pour un fichier de 10 octets)

6. Sources

- http://www.ubiqx.org/cifs/ → Excellent lien sur le protocole smb
- http://www.windowsitpro.com/Article/ArticleID/7609/7609.html --> Article pertinent sur le mécanisme hors ligne sous W2K (même principe que sous XP)
- Ressource Kit XP page 252
- http://www.microsoft.com/resources/documentation/Windows/XP/all/reskit/en-us/Default.asp?url=/resources/documentation/Windows/XP/all/reskit/en-us/prde_ffs_rljk.asp?frame=true

Partie 4 :

Mécanisme de restauration système Windows XP

(2 semaines d'étude)

1. Introduction

Ce mécanisme XP permet uniquement à un administrateur et non un simple utilisateur de rétablir l'état du système d'exploitation d'une machine tel qu'il était à une certaine date, sans que les documents créés après cette date ne soient effacés. Par exemple, si Bob, membre du groupe `Administrateurs`, installe un logiciel qui provoque des perturbations du système, il peut restaurer le système à un état antérieur. Avec les anciennes versions de *Windows*, s'il n'avait pas fait de sauvegardes sur *CD* ou clé *USB*, toutes ses données seraient perdues. Avec *Windows XP*, grâce aux points de restauration, ce n'est pas le cas.

Ces points de restauration sont créés automatiquement par le système d'exploitation à chaque évènement important survenant sur la machine comme, par exemple, la suppression ou l'installation d'un programme. Il est également possible de les créer manuellement. Pour cela, Bob dispose de l'interface graphique d'assistance disponible dans :
« `Démarrer→Programmes→Accessoires→Outils Systèmes→Restauration du système` »

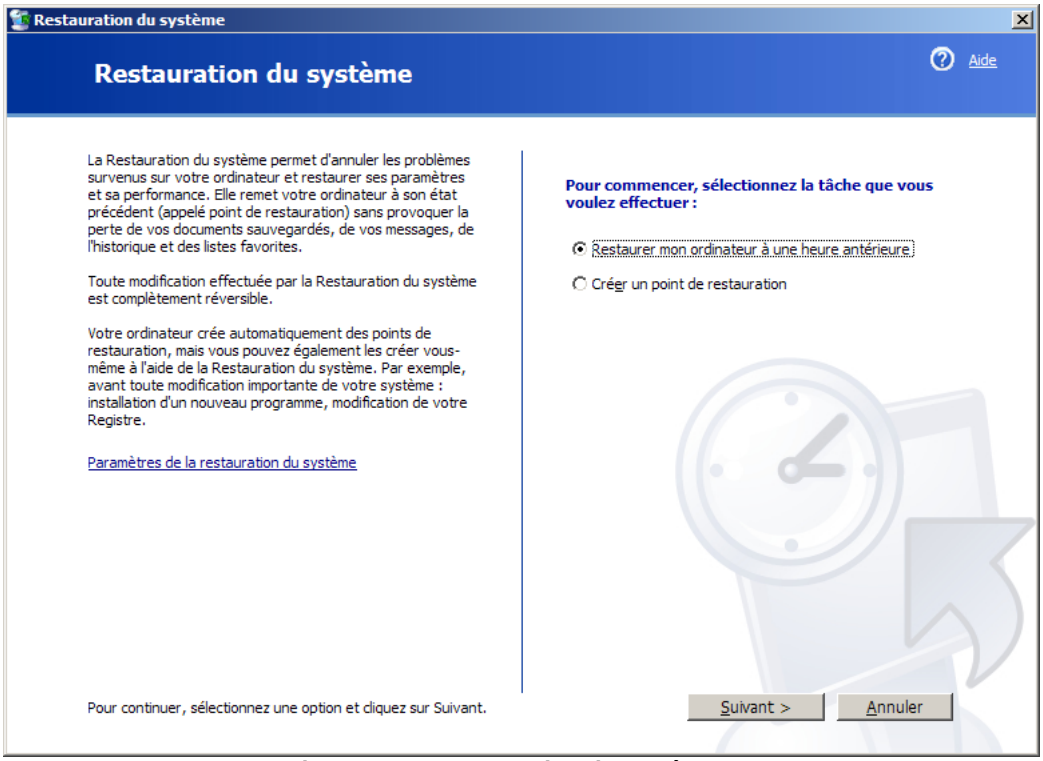


Fig. 55 : GUI Restauration du système

Bob peut, avec cet outil, créer lui-même ses points de restauration ou choisir un des points créé automatiquement chaque 24 heure ou à chaque installation ou suppression d'un programme compatible *Windows XP*.

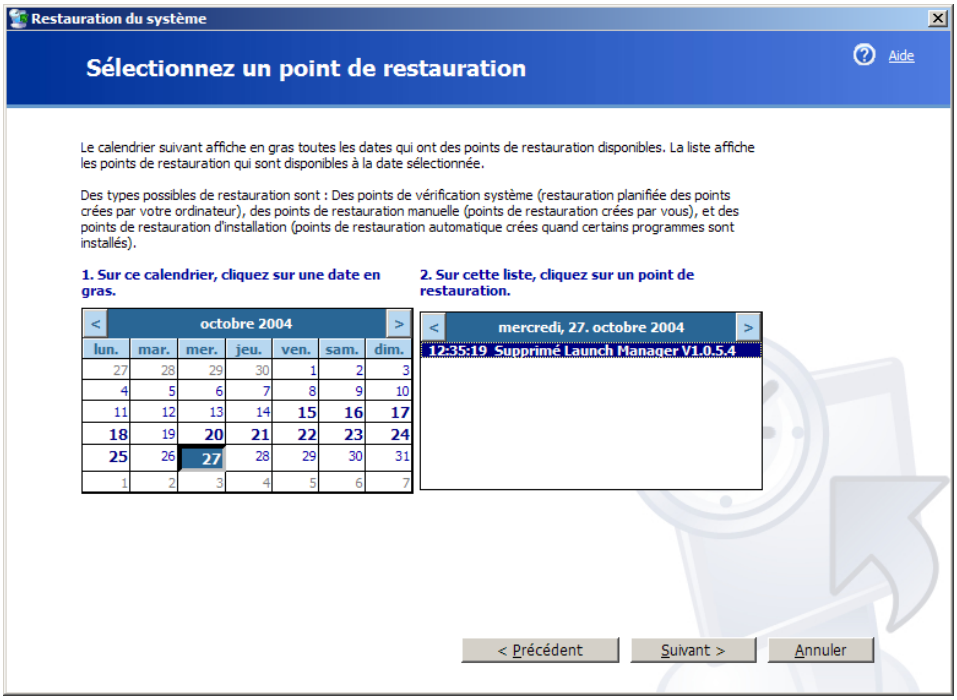


Fig. 56 : Sélection du point de restauration avec GUI

On voit, sur la figure 2, un point de restauration créé le 27 Octobre 2004 à l'occasion de la suppression d'un programme.

2. Mécanismes

2.1. Création automatique de points de restauration

Il y a création automatique de points de restauration quand un des évènements suivant intervient :

- L'installation d'un logiciel ou d'un driver (l'installeur de l'application doit être compatible avec l'API de point de restauration).
- L'installation d'une mise à jour automatique de Windows XP.
- Une restauration système.
- La création manuelle d'un point de restauration.
- L'installation d'un driver dont la signature numérique n'est pas reconnue par Microsoft.

Le processus est le suivant :

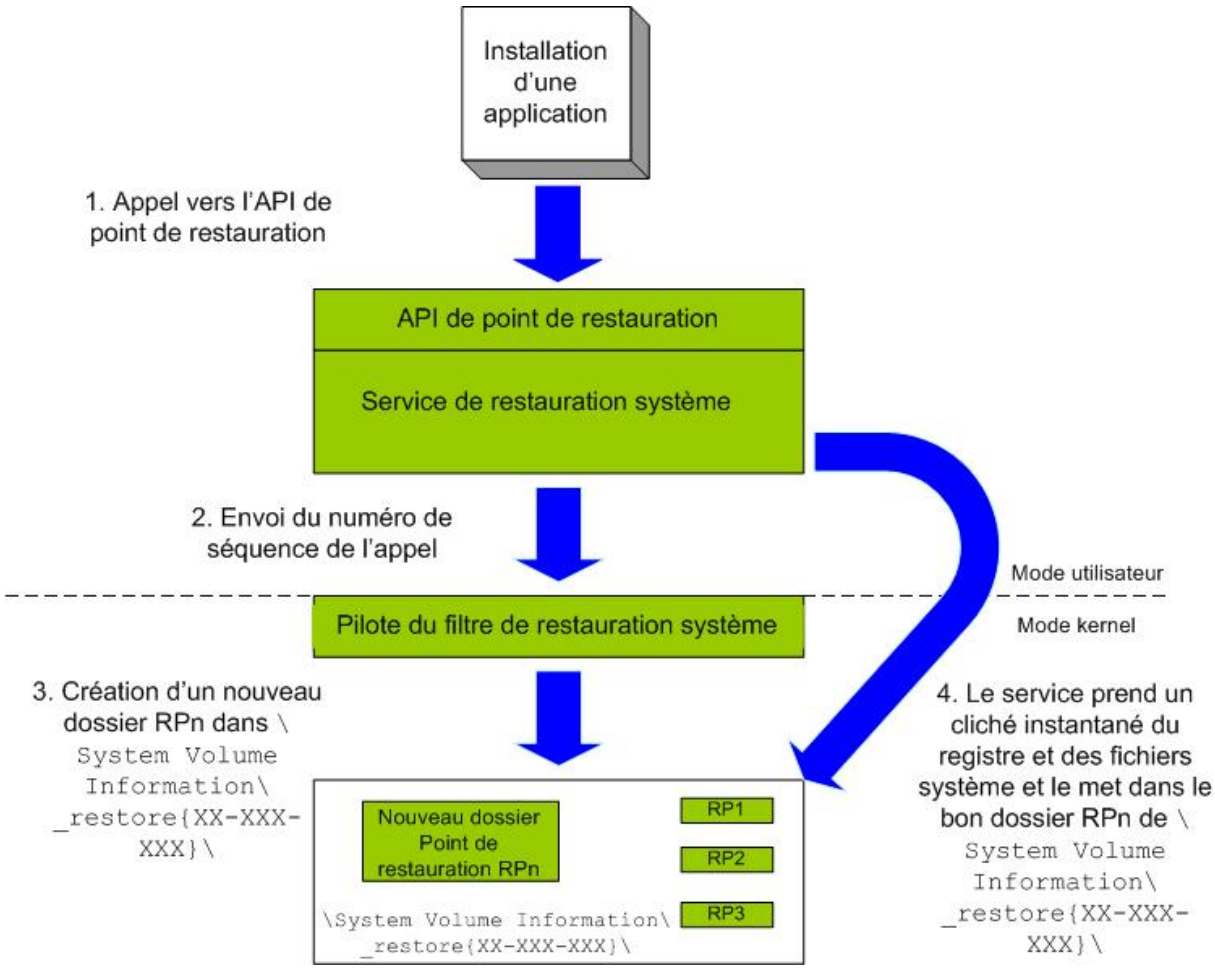


Fig. 57 : Création d'un point de restauration

Prenons comme exemple l'installation de l'application *Microsoft Office* sur ma machine qui a engendré la création du point de restauration numéro 6 :

1. L'installation d'une application ou d'un driver quelconque engendre automatiquement un appel vers l'API de point de restauration. (Ici *Microsoft Office*).

2. Le service de restauration système (`SrService` utilisant la *DLL* `\Windows\System32\Srsvc.dll`)envoie le numéro de séquence de l'appel à son pilote au niveau kernel.

3. Le pilote du service de restauration système (\Windows\System32\Drivers\Sr.sys) crée les dossiers *RP*n dans `\System Volume Information\_restore{`*XX-X-X-X-XXXX*`}`qui est un dossier système caché. n est l'identificateur décimal du point de restauration. {*XX-X-X-X-XXXX*} représente le *GUID* (*Globally Unique Identifier*) de la

machine. (Ici *RP*6 dans `\System Volume Information\_restore`{`CF106E14-791B-46F6-8501-1D89683FCE68`})

4. Le service de restauration système (`SrService`) prend un cliché instantané (*snapshot*) des clés de registre `HKEY_LOCAL_MACHINE` et `HKEY_USERS` et aussi de quelques fichiers système dynamiques (*DLLs*) et les stockent dans le point de restauration. Cette technologie de clichés instantanés permet à l'*OS* de collaborer avec l'application pour déterminer le moment temporel exact de création du point de restauration.

2.2. Base de registre

Afin de comprendre pourquoi Le mécanisme de restauration système prend un cliché instantané de la base de registre, il est important de comprendre comment fonctionne cette base de registre. Pour y accéder : `Start/Run/regedit` ou `Start/Run/regedt32.`

2.2.1. Les 5 clés racines

- **`HKEY_LOCAL_MACHINE`**

Renferme les informations globales sur la matériel et l'*OS*, type de bus, mémoire système, pilotes de périphériques et toutes informations requises lors de la séquence de démarrage de l'*OS*. Ces paramètres s'appliquent à l'ensemble des utilisateurs de la machine.
Cette clé ne correspond à aucun fichier stocké sur le disque dur, mais certaines de ses sous-clés correspondent bien à des fichiers ("ruches") :
`HKEY_LOCAL_MACHINE\SYSTEM, HKEY_LOCAL_MACHINE\SAM, HKEY_LOCAL_MACHINE\SECURITY, HKEY_LOCAL_MACHINE\SOFTWARE.`

- **`HKEY_CLASSES_ROOT`**

Comme pour les versions précédentes, elle contient les informations relatives aux associations de fichiers, aux informations *OLE* associés aux objets *COM* et aux classes d'association de fichiers (que l'on retrouve sous `HKEY_LOCAL_MACHINE\Software\Classes`).

