

VPN et tunnel au niveau de la couche 2

Le but d'un tunnel est d'assurer la sécurité d'une liaison dédiée au coût d'une infrastructure partagée. Ceci peut se réaliser à différents niveaux, des couches les plus basses jusqu'à l'application. En voici quelques exemples :

- Utilisation d'une paire de fibres optiques dans un câble en comportant plusieurs.
- Multiplexage en longueurs d'onde sur une fibre optique
- Multiplexage temporel
- Marquage des flux : MPLS, ATM, 802.1q
- Réseau : IPSec
- Transport : TLS
- Application : SSH, relais HTTP inverse sécurisé¹

Un VPN (Virtual Private Network) ou réseau privé virtuel permet deux choses :

- Constituer un groupe fermé de machines (réseau privé).
- Étendre ce réseau bien au delà d'une stricte localisation géographique comme un bâtiment. Ce qui repose sur l'utilisation de tunnels.

Dans ce qui suit nous nous intéresserons aux VPN et tunnels sur un réseau Ethernet. Nous étendrons nos propos à la façon de sécuriser un réseau Ethernet en le cloisonnant.

Rappels concernant Ethernet

Historique

Ethernet a été développé dans les années 1970 dans les laboratoires de Xerox. Depuis il a connu bien des évolutions :

- débits
 - 10Mbit/s à l'origine
 - 100Mbit/s
 - 1Gbit/s
 - 10Gbit/s depuis peu
- média
 - coaxial + prises vampire
 - coaxial + prises BNC
 - paires torsadées
 - fibres optiques
 - sans fil
- topologies du câblage
 - câble serpentant de machines en machines
 - câblage en étoile
- méthodes d'accès
 - bus partagé
 - commutation
- étendue du réseau
 - bâtiment
 - campus
 - métropole

De fait aujourd'hui Ethernet règne en maître pour tout ce qui concerne la périphérie du réseau c'est à dire la connexion des machines. Les autres protocoles ne subsistant qu'au niveau du cœur du

¹ Les accès au service Web se font par l'intermédiaire d'un relais HTTP inverse. Ce relais est sécurisée (chiffrement, authentification) en utilisant le protocole HTTPS. Il transmet les requêtes au serveur Web et en reçoit les réponses. Ce dernier serveur Web n'a pas nécessairement à être sécurisé par HTTPS.

réseau, ce qui est géré par les opérateurs. Dans tout ce qui va suivre on s'intéressera uniquement à Ethernet même si certains points s'appliquent aussi à d'autres protocoles.

Format des trames

Une trame Ethernet, en ne tenant pas compte des signaux qui servent à la délimiter, présente la structure suivante :

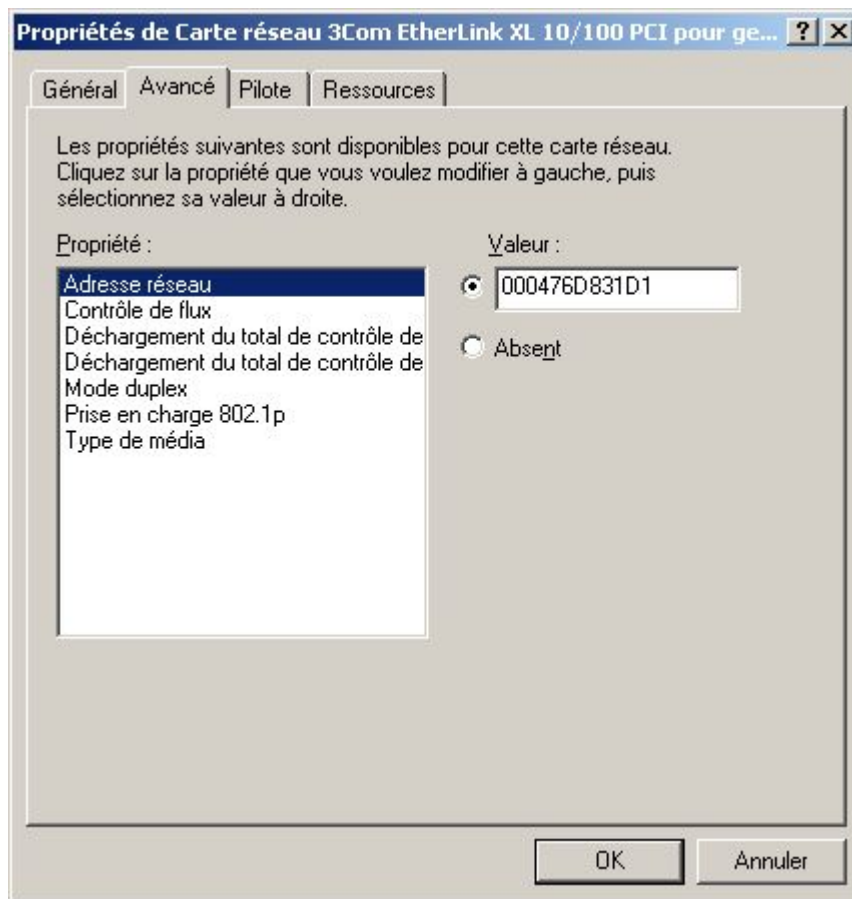
@ destination	@ source	type	données	CRC
---------------	----------	------	---------	-----

@ destination et @ source sont respectivement les adresses des matériels émetteur et destinataire. Une adresse Ethernet comprend 6 octets. Les 3 premiers correspondent au fabricant du matériel, les 3 derniers sont attribués par celui-ci de telle façon qu'il ne puisse y avoir au monde 2 machines portant la même adresse.

Cependant il n'y a aucune difficulté à attribuer à une interface n'importe quelle adresse. Par exemple sous Linux avec la commande :

```
ifconfig eth0 hw ether 00:04:76:D8:31:D1
```

Sous Windows avec certains pilotes de carte réseau il est possible de définir l'adresse MAC très simplement :



Ce qui comme nous le verrons par la suite a d'importantes implications sur la sécurité. Une adresse destination constituée uniquement de bits positionnés à 1 (FF:FF:FF:FF:FF:FF) est l'adresse de diffusion (« broadcast ») indiquant que la trame est destinée à tous les matériels connectés sur le réseau Ethernet. Plus généralement une adresse dont le premier octet est impair, est une adresse « multicast » destinée à un groupe de machines. Le fait que l'adresse destination soit en tête de la trame permet à l'électronique d'une interface Ethernet de rapidement reconnaître si le paquet lui est destiné ou non.

Le champ type a une longueur de 2 octets et indique dans le cas d'Ethernet le numéro du protocole au-dessus. Par exemple :

- 0x0800 : IP
- 0x0806 : ARP

Dans le cas du protocole 802.3 il s'agit de la longueur des données. Pratiquement ce dernier protocole est relativement peu employé face à Ethernet. Cependant Appletalk, IPX (NetWare) sont basés sur 802.3. La longueur des données étant limitée à 1500, si le champ a une valeur supérieure à 1500 il s'agit d'Ethernet et du type.

Les données dépendent du protocole de la couche située au-dessus, IP par exemple. Cependant comme une trame a une longueur minimale de 64 octets, les données peuvent être complétées par des octets de bourrage (« padding ») pour atteindre cette valeur.

Le CRC (Cyclic Redundancy Code) est utilisé pour contrôler l'absence d'erreur dans la transmission. Attention il s'agit uniquement de se prémunir contre des problèmes d'électronique, de parasites dans les signaux. Il n'a pas été conçu pour assurer la non-corruption d'une communication par des personnes malveillantes comme le fait IPsec avec AH (Authenticated Header).

Une trame Ethernet a une longueur maximale de 1518 octets ce qui laisse une taille maximale de 1500 pour les données (c'est l'explication de la valeur 1500 fréquemment employée pour le MTU sous IP).

La commutation

Ethernet a été basé sur une architecture de bus. A un moment donné seul un couple de machines (l'expéditeur et le destinataire) participe aux échanges. Toutes les machines ont donc à se partager la bande passante. L'arrivée d'un câblage en étoile autour d'éléments actifs appelés « hubs » n'a pas fondamentalement changé la donne. Le « hub » réémet un paquet reçu sur un port vers tous les autres.

En terme de câblage un commutateur se présente comme un « hub ». Son électronique est plus élaborée. Il maintient une table associant adresse MAC d'une machine et port sur lequel cette machine est connectée. Lorsqu'il reçoit un paquet sur un port, il analyse les adresses de l'expéditeur et du destinataire. Si l'adresse du destinataire est dans la table le commutateur sait sur quel port se trouve la machine et n'envoie le paquet qu'à travers ce port. Si l'adresse n'est pas dans la table ou bien s'il s'agit d'une adresse de diffusion (« broadcast »), le paquet est envoyé à travers tous les ports du commutateur. La réception d'un paquet permet de mettre à jour la table avec l'adresse de la machine expéditrice. Grâce à cet auto-apprentissage, la table est rapidement à jour et peu de paquets ont à être diffusés à l'ensemble des machines.

Pratiquement chaque couple de machines peut échanger des informations indépendamment des autres ce qui augmente de façon considérable la bande passante globale du réseau. Le fait que le câblage utilise généralement (ce n'est plus vrai pour le 1Gbit/s sur 4 paires torsadées) un support différent (fibre optique, paire torsadée) pour l'émission et la réception permet d'avoir des communications bidirectionnelles (« full duplex ») et améliore encore le débit.

Un des effets induits de l'introduction de la commutation est l'amélioration de la sécurité. En effet le trafic entre deux machines n'est pas visible des autres machines ce qui en assure la confidentialité. Évidemment les paquets avec une adresse de diffusion (« broadcast ») sont visibles par toutes les machines, mais n'est-ce pas le but ? Par ailleurs il faut attendre la mise à jour de la table et les premiers paquets peuvent être envoyés à l'ensemble des machines. Comme généralement, ce sont les paquets qui servent à l'établissement d'une connexion entre machines il n'y a pas encore réellement d'informations sensibles.

Une évolution de la commutation a consisté à introduire des réseaux locaux privés virtuels (« Virtual Local Area Network » ou VLAN). Avec l'augmentation du nombre de machines le trafic en mode diffusion (« broadcast ») prend une importance de plus en plus grande. Il faut dire que certains protocoles comme Netbios utilisent beaucoup ce mode y compris pour s'adresser à un seul

interlocuteur. Aussi on a eu l'idée de segmenter le réseau, regrouper les machines qui ont à communiquer ensemble dans des réseaux virtuels. Ceci est généralement réalisé en affectant les ports d'un commutateur à l'un ou l'autre VLAN. On s'affranchit ainsi des contraintes physiques du câblage, il suffit de modifier la configuration du commutateur pour déplacer logiquement une machine d'un réseau virtuel à un autre. Afin de pouvoir étendre les VLAN au delà d'un seul commutateur, il a été défini un protocole le 802.1q² permettant de faire transiter l'information concernant les VLAN sur les liens entre commutateurs.

Extensions 801.1q (VLAN) et 802.1p (qualité de service)

Les normes 801.1q et 802.1p introduisent un champ supplémentaire dans la trame Ethernet qui présente alors la structure suivante :

@ destination	@ source	0x8100	tag	type	données	CRC
---------------	----------	--------	-----	------	---------	-----

Le nouveau champ extension a une longueur de 4 octets, ce qui porte la longueur maximale de la trame à 1522 octets. Ces 2 premiers octets ont la valeur 0x8100 qui n'est jamais utilisé comme type, ce qui permet de distinguer une trame 802.1q ou 802.1p des autres. Les deux octets suivants sont une étiquette (« tag ») qui sert à spécifier le numéro de VLAN et des informations destinées à gérer une certaine qualité de service. Ces informations sont codées de la façon suivante :

Priorité	CFI	VLAN ID
----------	-----	---------

Priorité a une longueur de 3 bits et permet de définir 8 niveaux de priorité dans les flux. La façon dont est gérée cette qualité de service est définie dans la norme 802.1p.

CFI a une longueur de 1 bit et n'est pas utilisé pour Ethernet (il l'est pour Token Ring). Nous n'entrerons donc pas en détail sur sa signification.

VLAN ID a une longueur de 12 bits et représente le numéro de VLAN de 1 à 4095³. 0 indiquant l'absence de VLAN.

Une autre façon de considérer ce nouveau format de trame est de le faire comme une trame Ethernet classique dont le type est 0x8100 et les données sont constituées de la séquence suivante :
tag + type + anciennes données

Cela permet de comprendre comment un matériel ignorant l'extension à la norme⁴ est capable de transmettre les trames étendues (sans en exploiter évidemment les informations de VLAN ou de qualité de service).

La conversion entre trames classiques et trames étendues se fait au niveau d'un commutateur. Celui ci ajoute ou supprime selon le cas l'étiquette (« tag ») et recalcule le CRC.

Nous pouvons dire qu'un VLAN est un VPN. Le marquage 802.1q des trames circulant entre deux matériels permet de réaliser un tunnel.

Si les trames marquées (802.1q ou/et 802.1p) ont d'abord été prévues pour assurer le transport de VLAN sur les liaisons entre commutateur, elles peuvent aussi être utilisées pour la connexion d'une machine à un commutateur. Il suffit que celle-ci sache gérer les protocoles correspondants. Dans ce cas le commutateur n'a plus à ajouter ou supprimer un en-tête, lorsqu'il reçoit ou envoie une trame. Par contre, il est alors impératif que le commutateur s'assure que la politique de sécurité et de qualité de service est bien respectée. Il doit vérifier le numéro de VLAN et la priorité fournis par la machine et le cas échéant refuser la trame ou modifier son en-tête pour imposer le respect des règles.

Tunnel 802.1q

Il est possible d'avoir un double niveau d'étiquette comme suit :

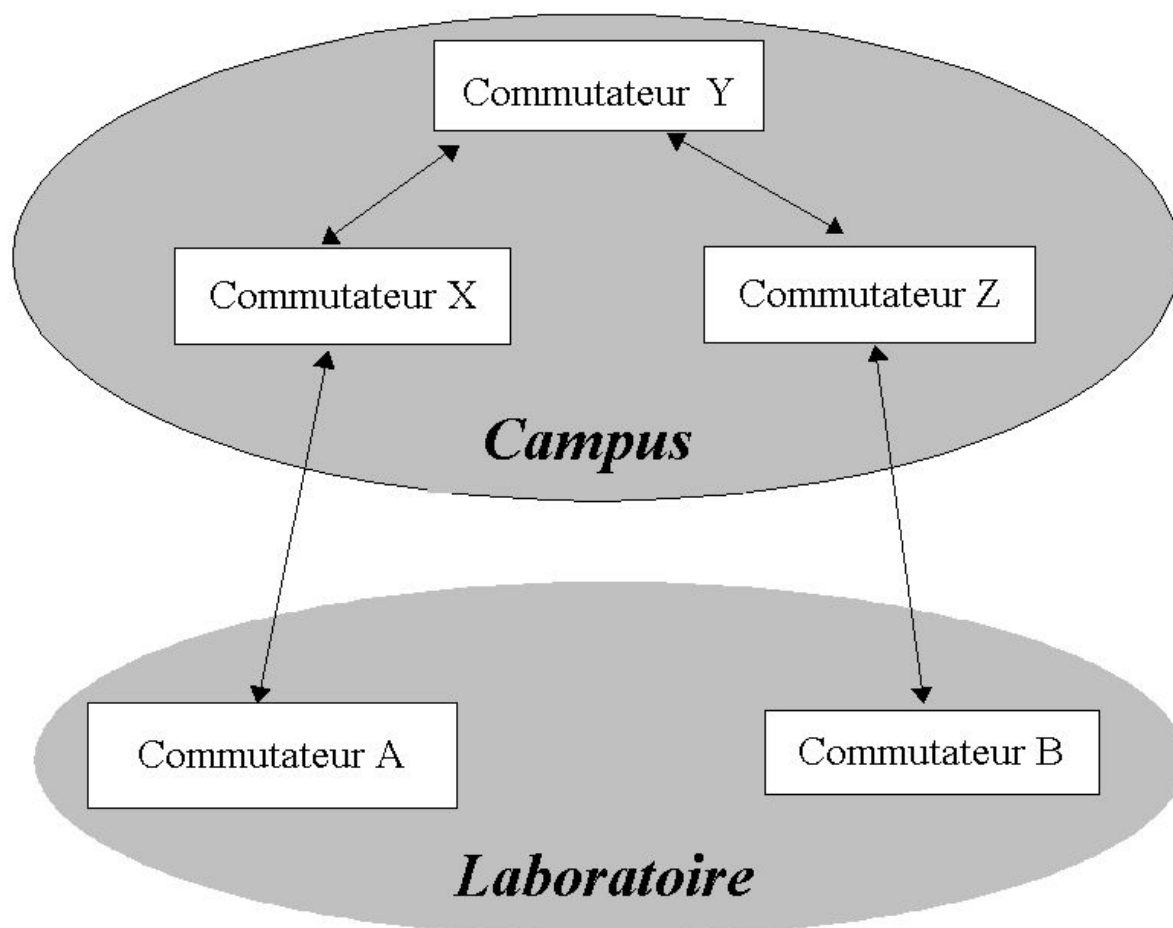
² A vrai dire il existe aussi des protocoles propriétaires comme ISL pour Cisco.

³ Il faut noter que certains matériels ne peuvent gérer qu'un nombre beaucoup plus limité de VLAN.

⁴ Il faut toutefois qu'il puisse accepter des trames d'une longueur supérieure à la normale (1522 au lieu de 1518).