- **`HKEY_CURRENT_CONFIG`**

Contient les paramètres de configuration du profil matériel actuel (en cours d'utilisation). Elle ne fait apparaître que les modifications apportées par rapport à la configuration de base qui est détaillée dans `HKEY_LOCAL_MACHINE` et ses sous-clés.

Ces mêmes modifications sont répertoriées dans `HKEY_LOCAL_MACHINE\System\CurrentControlSet\HardwareProfiles\Current key`.

- **`HKEY_CURRENT_USER`**

Renferme le profil utilisateur de l'utilisateur actuellement connecté. Clé qui fait référence à la clef `HKEY_USERS\user_SID`.

- **`HKEY_USERS`**

Contient tous les profils actifs des utilisateurs de la machine, sauf ceux accédant via le *LAN* puisque leurs profiles sont alors stockés sur leurs stations respectives

Des cinq clés racines que l'on a vues, HKLM et HKU sont de loin les plus importantes. En fait, les autres racines ne sont que des liens vers l'une des deux première clés racines. HKCU est un lien vers une sous-clé de HKU. HKCR et HKCC sont des liens vers des sous-clés de HKLM, le dernier étant par exemple un lien vers HKLM\SYSTEM\CurrentControlSet\Hardware Profiles\Current.

2.2.2. Stockage physique de la base de registre

La base de registre est constituée de ruches (*Hive*) auxquelles correspondent des fichiers stockés dans WINDOWS\System32\config pour la plupart (HKLM) et dans les dossiers des profils utilisateurs (pour HKU).

On trouve la liste des ruches chargées dans la base de registre à la clé : HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\hivelist. Les fichiers ruches ne comportent pas d'extension. Pour chaque ruche, *XP* crée des fichiers de sauvegarde avec l'extension .SAV. L'extension .LOG correspond aux fichiers qui retracent les changements affectant la ruche en cause. On ne trouve des ruches que dans HKLM et **HKU**, les autres clés racines étant des liens vers des sous clés de HKLM et HKU. Les ruches relatives à HKU sont dans les dossiers utilisateurs, dans les fichiers Ntuser.dat et UsrClass.dat. En fait, HKU\.DEFAULT a sa ruche dans WINDOWS\System32\config\default mais les autres ruches de HKU vont se trouver dans les fichiers cités précédemment.

La plupart des clés de la base de registre dont le nom figure en lettres capitales sont en fait des ruches.

Il n'existe pas sous *Windows XP* de limite physique pour la taille de la base de registre contrairement à *Windows 2000*. Pour connaître la taille occupée par la base de regsitre, un utilitaire issu du Ressource Kit de Windows 2000 fonctionne très bien. Il s'agit de *DuReg.EXE*, "*Registry Size Estimator*" (disponible en téléchargement au lien http://www.jurixt.com/informatique/menuftp.htm).

Il s'agit d'un utilitaire en ligne de commande, ne donnant que la taille occupée par les données contenues dans la base de registre et ne tient donc pas compte de l'espace libre également contenu dans la base de registre.

2.2.3. Clés de registre relatives à la restauration système

- **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Sr :** Cette clé est relative au filtre de restauration système.

- **HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Srservice:** relative au service de restauration système.

- **HKEY_LOCAL_MACHINE\Software\Microsoft\WindowsNT\CurrentVersion\SystemRestore :** Il s'agit de la clé où il est possible de changer des valeurs *DWORD* sans risque d'endommager le système. Voici les valeurs intéressantes :

  CompressionBurst → représente des secondes et spécifie le temps qu'il faut pour compresser les données du point de restauration après la machine ait atteint son état de repos.

  DiskPercent → permet de fixer le pourcentage d'espace disque utilisé par la restauration système (12% par défaut). La taille du magasin de données \System Volume Information_restore{*XX-X-X-X-XXXX*} est toujours calculée comme étant le maximum de (12%, DSMax).

`DSMax` → Spécifie la taille max du magasin de données. La taille par défaut est 400 *MBytes*.

`RPGlobalInterval` → représente des secondes et spécifie l'intervalle de temps entre chaque création automatique de points de restauration quand l'ordinateur est sous tension et au repos (24 heures par défaut).

`RPLifeInterval` → Spécifie le temps de vie des points de restauration en secondes avant qu'ils soient effaces. (Valeur par défaut 7776000 secondes, c'est-à-dire 90 jours).

Sources : http://www.jurixt.com/bdr/registre.htm, http://support.microsoft.com/default.aspx?scid=kb;en-us;295659

2.3. Monitorage des changements dans les fichiers système et d'applications

Pour dépister et copier les fichiers avant qu'ils soient modifiés, la restauration système utilise le pilote de filtre de restauration qui est au niveau *kernel*. Il permet d'interrompre momentanément une modification de fichier pour le copier sous sa forme originale dans le point de restauration. Une fois que la modification est effectuée, les changements sont notés dans le fichier *change.log* qui est aussi stocké dans le dossier point de restauration.

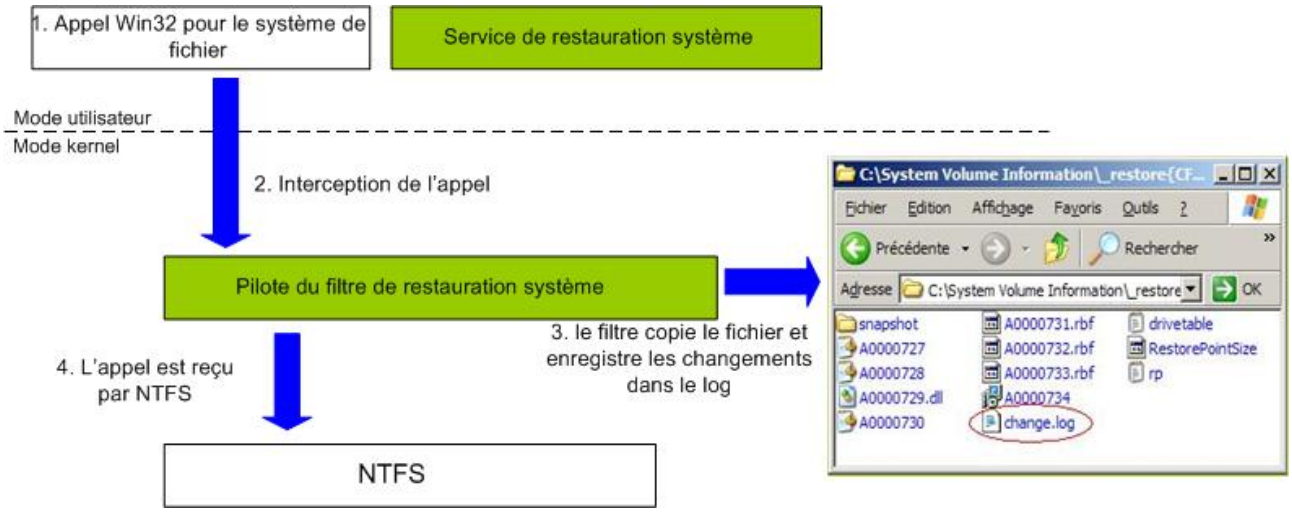


Fig. 58 : Copie de fichier et enregistrement des changements dans point de restauration

Le fichier *C:\WINDOWS\system32\Restore\filelist.xml* inclut toutes les extensions des fichiers devant être sauvegardés dans le point de restauration s'ils subissent un changement.

Cette liste, qui est documentée dans la plateforme SDK, impose à la restauration sytème de ne dépister que les fichiers ne contenant pas de données personnelles.

Je n'aimerais pas, par exemple, qu'un document Word important soit effacé juste parce que j'ai remis mon système dans un état antérieur afin de corriger un problème de carte son.

2.4. Le processus de restauration

Quand l'utilisateur (administrateur) ordonne au système d'effectuer une restauration, l'interface utilisateur (`\Windows\System32\Restore\Rstrui.exe`) crée une valeur *DWORD* nommée *`RestoreInProgress`* dans la clé de registre (`Hkey Local Machine\Software\Microsoft\System Restore`) et la met à 1. Le module de restauration initie alors une fermeture de l'*OS* avec redémarrage en appellant l'*API Win32 `ExitWindowsEx`*. Une fois le redémarrage commencé, le processus *`WinLogon`* (`\Windows\System32\Winlogon.exe` → premier processus exécuté lors du démarrage de *Windows*) comprend qu'il faut effectuer une restauration. *`Winlogon`* copie les fichiers sauvegardés dans le point de restauration approprié (RP6 dans notre exemple) et utilise les *change.logs* pour empêcher que les fichiers et dossiers personnels (.doc etc…) ne soient effacés. Quand le processus de restauration est terminé, le démarrage de l'*OS* continue normalement afin que le registre soit totalement restauré.

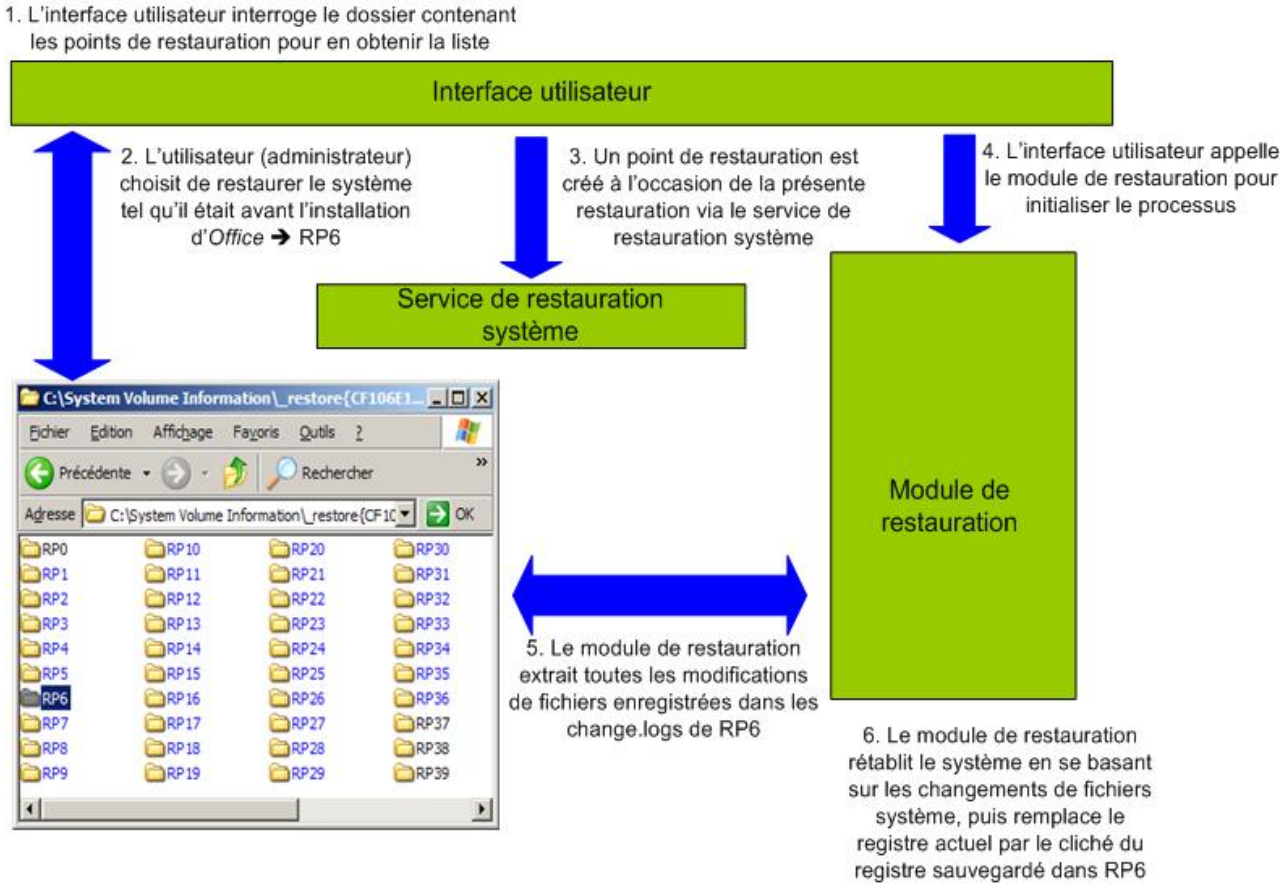


Fig. 59 : Processus de restauration

Remarque :

Les fichiers contenants les données des utilisateurs (.doc etc…) ne doivent pas avoir les mêmes extensions que celles des fichiers monitorés par la restauration système. Les utilisateurs pourraient perdre leurs données lors d'une restauration.

2.5. Ce qui est restauré

- Base de registre
- Profils utilisateurs locaux (local only—roaming user profiles not impacted by restore)
- *COM+ DB*
- *WFP.dll cache*
- *WMI DB*
- *IIS Metabase*

La liste des types de fichiers restaurés suivant leurs extensions est disponible dans le fichier `filelist.xml` situé dans `\WINDOWS\system32\Restore`.

La liste est disponible en annexe.

2.6. Ce qui n'est pas restauré

- La ruche *SAM* : Les mots de passé ne sont pas restaurés.
- Tous les documents personnels que l'on appel communément *user datas*.
- Tous les fichiers dont l'extension ne figure pas dans `filelist.xml`.

3. Conclusion

L'utilisation du mécanisme de restauration système *Windows XP* donne la possibilité de revenir à un état précédent de l'*OS* où tout marchait correctement.

Il faut cependant rester vigilant et ne surtout pas penser que faire des sauvegardes régulières sur support amovible devient inutile. Cela reste le seul remède contre les *crash* système ou les vols de machines.

La restauration système doit être utilisée quand des problèmes de configuration ou d'incompatibilités sont rencontrés.

Enfin, la philosophie de cet outil est avant tout d'apporter un complément aux sauvegardes régulières des disques durs.

4. Sources

- Certainement le meilleur site technique (Première source notamment pour les schémas) :

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwxp/html/windowsxpsystemrestore.asp

- Site de bonne qualité apportant des informations sur l'emplacement des services, pilotes et clés système :

http://www.kellys-korner-xp.com/xp_restore.htm

- Site intéressant mais trop orienté utilisation :

http://www.microsoft.com/technet/prodtechnol/winxppro/maintain/xpsysrst.mspx

- Informations sur la base de registre

http://www.jurixt.com/bdr/registre.htm,

- Liste des clés de registre modifiables

http://support.microsoft.com/default.aspx?scid=kb;en-us;295659

Partie 5 :

Permettre utilisation d'une souris usb et interdire mémoire de masse USB

(1 semaine d'étude)

1. Introduction

Une mémoire de masse USB est devenue un outil de tous les jours pour toute personne travaillant sur un PC mais aussi pour toute personne mal intentionnée désirant introduire des virus ou voler des données confidentielles disponibles sur une machine.
Comment faire donc pour que seulement des utilisateurs dignes de confiance puisse utiliser les mémoires de masse USB et les interdire à tout autre utilisateur. D'autre part, est-il possible de désactiver uniquement ce type de périphérique USB sans pour autant désactiver tous les autres, comme par exemple la souris, ce qui serait fort gênant.

2. Empêcher toute installation d' un périphérique de stockage USB

Il suffit à l'administrateur de refuser l'autorisation " Contrôle total " pour le groupe d' utilisateurs concernés, et ce, en accédant à l' onglet " Sécurité " de ces deux fichiers : `Usbstor.inf` et `Usbstor.pnf` situés dans `C:\WINDOWS\inf`.

Fig. 60 : Localisation des fichiers `usbstor.inf` et `usbstor.pnf`

- **fichier de type *inf* :** C'est un fichier qui fournit aux programmes d'installation les informations nécessaires à l'installation de matériel. C'est une sorte de liste de toutes les configurations logiques valides et des pilotes de matériel périphérique.

- **Fichier de type *pnf* :** Il s'agit d'un fichier *inf* précompilé. Le système d'exploitation *Windows* crée un fichier *pnf* pour chaque fichier *inf* afin d'optimiser les processus liés aux fichiers *inf*. Si un fichier *pnf* n'existe pas, le programme d'installation en génère un automatiquement pour le fichier *inf* correspondant.

3. Rôles

Comme nous l'avons à la section 2, il faut définir une stratégie de sécurité sur l'accès aux fichiers usbstor.inf et usbstor.pnf :

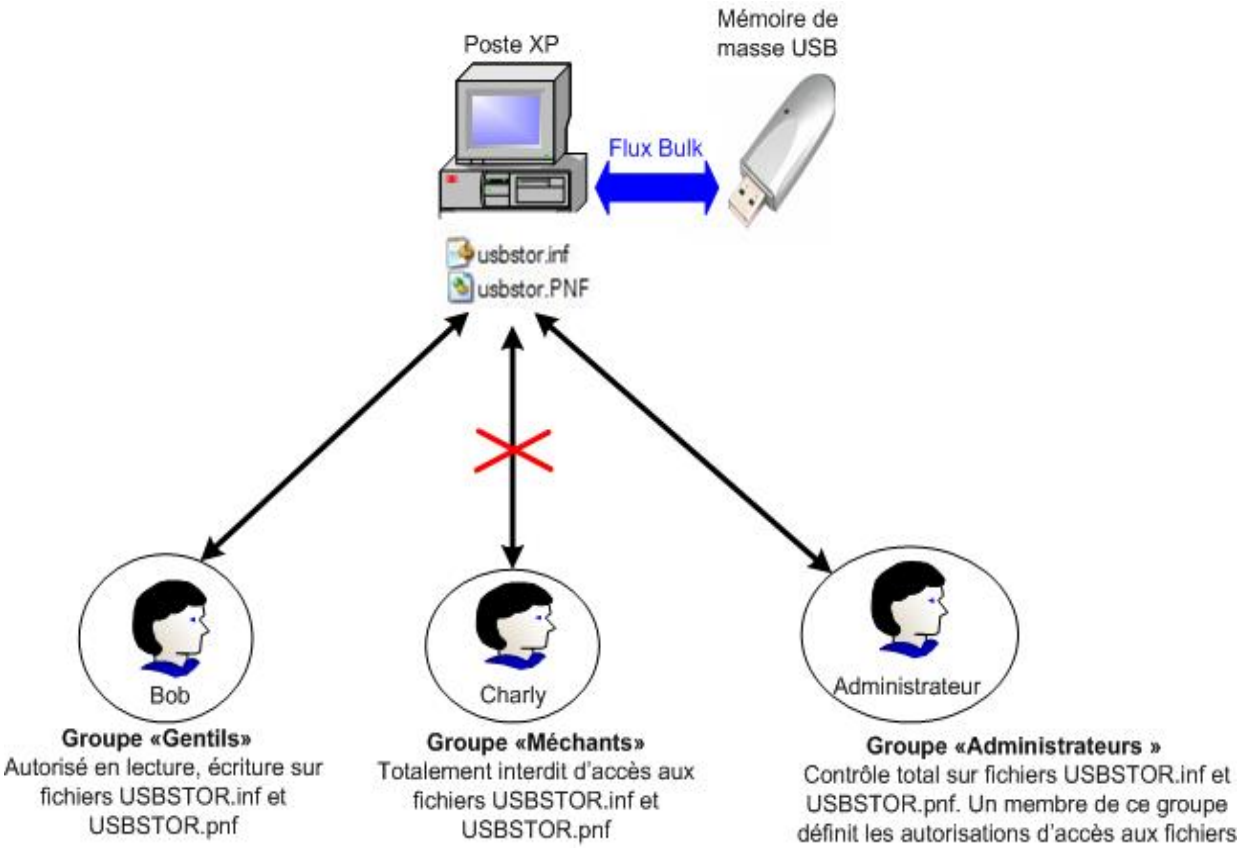


Fig. 61 : Scénario de stratégie de sécurité pour accès à la mémoire de masse USB

4. Périphérique de stockage déjà installé : que faire ?

Si le périphérique est déjà installé, l'administrateur doit aller dans la base de registre et modifier la valeur *DWORD* `start` de la clé `USBSTOR` située dans `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\`.

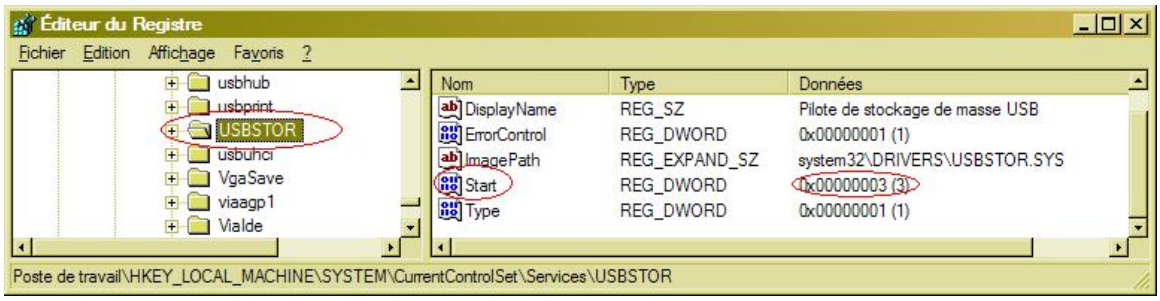


Fig. 62 : valeur *DWORD* `start` de la clé `USBSTOR` située dans `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\`

Il faut remplacer la valeur 3 par 4, ce qui va empêcher le démarrage de ce périphérique.
Dans ce cas, même l'administrateur ne pourra plus utiliser le périphérique.

Le test a été effectué sur ma machine. Le résultat a été convaincant : Je ne pouvais plus utiliser ma propre clé USB !

5. Empêcher toute écriture sur des périphériques de stockage USB

Il est parfois nécessaire que tous les utilisateurs, même ceux qui ne sont pas dignes de confiance, puissent lire les informations stockées sur le périphérique de stockage. Par contre afin d'empêcher tout vol de donnée confidentielles, il serait intéressant de les empêcher d'écrire sur le périphérique.

L'administrateur doit effectuer les manipulations suivantes :

Ouvrir la clé `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control` dans la base de registre.
Créer une clé nommée `StorageDevicePolicies`.
Dans cette clé créer une valeur *DWORD* nommée `WriteProtect` et lui affecter la valeur 1.

Fig. 63 : valeur *DWORD* `WriteProtect`

Le test a été effectué sur ma machine. Effectivement, il est impossible de déplacer un fichier présent sur une partition de la machine vers le périphérique de stockage : L'écriture est refusée.

6. Sources

http://support.microsoft.com/?kbid=823732 → Lien donnant la marche à suivre. Très synthétique.

L'excellente revue MISC numéro 16 (Novembre 2004) p. 10 à 16 (Article intitulé « La clé USB : votre nouvel ennemi »).

Partie 6 :

Fonctionnalités de VIRTUAL PC 2004 sur une base XP

(3 semaines d'étude)

1. Introduction

Virtual PC 2004 permet de créer un ou plusieurs ordinateurs virtuels, chacun exécutant son propre système d'exploitation, sur un ordinateur physique unique. C'est la réponse de Microsoft au produit *VMWare*. L'ordinateur virtuel émule un ordinateur standard basé sur un processeur x86, qui contient tous les composants matériels de base à l'exception du processeur. En utilisant du matériel émulé et le processeur de l'ordinateur physique, chaque ordinateur virtuel fonctionne comme un ordinateur physique distinct. Étant donné que chaque ordinateur virtuel possède son propre système d'exploitation, on peut exécuter simultanément plusieurs systèmes d'exploitation sur un même ordinateur.

Il faut bien distinguer l'ordinateur physique de l'ordinateur virtuel. Ce dernier tourne en tant qu'application Virtual PC dans l'OS de l'ordinateur physique.

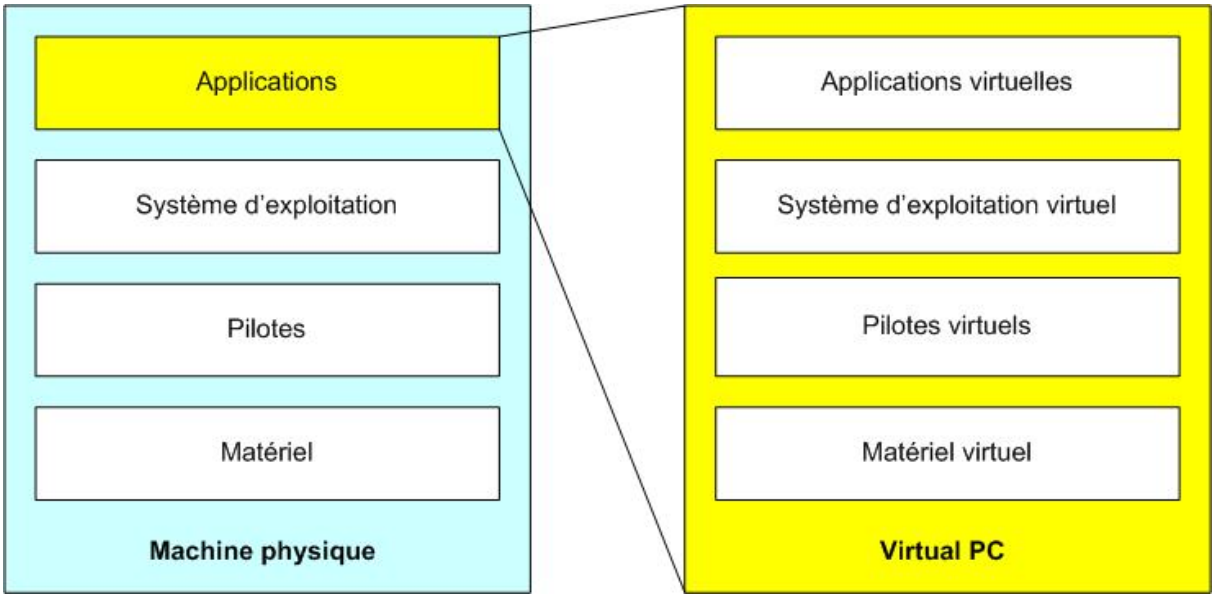


Fig. 64 : Virtual PC dans machine physique

2. Scénario

Le but de cette partie *Virtual PC* est de configurer le labo 4 *XP* dans un seul ordinateur.