@ destination	@ source	0x8100	Tag1	0x8100	Tag2	type	données	CRC
---------------	----------	--------	------	--------	------	------	---------	-----

Cela peut être utilisé pour réaliser un tunnel à travers un réseau local d'interconnexion pour relier deux parties d'un réseau local. Considérons un laboratoire situé en deux emplacements distincts sur un campus. Ce laboratoire possède ses propres commutateurs et utilise un certain nombre de VLAN. Il souhaite prolonger ces VLAN sur les deux implantations mais il n'a pas la possibilité d'établir une liaison directe (en utilisant une fibre optique par exemple) seulement de se connecter en deux points au réseau local du campus. Voici le schéma d'une architecture possible :



Les commutateurs X, Y, Z appartiennent au réseau du campus, A et B à celui du laboratoire. X reçoit de A les trames étiquetées (tag2) avec les numéros de VLAN du laboratoire. Il y ajoute une nouvelle étiquette (tag1) portant le numéro de VLAN réservé à l'interconnexion du laboratoire. Les trames doublement étiquetées sont envoyées à Y. Y ne considère que l'étiquette extérieure (tag1), celle contenant le numéro de VLAN réservé pour l'interconnexion du laboratoire et transmet à Z les trames reçues. Z supprime l'étiquette extérieure (tag1) avant de transmettre à B les trames reçues. B reçoit des trames avec un seul niveau d'étiquette contenant les numéros de VLAN du laboratoire. Tout se passe comme si les trames provenaient directement de A.

Il n'y a ainsi aucun conflit entre les numéros de VLAN du campus et ceux du laboratoire. On peut utiliser un même numéro de VLAN pour le laboratoire et le campus ou entre deux laboratoires. En l'absence de tunnel il faudrait se mettre d'accord entre les différents réseaux pour l'attribution

des numéros de VLAN. Ce n'est pas impossible mais devient vite pénible avec la multiplication des réseaux et ceci d'autant plus que le nombre relativement limité 4095 des VLAN oblige à une gestion fine.

En outre le tunnel permet de faire transiter correctement et sans interférence les protocoles de niveau 2 qui servent à l'administration du réseau comme STP (Spanning Tree Protocol), VTP (VLAN Trunk Protocol) ou CDP (Cisco Discovery Protocol), les deux derniers étant des protocoles propriétaire de Cisco.

Les commutateurs A et B du réseau du laboratoire sont configurés de la façon habituelle comme s'il y avait une connexion directe entre eux.

Les commutateurs X, Y et Z sont configurés normalement pour faire transiter le VLAN réservé à l'interconnexion du laboratoire. Les seules particularités concernent le port de X en face de A et celui de Z en face de B.

Ce type de tunnel n'offre aucune confidentialité car il n'y a pas de chiffrement. Il n'y a pas non plus de contrôle de l'intégrité des informations transférées ce qui ouvre la porte à des attaques du type entremetteur⁵. Par contre le traitement nécessaire à la gestion de ce tunnel est insignifiant (4 octets à insérer ou à supprimer) ce qui permet d'avoir des débits très élevés. Faire du chiffrement à 1 Gbit/s exige de coûteux circuits spécialisés. Si on fait confiance au réseau d'interconnexion et il n'y a aucune raison de ne pas faire confiance à un réseau de campus correctement administré cela est une excellente solution.

Cependant il ne faut pas chercher à tout prix à utiliser des VLAN. Lorsque cela est possible, pour des raisons de performance, de sécurité et de fiabilité, il est très largement préférable d'utiliser une liaison directe.

VLAN par adresse MAC, VMPS

Comme nous l'avons vu précédemment l'implémentation classique des VLAN est celle qui associe à un port donné sur un commutateur un numéro de VLAN. La gestion manque cependant de souplesse, il faut parfaitement connaître le câblage, l'implantation des prises dans les locaux. A chaque installation de nouvelles machines ou simplement à chaque déplacement d'une machine existante, il faut modifier la configuration du commutateur. C'est une opération qui est loin d'être anodine. En effet il faut se connecter sur le commutateur, lancer les commandes pour mettre à jour la configuration et ne pas oublier de sauvegarder la nouvelle configuration dans la mémoire non volatile qui sera utilisée en cas de redémarrage.

Ce serait bien plus simple si on pouvait définir dans un simple fichier la correspondance entre machines et numéros de VLAN. Nous allons décrire une réalisation possible sur du matériel Cisco à l'aide d'un protocole propriétaire VQP (VLAN Query Protocol). Ce protocole assure les échanges entre le commutateur et un serveur VMPS (VLAN Management Policy Server). Le principe en est le suivant :

- La machine est connectée au commutateur, elle est mise sous tension, son interface Ethernet est activée.
- Différents paramètres comme le débit, half ou full duplex, le spanning tree sont négociés entre le commutateur et la machine.
- Une fois la liaison établie (link) le commutateur attend une trame provenant de la machine.
- Il récupère l'adresse source de la trame Ethernet.
- Il envoie cette adresse ainsi que le numéro du port à un serveur VMPS qui soit lui retourne le numéro de VLAN à utiliser, soit lui demande de refuser de connexion.
- Il affecte ce numéro au port considéré.

⁵ Parfois appelé en anglais « Man in the Middle ».

- Périodiquement le commutateur fait reconfirmer par le serveur le numéro de VLAN en fonction de l'adresse MAC qui arrive sur le port considéré.
- L'interface Ethernet de la machine est désactivée, la liaison (link) est coupée. Le port du commutateur bascule dans un état inactif.

Évidemment pour que cela puisse fonctionner correctement, il faut qu'il n'y ait qu'une seule machine par port ou à la rigueur que toutes les machines appartiennent au même VLAN.

Nous ne décrivons pas VQP, le protocole employé, car nous n'avons pas trouvé de document donnant ses spécifications. De toute façon pour un protocole propriétaire et devenant obsolète vis à vis de 802.1x cela présente un intérêt assez limité. Nous en résumerons les quelques caractéristiques :

- Il utilise le port UDP 1589.
- Il ne comporte aucun mécanisme de sécurisation.

Mise en œuvre de VMPS

Commutateur

Il faut définir les serveurs VMPS utilisés :

```
vmps server 10.254.254.1 primary
vmps server 10.254.254.2
```

Les commutateurs envoyant dans leurs requêtes au serveur VMPS un nom de domaine qui est celui utilisé par VTP (VLAN Trunking Protocol) il faut éventuellement le définir :

```
vtp domain JRES
```

Chaque interface gérant des VLAN dynamiques doit être déclarée comme suit :

```
interface FastEthernet0/1
switchport mode access
switchport access vlan dynamic
```

Serveur VMPS

Le serveur VMPS est disponible sur différents matériels Cisco comme les Catalyst 5000 ou 6000.

Il existe une implémentation libre openVMPS d'un serveur VMPS⁶. C'est celle que nous utilisons.

Un exemple de fichier de configuration est donné ci-dessous. Il s'agit essentiellement d'une table de correspondance entre adresse MAC et adresse VLAN.

```
!vmps domain <domain-name>
! The VMPS domain must be defined.
vmps domain JRES
!vmps mode { open | secure }
! The default mode is open.
!vmps fallback <vlan-name>
!vmps no-domain-req { allow | deny }
! The default value is allow.
vmps mode open
vmps fallback VLAN0601
vmps no-domain-req deny
!
!
!MAC Addresses
!
vmps-mac-addr
! address <addr> vlan-name <vlan_name>
! [speed {10 | 100 | auto}]
! [duplex {half|full}]
address 0005.02b3.18e9 vlan-name VLAN0605
address 0000.015b.b800 vlan-name VLAN0605
address 0007.e96c.ca60 vlan-name VLAN0606
address 0030.65d4.bb6a vlan-name VLAN0607
```

Il est possible, en outre, de définir des règles permettant de restreindre les ports autorisés pour un

⁶ En l'absence de spécifications publiques, il s'agit d'ingénierie inverse à partir des résultats d'un analyseur de réseau.

VLAN donné.

Dans cet exemple si l'adresse MAC n'est définie dans le fichier de configuration le serveur attribue le VLAN VLAN0601. Cela peut être intéressant si on souhaite avoir un VLAN ayant des droits très restreints pour tous les inconnus.

Station de travail

Il n'y a rien de spécifique à mettre en œuvre sur les stations de travail.

Protocole 802.1x

Le nouveau standard 802.1x est une réponse au besoin d'authentifier les machines ou les utilisateurs connectés sur un réseau local. Il a été défini avant l'arrivée du Wi-Fi. Aujourd'hui les trop nombreuses faiblesses dans la sécurité du Wi-Fi en limitent son déploiement. Afin de conserver leur business, les différents acteurs du marché ont poussé au développement de mesures permettant la sécurisation du Wi-Fi. Ces mesures impliquent, en autres choses, l'utilisation du standard 802.1x. Désormais on associe généralement le 802.1x au Wi-Fi, mais on oublie trop souvent que ce standard peut aussi être utilisé pour sécuriser les connexions Ethernet câblées.

Le protocole 802.1x permet d'authentifier les éléments connectés sur le réseau. Il offre aussi un mécanisme pour échanger les clés qui vont être utilisées pour chiffrer les communications et en contrôler l'intégrité ce qui est particulièrement important pour le Wi-Fi.

Le protocole 802.1x définit 3 catégories d'acteurs jouant chacun un rôle différent. Ce sont le requérant (« supplicanant »), le certificateur⁷ (« authenticator ») et le serveur d'authentification (« authentication server »).

- Le supplicanant est le poste travail demandant à accéder au réseau.
- L'authenticator est le dispositif, commutateur ou borne d'accès Wi-Fi, fournissant la connexion au réseau. Un port sur ce dispositif peut avoir 2 états :
 - Non autorisé. Tant que le client n'a pas été authentifié le port reste dans l'état non autorisé. Le seul le trafic permis est alors celui entre le requérant et l'authenticator afin de pouvoir effectuer l'authentification.
 - Autorisé. Après une authentification fructueuse le port bascule dans l'état autorisé. La station de travail est alors autorisée à avoir un accès complet au réseau.
 - L'authenticator n'effectue aucune authentification, il se contente de jouer le rôle de relais pour transmettre les messages d'authentification en provenance du supplicanant vers le serveur d'authentification et réciproquement.
 - Si le serveur d'authentification valide la demande alors le port est commuté dans l'état autorisé et alors la station de travail est autorisée à avoir un accès complet au réseau.
- Le serveur d'authentification. Il s'agit d'une machine implémentant un serveur RADIUS. C'est lui qui va authentifier le supplicanant. En fonction du résultat de l'authentification ce serveur va envoyer à l'authenticator un message comme quoi ce dernier peut autoriser ou non le port.

Par abus de langage et pour mieux correspondre à la réalité des matériels utilisés, nous appellerons parfois, par la suite, supplicanant, authenticator et serveur d'authentification respectivement client ou station de travail, commutateur et serveur.

Il faut noter un certain nombre de points dans cette architecture :

- L'authenticator ne gère pas directement la tâche complexe de l'authentification, il la délègue à un serveur. Il se limite à un rôle de relais. C'est donc relativement facile à intégrer à un élément de réseau comme un commutateur ou une borne d'accès Wi-Fi.
- Le serveur RADIUS a été initialement conçu pour authentifier des connexions par modem

⁷ Les termes requérant et certificateur font partie du vocabulaire juridique. Mais comme ils restent trop loin des termes d'origine (suppliant est vraiment inapproprié et « authenticateur » n'existe pas) nous emploierons par la suite les mots anglais pour faciliter la compréhension.

(PPP).

- Le serveur RADIUS outre le résultat de l'authentification peut envoyer à l'authenticator des informations supplémentaires comme un numéro de VLAN à utiliser, des clés WEP pour le Wi-Fi, des règles filtrage (ACL) à appliquer. Il reste qu'il s'agit très souvent d'extensions propriétaires.
- Les méthodes d'authentification ne sont pas fixées par la norme qui ne définit qu'un mécanisme d'échange des messages d'authentification. Cela offre une grande souplesse mais pose aussi de sérieux problèmes d'interopérabilité.
- Si on veut évaluer la qualité globale de l'authentification il ne faut pas s'intéresser uniquement au protocole 802.1x stricto sensu ou à EAP qui reste relativement simple mais aussi aux protocoles RADIUS, TLS et surtout à leurs implémentations. On a affaire à des protocoles complexes dont la validité est dure à montrer et dont la réalisation est nécessairement accompagnée de bogues.
- Pour des raisons de disponibilité le serveur d'authentification doit être redondant, au minimum 2 machines. En effet sans serveur disponible aucune machine ne peut accéder au réseau. Pour équilibrer la charge entre ces différents serveurs, il suffit de ne pas déclarer sur tous les authenticators la même machine comme serveur primaire mais au contraire de les répartir.

A notre connaissance les systèmes d'exploitation implémentant le standard 802.1x (dans le rôle supplicant) sont actuellement :

- Windows 2000 SP4, Windows XP, Windows 2003
- MacOS X, FreeBSD, OpenBSD, Linux avec open1x

La majorité des constructeurs de matériels de réseau proposent désormais des modèles de commutateurs et de bornes d'accès Wi-Fi supportant le protocole 802.1x.

Protocole EAP

Le protocole utilisé pour assurer l'authentification est EAP (Extensible Authentication Protocol) défini dans le RFC 2284. Il a été développé à l'origine pour PPP. Ce n'est pas, en soi, un protocole d'authentification, seulement un transport optimisé des informations nécessaires à l'authentification.

Le format d'un paquet est le suivant⁸ :

Code	Identifiant	Length	Data
------	-------------	--------	------

- Code a une taille de 1 octet et définit le code du paquet
 - 1 Request
 - 2 Response
 - 3 Success
 - 4 Failure
- Identifiant a une taille de 1 octet et permet d'associer les requêtes avec les réponses.
- Length a une taille de 2 octets et définit la longueur totale du paquet incluant code, identifiant, length et data.
- Data : zéro ou plusieurs octets de données dont le format dépend du type de paquet.

Les paquets Request et Response ont le format suivant :

Code	Identifiant	Length	Type	Type-Data
------	-------------	--------	------	-----------

- Code a une taille de 1 octet et définit le code du paquet
 - 1 Request
 - 2 Response
- Identifiant a une taille de 1 octet et permet d'associer les requêtes avec les réponses.
- Length a une taille de 2 octets et définit la longueur totale du paquet incluant code, identifiant, length, type et type-data.
- Type a une taille de 1 octet et indique le type de la requête ou de la réponse.
- Type-Data : le format varie en fonction du type de requête et de la réponse associée.

⁸ Nous préférons pour éviter des confusions ne pas traduire les termes employés dans les différentes normes.

Les paquets Success et Failure ont le format suivant :

Code	Identifiant	Length
------	-------------	--------

- Code a une taille de 1 octet et définit le code du paquet
 - 3 Success
 - 4 Failure
- Identifier a une taille de 1 octet et permet d'associer les requêtes avec les réponses.
- Length a une taille de 2 octets et a la valeur 4 puisqu'il n'y a pas de données.

Un certain nombre de types pour les paquets request et response ont été initialement définis dans le RFC mais il est prévu de pouvoir les étendre.

- 1 Identity. Permet de demander l'identité de l'interlocuteur (request) et la fournit (response).
- 2 Notification
- 3 Nak (Response seulement)
- 4 MD5 Challenge
- 5 One Time Password (OTP) (RFC 1938)
- 6 Generic Token Card
- 13 TLS (extension non définie dans le RFC 2284)

Le standard 802.1x définit entre autres choses EAPOL une méthode d'encapsulation d'EAP sur un réseau local. C'est ce qui est utilisé pour le dialogue entre le supplicatif et l'authenticator tant que le port reste dans l'état unauthorized. Pour Ethernet on utilise des paquets ayant un type de 0x888e. Pour dialoguer avec l'authenticator on utilise l'adresse 01:80:C2:00:00:03. Elle fait partie d'un groupe d'adresses réservées ne devant jamais être retransmise par un pont ou commutateur. Une trame EAPOL se présente sous la forme suivante en ignorant les adresses MAC :

0x888E Protocol Version Packet Type Packet Body Length Packet Body

- 0x888E est le type Ethernet pour EAPOL
- Protocol Version a une taille de 1 octet et vaut aujourd'hui 1
- Packet Type a une taille de 1 octet et peut prendre les valeurs suivantes :
 - 0 EAP-Packet. Pour encapsuler un paquet EAP.
 - 1 EAPOL-Start. Permet au supplicatif de démarrer une séquence d'authentification.
 - 2 EAPOL-Logoff. Lorsque le supplicatif envoie ce paquet l'authenticator bascule immédiatement le port dans l'état unauthorized.
 - 3 EAPOL-Key. Prévu⁹ pour la gestion des clés de chiffrement.
 - 4 EAPOL-Encapsulated-ASF-Alert. Permet de transmettre des alertes (SNMP traps) lorsque le port est dans l'état unauthorized.
- Packet Body Length a une taille de 2 octets et définit la longueur en octets du champ Packet Body.
- Packet Body est présent uniquement pour les types EAP-Packet, EAPOL-Key ou EAPOL-Encapsulated-ASF-Alert et contient les données encapsulées.

Protocole RADIUS

Le protocole RADIUS assurant les échanges entre l'authenticator et le serveur d'authentification est décrit dans les RFC 2865, 2866, 2867, 2868, 2869, 3162. Il s'agit d'un protocole UDP utilisant les ports 1812 pour l'authentification et 1813 pour la comptabilité. Le format d'un paquet RADIUS est le suivant :

Code	Identifiant	Length	Authenticator	Attributes
------	-------------	--------	---------------	------------

- Code a une taille de 1 octet et identifie le type de paquet :
 - 1 Access-Request. Du client vers le serveur pour demander l'authentification et l'autorisation d'une connexion.
 - 2 Access-Accept. Du serveur vers le client en réponse à un message Access-Request

⁹ On verra par la suite que ce type de paquet n'est pas utilisé par le Wi-Fi mais a été remplacé par un autre. (254)

- pour indiquer que la connexion est authentifiée et autorisée.
- 3 Access-Reject. Du serveur vers le client en réponse à un message Access-Request pour indiquer que la connexion est refusée.
- 4 Accounting-Request
- 5 Accounting-Response
- 11 Access-Challenge. Du serveur vers le client en réponse à un message Access-Request pour envoyer un défi au client qui devra répondre.
- 12 Status-Server
- 13 Status-Client
- 255 Réserve
- Identifier a une taille de 1 octet et permet d'associer requêtes et réponses.
- Authenticator a une taille de 16 octets. Ce champ est utilisé pour authentifier et contrôler l'intégrité des messages RADIUS transmis. Pour simplifier dans une requête il s'agit d'une valeur non prévisible (aléatoire) et globalement unique. Dans la réponse associée c'est une empreinte (hash) de la requête + la réponse + le secret partagé. L'émetteur est alors sûr que la réponse provient du bon interlocuteur (il connaît le secret) et que ni la requête ni la réponse n'ont été altérées.
- Attributes. Une liste d'attributs dont la fin est indiquée par la longueur du paquet.

Un attribut a le format suivant

Type	Length	Value
------	--------	-------

- Type a une longueur de 1 octet et définit l'attribut. Exemples :
 - 1 User-Name
 - 2 User-Password
 - 3 CHAP-Password
 - 79 EAP-Message
 - 80 Message-Authenticator
- Length a une longueur de 1 octet et indique la taille en incluant les champs type, length et value.
- Value. Contient les informations relatives à l'attribut. Le format et la taille de la valeur sont déterminés par les champs type et length. Il y a 5 formats possibles :
 - Text : 1-253 octets contenant une chaîne de caractères encodés en UTF-8 (ISO 10646)
 - String : 1-253 octets contenant des données binaires.
 - Address : 32 bits
 - Integer : entier 32 bits non signé
 - Time : nombre de secondes (32 bits) depuis le 01/01/1970 00:00:00

Les messages EAP en provenance du supplicand sont encapsulés par l'authenticator dans un message RADIUS de type EAP-Message (79) pour être transmis au serveur d'authentification.

L'attribut Message-Authenticator est une empreinte MD5 de l'ensemble du message en utilisant le secret partagé comme clé. Cet attribut a été ajouté par la suite, afin d'améliorer la sécurité du protocole RADIUS. Cela permet de se prémunir contre un client pirate effectuant une attaque par dictionnaire contre le serveur RADIUS. Mais ne protège pas contre une attaque par dictionnaire hors ligne par un pirate qui aurait récupéré le défi et la réponse.

On peut légitimement s'interroger si les différentes mesures qui ont été prises pour sécuriser le protocole RADIUS sont suffisantes. Dans le cas du 802.1x il faut cependant très largement tempérer ces craintes par les faits suivants :

- Dans une architecture bien conçue les serveurs RADIUS, les commutateurs ou bornes d'accès sans fil sont connectés sur un réseau isolé (VLAN spécifique).
- Avec les méthodes d'authentification EAP-TLS ou EAP-TTLS le protocole RADIUS ne sert qu'à transporter du TLS et c'est ce dernier qui assure la sécurité de bout en bout.

Méthodes d'authentification EAP

Il existe différentes méthodes d'authentification pour EAP :

- EAP-MD5. C'est la plus simple. Le client est authentifié par le serveur en utilisant un mécanisme de défi réponse. C'est à dire le serveur envoie une valeur aléatoire (le défi), le client concatène à ce défi le mot de passe et en calcule, en utilisant l'algorithme MD5, une empreinte (« hash ») qu'il renvoie au serveur. Le serveur qui connaît le mot de passe calcule sa propre empreinte, compare les deux et en fonction du résultat valide ou non l'authentification. Une écoute du trafic peut dans le cas d'un mot de passe mal choisi permettre de le retrouver par une attaque par dictionnaire ou par force brute. Il n'y a pas d'authentification mutuelle, le serveur n'est pas authentifié par le client. La future norme 802.11i pour sécuriser le Wi-Fi impose une authentification mutuelle. Il ne contient pas non plus de « nonce¹⁰ » adéquat pour dériver les clés de session permettant de sécuriser le trafic sans fil (WEP). Cette méthode est à exclure totalement avec le Wi-Fi.
- EAP-TLS. C'est la plus sûre. Le serveur et le client possèdent chacun leur certificat qui va servir à les authentifier mutuellement. Cela reste relativement contraignant du fait de la nécessité de déployer une IGC (Infrastructure de Gestion de Clés). Rappelons que TLS, la version normalisée par l'IETF de SSL (Secure Socket Layer), est un transport sécurisé (chiffrement, authentification mutuelle, contrôle d'intégrité). C'est lui qui est utilisé de façon sous-jacente par HTTPS, la version sécurisée de HTTP, pour sécuriser le Web. Il faut noter cependant que l'identité de l'utilisateur (paquet EAP-Response/Identity) n'est pas protégée.
- EAP-TTLS (tunneled TLS) utilise TLS comme un tunnel pour échanger des couples attribut-valeur à la manière de RADIUS¹¹ servant à l'authentification. Pratiquement n'importe quelle méthode d'authentification peut être utilisée.
- PEAP (Protected EAP) est une méthode très semblable dans ses objectifs et voisine dans la réalisation à EAP-TTLS. Elle est développée par Microsoft. Elle se sert d'un tunnel TLS pour faire circuler de l'EAP. On peut alors utiliser toutes les méthodes d'authentification supportées par EAP.
- EAP-SIM utilise la carte SIM d'un téléphone GSM.
- LEAP (Lightweight EAP) est un méthode propre à Cisco qui repose sur l'utilisation de secrets partagés pour authentifier mutuellement le serveur et le client. Elle n'utilise aucun certificat et est basé sur l'échange de défi et réponse.

Si on veut classer les différentes méthodes, on a de la moins sûre à la plus sûre :

- EAP-MD5. Ne doit absolument pas être utilisé pour le Wi-Fi.
- LEAP
- EAP-TTLS, PEAP
- EAP-TLS

Pour EAP-TTLS et PEAP, le serveur est authentifié par son certificat. Le client est authentifié par l'un des mécanismes classiques (identifiant/mot de passe par exemple). On n'a pas à déployer d'IGC. Il suffit d'acheter un certificat pour le serveur. Il est possible de conserver son mécanisme habituel d'authentification. Il faut noter que l'identité de l'utilisateur est protégée par TLS. Si on veut éviter d'avoir à gérer des certificats client, c'est la méthode à utiliser¹².

EAP-TLS impose l'utilisation d'un certificat, ce qui n'est plus vraiment une contrainte lorsque comme c'est le cas du CNRS on dispose d'une IGC bien rodée. La gestion du serveur d'authentification est alors très largement simplifiée car il n'a plus à stocker les mots de passe des utilisateur, toute l'intendance s'effectue au niveau l'IGC. Cette méthode révèle tout son intérêt avec l'utilisation de certificats sur carte à puce. Comme nous le verrons par la suite, il n'est pas certain

¹⁰ Il ne s'agit pas du légat du pape. Le terme anglais nonce signifie: mot créé pour l'occasion.

¹¹ Le format utilisé pour coder ces paires est en réalité très proche de RADIUS.

¹² Avec une petite préférence pour EAP-TTLS qui est un peu plus souple et aujourd'hui plus répandu. PEAP n'est implémenté que chez Microsoft.

que l'utilisation d'un certificat stocké sur le disque de l'ordinateur offre un réel accroissement de sécurité par rapport à un mot de passe.

Séquence d'authentification

On peut résumer une séquence d'authentification utilisant EAP-MD5 comme méthode d'authentification dans le schéma suivant :

<i>Supplicant <-> authenticator</i>	<i>Authenticator <-> RADIUS</i>
EAPOL-Start ->	
EAP-Request/Identity <-	
EAP-Response/Identity ->	
	RADIUS Access-Request ->
	RADIUS Access-Challenge <-
EAP-Request/MD5 <-	
EAP-Response/MD5 ->	
	RADIUS Access-Request ->
	RADIUS Access-Accept <-
EAP-Success <-	
EAP-Logoff ->	

Voici avec un peu plus de détails comment les choses se déroulent :

1. Le client se connecte et envoie au commutateur une trame EAPOL-Start
2. Le commutateur répond par une demande d'identité EAP-Request/Identity
3. Le client fournit son identité au commutateur qui la transmet au serveur RADIUS dans un paquet Access-Request sous la forme d'un attribut EAP-Message.
4. Le serveur RADIUS consulte ses tables et si l'identité est valide il retourne un paquet EAP-Request/MD5 sous forme d'un attribut EAP-Message d'un paquet RADIUS Access-Challenge.
5. Le commutateur extrait le paquet EAP-Request/MD5 et le transmet au client.
6. Le client à partir du challenge et du mot de passe fourni par l'utilisateur forme une réponse EAP-Response/MD5. Le commutateur transmet ce paquet sous forme d'un attribut EAP-Message d'un paquet RADIUS Access-Request
7. Le serveur vérifie dans ses tables que le client a fourni le bon mot de passe. Il valide alors la demande et envoie un paquet Access-Accept au commutateur.
8. Le commutateur envoie un message EAP-Success au client et bascule le port dans l'état authorized.
9. Plus tard lorsque le client n'a plus besoin de la connexion il envoie au commutateur une trame EAPOL-Logoff. Le commutateur bascule alors le port dans l'état unauthorized.

Avec d'autres méthodes d'authentification comme EAP-TLS les échanges sont un peu plus complexes. Mais répétons le, seuls le client et le serveur sont affectés par cet accroissement de complexité, le commutateur se contentant de relayer les messages. Ce qui explique, comme nous le constaterons par la suite, que les difficultés rencontrées lors du déploiement de 802.1x proviennent essentiellement des clients, du serveur RADIUS et non pas du commutateur.

On peut résumer dans le tableau suivant les différents protocoles utilisés :

Supplicant		Authenticator		Authentication Server
EAP-TLS				
EAP				
EAPOL			RADIUS	
Ethernet	802.3	802.11	UDP	

Mise en œuvre du protocole 802.1x

En ce qui concerne la mise en œuvre du protocole 802.1x et afin de rester concret nous décrivons en détail ce que nous avons fait dans notre environnement. Les principes doivent pouvoir s'appliquer mutatis mutandis à d'autres matériels et logiciels.

Commutateurs

Les commutateurs que nous utilisons sont des Cisco Catalyst 3550 avec une version d'IOS 12.1 (14)EA1. Attention les versions antérieures offrent moins de fonctionnalités en matière de 802.1x et se configurent légèrement différemment.

La première étape consiste à activer l'authentification :

```
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
```

La dernière directive est nécessaire seulement si on veut attribuer un numéro de VLAN ou des ACL par utilisateur.

Attention le fait d'activer «aaa new-model» change le comportement par défaut en ce qui concerne le «login» sur le commutateur ou le passage en mode privilégiée. Les directives suivantes rétablissent le comportement par défaut :

```
aaa authentication login default line
aaa authentication enable default enable
```

La deuxième étape consiste à définir les serveurs radius utilisés :

```
radius-server host 10.254.254.1 auth-port 1812 acct-port 1813 key secret
radius-server host 10.254.254.2 auth-port 1812 acct-port 1813 key secret
```

La valeur du paramètre key est le secret partagé entre le commutateur et le serveur radius. Il sert à sécuriser les communications entre eux. Il faut reporter cette valeur dans la configuration du serveur.

La troisième étape consiste à activer l'authentification 802.1x d'abord globalement sur le commutateur :

```
dot1x system-auth-control
```

et ensuite sur chacun des ports où une authentification 802.1x doit être effectué :

```
interface FastEthernet0/1
switchport mode access
dot1x port-control auto
```

Dans le cas où on utilise les informations fournies par les serveurs radius pour affecter le numéro de VLAN il est possible de définir un VLAN invité qui sera utilisé pour connecter les matériels ne supportant pas le protocole 802.1x. Évidemment il faudra définir par ailleurs un routage et un filtrage pour ce VLAN invité qui donnent des droits réduits.

```
dot1x guest-vlan 601
```

Dans le cas où le numéro de VLAN ne serait pas attribué dynamiquement il faut le définir. Même si le VLAN est défini dynamiquement, pour des raisons de sécurité, il est très vivement conseillé d'affecter le numéro d'un VLAN spécial. En effet tant que le port n'est pas autorisé, le VLAN par défaut soit numéro 1 qui sert à l'administration du commutateur est affecté au port. Certes le port est alors en mode «shutdown» et théoriquement aucun trafic ne peut circuler. Mais il faut se méfier des erreurs toujours possibles lorsque l'on manipule le fichier de configuration, comme de supprimer «dot1x port-control auto» en oubliant de fermer le port ou d'attribuer un numéro de VLAN. Par ailleurs on n'est jamais à l'abri de failles de sécurité dans le commutateur. Donc par

prudence il faut toujours définir un numéro de VLAN :

```
switchport access vlan 600
```

Pour vérifier la configuration du commutateur on peut utiliser les commandes :

```
show dot1x all
```

Pour la configuration définie ci-dessus on obtient :

```
Dot1x Info for interface FastEthernet0/1
```

```
-----  
Supplicant MAC 0004.76d8.31d0  
AuthSM State      = AUTHENTICATED  
BendSM State      = IDLE  
PortStatus        = AUTHORIZED  
MaxReq            = 2  
HostMode          = Single  
Port Control      = Auto  
QuietPeriod       = 60 Seconds  
Re-authentication = Disabled  
ReAuthPeriod      = 3600 Seconds  
ServerTimeout     = 30 Seconds  
SuppTimeout       = 30 Seconds  
TxPeriod          = 30 Seconds  
Guest-Vlan        = 601
```

Serveur Radius

Il existe différentes implémentations du serveur Radius (Remote Authentication Dial-in User Service). Chez Microsoft il s'agit de IAS (Internet Authentication Service).

Dans tout ce qui nous supposera que le serveur Radius utilisé est FreeRadius sur une plateforme Linux. C'est la configuration que nous utilisons. N'ayant pas de serveur RADIUS existant, nous avons choisi cette solution qui possède l'avantage de pas être trop onéreuse à mettre en œuvre, le logiciel est gratuit et une machine disponible pour l'installer se trouve assez facilement. Pour pouvoir utiliser EAP-TLS il faut une version suffisamment récente de freeradius (09.0) et openssl (0.9.7).

EAP-MD5

La configuration par défaut fourni avec la version de freeradius que nous utilisons (0.9.0) permet d'utiliser EAP avec MD5 comme méthode d'authentification sans rien à avoir à modifier dans le fichier de configuration radiusd.conf. L'autre méthode d'authentification EAP-TLS qui permet d'utiliser des certificats est nettement plus complexe à mettre en œuvre. C'est pourquoi, même s'il est largement préférable d'utiliser EAP-TLS, nous commencerons par décrire EAP MD5. Croyez en notre expérience, commencez les essais avec EAP MD5. Le fichier de configuration est plutôt volumineux et assez complexe. Aussi nous en avons extrait les directives portant spécifiquement sur EAP.

```
# MODULE CONFIGURATION  
#  
# The names and configuration of each module is located in this section.  
#  
# After the modules are defined here, they may be referred to by name,  
# in other sections of this configuration file.  
#  
modules {  
    #  
    # Each module has a configuration as follows:  
    #  
    #     name [ instance ] {  
    #         config_item = value  
    #         ...  
    #     }  
    #  
    # The 'name' is used to load the 'rlm_name' library  
    # which implements the functionality of the module.  
    #  
    # The 'instance' is optional. To have two different instances
```

```

# of a module, it first must be referred to by 'name'.
# The different copies of the module are then created by
# inventing two 'instance' names, e.g. 'instance1' and 'instance2'
#
# The instance names can then be used in later configuration
# INSTEAD of the original 'name'. See the 'radutmp' configuration
# below for an example.
#

# Extensible Authentication Protocol
#
# For all EAP related authentications
eap {
    # Invoke the default supported EAP type when
    # EAP-Identity response is received.
    #
    # The incoming EAP messages MAY NOT specify which EAP
    # type they will be using, so it MUST be set here.
    #
    # For now, only one default EAP type may be used at a
    # time.
    #
    default_eap_type = md5

    # Default expiry time to clean the EAP list,
    # It is maintained to correlate the
    # EAP-response for each EAP-request sent.
    timer_expire     = 60

    # Supported EAP-types
    md5 {
    }
}

# Authorization. First preprocess (hints and huntgroups files),
# then realms, and finally look in the "users" file.
#
# The order of the realm modules will determine the order that
# we try to find a matching realm.
#
# Make *sure* that 'preprocess' comes before any realm if you
# need to setup hints for the remote radius server
authorize {
    #
    # This module takes care of EAP-MD5, EAP-TLS, and EAP-LEAP
    # authentication.
    eap
}

# Authentication.
#
# This section lists which modules are available for authentication.
# Note that it does NOT mean 'try each module in order'. It means
# that you have to have a module from the 'authorize' section add
# a configuration attribute 'Auth-Type := FOO'. That authentication type
# is then used to pick the appropriate module from the list below.
#
# The default Auth-Type is Local. That is, whatever is not included inside
# an authtype section will be called only if Auth-Type is set to Local.
#
# So you should do the following:
# - Set Auth-Type to an appropriate value in the authorize modules above.
# - For example, the chap module will set Auth-Type to CHAP, ldap to LDAP, etc.
# - After that create corresponding authtype sections in the
#   authenticate section below and call the appropriate modules.
authenticate {
    #
    # Allow EAP authentication.
    eap
}