Voir annexe pour acquisitions d'écrans.

L'ordinateur à disposition est un DELL portable possédant 1 *GBytes* de *RAM*, afin de pouvoir supporter les 3 machines virtuelles de test. Windows XP SP2 est l'OS de cette machine physique ou machine hôte.

La configuration désirée est la suivante :

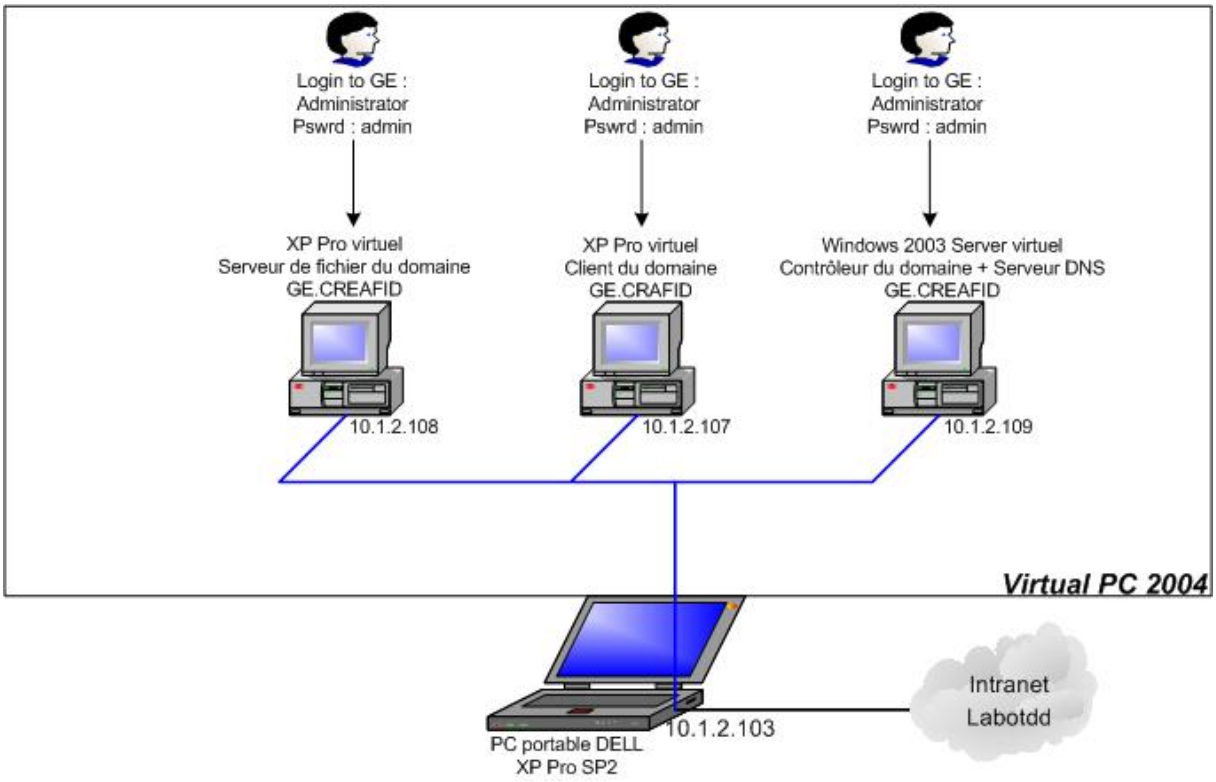


Fig. 65 : Configuration du labo 4 XP

Pour réaliser cette configuration, j'ai suivi pas à pas les instructions du Labo 4 *XP*. Le résultat est satisfaisant.

Pour arriver à ce résultat, il a d'abord fallu étudier les exigences matérielles requises, le mécanisme d'installation puis les paramètres réseaux pour que les machines virtuelles puissent communiquer entre elles.

3. Configuration et mécanisme d'installation

3.1. Installation de Microsoft Virtual PC 2004

L'installation de *Virtual PC* s'effectue comme l'installation de n'importe quelle application. Il suffit d'insérer le cd *Virtual PC* et suivre les étapes d'installation. A noter que la taille de l'exécutable d'installation est de 4,27 MB.

Lors de l'installation, un dossier *Microsoft Virtual PC* est créé dans le dossier `Program Files` de la partition où l'OS a été installé. L'exécutable se nomme `Virtual PC.exe` et pèse 3,81MB.

3.2. Exigences requises pour faire tourner une machine virtuelle

3.2.1. Matériel requis

- Architecture x86 de processeur ➔ un des processeurs suivants : AMD Athlon/Duron, Intel Celeron, Pentium 2,3 ou 4.
- 400 MHz minimum (1GHz recommandé).
- Lecteur CD ou DVD.
- Ecran super VGA (800 x 600) minimum.
- Souris et clavier compatibles Microsoft.
- Système d'exploitation : Windows XP Pro, Windows 2000 Pro ou Windows XP Tablet Edition.

3.2.2. Espace disque et RAM nécessaires suivant l'OS virtuel

Virtual PC supporte tous les systèmes d'exploitation pouvant tourner sur un processeur x86 32 bits.

Tous les tests effectués dans cette partie *Virtual PC* ont été effectués sur des *OS* virtuels *Windows server 2003* et *Windows Xp Pro*. Voici donc la *RAM* et l'espace disque nécessaires à leur installation.

OS virtuel	RAM minimum	Espace disque minimum
Windows XP Pro	128 MB	2 GB
Windows Server 2003	256 MB	2 GB

Fig. 66 : Espace disque et RAM nécessaires

N.B : *Windows 2000 Pro* nécessite seulement 96 MB de *RAM*, l'espace disque minimum étant de 2 GB (source : *Microsoft Virtual PC 2004 Technical Overview*).

Dans la partie tests on aura 1 *Windows Server 2003* et 2 *XP Pro*. La quantité de RAM nécessaire sera : 1x256 MB + 2x128 MB = 512 MB.

Sachant que l'*OS* de la machine physique étant un *XP Pro SP2* utilisant au minimum 150 MB de *RAM*, il est indispensable de posséder une machine possédant 1GB de *RAM*. Les nouvelles machines du labo de TDD le permettent.

Une machine virtuelle utilise uniquement la *RAM* physique libre contiguë de la machine physique :

hors fonctionnement **En fonctionnement**

Fig. 67 : RAM disponible en KB hors et en fonctionnement de XP Pro virtuel

La figure 3 nous montre bien que 361704 – 234504 = 127200 KB donc environ 128 MB utilisés pour un XP Pro virtuel.

3.3. Mécanismes d'installation

Source : Ressource kit virtual PC disponible dans le dossier créé par l'exécutable d'installation `\Program Files\Microsoft Virtual PC\Documentation\French`

Pour configurer un ordinateur virtuel reconnu par Virtual PC et figurant dans la liste de la Console Virtual PC, il faut utiliser l'Assistant Nouvel ordinateur virtuel afin de créer un ordinateur virtuel.

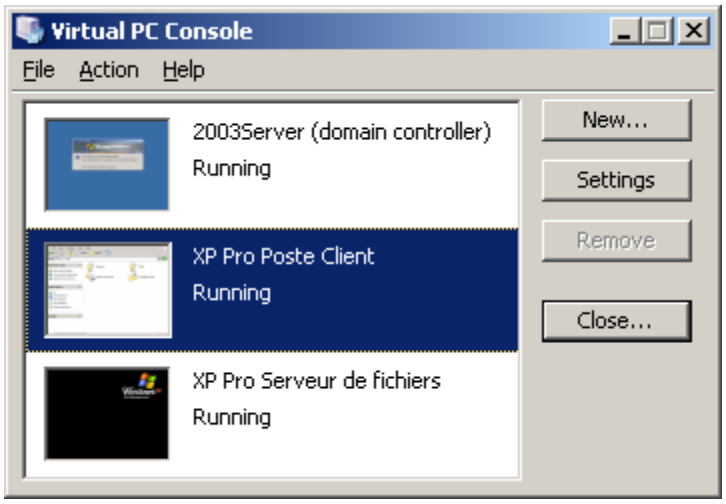


Fig. 68 : Console virtual PC

L'installation d'un nouvel OS virtuel entraîne les actions suivantes :

L'assistant crée le répertoire par défaut `Mes ordinateurs virtuels/nom_de_l'OS_virtuel` dans un dossier choisi par l'utilisateur. Il crée un fichier de configuration (`.vmc`) au format *XML*, contenant les informations de configuration de l'ordinateur virtuel, notamment tous les paramètres de l'ordinateur virtuel ainsi que la définition du matériel émulé, qu'il place dans le répertoire créé. Un disque dur virtuel permet d'exécuter le système d'exploitation de l'ordinateur virtuel et d'enregistrer d'autres données. Il est enregistré sous forme de fichier `.vhd` dans le répertoire créé. Le système d'exploitation le traite comme un disque dur physique.

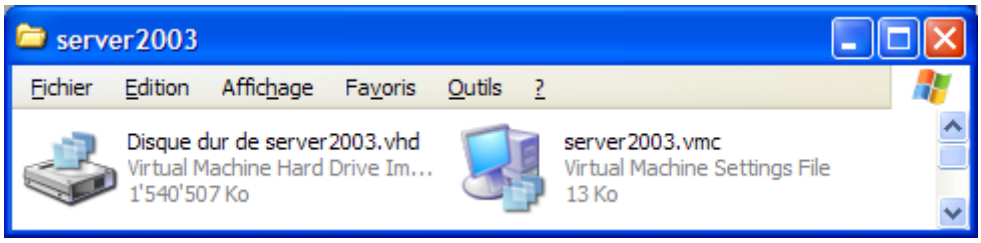


Fig. 69 : exemple de fichiers .vmc et .vhd dans `Mes ordinateurs virtuels/nom_de_l'OS_virtuel`

L' assistant utilise le type du système d'exploitation pour définir la quantité de mémoire allouée par défaut, que l'on peut modifier soit dans l'assistant, soit après avoir créé l'ordinateur virtuel.

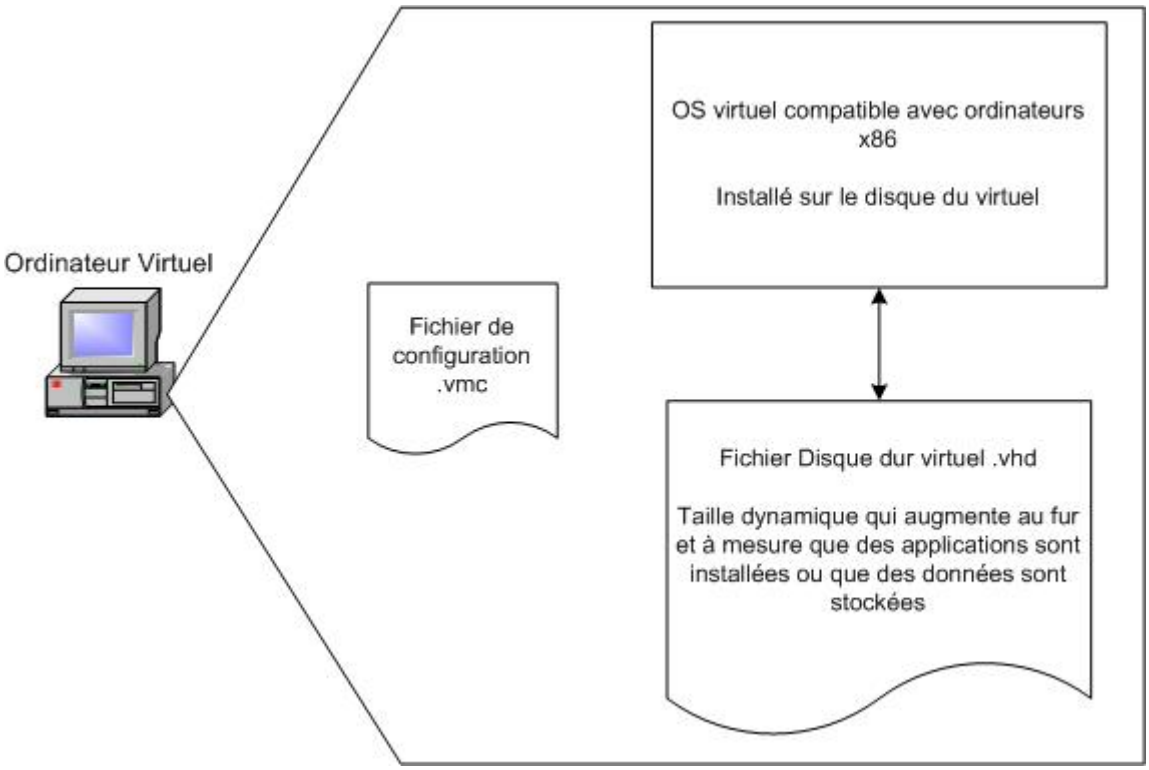


Fig. 69 : Composants d'un ordinateur virtuel

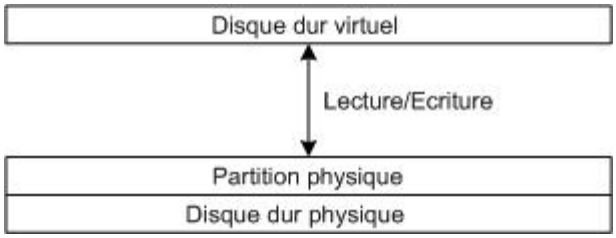


Fig. 70 : Relation entre disque dur virtuel et disque dur physique

3.4. Stockage des bases de registre virtuelles

La base de registre d'un *OS* virtuel est intégralement stockée dans le fichier disque dur virtuel .vhd. Ce fichier contient également la totalité de l'*OS* virtuel. Afin de le prouver, j'ai essayé d'ouvrir ce fichier avec un éditeur de texte standard mais ce fut impossible car il pèse 1,5 GB. J'ai donc installé l'outil *HEX WORKSHOP* disponible sur le *cd* de mon travail de diplôme. Il permet d'ouvrir le fichier .vsd en hexadécimal et afficher en ascii les parties lisibles. La lecture est fastidieuse mais on peut observer quand même du code d'implémentation *xp* et des parties de base de registre.

Afin d'éviter de mettre en annexe un fichier de 1,5 GB, on trouvera sur le cd annexe l'exécutable de *HEX WORKSHOP* qui permet d'ouvrir le fichier .vhd.

Voici quelques bribes de ce que l'on peut trouver dans ce fichier :

.C.u.r.r.e.n.t...S.y.s.t.e.m.\.C.u.r.r.e.n.t.C.o.n.t.r.o.l.S.e.t.\.C.o.n.t.r.o.l.\.I.D.C.o.n.f.i.g.D.B...C.u.r.r.e.n.t.D.o.c.k.I.n.f.o...D.o.c.k.i.n.g.S.t.a.t.e.....F.r.i.e.n.d.l.y.N.a.m.e.....D.e.v.i.c.e.D.e.s.c.....
.....H.a.r.d.w.a.r.e.I.D.....D.e.v.i.c.e.
.P.a.r.a.m.e.t.e.r.s.....C.o.m.p.a.t.i.b.l.e.I.D.s.....C.u.s.t.o.m.P.r.o.p.e.r.t.y.C.a.c.h.e.D.a.t.e...C.u.s.t.o.m.P.r.o.p.e.r.t.y.H.w.I.d.K.e.y...L.a.s.t.U.p.d.a.t.e.T.i.m.e.....H.T.R.E.E.\.R.O.O.T.\.0.....C.S.C.o.n.f.i.g.F.l.a.g.s...P.h.a.n.t.o.m.....C.l.a.s.s.....xg.u.....
.....S.y.s.t.e.m.\.C.u.r.r.e.n.t.C.o.n.t.r.o.l.S.e.t.\.C.o.n.t.r.o.l.\.C.l.a.s.s.....S.y.s.t.e.m.\.C.u.r.r.e.n.t.C.o.n.t.r.o.l.S.e.t.\.S.e.r.v.i.c.e.s.....S.o.f.t.w.a.r.e.\.M.i.c.r.o.s.o.f.t.\.W.i.n.d.o.w.s.
.N.T.\.C.u.r.r.e.n.t.V.e.r.s.i.o.n.\.P.e.r.H.w.I.d.S.t.o.r.a.g.e.....
S.e.r.v.i.c.e.....S.y.s.t.e.m.\.C.u.r.r.e.n.t.C.o.n.t.r.o.l
.S.e.t.....S.y.s.t.e.m.\.C.u.r.r.e.n.t.C.o.n.t.r.o.l.S.e.t.\.C.o.n.t.r.o.l.\.D.e.v.i.c.e.C.l.a.s.s.e.s.....S.y.s.t.e.m.....H.a.r.d.w.a.r.e. .P.r.o.f.i.l.e.s.....D.e.l.e.t.e.d. .D.e.v.i.c.e.
.I.D.s.....L.o.g.C.o.n.f...P.r.o.p.e.r.t.i.e.s.....D.e.v.i.c.e.I.n.s.t.a.n.c.e.....N.e.w.D.e.v.i.c.e.D.e.s.c...P.o.r.t.N.a.m.e.....D.e.t.e.c.t.S.i.g.n.a.t.u.r.e...D.i.s.a.b.l.e.C.o.u.n.t.....C.u.r.r.e.n.t.C.o.n.f.i.g.....E.j.e.c.t.a.b.l.e.D.o.c.k.s.....F.i.r.m.w.a.r.e.I.d.e.n.t.i.f.i.e.d.....F.i.r.m.w.a.r.e.M.e.m.b.e.r.....M.f.g...B.o.o.t.C.o.n.f.i.g.....A.l.l.o.c.C.o.n.f.i.g...F.o.r.c.e.d.C.o.n.f.i.g.....O.v.e.r.r.i.d.e.C.o.n.f.i.g.V.e.c.t.o.r.....B.a.s.i.c.C.o.n.f.i.g.V.e.c.t.o.r...F.i.l.t.e.r.e.d.C.o.n.f.i.g.V.e.c.t.o.r.....L.o.c.a.t.i.o.n.I.n.f.o.r.m.a.t.i.o.n...U.I.N.u.m.b.e.r.....U.I.N.u.m.b.e.r.D.e.s.c.F.o.r.m.a.t.....R.e.m.o.v.a.l.P.o.l.i.c.y...U.p.p.e.r.F.i.l.t.e.r.s.....L.o.w.e.r.F.i.l.t.e.r.s.....S.e.c.u.r.i.t.y.....D.e.v.i.c.e.T.y.p.e.....E.x.c.l.u.s.i.v.e...D.e.v.i.c.e.C.h.a.r.a.c.t.e.r.i.s.t.i.c.s...M.i.g.r.a.t.e.d.....C.o.n.t.r.o.l.F.l.a.g.s.....I.n.t.e.r.a.c.t.i.v.e.I.n.s.t.a.l.l.....S.o.f.t.w.a.r.e.\.M.i.c.r.o.s.o.f.t.\.W.i.n.d.o.w.s.\.C.u.r.r.e.n.t.V.e.r.s.i.o.n.\.P.o.l.i.c.i.e.s.....U.n.d.o.c.k.W.i.t.h.o.u.t.L.o.g.o.n.....P.l.u.g.P.l.a.y.\.P.a.r.a.m.e.t.e.r.s

Fig. 71 : extrait du fichier disque dur .vsd → visiblement une partie de la base de registre.

Description..Indicates the Win32 Configuration Manager error code. The following values may be returned: .0.This device is working properly. .1.This device is not configured correctly. .2.Windows cannot load the driver for this device. .3.The driver for this device might be corrupted, or your system may be running low on memory or other resources. .4.This device is not working properly. One of its drivers or your registry might be corrupted. .5.The driver for this device needs a resource that Windows cannot manage. .6.The boot configuration for this device conflicts with other devices. .7.Cannot filter. .8.The driver loader for the device is missing. .9.This device is not working properly because the controlling firmware is reporting the resources for the device incorrectly. .10.This device cannot start. .11.This device failed. .12.This device cannot find enough free resources that it can use. .13.Windows cannot verify this device's resources. .14.This device cannot work properly until you restart your computer. .15.This device is not working properly because there is probably a re-enumeration problem. .16.Windows cannot identify all the resources this device uses. .17.This device is asking for an unknown resource type. .18.Reinstall the drivers for this device. .19.Your registry might be corrupted. .20.Failure using the VxD loader. .21.System failure: Try changing the driver for this device. If that does not work, see your hardware documentation. Windows is removing this device.

Fig. 72 : extrait du fichier disque dur .vsd → messages relatifs à certains codes d'erreur (stockés en clair)

4. Réseau virtuel

4.1. Paramètre réseau d'un OS vituel

Pour accéder au réseau physique, il faut configurer le paramètre Réseau de l'ordinateur virtuel :

- Nombre de cartes réseau . Maximum de quatre cartes réseau émulées qui seront utilisées par l'ordinateur virtuel.
- Configuration des cartes . On peut affecter chaque carte émulée à n'importe quelle carte réseau installée dans l'ordinateur physique :

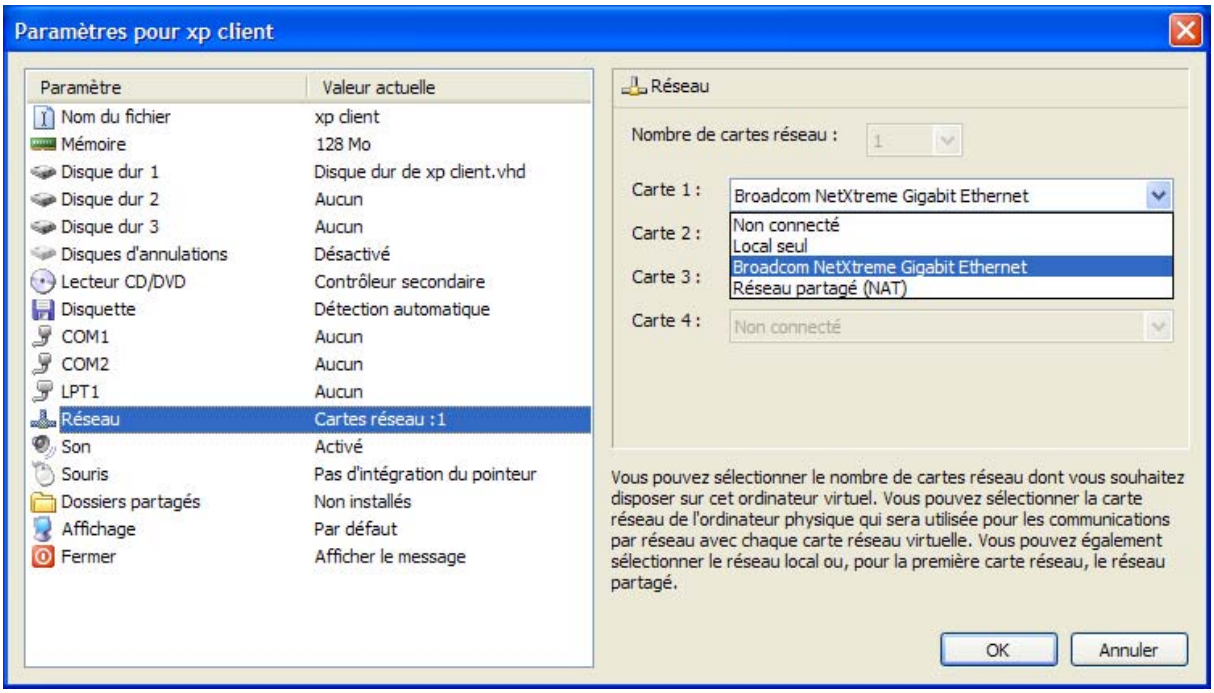


Fig. 73 : Fenêtre des paramètres d'un *OS* virtuel

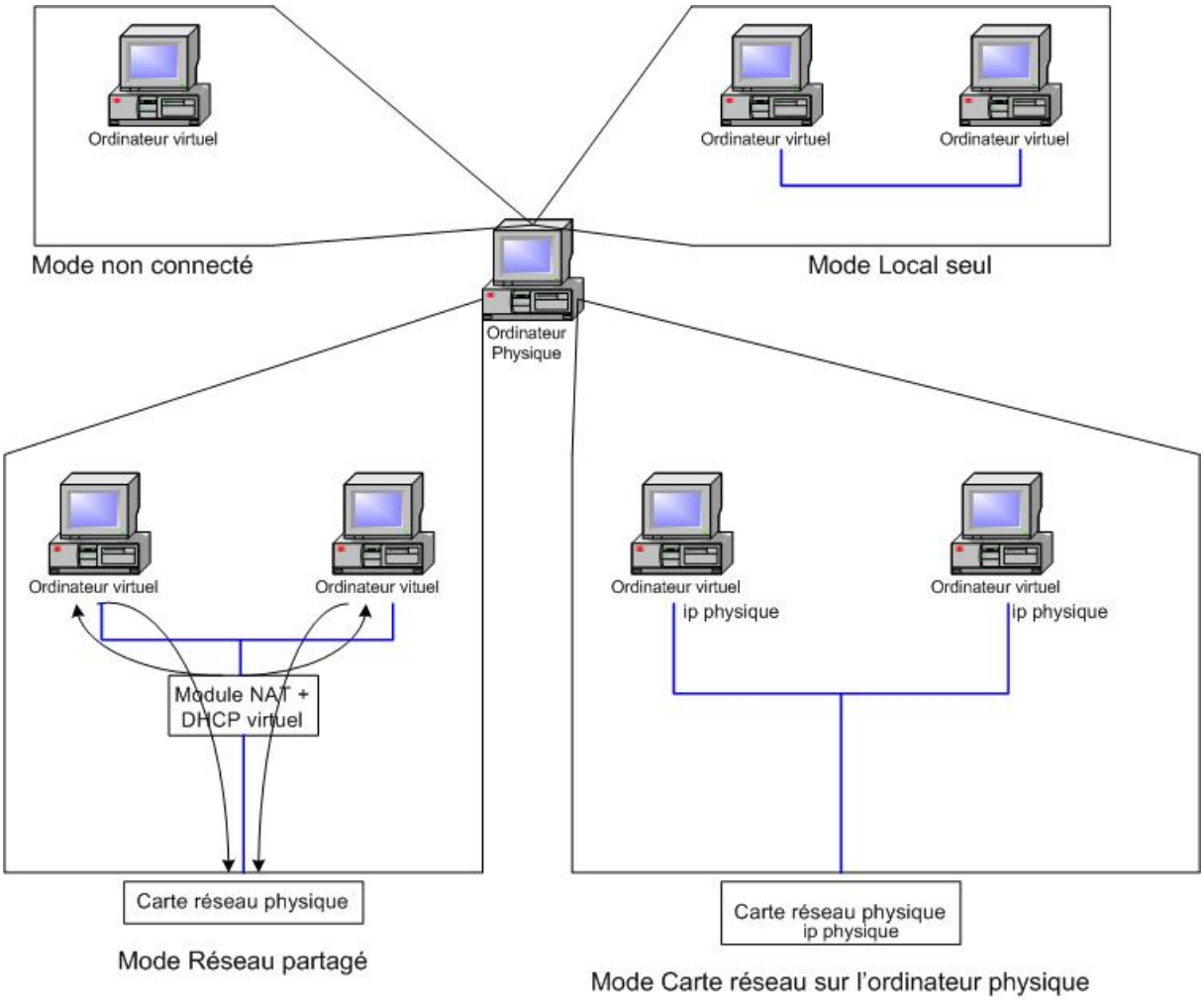


Fig. 74 : Description du paramètre réseau

Remarque :

J'ai volontairement choisi d'utiliser le mode `Carte réseau sur l'ordinateur virtuel` pour effectuer mes tests. Il s'agit du cas le plus courant dans un *LAN* bien que le mode *dhcp* le soit également.