```

```

#
# When the server receives a reply to a request it proxied
# to a home server, the request may be massaged here, in the
# post-proxy stage.
#
post-proxy {
    # attr_rewrite

    #
    # If you are proxying LEAP, you MUST configure the EAP
    # module, and you MUST list it here, in the post-proxy
    # stage.
    #
    # You MUST also use the 'nostrip' option in the 'realm'
    # configuration. Otherwise, the User-Name attribute
    # in the proxied request will not match the user name
    # hidden inside of the EAP packet, and the end server will
    # reject the EAP request.
    #
    eap
}

```

EAP-TLS

```

modules {
    # Extensible Authentication Protocol
    #
    # For all EAP related authentications
    eap {
        # Invoke the default supported EAP type when
        # EAP-Identity response is received.
        #
        # The incoming EAP messages MAY NOT specify which EAP
        # type they will be using, so it MUST be set here.
        #
        # For now, only one default EAP type may be used at
        # a time.
        #
        default_eap_type = tls

        # Default expiry time to clean the EAP list,
        # It is maintained to correlate the
        # EAP-response for each EAP-request sent.
        timer_expire      = 60

        # Supported EAP-types
        md5 {
        }

        ## EAP-TLS is highly experimental EAP-Type at the moment.
        # Please give feedback on the mailing list.
        tls {
            private_key_password = secret
            private_key_file = /etc/raddb/certs/radius.pem

            #
            # If Private key & Certificate are located in the
            # same file, then private_key_file &
            # certificate_file
            # must contain the same file name.
            certificate_file = /etc/raddb/certs/radius.pem

            #
            # Trusted Root CA list
            CA_file = /etc/raddb/certs/root.pem

            dh_file = /etc/raddb/certs/dh
            random_file = /dev/urandom

            #
            # This can never exceed MAX_RADIUS_LEN (4096)
            # preferably half the MAX_RADIUS_LEN, to
            # accomodate other attributes in RADIUS packet.
            # On most APs the MAX packet length is configured
            # between 1500 - 1600. In these cases, fragment
            # size should be <= 1024.
        }
    }
}

```

```

#
#           fragment_size = 1024
#           include_length is a flag which is by default
#           set to yes
#           If set to yes, Total Length of the message is
#           included
#           in EVERY packet we send.
#           If set to no, Total Length of the message is
#           included
#           ONLY in the First packet of a fragment series.
#
#           include_length = yes
#
#
}
}
}

```

Authenticator

Il faut déclarer les clients en l'occurrence les commutateurs autorisés à se connecter sur le serveur radius et spécifier le secret partagé pour sécuriser les échanges. Cela s'effectue en ajoutant dans le fichier clients.conf les directives suivantes :

```

# Commutateurs
clients 10.254.254.11 {
    secret    = secret1
    shortname = commutateur1
}
clients 10.254.254.12 {
    secret    = secret2
    shortname = commutateur2
}

```

Si tous les commutateurs ont le même secret et appartiennent au même réseau il est possible de regrouper en :

```

# Commutateurs
clients 10.254.254.0/24 {
    secret    = secret
    shortname = commutateurs
}

```

Attention le secret utilisé est le même que celui qui est déclaré dans la configuration du commutateur (paramètre key) de radius-server.

Il faut ensuite définir dans le fichier users les paramètres d'authentification des différents utilisateurs. Dans le de EAP-MD5 comme méthode d'authentification les informations relatives à un utilisateur donné se présentent sous la forme suivante :

```

martin    Auth-Type := EAP, User-Password == "secret",
          Service-Type == Framed-User,
          Tunnel-Type = VLAN,
          Tunnel-Medium-Type = 6,
          Tunnel-Private-Group-ID = "VLAN0602",
          Cisco-AVPair = "ip:inacl#1=permit ip any 10.1.1.0 0.0.0.255",
          Cisco-AVPair += "ip:inacl#2=deny ip any 10.1.2.0 0.0.0.255",
          Cisco-AVPair += "ip:inacl#3=permit ip any any"

```

La première ligne définit l'utilisateur ici « martin », son mot de passe ici « secret », ainsi que le type d'authentification (EAP). La deuxième définit le type de service, c'est obligatoirement « Framed-User ».

Les trois lignes suivantes ne sont nécessaires que si on affecte dynamiquement le numéro de VLAN à l'utilisateur. Dans ce cas Tunnel-Type vaut toujours VLAN et Tunnel-Medium-Type 6. Tunnel-Medium-Type définit le nom du VLAN ici « VLAN0602 » comme le type associé est une chaîne de caractères il doit obligatoirement être mis entre des guillemets. Cette possibilité de définir un VLAN et la syntaxe associée est, lorsqu'elle existe, a priori indépendante du matériel utilisé.

Les trois dernières lignes sont optionnelles, elles permettent d'associer dynamiquement une ACL au port utilisé pour la connexion de l'utilisateur. Il s'agit d'une fonctionnalité spécifique au matériel

Cisco. Nous ignorons s'il existe une fonctionnalité analogue sur des matériels d'autres fournisseurs. De toute façon la syntaxe utilisée ici est propre à Cisco. Dans l'exemple ci-dessus ip indique qu'il s'agit d'une ACL IP, il existe aussi des ACL MAC. Inacl indique que l'ACL s'applique dans le sens « ingress ». Les différents numéros #1, #2, #3 permettent de définir l'ordre des différentes règles. Attention si pour la première directive Cisco-AVPair l'opérateur d'affectation est « = », il est de « += » pour les suivantes qui signifie que l'on ajoute les informations à la suite des précédentes. Sans entrer dans le détail, pour cela on se reportera aux manuels de Cisco, l'ACL défini ci-dessus permet à la machine de l'utilisateur d'envoyer des paquets IP vers les machines de 10.1.1.0 à 10.1.1.255, aux adresses externes, ainsi qu'à l'adresse de broadcast IP (255.255.255.255) et de rejeter tout paquet IP ayant une autre destination sur le réseau interne. Supposons que dans le réseau les serveurs, la passerelle vers l'extérieur soient dans la plage d'adresse 10.1.1.0/24 et que l'on attribue aux machines clientes des adresses dans la plage 10.1.2.0/24 alors on a un moyen simple de permettre à une machine d'un utilisateur de se connecter à un serveur sur le réseau interne ou à l'extérieur tout en lui interdisant d'échanger des informations avec une machine d'un autre utilisateur. Il faut aussi permettre l'adresse de broadcast, notamment pour le protocole DHCP. Il est évidemment possible d'établir des règles beaucoup plus élaborées. Nous reviendrons par ailleurs sur les politiques de sécurité qu'il est possible de mettre en œuvre.

Pour EAP-TLS comme méthode d'authentification il suffit de supprimer « User-Password == "secret" », l'authentification étant désormais faite au niveau du protocole TLS.

Pour démarrer le serveur pendant la période de mise au point il est conseillé d'utiliser la commande avec une option permettant la sortie de nombreux messages.

```
radiusd -x
```

Si la commande n'est pas dans le PATH il faut spécifier son nom complet (/usr/local/sbin/radiusd par exemple).

Station de travail sous Windows

Il faut une version de Windows suffisamment récente c'est à dire au moins Windows XP ou Windows 2000 avec le « service pack 4 »¹³. Les exemples qui suivent ont été produits avec Windows 2000 SP4.

Il est indispensable de faire afficher l'icône « Connexion au réseau local » dans la barre des tâches. En effet la progression dans les différentes étapes de la connexion et les éventuels messages d'erreur se présentent sous forme de bulles associées à cette icône.

Pour la mise au point il peut être utile d'activer les traces. Pour cela il faut à l'aide de l'éditeur de registre mettre à 1 la clé « EnableFileTracing » dans les différents nœuds de l'arbre portant les noms EAP*, RAS*¹⁴ situés à partir de

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Tracing
```

Le nom du fichier contenant les traces est défini dans une clé au même niveau dans l'arborescence.

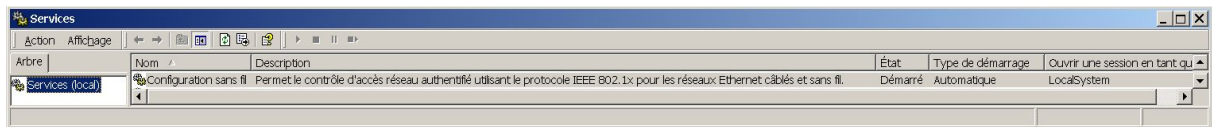
Pour forcer la séquence d'authentification on peut débrancher et rebrancher le câble Ethernet ou tous simplement désactiver puis activer à nouveau la connexion¹⁵.

Le service « Configuration sans fil » doit être démarré. Attention ce service est très mal nommé. Il concerne la gestion du protocole 802.1x et s'applique aussi bien comme indiqué dans le champ description aux réseaux Ethernet câblé et sans fil.

¹³ Windows 2000 SP3 avec le patch 313664 (<http://support.microsoft.com/default.aspx?scid=kb:en-us:313664>)

¹⁴ Nous avons trouvé que EAPOL, RASTLS étaient parmi les plus utiles.

¹⁵ Cliquer avec le bouton de droite sur l'icône « Connexion réseau ».

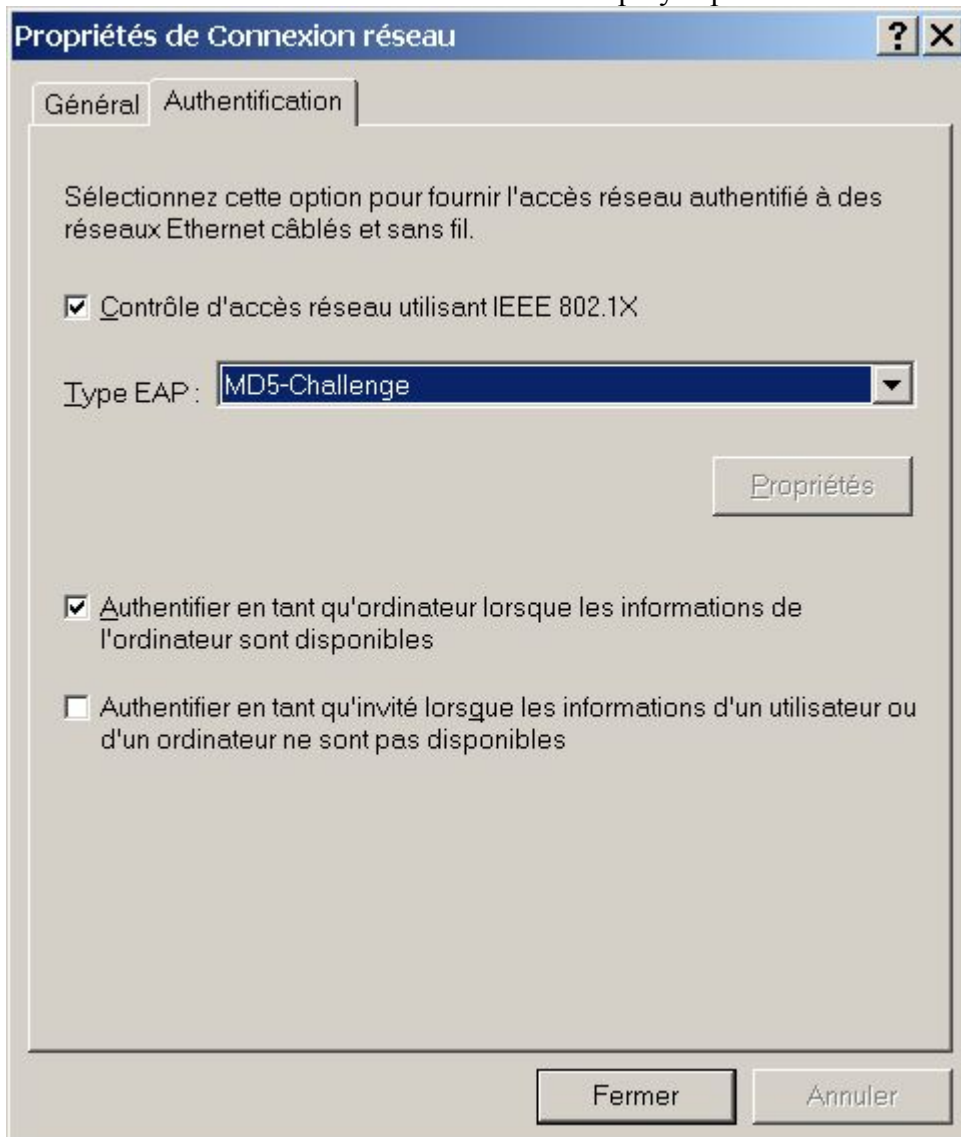


Il apparaît alors dans la fenêtre « Propriétés de Connexion au réseau local » un onglet nommé « Authentification ». Il faut cocher la case « Contrôle d'accès réseau utilisant IEEE 802.1x ».

Plusieurs méthodes d'authentification sont disponibles. Elles apparaissent sous le libellé « Type EAP ». Pour débiter, il est conseillé de choisir « MD5-Challenge » qui est une méthode nettement plus facile à mettre en œuvre que celle impliquant des certificats.

Authentification par mot de passe

Cette méthode ne peut être envisagée dans un contexte opérationnel que pour des liaisons Ethernet câblées. Elle ne doit en aucun cas être employée pour le Wi-Fi.



Il apparaît une fenêtre intitulée « Connexion au réseau local »



L'utilisateur donne alors son nom et son mot de passe. Le plus simple est de ne jamais utiliser de nom de domaine. Lorsque l'on en utilise un, le nom de l'utilisateur dans la requête envoyé au serveur radius est préfixé par le nom de domaine avec «\» comme séparateur. Il faut alors configurer le serveur radius en conséquence.

Authentification par certificat

La gestion des certificats sous Windows est relativement complexe et fait appel à une terminologie propre à Microsoft qui est parfois déroutante. Il n'est donc pas inutile de faire quelques rappels.

Un certificat peut être associé à :

- Un utilisateur
- Une machine
- Un service

Il existe différents types de certificats :

- Personnel. C'est le certificat permettant d'identifier un individu mais aussi une machine ou un service.
- Autorité de certification racine de confiance. Il s'agit toujours d'un certificat auto-signé.
- Autorité intermédiaire. Cela concerne la chaîne des certificats situés entre la racine et le certificat personnel.

A chacun de ces types est associé un magasin de certificats différent.

Il existe 2 catégories de magasins :

- Magasins logiciels. Il s'agit de zones systèmes (sur disque) où sont stockés les certificats ainsi que des fonctions logicielles permettant d'y accéder¹⁶. Les clés pour les utilisateurs et les machines sont rangées respectivement dans les répertoires¹⁷
\\Documents and setting\et
\\Documents and setting\All Users\Application Data\Microsoft\Crypto\RSA\Machinekeys
- Magasins physiques : cartes à puce, « token » USB.

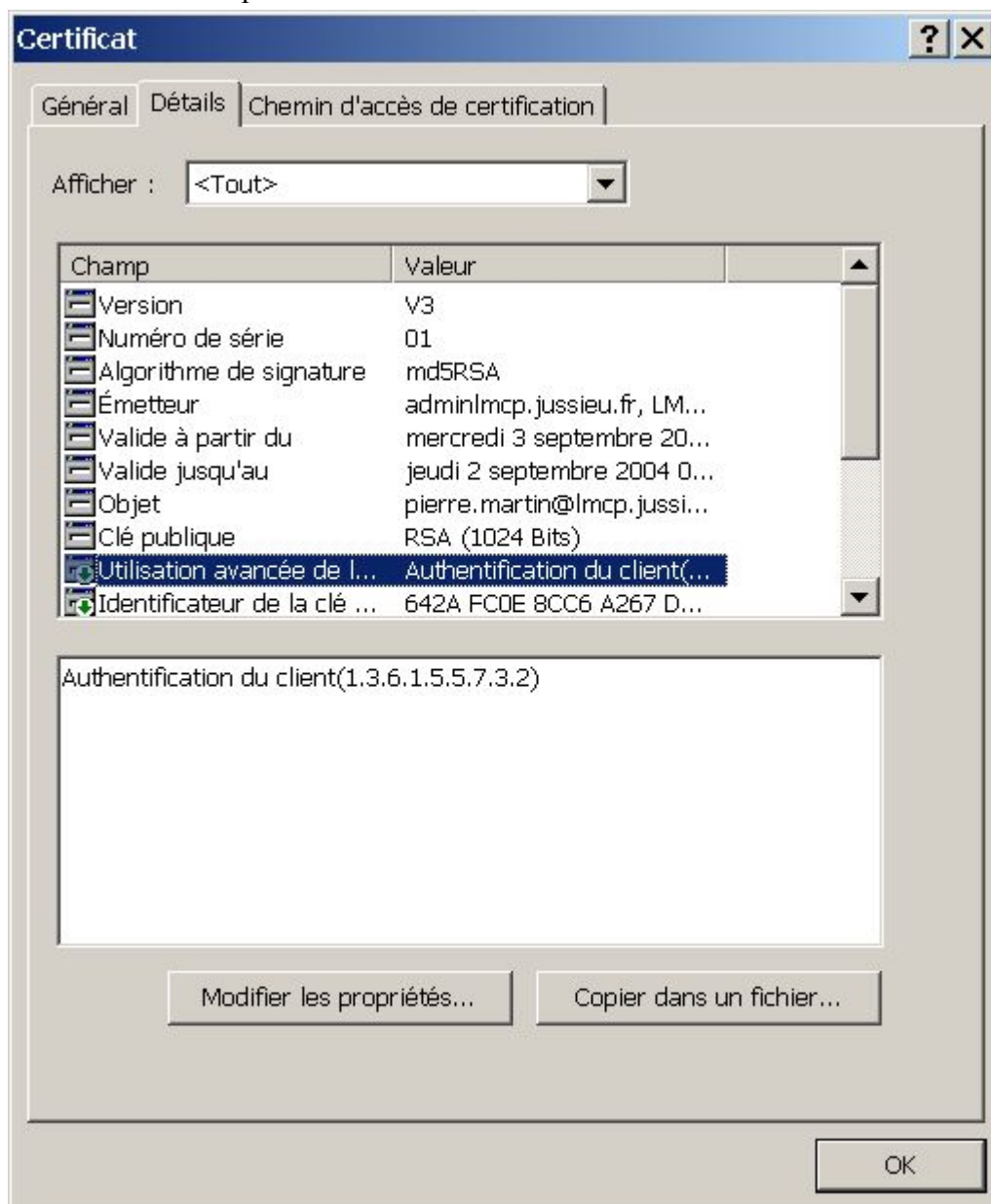
Avec le certificat peut aussi être stockée une clé privée. De fait cela concerne uniquement les certificats personnels. Pour les autorités de certification racine et intermédiaires on ne stocke

¹⁶ Récupérer la clé privée sur un disque doit être théoriquement possible mais il faut posséder des informations sur les mécanismes employés car la clé n'est certainement pas rangée en clair.

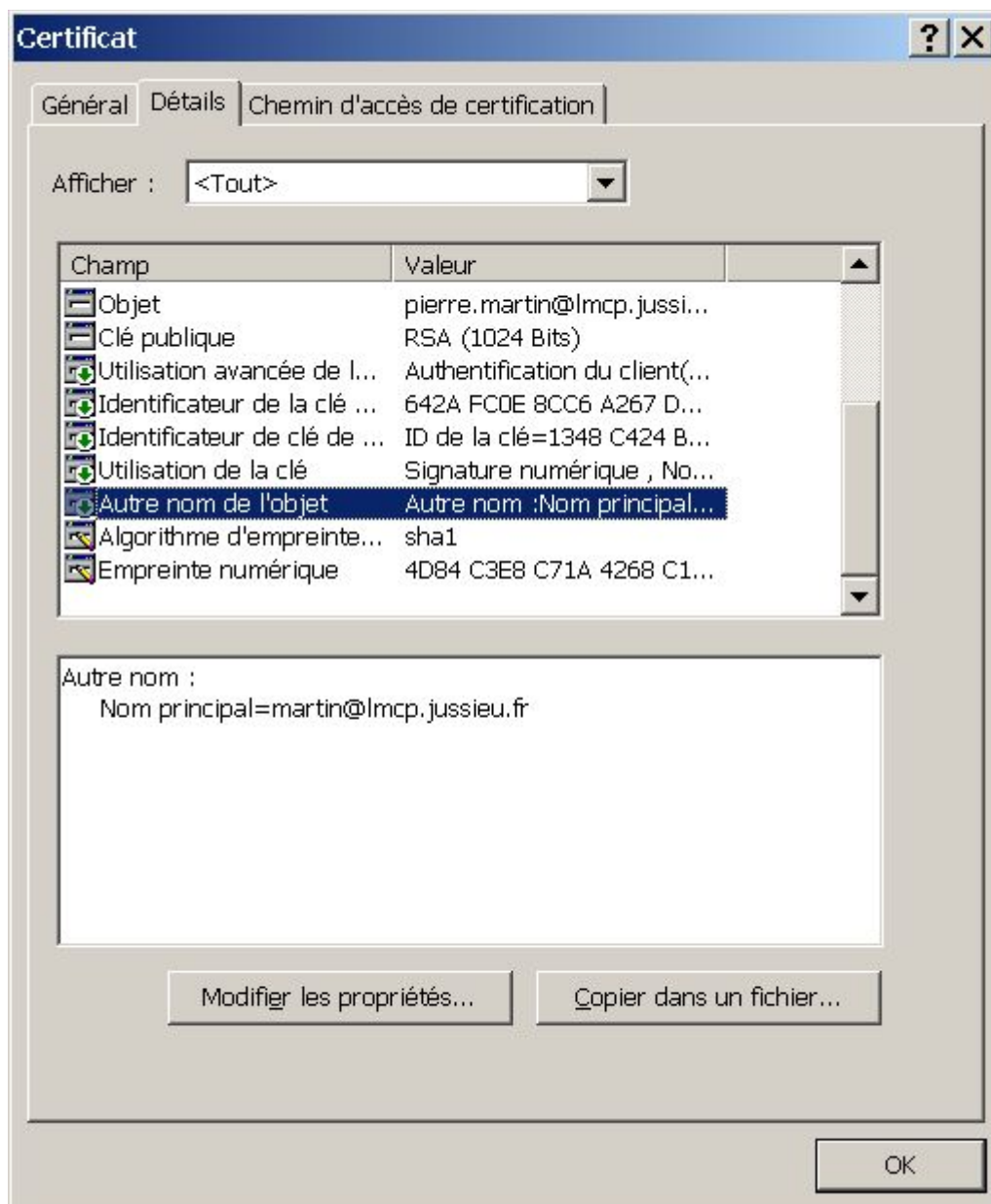
¹⁷ Sous Windows NT les clés privées sont rangées dans le registre.

évidemment jamais la clé privée. Ce qu'il faut protéger c'est la clé privée, le certificat étant public par nature. De fait tout le mécanisme des magasins de certificats est destiné à sécuriser la clé privée. Ceci explique que seuls les certificats personnels et leur clé privée peuvent être stockés dans des magasins physiques, en effet ils sont les seuls à avoir vraiment besoin de la sécurité supplémentaire offerte par une carte à puce pour protéger leur clé privée.

Du moins avec Windows, les certificats utilisés pour le serveur et le client doivent avoir des attributs particuliers. Cela concerne en particulier l'attribut «Extended Key Usage» que Microsoft appelle «Advanced Key Usage» qui pour le serveur RADIUS et le client doivent avoir respectivement les valeurs «TLS Web Server Authentication» et «TLS Web Client Authentication». Voici un exemple de certificat client :



L'attribut « Alternative Subject Name » est utilisé dans le certificat du client pour définir le nom d'utilisateur qui va servir pour l'authentification. En voici un exemple :



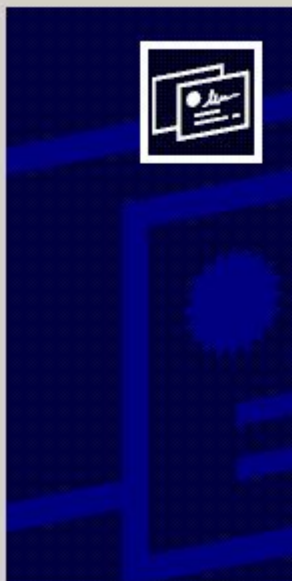
Attention cet usage de l'attribut « Subject Alternative Name » n'est pas standard mais propre à Microsoft. Son codage ASN1 peut être résumé de la façon suivante :

```
SEQUENCE
  cont [ 0 ]
  OBJECT: 1.3.6.1.4.1.311.20.2.3
  cont [ 0 ]
  UTF8STRING: login@domain
```

Pour installer les certificats, différentes méthodes sont possibles. Nous en décrivons une en supposant que dans un fichier au format PKCS#12 sont stockés le certificat avec sa clé privée, ainsi que les certificats des autorités de certification racine et intermédiaires¹⁸.

¹⁸ Dans l'exemple présenté ci-dessous, qui correspond à des essais, nous utilisons notre propre autorité de certification afin de pouvoir générer des certificats avec les attributs requis, l'IGC du CNRS ne les gérant pas encore à l'époque des essais.

Assistant Importation de certificat



Bienvenue !

Cet Assistant vous aide à copier des certificats, des listes de certificats de confiance et des listes de révocation de certificats depuis votre disque vers un magasin de certificats.

Un certificat, émis par une Autorité de certification, est une confirmation de votre identité et contient des informations utilisées pour protéger vos données ou établir des connexions réseau sécurisées. Le magasin de certificats est la zone système où les certificats sont conservés.

Pour continuer, cliquez sur Suivant.

< Précédent

Suivant >

Annuler

Assistant Importation de certificat



Fichier à importer

Spécifiez le fichier à importer.

Nom du fichier :

Parcourir...

Remarque : plusieurs certificats peuvent être stockés dans un seul fichier aux formats suivants :

Échange d'informations personnelles - PKCS #12 (.PFX,.P12)

Standard de syntaxe de message cryptographique - Certificats PKCS #7

Magasin de certificats sérialisés Microsoft (.sst)

< Précédent

Suivant >

Annuler

Assistant Importation de certificat [X]

Mot de passe

Pour maintenir la sécurité, la clé privée a été protégée avec un mot de passe.

Entrez le mot de passe de la clé privée.

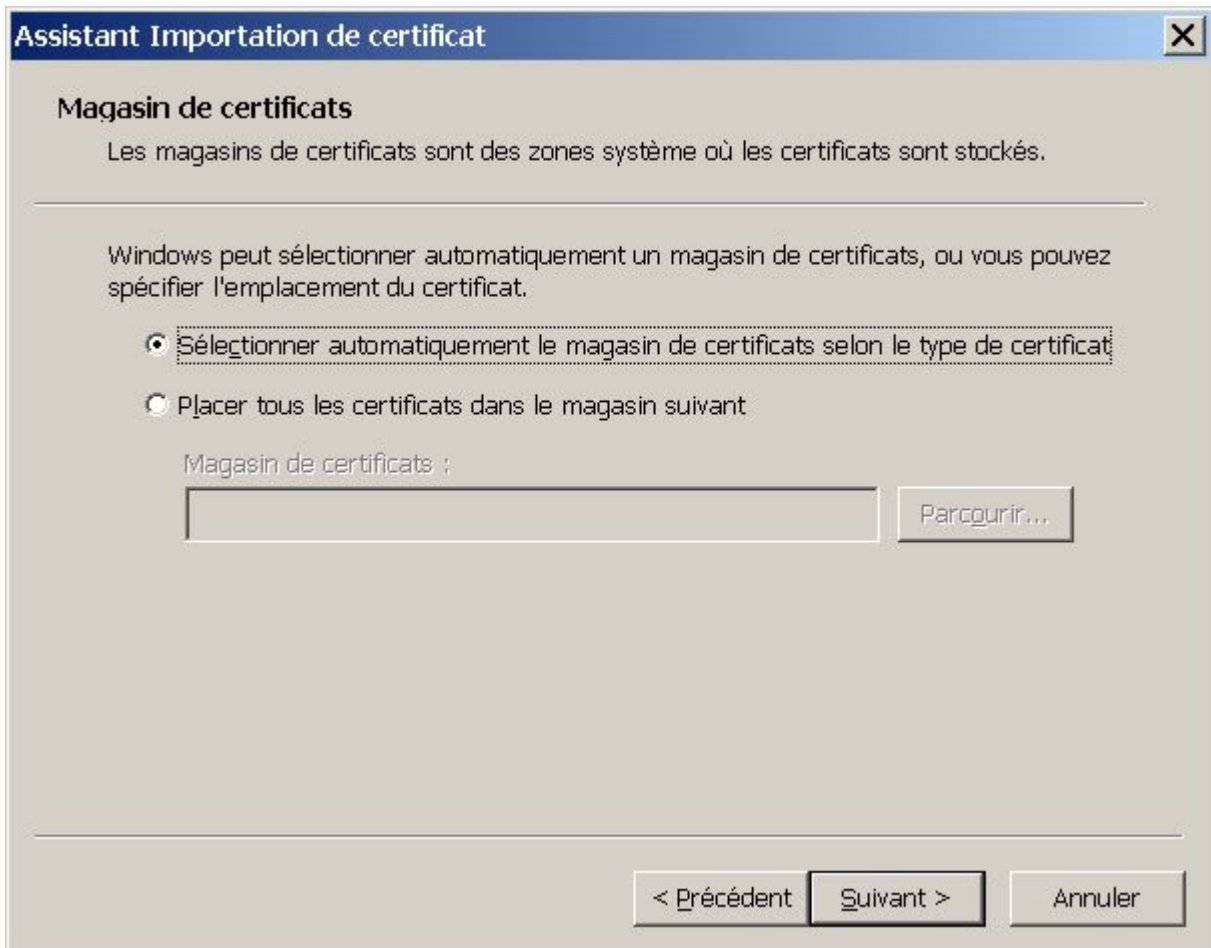
Mot de passe :

Activer la protection renforcée de clés privées. La clé privée vous sera demandée chaque fois qu'elle est utilisée par une application si vous activez cette option.

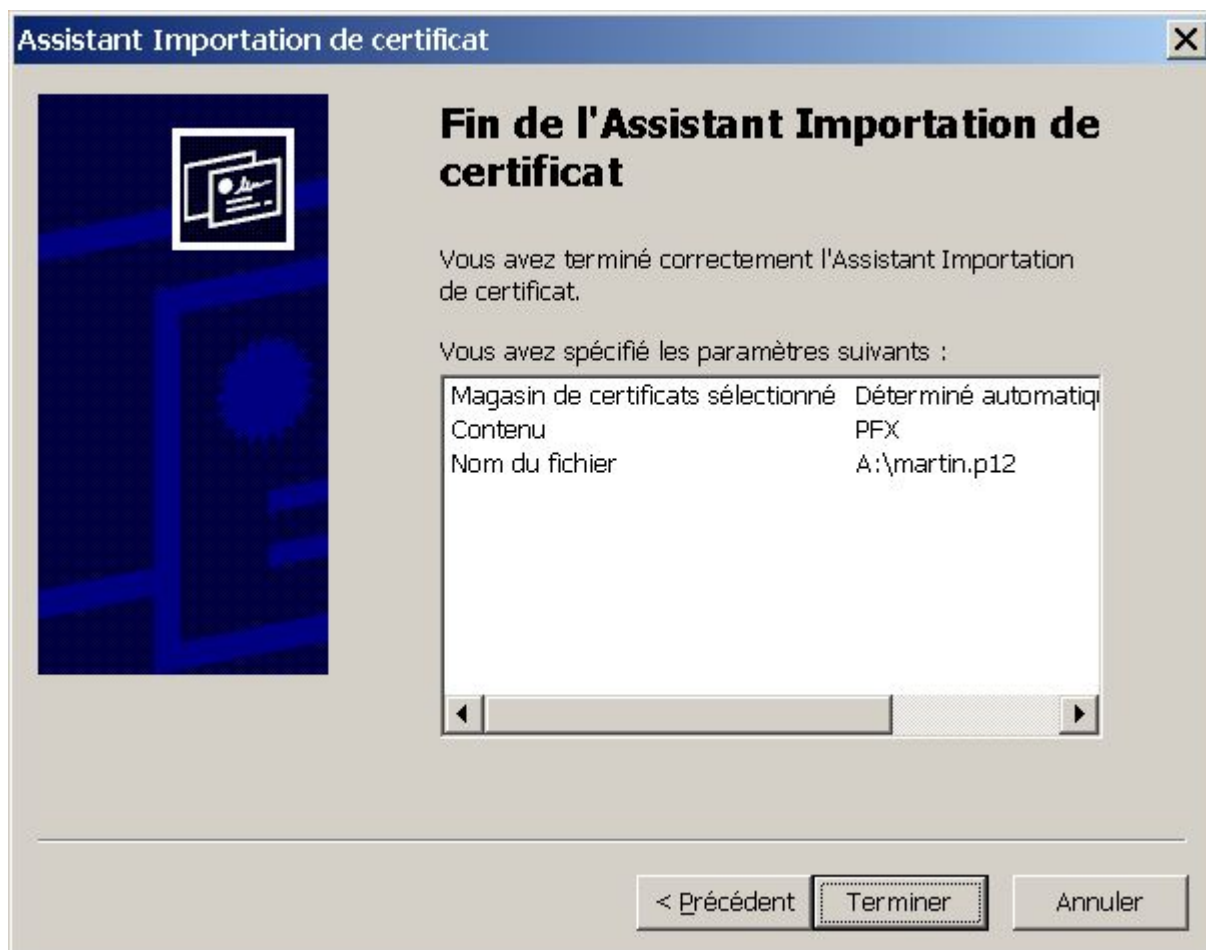
Marquer la clé privée comme étant exportable

< Précédent Suivant > Annuler

Attention il ne faut surtout pas cocher la case «Activer la protection renforcée de clés privées». Sinon l'authentification 802.1x ne fonctionne pas. Si cette protection a été activée, la seule solution consiste à effacer le certificat en utilisant le programme mmc («Microsoft Management Console»). Cela signifie que jamais un mot de passe ne sera demandé ultérieurement pour accéder à la clé privée. Il faut donc faire confiance à Windows pour assurer la sécurité.



La méthode conseillée consiste à cocher « Sélectionner automatiquement... ».



Ce message demande si on veut installer le certificat de l'autorité de certification racine. Il faut répondre « Oui ».



Avec la méthode EAP-TLS l'authentification du «suppliquant» peut se faire de différentes façons :

- A l'aide d'un certificat personnel associé à la machine, l'authentification a lieu au démarrage de la machine.
- A l'aide d'un certificat personnel associé à l'utilisateur, l'authentification a lieu après l'entrée en session (« logon ») de l'utilisateur.
- Il est possible de combiner les deux précédentes méthodes.

La valeur de la clé de registre

HKEY_LOCAL_MACHINE\Software\Microsoft\EAPOL\Parameters\General\Global\AuthMode

permet de modifier ce comportement :

- 0 authentification de la machine au démarrage, en cas d'échec authentification de l'utilisateur au logon
- 1 authentification de la machine au démarrage, puis authentification de l'utilisateur au logon
- 2 uniquement authentification de la machine

Il est aussi possible, comme nous le verrons par la suite, de supprimer l'authentification de la machine.

Dans le cas où l'authentification de l'utilisateur au logon se fait via le réseau (contrôleur de domaine, active directory), il faut évidemment avoir une connexion établie lors de cette phase. Cela implique une authentification réussie de la machine au démarrage. Pour les matériels qui permettent l'attribution d'un VLAN particulier en cas de non authentification 802.1x, on peut envisager de ne pas faire cette authentification de la machine au démarrage.

Pour les serveurs qui fonctionnant en l'absence d'utilisateur en session, il faut évidemment effectuer une authentification au démarrage et uniquement celle-ci. Il reste qu'un serveur est en principe une machine correctement administrée, située dans un local dont l'accès est contrôlé et ne pose pas les mêmes problèmes de sécurité ou de mobilité. Plutôt qu'une authentification 802.1x, il semble plus simple et aussi sûr de configurer les commutateurs de telle sorte que :

- sur le port auquel est connecté le serveur, seule est acceptée l'adresse MAC correspondant à celui-ci
- sur tout autre port et tout autre commutateur¹⁹, cette même adresse MAC est rejetée.

Pour une station de travail l'authentification de l'utilisateur permet théoriquement d'appliquer des règles différentes à chacun. Cependant comme il s'agit généralement d'ordinateur personnel l'avantage peut se révéler assez illusoire.

Si on n'utilise pas de carte à puce les certificats machine ou utilisateur et leur clé privée sont rangés dans des fichiers sur disque. L'implémentation est telle qu'en aucun cas un mot de passe est demandé pour déverrouiller la clé privée. Le seul petit plus apporté par un certificat utilisateur est qu'il a fallu réussir une ouverture de session, mais est-ce une réelle sécurité sur un ordinateur personnel ?

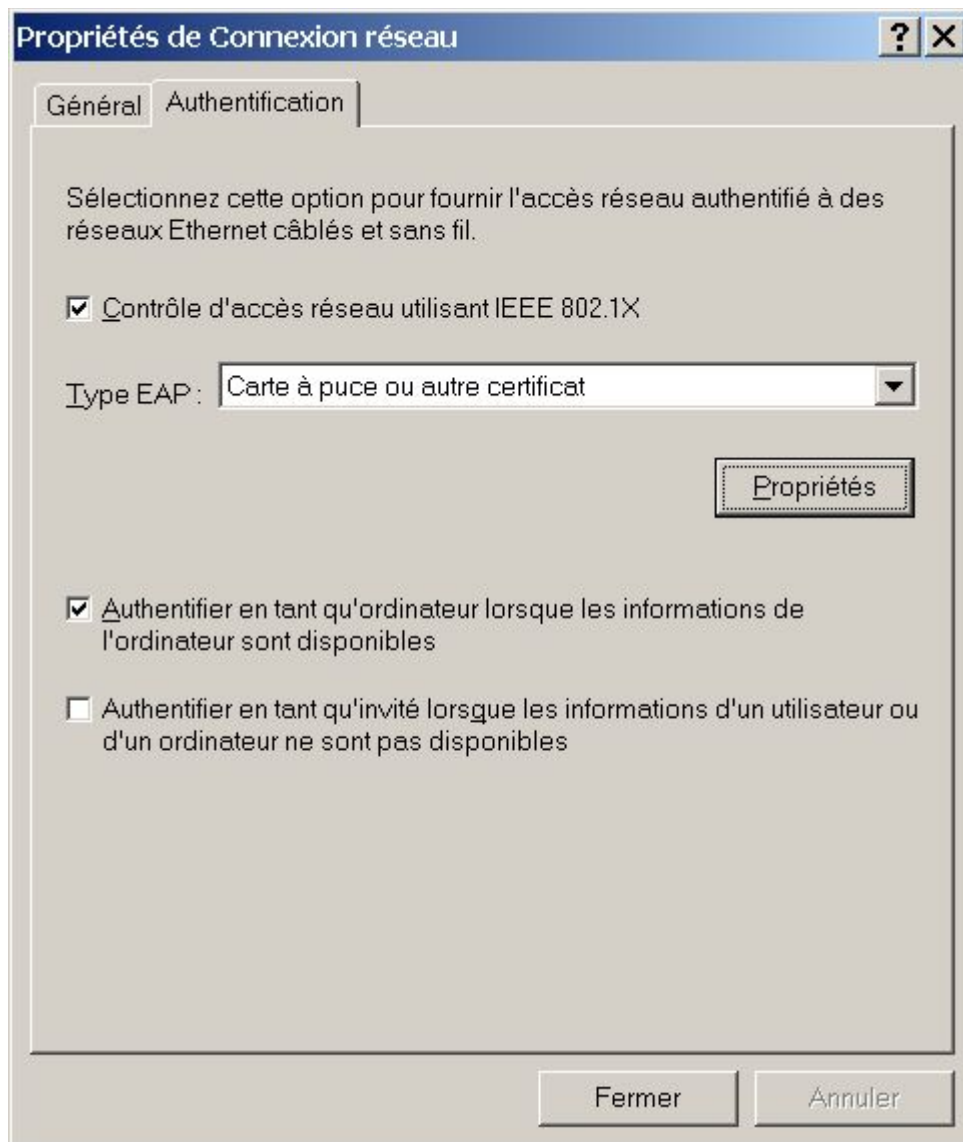
Seul un certificat utilisateur sur carte à puce offre une sécurité forte, l'individu devant à la fois posséder la carte à puce et fournir un code personnel.

Des réflexions précédentes on peut tirer les conclusions suivantes :

- Pour authentifier un utilisateur utiliser un certificat sur carte à puce.
- A défaut utiliser la méthode PEAP décrite plus loin.
- Un certificat machine est incomparablement plus sûr qu'une adresse MAC pour identifier une machine.

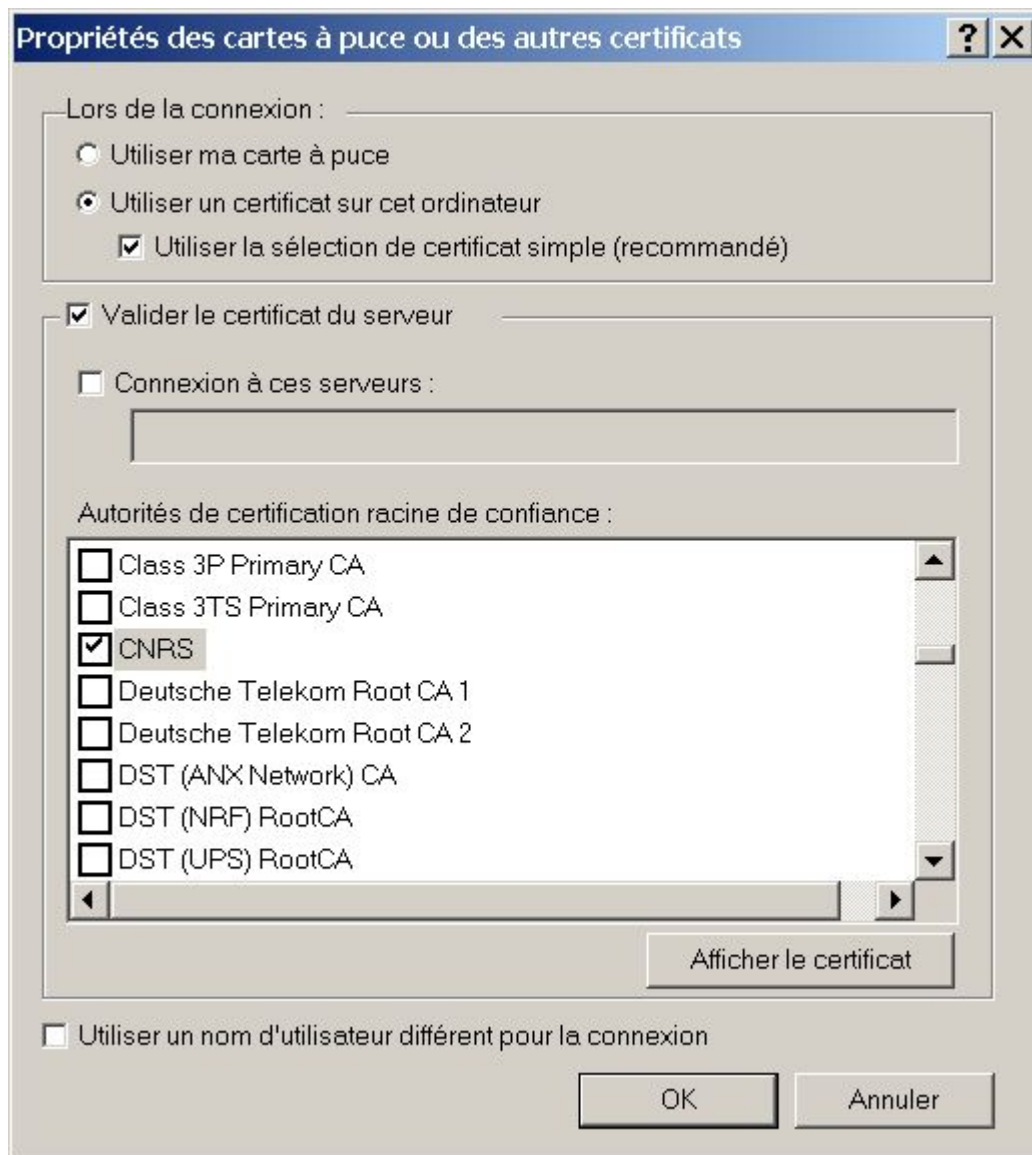
Pour utiliser la méthode d'authentification EAP-TLS il faut dans la fenêtre «Propriétés de Connexion réseau», sous l'onglet « Authentification » choisir « Carte à puce ou autre certificat » pour le champ « Type-EAP ».

¹⁹ A l'exception, si nécessaire, des ports «trunk» assurant la liaison entre commutateurs.



Cocher « Authentifier en tant qu'ordinateur... » permet d'utiliser un éventuel certificat associé à l'ordinateur pour effectuer une authentification au moment du démarrage. L'utilisateur sera ou non authentifié par la suite en fonction de la valeur de la clé de registre AuthMode comme expliqué précédemment. Si on veut uniquement authentifier l'utilisateur, il ne faut pas cocher cette case.

Il faut alors cliquer sur le bouton « Propriétés » pour faire apparaître la fenêtre « Propriétés des cartes à puce ou des autres certificats ». On définit alors l'autorité de certification racine.