4.2. Changer l'adresse *mac* virtuelle

Le fichier `.vmc` de configuration (d'une machine virtuelle), contient l'adresse *MAC* de la carte *ethernet* virtuelle. La valeur de cette adresse est par défaut la même pour chaque machine virtuelle, il faut donc la changer afin qu'il n'y est pas de conflits.
Il faut ouvrir le fichier *xml* `.vmc` de chaque ordinateur virtuel avec le bloc note et effectuer le changement suivant :

Remplacer :	**Par :**
<Ethernet_card_address type="bytes">0003FF01C5CF</ thernet_card_address>	<Ethernet_card_address type="bytes"></ thernet_card_address>

Fig. 75 : Changement de l'adresse *mac*

Il s'agit de supprimer la *mac address*. Ainsi, *Virtual PC* en créera une nouvelle au prochain redémarrage de la machine virtuelle correspondante.

4.3. Configuration réseau pour les tests

Afin d'analyser le comportement réseau des machines virtuelles, j'ai mis en place 3 machines physiques, chacune tournant sous *Windows XP SP2* (10.1.2.101, 10.1.2.102 et 10.1.2.103). *Virtual PC* est installé sur 10.1.2.103 qui est le PC portable *DELL INSPIRON 8600* que M.Litzistorf m'a demandé d'utiliser. Cette machine supporte 3 machines virtuelles : 1 *Windows 2003 Server* (10.1.2.109), 2 *Windows XP Pro* de base (10.1.2.107 et 10.1.2.108). Virtual PC est également installé sur 10.1.2.101 et supporte 1 Windows XP Pro de base (10.1.2.104). Ethereal est installé sur 1 machine physique (10.1.2.102) et sur 1 machine virtuelle (10.1.2.107).

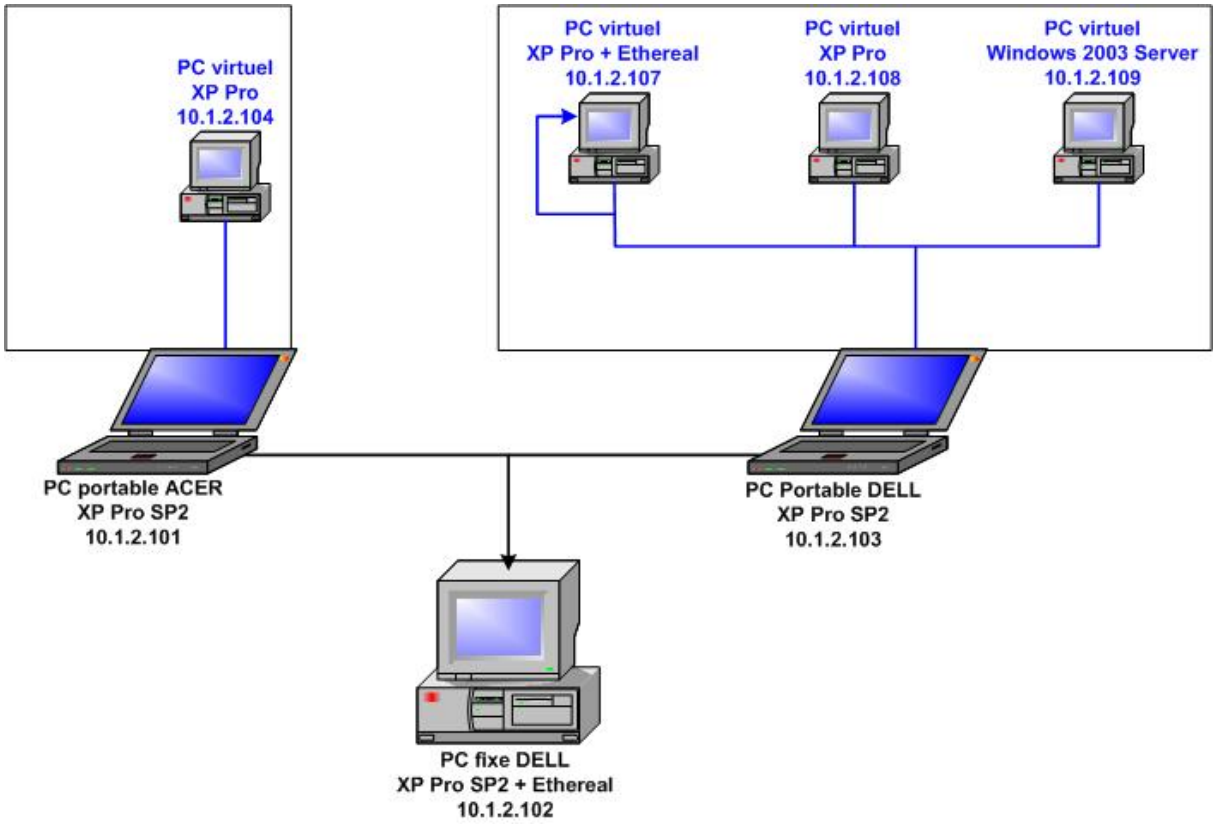


Fig. 76 : Schéma de configuration des tests

4.4. Tests

4.4.1. Test 1 : `ping 10.1.2.107` depuis 10.1.2.101

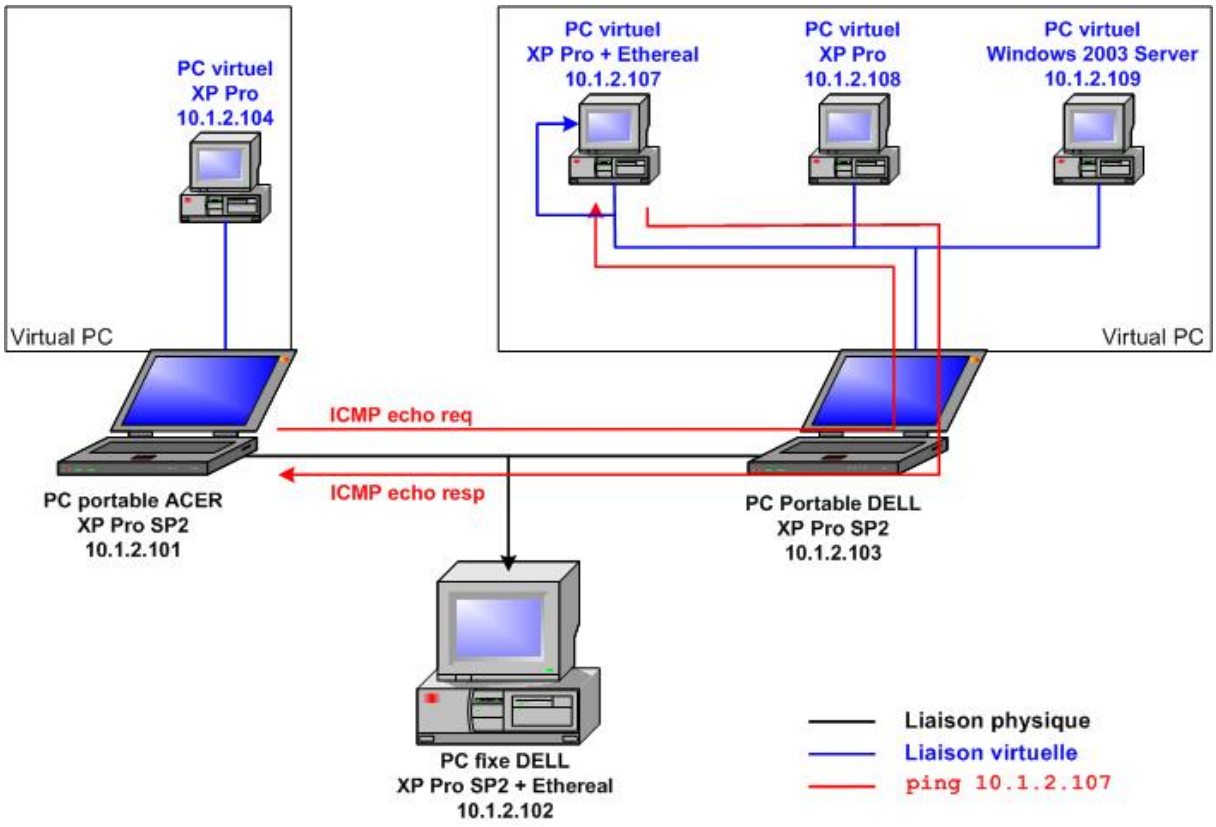


Fig. 77 : Test 1

Il s'agit de tester si une machine virtuelle est atteignable depuis une machine physique.

Observons l'acquisition *Ethereal* effectuée depuis le réseau physique par la machine 10.1.2.102 :

```
No.     Time        Source                Destination
Protocol Info
  3 0.000735    10.1.2.101            10.1.2.107            ICMP
Echo (ping) request

No.     Time        Source                Destination
Protocol Info
  4 0.001702    10.1.2.107            10.1.2.101            ICMP
Echo (ping) reply
```

Fig. 78 : Résultat du test 1 sur 10.1.2.102

On s'aperçoit que la machine virtuelle a un comportement identique à celui d'une machine physique. On peut en conclure qu'une machine virtuelle disposant d'une adresse *ip* statique valable dans le réseau physique est vue depuis ce réseau comme une machine physique.

Le résultat est le même quand cette communication est sniffée sur le réseau virtuel.

4.4.2. Test 2 : Prise d'empreinte active de 10.1.2.107 depuis 10.1.2.101

Il s'agit de déterminer l'*OS* d'une machine virtuelle (*XP Pro*) depuis une machine physique distante avec l'outil *nmap* version 3.75 (dernière version).