Il est conseillé de cocher « Valider le certificat du serveur ». Il faut aussi cocher les autorités de certification racine auxquelles on fait confiance. En effet à partir du moment où on a décidé, avec tout ce que cela implique en matière de déploiement, d'effectuer une authentification forte du client par certificat, cela n'a aucun sens de faire les choses à moitié, surtout qu'il n'y a quasiment rien de plus à faire. Le « supplicat » va vérifier que le certificat fourni par le serveur RADIUS et transmis par l' « authenticator » est bien valide. Pour cela il faut que les certificats de l'autorité de certification racine et ses éventuelles intermédiaires soient installés dans les magasins correspondants. Il faut en outre que le certificat du serveur RADIUS ait les bons attributs. En particulier :

Ne pas cocher « Valider le certificat du serveur » peut s'avérer utile pour la mise au point en particulier si on a constaté dans le journal d'événements que le certificat du serveur a été refusé.

Spécifier « Connexion à ces serveurs » n'offre pas réellement d'intérêt.

« Utiliser un nom d'utilisateur différent pour la connexion » permet de présenter au serveur RADIUS un autre nom d'utilisateur que celui qui est défini dans le certificat. Nous sommes enclins à penser que cela ne présente pas beaucoup d'intérêt et surtout ouvre une faille dans la sécurité. En fait l'attribut « User-Name » envoyé au serveur RADIUS est par ordre de préférence :

1. Celui spécifié si on utilise « Utiliser un nom d'utilisateur différent pour la connexion »
2. Sinon la valeur du « Principal Name » trouvé dans champ Subject Alternative Name du

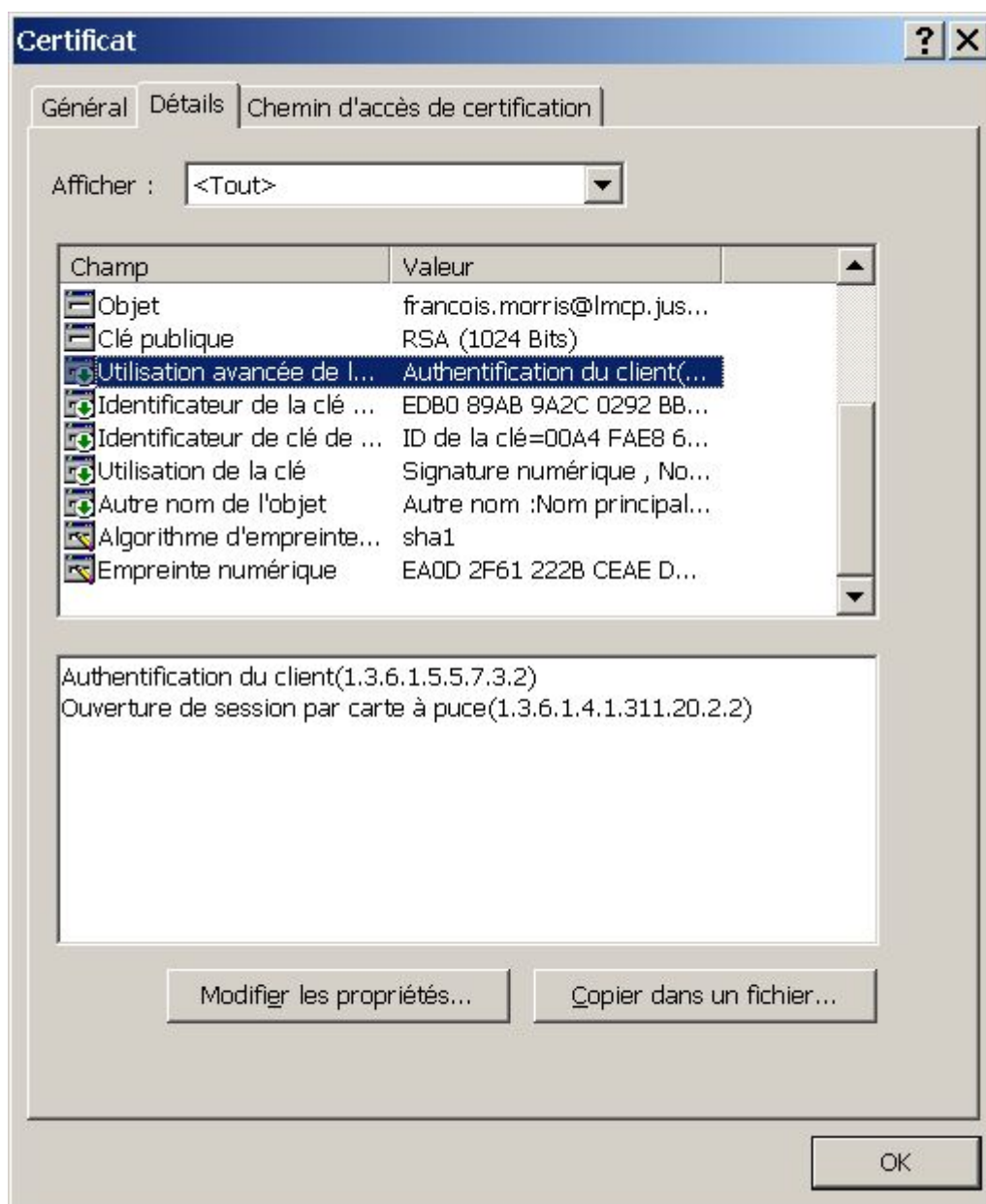
certificat

3. Sinon le nom utilisé au « logon » Windows

Il faut que le serveur RADIUS prenne sa décision d'authentification en s'assurant tout d'abord que le certificat présenté est bien valide, mais aussi en vérifiant que le certificat présenté est bien celui associé au nom d'utilisateur. D'après la documentation de Microsoft son serveur IAS (RADIUS) vérifie que le nom dans le champ « Subject Alternative Name » du certificat est bien le même que celui qui a été présenté dans la requête EAP-Response/Identity²⁰.

Une fois ces opérations effectuées, il n'y a plus rien à faire. L'authentification 802.1x se fait automatiquement sans intervention de l'utilisateur.

Nous avons aussi effectué des essais avec un « token USB » eToken de Aladdin. Ce dispositif est tout à fait analogue à une carte à puce²¹. Il faut cocher la case « Utiliser ma carte à puce ». L'attribut « Extended Key Usage » du certificat doit indiquer une utilisation supplémentaire. C'est propre à Microsoft et défini par un OID valant 1.3.6.1.4.311.20.2.2.



Lors de l'importation d'un tel certificat dans le magasin physique se fait de la façon suivante :

²⁰ Dans la version actuelle (0.9.0) du serveur freeradius cette vérification n'est pas effectuée. Ce ne serait pas compliqué à ajouter.

²¹ La puce utilisée en interne est celle d'une carte à puce.

Assistant Importation de certificat



Fichier à importer

Spécifiez le fichier à importer.

Nom du fichier :

Remarque : plusieurs certificats peuvent être stockés dans un seul fichier aux formats suivants :

Échange d'informations personnelles - PKCS #12 (.PFX, .P12)

Standard de syntaxe de message cryptographique - Certificats PKCS #7

Magasin de certificats sérialisés Microsoft (.sst)

Assistant Importation de certificat



Mot de passe

Pour maintenir la sécurité, la clé privée a été protégée avec un mot de passe.

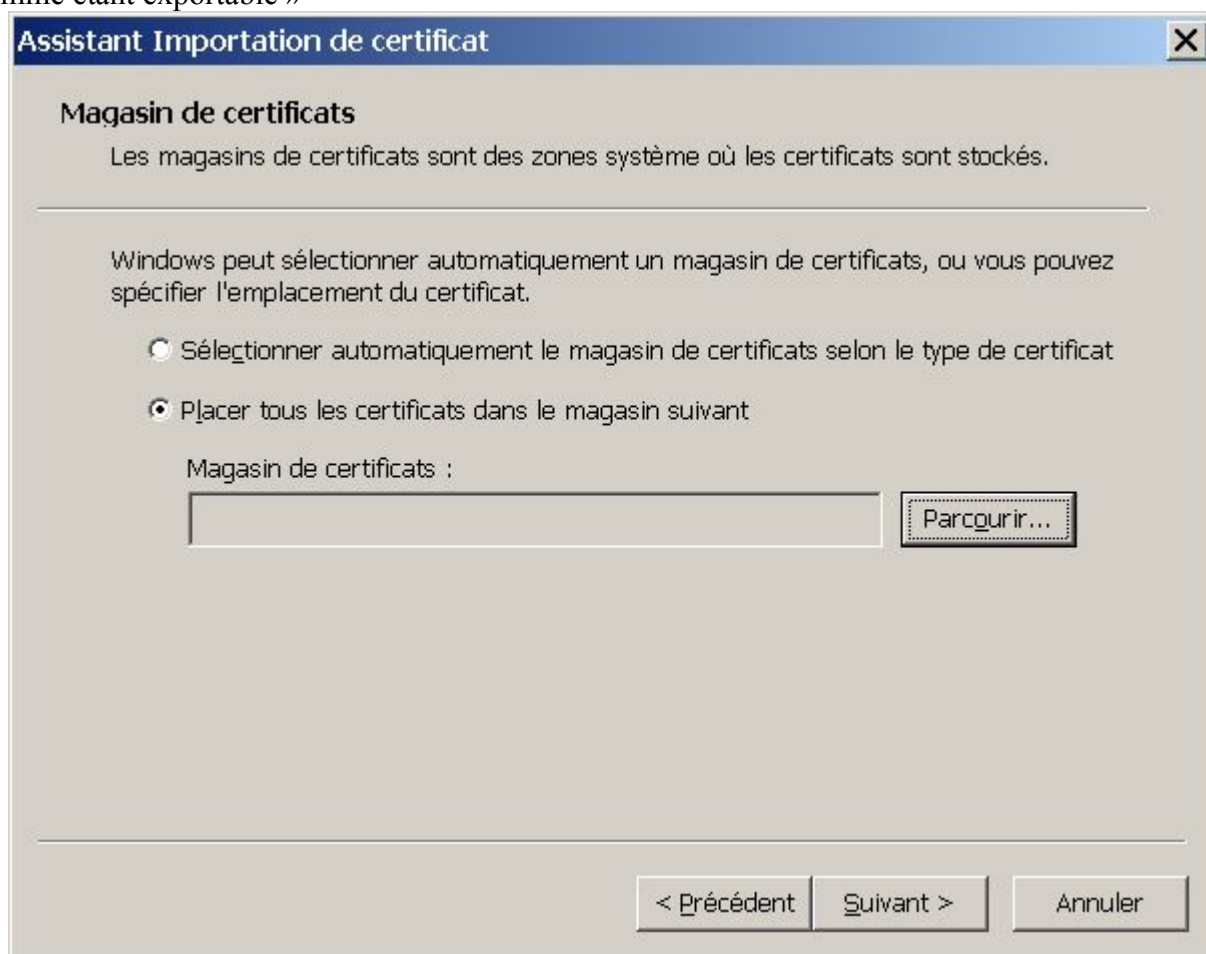
Entrez le mot de passe de la clé privée.

Mot de passe :

Activer la protection renforcée de clés privées. La clé privée vous sera demandée chaque fois qu'elle est utilisée par une application si vous activez cette option.

Marquer la clé privée comme étant exportable

Il faut impérativement, du moins pour l'eToken d'Aladdin, cocher la case « Marquer la clé privée comme étant exportable »²²

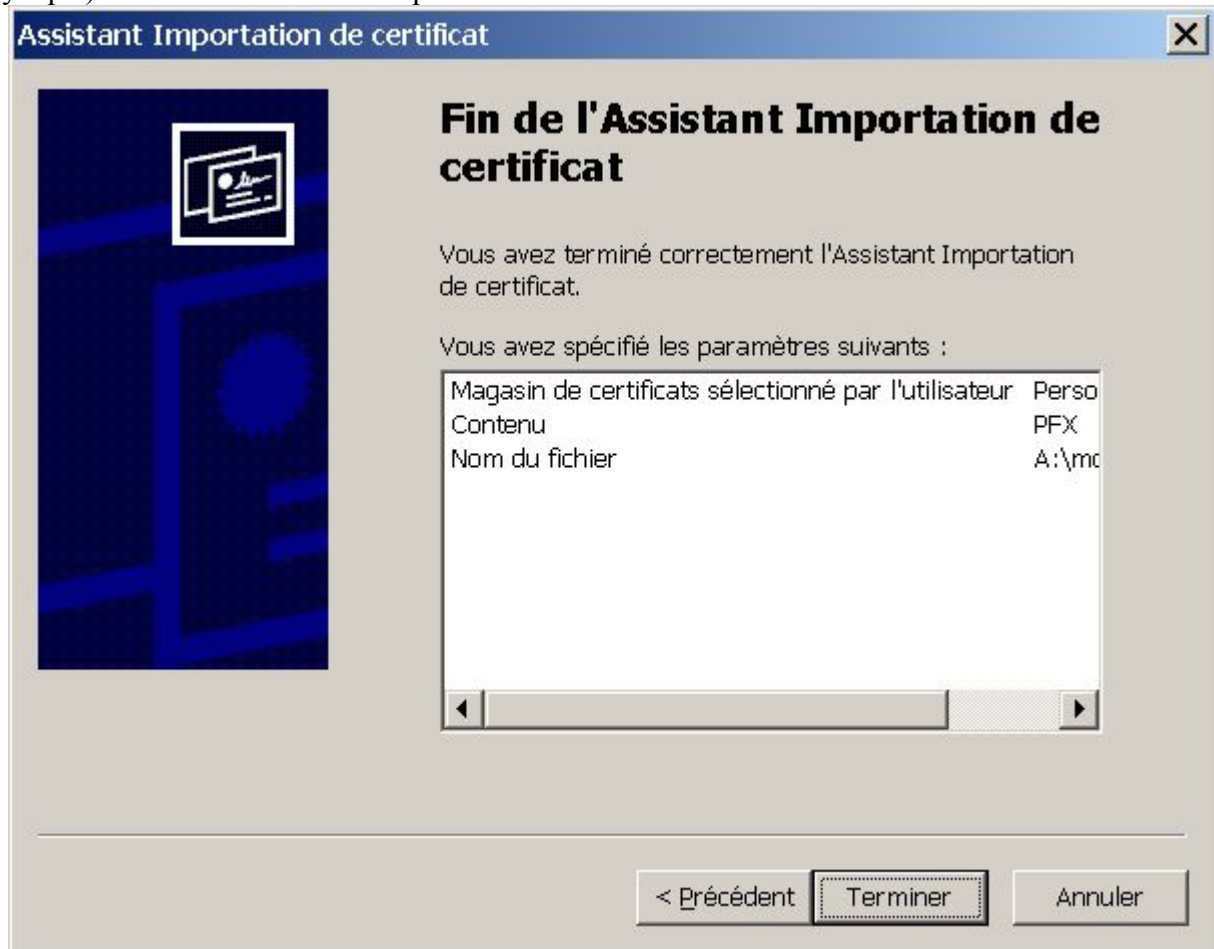


Il faut cocher « Placer tous les certificats dans le magasin suivant » et cliquer sur « Parcourir ».



²² Cela concerne eToken RTE version 3.50. Dans le README il est indiqué qu'il FAUT marquer la clé comme étant exportable mais qu'une fois importée dans l'eToken la clé ne sera plus exportable.

Il faut cocher « Afficher les magasins physiques » et sélectionner « eToken » (le nom du magasin physique)²³. On valide alors en cliquant sur « OK ».



On clique sur « Terminer »

²³ Il semblerait que « Registre » soit une scorie de Windows NT puisque pour Windows 2000, les clés privées sont rangées dans des fichiers et non plus dans le registre.

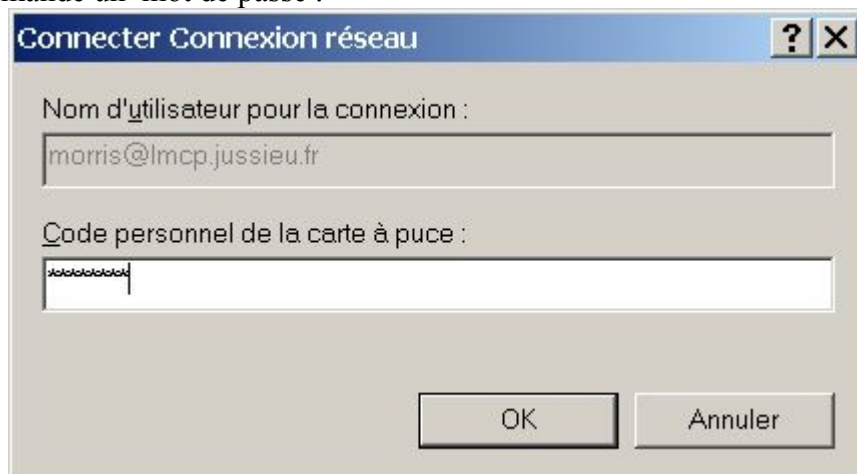


Il faut donner le mot de passe protégeant le magasin physique. Après quelques secondes apparaît la fenêtre suivante :



Il faut répondre « OK ».

Ce qui précède est effectué une fois pour toute. Par contre à chaque connexion au réseau il est demandé un mot de passe :



Authentification PEAP

PEAP (Protected EAP) est un équivalent par Microsoft de la méthode EAP-TTLS. L'authentification du serveur RADIUS se fait à l'aide d'un certificat. L'authentification du client utilise une méthode classique par mot de passe. Il s'agit en l'occurrence de «EAP-MSCHAP v2». La séquence d'authentification est alors sécurisée par TLS.

Côté client il y a quasiment rien à faire, sinon installer le certificat de l'autorité de certification qui a délivré le certificat du serveur d'authentification au cas où il ne le serait pas déjà.

Dans le cas où il existe déjà une structure d'authentification (domaine, «active directory»), le serveur RADIUS délègue à celle-ci la vérification du mot de passe.

A notre avis c'est une excellente solution dans un environnement homogène Windows²⁴ qui permet à moindre frais (pas besoin d'IGC) de s'intégrer dans l'existant.

Station de travail sous Unix

Il existe une implémentation du protocole 802.1x en logiciel libre qui s'installe les différents avatars d'Unix y compris MacOS 10. Il s'agit de open1x. C'est celle que nous utilisons. Nous ne détaillerons pas la procédure pour compiler et installer le produit. Pour cela on se rapportera à la documentation qui accompagne le produit. Nous décrirons son utilisation dans un environnement Linux, pour les autres les changements devraient être mineurs.

Il faut prendre garde au fait que généralement l'interface Ethernet est activée au démarrage du système en utilisant le protocole DHCP pour récupérer une adresse IP. Ce comportement est incompatible avec le fait que l'authentification 802.1x doit être préalable à tout trafic. Il est évidemment possible de modifier le script d'activation de l'interface pour y intégrer à la bonne place l'authentification 802.1x. Dans la suite nous supposons que l'interface Ethernet n'est pas activée au démarrage mais qu'un utilisateur a le droit de l'activer. L'authentification se fera après l'ouverture de session («login»).

Le fichier de configuration situé par défaut sous /etc/1x/1x.conf définit le comportement du produit. En voici un exemple :

Le fichier de configuration situé par défaut sous /etc/1x/1x.conf définit son comportement.

```
## This is a sample configuration file for xsupplicant that explains
## All currently configurable functionality. In general, this file is a
## series of tag-value pairs. In addition to a tag and a value, there is
## also a "network id" to group different tag-value pairs together.
## the file is parsed linearly, so redundant tags with the same network
## id will take the value of the last line. If no network name is provided
## on the command line (using the -n flag) then the network id "default"
## is parsed.

# the id tag indicates what value to return for an EAP Identity request
default:id = martin@lmcp.jussieu.fr #comment here

## spaces don't matter, this will work too
# default : id = xsupplicant-uesr@somedomain.com

## the path to the certificate file to be used for the above user
default : cert = /home/martin/martin.pem

## the path to the private key of the user for that cert
default : key = /home/martin/martin.pem

## the path to file containing all valid CA roots
default :root = /etc/1x/certs/CAroot.pem

## I have no idea if this does anything
default :auth = none
#default:auth = EAP

## Force this connection to wired or wireless.
```

²⁴ Nous n'avons pas effectué d'essais car nous n'avons ni IAS (actuellement freeradius ne supporte pas EAP-TTLS mais la versions en cours de développement l'intègre), ni Active Directory.

```

## Needed in situations where wired drivers answer ioctl's for wireless cards.
## Specifically, some intel cards with current drivers.
default:type = wireless
#default:type = wired

## preferred auth type
default : pref = tls

## password for the connection. This is optional, if you want the supplicant
## to authenticate without prompting for a password.
#default : password = <password>

## Phase 2 auth method for TTLS. (Currently, PAP, CHAP, MS-CHAP, or MS-CHAPv2)
## For PEAP, there is only MS-CHAPv2, so this does nothing.
default : phase2auth = PAP

## Phase 2 username (for using anonymous in the phase 1 piece).
## If this isn't defined, it defaults to the same as the phase 1 piece.
#default : phase2id = username@domain.org

## chunk size
default : chunk_size = 1398

## random file to use
default : random_file = /dev/random

## Shell command to run after the FIRST successful authentication
## command MUST begin with a "/" (absolute path)
default : first_auth = "/sbin/dhclient eth0"

## shell command to run after ALL successful authentications
## the current semantics are that if first_auth is also defined,
## only it is run the first time and after_auth is run ever other time
## if first_auth is not defined, after_auth is run after ALL authentications
## including the first.
## command MUST begin with a "/" (absolute path)
default : after_auth = "/bin/echo I authenticated"

```

La première action à effectuer est l'activation de l'interface Ethernet :

```
ifconfig eth0 up
```

Il faut ensuite utiliser la commande `xsupplicant` pour lancer l'authentification 802.1x. Le plus simple est comme toujours, de débiter avec une authentification MD5. Comme c'est EAP-TLS qui est défini par défaut dans la configuration on doit spécifier la méthode dans la commande.