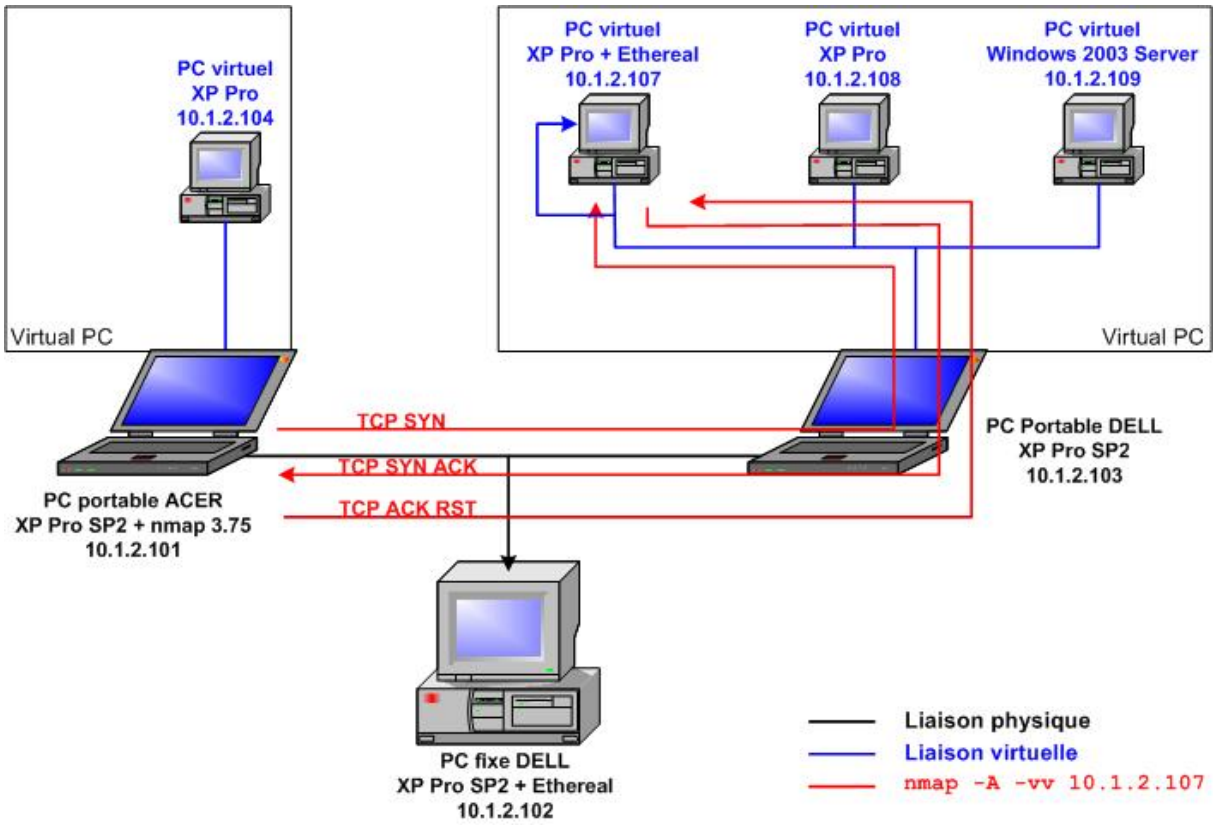


Fig. 79 : Test 2

```
MAC Address: 00:03:FF:A9:75:72 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 95/98/ME|NT/2K/XP
OS details: Microsoft Windows Millennium Edition (Me), Windows 2000
Pro
ced Server, or Windows XP, Microsoft Windows XP SP1
OS Fingerprint:
TSeq(Class=RI%gcd=1%SI=377F%IPID=I%TS=0)
T1(Resp=Y%DF=Y%W=FAF0%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=FAF0%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Fig. 80 : Résultat du Test 2

On constate que l'empreinte du *XP Pro* de la machine virtuelle est identique à celle d'un *XP Pro* d'une machine physique. (Voir annexe p. pour empreinte d'un XP Pro physique)

Voyons si, en changeant le *TTL* ainsi que la taille de la fenêtre *TCP* du *XP Pro* dans la base de registre de la machine virtuelle, *nmap* arrive à le détecter.

Changements effectués sur machine virtuelle :

- Ajout de la valeur *DWORD* `DefaultTTL` dans `HKLM/SYSTEM/CurrentControlSet/Services/Tcpip/Parameters`
  → fixée à 64 au lieu de 128 par défaut.
- Ajout de la valeur *DWORD* `TcpWindowSize` dans `HKLM/SYSTEM/CurrentControlSet/Services/Tcpip/Parameters`
  → fixée à 13140 au lieu de 8760 par défaut.

```
MAC Address: 00:03:FF:A9:75:72 (Microsoft)
No exact OS matches for host
TCP/IP fingerprint:
TSeq(Class=RI%gcd=1%SI=2559%IPID=I%TS=0)
TSeq(Class=RI%gcd=1%SI=4310%IPID=I%TS=0)
TSeq(Class=RI%gcd=1%SI=1BBE%IPID=I%TS=0)
T1(Resp=Y%DF=Y%W=3354%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=3354%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=38%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)
```

Fig. 81 : Résultat du test 2 avec TTL et et Taille fenêtre TCP modifiée

nmap ne peut plus reconnaître l'*OS* de la machine virtuelle 10.1.2.107. Il est donc possible de le camoufler. Seules les versions des services permettent d'avoir une idée sur le type d'*OS*.

4.4.3. Test 3 : `ping 10.1.2.107` par 10.1.2.108

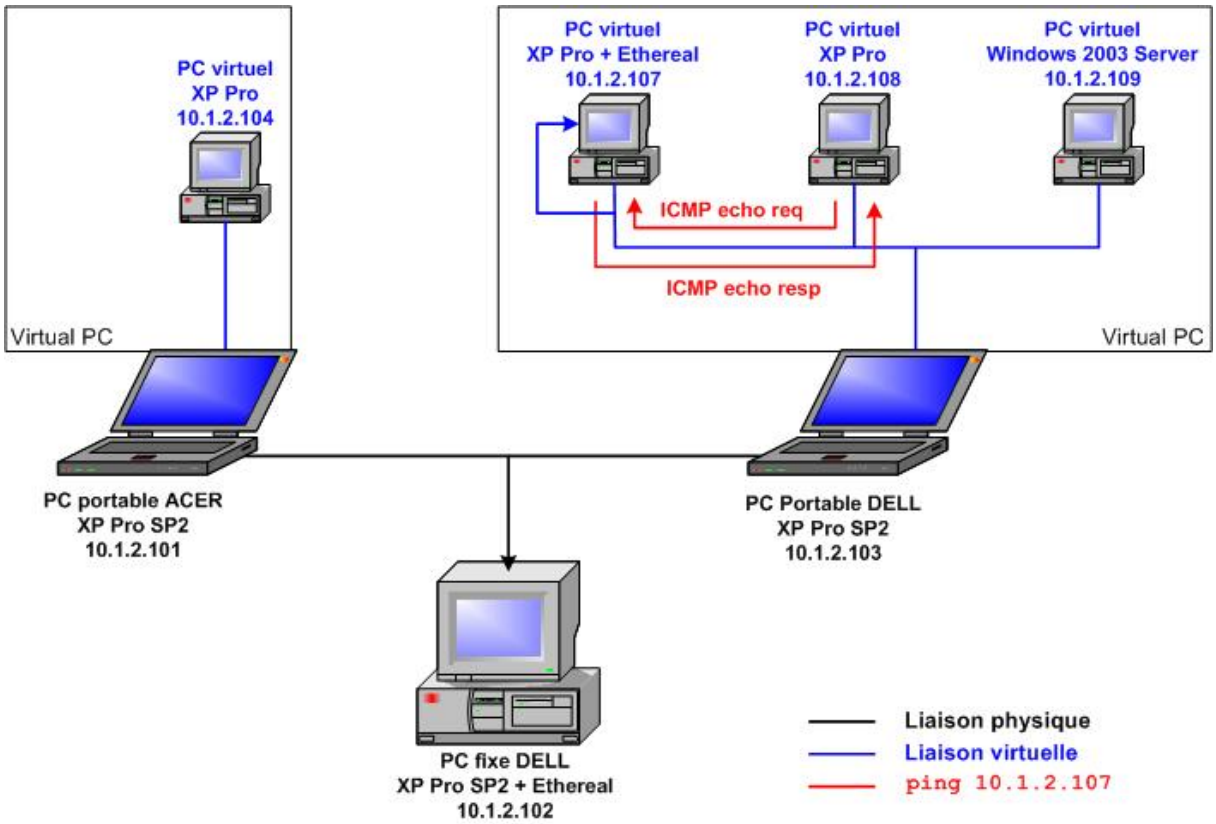


Fig. 82 : Test 3

Ce test va permettre de déterminer si une conversation *ICMP* entre 2 machines virtuelles du même ordinateur physique est détectable par un sniffer du réseau physique.

Le *sniffer* du réseau physique n'a rien pu observer. On peut en conclure qu'une communication entre machines virtuelles du même réseau virtuel n'utilise pas l'interface *ethernet* de la machine hôte.

Le sniffer du réseau virtuel intercepte sans problème la communication.

4.4.4. Test 4 : Ping 10.1.2.107 depuis 10.1.2.104

On veut analyser une communication *ICMP* entre 2 machines virtuelles située dans 2 machines physiques différentes. La communication est sniffée depuis le réseau physique :

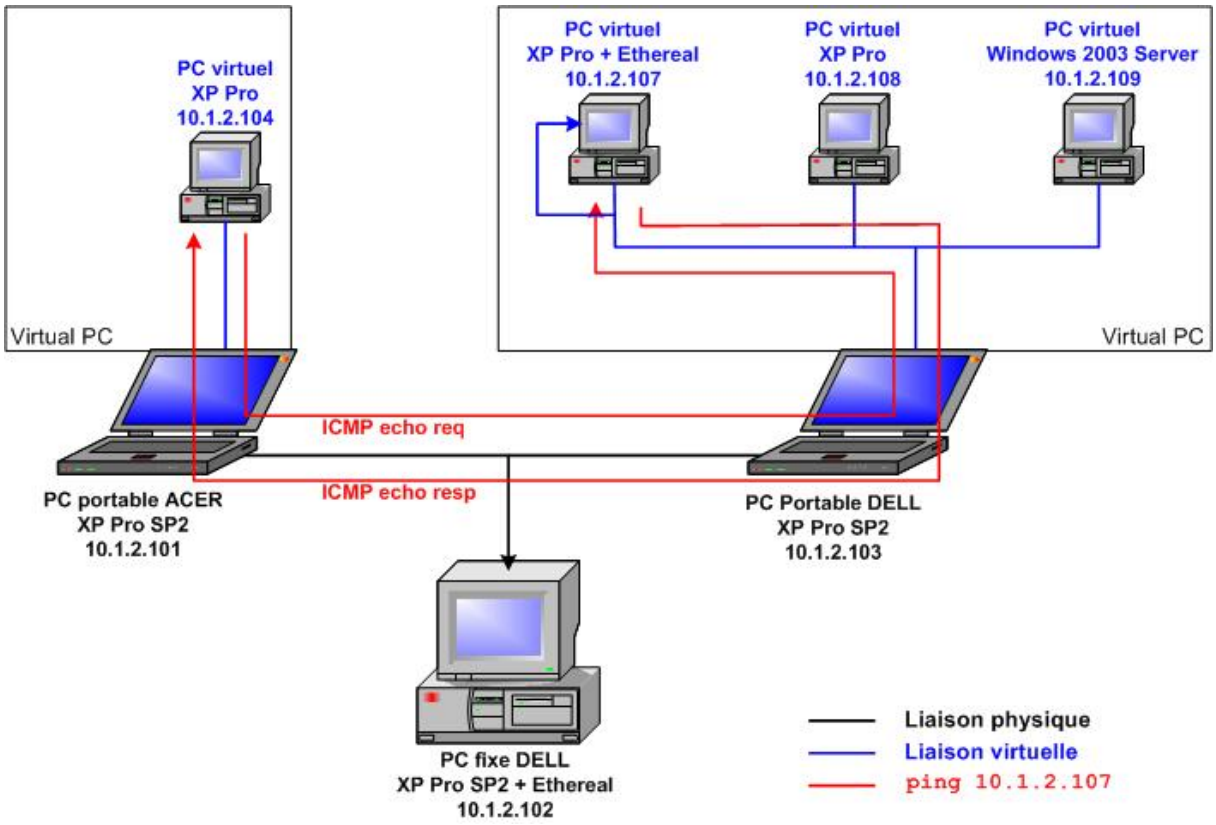


Fig. 83 : Test 4

```
No.     Time        Source                Destination
Protocol Info
      3 0.000729    10.1.2.104            10.1.2.107            ICMP
Echo (ping) request

Frame 3 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: 00:03:ff:4f:f5:b9, Dst: 00:03:ff:a9:75:72
Internet Protocol, Src Addr: 10.1.2.104 (10.1.2.104), Dst Addr:
10.1.2.107 (10.1.2.107)
Internet Control Message Protocol

No.     Time        Source                Destination
Protocol Info
      4 0.000972    10.1.2.107            10.1.2.104            ICMP
Echo (ping) reply

Frame 4 (74 bytes on wire, 74 bytes captured)
Ethernet II, Src: 00:03:ff:a9:75:72, Dst: 00:03:ff:4f:f5:b9
Internet Protocol, Src Addr: 10.1.2.107 (10.1.2.107), Dst Addr:
10.1.2.104 (10.1.2.104)
Internet Control Message Protocol
```

Fig. 84 : Résultat du test 4 sur 10.1.2.102

Le résultat est le même sur le sniffer du réseau virtuel.

4.4.5. Test 5 : `nmap –A –vv 10.1.2.109` depuis 10.1.2.101

On veut déterminer l'empreinte d'un *Windows Server 2003*.

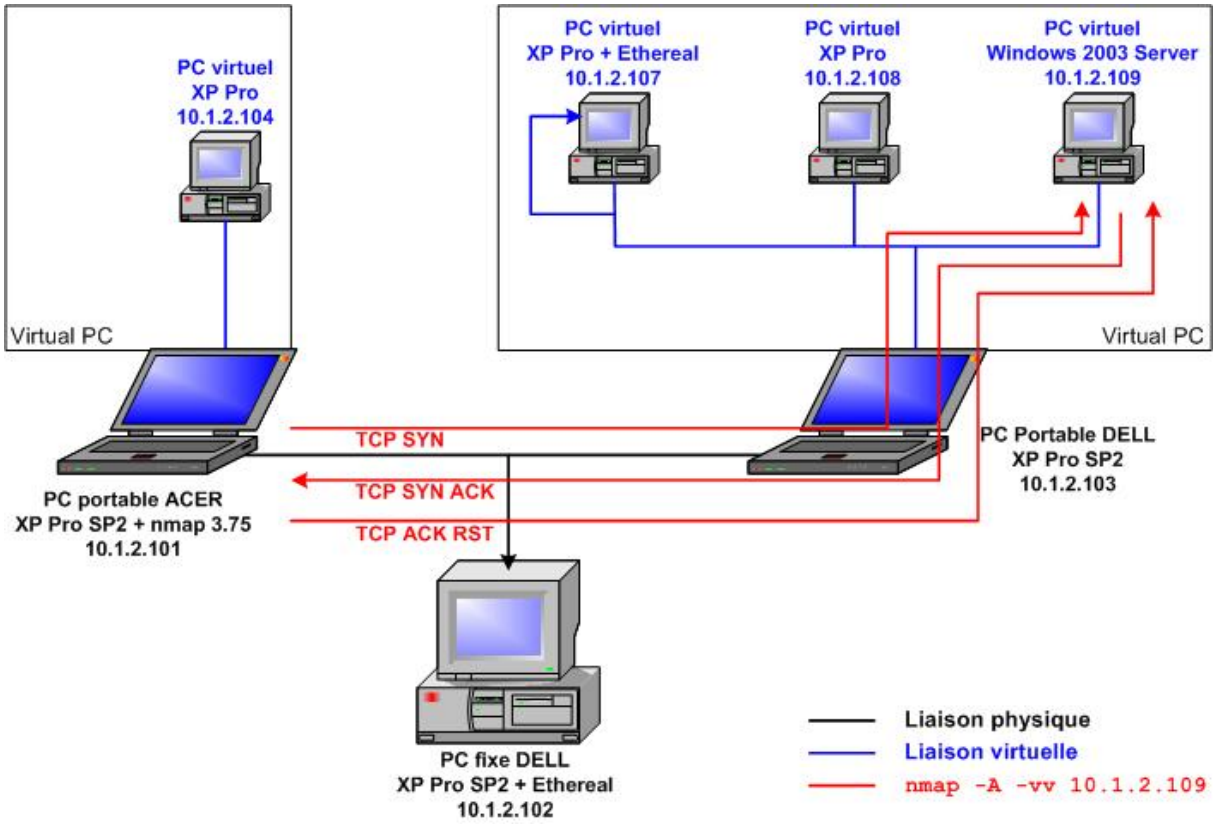


Fig. 85 : Test 5

```
              MAC Address: 00:03:FF:A8:75:72 (Microsoft)
Device type: general purpose
Running: Microsoft Windows 2003/.NET|NT/2K/XP
OS details: Microsoft Windows 2003 Server or XP SP2
OS Fingerprint:
TSeq(Class=TR%IPID=I%TS=0)
T1(Resp=Y%DF=Y%W=402E%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=402E%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T5(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(Resp=Y%DF=N%W=0%ACK=O%Flags=R%Ops=)
T7(Resp=Y%DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(Resp=Y%DF=N%TOS=0%IPLEN=B0%RIPTL=148%RIPCK=E%UCK=E%ULEN=134%DAT=E)

TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
TCP ISN Seq. Numbers: BD59C154 1F063C57 3F43FE6A 5D4F9B99 2E2FD5C3
6D4170CB
IPID Sequence Generation: Incremental
```

Fig. 86 : Empreinte Windows Server 2003

4.4.6. Test 6 : ping 10.1.2.108 depuis 10.1.2.103

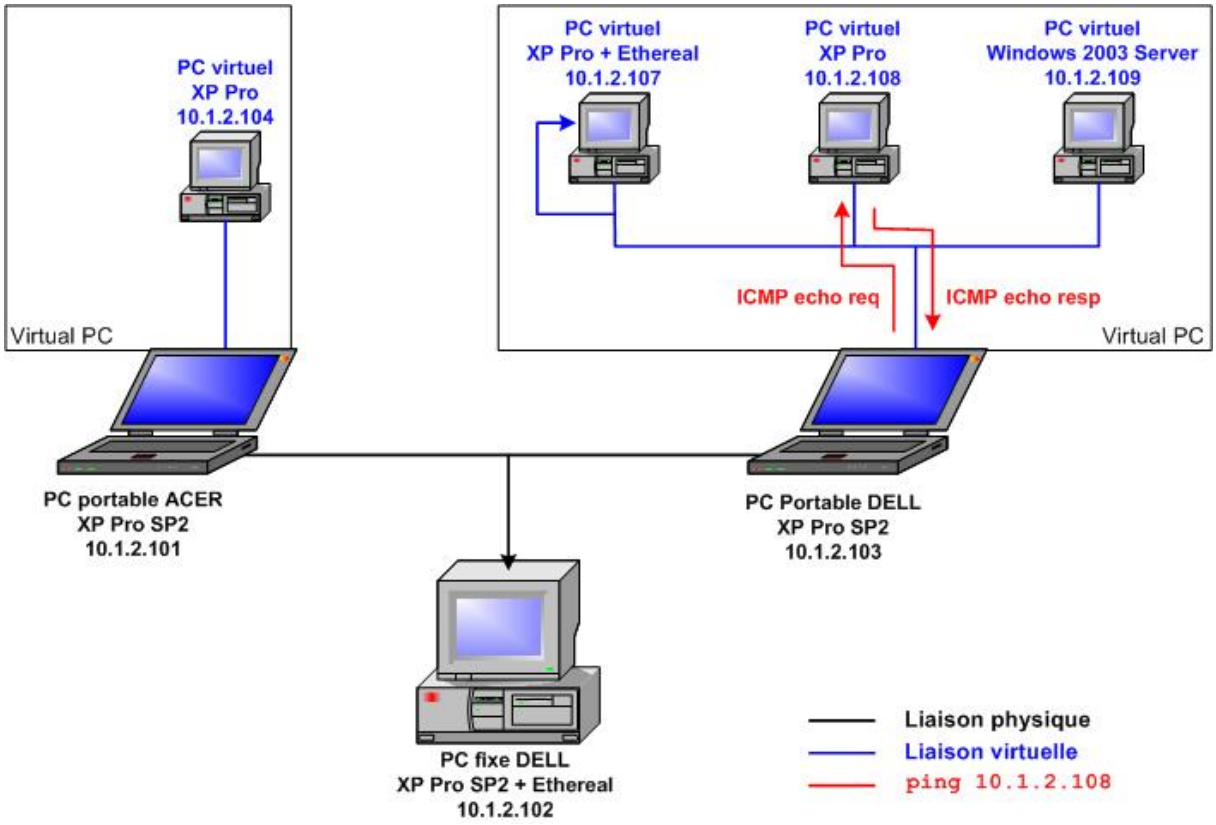


Fig. 87 : Test 6

Ce test va nous permettre de determiner si du traffic ICMP entre 1 machines virtuelles et la machine physique hôte peut être intercepté sur le réseau physique.

```
No.     Time        Source        Destination           Protocol Info

1       0.000000    10.1.2.103    Broadcast             ARP      Who
has 10.1.2.107?  Tell 10.1.2.103

Frame 1 (60 bytes on wire, 60 bytes captured)
Ethernet II, Src: 00:0d:56:af:75:72, Dst: ff:ff:ff:ff:ff:ff
Address Resolution Protocol (request)
```

Fig. 88 : Résultat du test 6 sur 10.1.2.102

La communication est interceptée normalement sur le réseau virtuel.

5. Principaux paramètres de sécurité des ordinateurs virtuels

Gérer la sécurité des ordinateurs virtuels consiste principalement à gérer les autorisations et accès utilisateur, selon le même processus pour le système d'exploitation d'un ordinateur virtuel que pour celui de tout autre ordinateur. Il faut recourir aux mêmes méthodes de verrouillage et de sécurité sur le système d'exploitation d'un ordinateur virtuel que pour le système d'exploitation hôte et les systèmes d'exploitation des autres ordinateurs de l'entreprise.

Il faut que les autorisations sur les dossiers partagés ne soient attribuées qu'aux utilisateurs qui ont besoin d'accéder aux informations de l'ordinateur physique. Bien qu'il soit possible de configurer plusieurs ordinateurs virtuels pour accéder au même dossier partagé sur le système d'exploitation hôte, cette pratique est déconseillée.

Si plusieurs disques différents font référence à un même disque parent, il vaut mieux empêcher les utilisateurs de modifier le disque parent. Il faut donc configurer le disque parent en lecture seule et définir des autorisations sur le fichier de disque dur virtuel afin d'empêcher les utilisateurs de modifier ses attributs.

Il ne faut pas configurer un disque dur virtuel en lecture seule, sauf s'il s'agit d'un disque parent. Tout autre disque dur virtuel configuré en lecture seule peut provoquer des problèmes graves, notamment la corruption de configurations ou la perte de données.

On peut également utiliser l'option `Sécurité` de la Console *Virtual PC* pour limiter l'accès à certaines fonctions et options à l'administrateur ou à un membre du groupe Administrateurs. Si cette option est sélectionnée, les utilisateurs qui ne sont ni administrateurs, ni membres du groupe Administrateurs ne peuvent pas accéder à la fonction ou à l'option. Il est possible de restreindre l'accès à chacune des fonctions décrites dans le tableau ci-dessous.

Application de restriction à	Résultat
Options	Limite au seul administrateur, ou à un membre du groupe Administrateurs, l'accès au menu `Options` de Virtual PC.
Paramètres de l'ordinateur virtuel	Limite au seul administrateur, ou à un membre du groupe Administrateurs, l'accès au menu `Paramètres` de Virtual PC.
Création ou suppression d'ordinateur virtuel	Limite au seul administrateur, ou à un membre du groupe Administrateurs, la possibilité de créer ou de supprimer un ordinateur virtuel.
Assistant Disque virtuel	Limite au seul administrateur, ou à un membre du groupe Administrateurs, l'accès au menu `Options` de Virtual PC.

Fig. 89 : Tableau des restrictions d'accès pour la Console *Virtual PC*

Important de stocker les ordinateurs virtuels dans un emplacement sécurisé sur l'ordinateur physique. Par défaut, les ordinateurs virtuels sont créés dans le dossier `Mes documents`, ce qui limite l'autorisation d'accès aux administrateurs. Si les ordinateurs virtuels sont déplacés vers un dossier dont l'accès n'est pas restreint, ils deviennent vulnérables. Par exemple, si un utilisateur malveillant accède à un ordinateur virtuel dans l'état enregistré et le restaure, il obtiendra les mêmes privilèges que l'utilisateur qui l'a placé dans l'état enregistré à l'origine.

6. Conclusion

Virtual PC aura permis la configuration du Labo *XP* 4 sur une seule machine. On a pu ainsi créer 3 machines virtuelles membres d'un domaine et effectuer tous les tests possibles.
On peut, avec le poste client, accéder au serveur de fichiers via l'annuaire d'*Active directory*.

Avec la mise en œuvre de cette configuration, on a pu observer les points suivants :

- Les OS virtuels nécessitent de la RAM libre contiguë (128 MB pour XP Pro, 256 MB pour Windows Server 2003)
- Un disque dur virtuel est stocké sous forme d'un seul fichier sur le disque dur physique dont la taille croît dynamiquement au fur et à mesure que des applications sont installées et que des documents sont créés
- Plusieurs modes réseau sont possibles → Choix d'adresses physiques statiques
- Les communications entre machines virtuelles s'effectuent comme si elles étaient dans un réseau physique. Ces communications virtuelles ne sont par contre pas vues depuis le réseau physique.
- Les communications entre machines virtuelles et machines physiques s'effectuent comme s'il n'y avait que le réseau physique.
- Les empreintes des *OS* de machines virtuelles sont les mêmes que celles d'*OS* de machines physiques.
- Les règles de sécurité Windows se font de la même manière que sur un Windows physique. Comme le disque dur virtuel est un fichier sur le disque dur physique, il est possible de gérer les autorisations d'accès en lecture et écriture à un *OS* virtuel.

Le produit *Virtual Server 2005* est sorti en version bêta. Je ne l'ai malheureusement pas testé.
La documentation est disponible sur le *CD* de mon travail de diplôme.

Conclusion travail de diplôme

Ce travail de diplôme m'aura permis de mieux comprendre ce qui se cache derrière Windows XP, système d'exploitation que nous utilisons presque tous les jours.

De plus, les différentes études m'ont permis de comprendre en profondeur les mécanismes de chiffrement en général, les mécanismes de restauration de système et les partages de fichiers entre machines d'un même réseau.

On aura ainsi étudié tout ce que proposait l'énoncé du diplôme, bien que le temps imparti ait été un peu juste. En effet, il y a eu une grande quantité de concepts à comprendre Je me suis ainsi rendu compte que plus les utilisations d'outils sont simples, plus les mécanismes qui se cachent derrière sont compliqués et obscures, surtout en ce qui concerne Microsoft.

J'ai eu du mal a trouvé de la documentation réellement technique. Il semblerait que Microsoft dissimule volontairement les informations concernant leurs produits, notamment les codes sources qui auraient été fort utile pour comprendre en profondeur les mécanismes.

Je regrette de ne pas avoir eu le temps d'aborder *LongHorn*. Cela dit, ce produit semblerait nécessiter des ressources processeur et RAM assez gigantesques, ce qui me réconforte dans le sens que la technologie actuelle ne permet pas encore de l'utiliser pleinement.

Enfin, l'utilisation de Virtual PC a été motivante. Ce produit permet de réaliser des tests tout à fait semblables à la réalité et serait parfait pour effectuer des laboratoires pédagogiques (notamment les labos XP).

Table des figures