```
xsupplicant -m MD5
```

Les messages suivants s'affichent alors :

```

Setup on device eth0 complete
Done with init.
Sending EAPOL-Start #1
Connection Established, authenticating...
Please Enter Your Password :
The authentication server requested a different style of authentication.
(Style 13)
Trying to force the authentication server to use our style.
Authentication Succeeded
Internet Software Consortium DHCP Client V3.0p11
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP

Listening on LPF/eth0/00:04:76:d9:61:d4
Sending on LPF/eth0/00:04:76:d9:61:d4
Sending on Socket/fallback
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.1.1.3
bound to 10.1.1.4 -- renewal in 21221 seconds.

```

Il faut noter que la requête DHCP a été automatiquement envoyée. Pour une authentification EAP-TLS il faut lancer la commande sans argument :

```
xsupplicant
```

On obtient les mêmes messages. Il faut cependant noter que le mot de passe à fournir est celui qui a servi à protéger la clé privée.

Wi-Fi

Le terme Wi-Fi est une appellation commerciale pour tout ce qui concerne les réseaux sans fil utilisant les normes de la série 802.11. Aujourd'hui la version 802.11b²⁵ utilisant une fréquence 2,4GHz et délivrant un débit théorique de 11Mbps/s est la plus utilisée. Le média utilisé est par nature partagé et on se retrouve dans la situation qui prévalait au début d'Ethernet où tout le monde peut écouter tout le monde. Il n'y a pas de commutation pour isoler les trafics. Le fait qu'il ne soit pas nécessaire de se connecter physiquement au réseau mais qu'il suffit d'être dans la zone de réception de la borne d'accès et l'on sait que les ondes électromagnétiques ne s'arrêtent pas aux murs d'un bâtiment, est un facteur aggravant. Avec le WEP (Wired Equivalent Privacy) les concepteurs ont voulu obtenir un niveau de confidentialité équivalant à celui fourni par un réseau Ethernet câblé. Cependant le WEP est né avec deux graves tares congénitales.

- Une faiblesse dans l'implémentation des algorithmes cryptographiques l'a rendu très vulnérable²⁶. Pratiquement il suffit dans le pire des cas de quelques heures d'écoute à un éventuel pirate pour casser les clés utilisées²⁷.
- Surtout l'objectif à atteindre était notoirement insuffisant. Pour le sans fil ne suffit pas de chiffrer les communications, il faut aussi impérativement authentifier la machine ou l'individu se connectant au réseau.

Ces défauts constituent un tel obstacle au développement du Wi-Fi que les différents fournisseurs dans le domaine ont poussé au développement rapide de nouvelles normes intégrant ces soucis de sécurité. Face à l'urgence, ils n'ont pas attendu la sortie des versions définitives de ces normes pour mettre sur le marché des produits les intégrant dans un état préliminaire ou bien en développant leurs propres mécanismes. Évidemment cela ne favorise pas l'interopérabilité ni la pérennité des investissements²⁸.

Un groupe de travail au sein de l'IEEE l'« IEEE 802.11 Task Group I » (TGi) travaille au développement d'une nouvelle norme de sécurisation des réseaux sans fil l'IEEE 802.11i. Les extensions à la norme telles qu'elles ont été défini par ce groupe de travail TGi ont été nommées « Robust Security Network » (RSN), l'achèvement des travaux est prévu pour dans quelques mois.

Un consortium le « Wi-Fi Alliance » dont le but est d'assurer l'interopérabilité des différents produits a défini des spécifications pour assurer la sécurité des réseaux sans fil, il s'agit « Wi-Fi Protected Access » (WPA). De fait il s'agit d'un sous ensemble des futures normes 802.11i.

Pour résoudre les problème liés à l'insuffisance du WEP la norme 802.11i définit 3 méthodes de chiffrement et de contrôle d'intégrité :

- TKIP (Temporal Key Integrity Protocol). Il a été conçu pour pouvoir fonctionner avec le matériel existant. Il peut être considéré comme une « rustine » sur le WEP. Il ajoute un contrôle d'intégrité (MIC)²⁹ sur les paquets transmis et introduit un mécanisme de gestion des clés.
- CCMP (Counter Mode CBC MAC). A priori plus puissant, c'est la solution à long terme. Cependant il est totalement incompatible avec ce matériel actuel. Il utilise AES comme algorithme de chiffrement, alors que le WEP utilise RC4.
- WRAP (Wireless Robust Authenticated Protocol) qui utilise AES a été proposé à l'origine mais a été abandonné depuis et remplacé par CCMP. Cependant il a été conservé car certains vendeurs en avaient déjà une implémentation matérielle.

²⁵ Le protocole 802.11a qui utilise une autre fréquence (5GHz) et permet un débit supérieur (54Mbit/s) est apparu après et est encore peu répandu. Quant au futur 802.11g, il permettra un débit de 54Mbit/s dans la bande des 2,4GHz.

²⁶ Cela démontre, par l'exemple, que définir un protocole de sécurité est une tâche difficile qui doit être exercée avec d'innombrables précautions par des spécialistes compétents, expérimentés et confrontés à la critique.

²⁷ Il n'est pas dans notre propos de détailler les failles, il existe une abondante littérature à ce sujet [3]. Disons qu'il s'agit essentiellement de problèmes de gestion et de génération des différentes clés de chiffrement utilisées.

²⁸ Lors d'une décision d'achat il est important de s'assurer que le matériel pourra être mis à niveau par un simple changement de son logiciel.

²⁹ L'acronyme MIC pour « Message Authentication Code » qui est généralement utilisé ne peut être employé ici car dans les normes IEEE 802.x il désigne « Medium Access Control ».

Le point important est que dans tous les cas l'utilisation de 802.1x, pour l'authentification et la génération des différents clés, est requise. Nous allons étudier comment s'effectue cette génération de clés.

- Après authentification mutuelle le serveur d'authentification et le client se mettent d'accord sur une clé symétrique maîtresse "Master Key" (MK)
- A partir de cette clé, chacun de leur côté, le client et le serveur calculent une clé appelée "Pairwise Master Key" (PMK) de la façon suivante :
 - $PMK = TLS-PRF (MK, \text{"client EAP encryption"} \mid \text{clientHello.random} \mid \text{serverHello.random})$
 - TLS-PRF (Pseudo Random Function) est une fonction pseudo aléatoire définie dans TLS et utilisant des algorithmes HMAC (Hashed Message Authentication Code) MD5 ou SHA1.
 - MK est la clé "Primary Key" définie précédemment.
 - clientHello.random et serverHello.random sont des nombres aléatoires générés respectivement par le client et le serveur et échangés lors de l'authentification TLS.
 - « | » est l'opérateur de concaténation.
- Le serveur RADIUS envoie cette clé PMK à la borne d'accès en utilisant l'attribut spécifique Microsoft MS-MPPE-Recv-Key (RFC 2548). La clé est chiffrée à l'aide du secret partagé entre le serveur et la borne d'accès. Les limites de ce chiffrement à l'aide d'un secret partagé pour la transmission de clé sont évidentes :
 - Il faut beaucoup de soins et de vigilance si on veut éviter que le secret ne finisse par être connu.
 - Si on utilise un réseau séparé physique ou virtuel pour les échanges entre la borne d'accès et le serveur RADIUS les risques sont limités.
- Le client et la borne dérive, chacun de leur côté, une clé symétrique de session appelée "Pairwise Transient Key" (PTK). Cette clé va être découpée en 3 parties pour définir les 3 clés de contrôle d'intégrité (MIC), de chiffrement et temporelle.
 - $PTK = EAPoL-PRF (PMK, AP \text{ Nonce} \mid STA \text{ Nonce} \mid AP \text{ MAC} \mid STA \text{ MAC})$
 - EAPoL-PRF est une fonction pseudo-aléatoire basée sur HMAC-SHA1
 - PMK est la clé "Primary Master Key" définie précédemment.
 - AP Nonce et STA Nonce sont des "nonces" correspondant respectivement à la borne d'accès et à la station.
 - AP MAC et STA MAC sont les adresses MAC de la borne d'accès et de la station
 - Les opérations exigent un mécanisme relativement complexe de 4 échanges de messages EAPoL-Key (4-Way Handshake) entre le supplicant et l'authenticator pour
 - Confirmer l'existence de PMK
 - Confirmer que cette PMK est la courante
 - Dériver la clé "Pairwise Transient Key" à partir de PMK
 - Installer les clés servant au chiffrement et au contrôle d'intégrité dans 802.11
- L'étape suivante est d'installer sur la station une clé partagée par toutes les machines connectée à la même borne appelée "Group Transient Key" (GTK) qui va servir à sécuriser le trafic en mode diffusion (broadcast/multicast). La transmission est chiffrée en utilisant la clé PTK.

Le message EAPoL-Key a un "Descriptor Type" qui a une valeur de 254, il remplace le message EAPoL-Key (Descriptor Type = 3) défini dans la norme 802.1x. En effet il faut transporter des informations différentes et le format de EAPoL-Key avait mal été conçu à l'origine.

Tout ceci montre que le mécanisme de génération et d'échanges de clés est très complexe et encore la présentation en a été considérablement simplifiée. Cette complexité, loin d'être gratuite, a été imposée pour se prémunir des différentes attaques possibles (écoute, rejeu, entremetteur, etc.). Encore une fois l'implémentation d'un système cryptographique n'est jamais simple, exige une étude préalable poussée et demande de le soumettre à la critique, ce qui n'avait pas été le cas pour le WEP.

Pour sécuriser le Wi-Fi, il existe une autre approche. C'est celle de VPN IPSec comme décrit

précédemment. La sécurisation se fait alors au niveau de la couche 3 et non plus 2.

Filtrage au niveau 2

Les normes prévoient uniquement l'utilisation de VLAN pour isoler le trafic au niveau 2. Pour pouvoir faire communiquer deux machines situées dans des VLAN différents il est nécessaire de passer par un élément au niveau 3 comme un routeur qui appliquera ses propres règles de filtrage. Si l'on veut contrôler finement le trafic sur un réseau local, il faut augmenter le nombre de VLAN. A la limite il faudrait un VLAN par machine.

Certains fournisseurs de commutateurs ont trouvé que cela manquait de souplesse, de performance. Ils ont proposé différentes méthodes permettant d'effectuer un filtrage plus fin au niveau d'un même VLAN.

Une situation typique est celle où l'on a un ensemble de serveurs et un ensemble de stations de travail. Lorsque l'on analyse la matrice des trafics, on s'aperçoit que les serveurs ont besoin d'échanger entre eux, une station de travail a seulement besoin de communiquer avec les serveurs. Le trafic entre stations de travail n'est généralement pas nécessaire et même souvent plus dangereux qu'utile. En effet le partage de ressources entre PC qui peut souvent être évité par l'utilisation de serveurs de fichiers, d'impression, ouvre une porte béante à la propagation des virus.

Une autre situation est celle d'une DMZ contenant plusieurs serveurs accessibles depuis l'extérieur. Le routeur d'entrée doit pouvoir communiquer avec chacun des serveurs, par contre les serveurs doivent être isolés entre eux pour éviter que la compromission de l'un permette d'attaquer les autres.

Nous allons nous intéresser à un certain nombre de fonctionnalités permettant de contrôler le trafic sur un réseau local. Pour être plus précis, généralement on ne se contentera pas uniquement des informations de la couche 2 (adresses MAC, protocole, etc.) qui n'offrent que des possibilités assez limitées en matière de filtrage mais on extraira de la trame Ethernet des éléments de la couche au dessus - IP en l'occurrence - (adresses IP, ports, etc.). Parler de filtrage au niveau 2 est alors un abus de langage, il s'agit plus exactement de filtrage par un commutateur. Évidemment ce qu'il est possible de faire dépend grandement du matériel utilisé. Cependant les avantages potentiels méritent qu'on en étudie les possibilités. Nous nous intéresserons uniquement au matériel Cisco car c'est celui que nous utilisons, d'autres matériels peuvent offrir des fonctionnalités analogues. Il ne s'agit aucunement de prétendre que ce matériel est supérieur à tel autre mais de donner des exemples de ce qu'il est possible de faire. C'est à chacun d'étudier, comment en fonction des possibilités de son matériel, il peut améliorer la sécurité de son réseau.

Contrôles des adresses MAC source

Il est possible de configurer le commutateur afin d'autoriser, sur un port donné, uniquement les paquets provenant d'une ou plusieurs adresses MAC prédéfinies. Cela permet, en partie, de se prémunir contre les usurpation d'adresses MAC. En effet si la machine usurpant l'adresse se branche sur tout autre port que celui auquel est connectée la machine légitime tous les paquets qu'elle va émettre vont être rejetés.

Dans le cas où l'on a affecté une politique de sécurité (numéro de VLAN, ACL) au niveau des ports, il faut impérativement contrôler qu'elle n'est pas contournée par un déplacement des machines. En effet il est vraiment difficile d'interdire à quelqu'un de connecter sa machine sur une autre prise.

Contrôler les adresses MAC est un moyen certes un peu brutal mais efficace pour connaître les machines connectées sur son réseau et leur emplacement. Comme il est possible de paramétrer les

commutateurs de telle sorte qu'une mauvaise adresse MAC génère une alarme, on est prévenu de tout comportement inhabituel.

Il existe deux méthodes pour définir les adresses MAC. La première consiste à les déclarer explicitement dans la configuration du commutateur. La seconde utilise l'auto-apprentissage, la première machine à être connectée sur un port donné impose son adresse MAC ce qui est plus simple à administrer.

Suivant le paramétrage choisi différentes actions sont possibles en cas d'adresse MAC invalide :

- On ignore le paquet.
- On bloque temporairement le port.
- On bloque définitivement le port.

Il est parfaitement envisageable d'utiliser ce mécanisme pour contrôler les paquets au niveau d'une liaison entre commutateurs.

Le principal obstacle au contrôle des adresses MAC est la complexité de sa mise en œuvre. Il est vrai que si on maintient dans une base de donnée, un état complet de son parc de machines, il est relativement facile d'en extraire le paramétrage à appliquer aux commutateurs. Mais il faut alors établir des procédures pour synchroniser la base de données avec la configuration des commutateurs, ce qui n'est généralement pas trivial.

Voici un exemple de configuration sur un commutateur Cisco Catalyst 3500XL :

```
interface FastEthernet0/1
port security max-mac-count 1
!
mac-address-table secure 0004.1234.5678 FastEthernet0/1 vlan 605
```

Sur un Cisco Catalyst 3550 cela se définit comme suit :

```
interface FastEthernet0/1
switchport port-security maximum 1
switchport port-security mac-address 0004.1234.5678
```

Cet exemple montre qu'avec un parc de commutateurs hétérogène, ce n'est pas simple à administrer et encore il s'agit ici du même constructeur.

Ports protégés

Cisco introduit la notion de port protégé («protected»). Aucun trafic n'est autorisé entre 2 ports protégés sur le même commutateur. Par contre le trafic entre 2 ports non protégés ou entre un port protégé et un port non protégé est autorisé. Cela s'applique parfaitement à la situation décrite ci-dessus. Un serveur aura un port non protégé, un PC aura un port protégé. Le principal inconvénient est que cette notion de protection ne s'étend pas au delà d'un commutateur, le trafic n'est pas interdit entre deux ports protégés situés sur deux commutateurs différents. Si les machines A et B connectées au commutateur X appartiennent au même VLAN que C connectée à Y la contamination directe de B par A n'est pas possible. Par contre on peut avoir la chaîne suivante : A contamine C, C contamine B.

Les ports protégés sont disponibles sur les Catalyst 2900XL/3500XL, 2950, 3550. La configuration se fait de la façon suivante :

- 2900XL/3500XL

```
interface FastEthernet0/1
port protected
```

- 2950, 3550

```
interface FastEthernet0/1
switchport protected
```

En dépit de ses limitations nous estimons que l'utilisation de ports protégés est une mesure

relativement simple à mettre en œuvre qui apporte un accroissement notable de la sécurité. Pour un VLAN destiné à connecter des ordinateurs portables c'est une excellente solution³⁰.

Private VLAN

Sur les Catalyst séries 4000, 6000 il existe la notion de VLAN privé qui permet d'isoler des machines situées sur le même VLAN³¹.

ACL

En fonction du matériel utilisé, il est possible ou non de définir des règles de filtrage (Access Control List ou ACL) applicables aux paquets. Ces filtres peuvent concerner :

- Soit à un port du commutateur.
- Soit à l'ensemble des paquets transitant sur un VLAN donné (VLAN map).

Les filtres utilisent les informations des en-têtes de paquets

- Niveau 2 : adresse MAC, type
- Niveau 3 : adresses IP, ports, protocole... Il faut noter que bien que travaillant au niveau 2, le commutateur est capable d'analyser des informations de la couche au-dessus.

Nous ne détaillerons pas tout ce qu'il est possible d'effectuer comme filtrage. Disons que c'est tout à fait semblable à ce que l'on peut faire sur un routeur sauf que dans ce cas cela se fait au niveau de la couche 2.

Avec le 802.1x il est possible d'associer dynamiquement une ACL à un port donné au moment de l'authentification de la machine. Cela permet d'avoir des règles de filtrage par machine.

Nous allons illustrer notre propos par un exemple. Nous voulons pouvoir connecter les ordinateurs portables du personnel du laboratoire sans prendre de risque. Nous appliquerons donc la politique de sécurité suivante :

- Accès à différents serveurs (10.1.1.0/28) supposés peu vulnérables aux différents codes malicieux (virus, ver, chevaux de Troie) car ils sont régulièrement mis à jour et correctement administrés.
- Accès à la passerelle vers l'Internet. (10.1.1.254)
- Broadcast UDP uniquement pour récupérer l'adresse IP par DHCP
- Aucun trafic n'est autorisé vers les autres machines (10.1.1.0/24) pour éviter d'infecter un PC où la dernière faille de sécurité ne serait pas corrigée.

```
access-list 102 permit udp any host 255.255.255.255 eq bootps
access-list 102 deny udp any host 255.255.255.255
access-list 102 permit ip any 10.1.1.0 0.0.0.15
access-list 102 permit ip any host 10.1.1.254
access-list 102 deny ip any 10.1.1.0 0.0.0.255
access-list 102 accept ip any any
```

Nous supposons que cette ACL qui concerne les paquets entrant est affectée dynamiquement lors de l'authentification 802.1x. Il faut alors définir l'attribut «Filter-Id» pour le serveur RADIUS :

```
Filter-Id = "102.in"
```

Recommandations

Le cloisonnement des réseaux locaux, l'utilisation de VLAN, l'authentification 802.1x permettent d'améliorer grandement la sécurité. Encore faut-il faire attention à respecter un certain nombre de règles pour éviter de laisser des trous béants dans la sécurité qui compromettent tous les efforts faits par ailleurs. Il ne sert à rien de mettre un gardien qui contrôle ceux qui passent par la porte d'entrée, si on n'a pas fermé la porte de derrière et les fenêtres.

³⁰ A priori, pas de besoin de partage entre portables. Risques très élevés de diffusion de virus.

³¹ Ne disposant pas de tel matériel qui est plutôt destiné à des réseaux de taille conséquente nous n'avons pas approfondi la question.

Voici donc quelques recommandations :

- Choisir des commutateurs et des bornes d'accès Wi-Fi gérant les VLAN et possédant des fonctions de sécurité³².
- Installer les commutateurs dans des locaux dont l'accès est contrôlé (fermé à clé)³³.
- Utiliser les mécanismes disponibles pour sécuriser les bornes d'accès Wi-Fi. Il faut éviter qu'un individu ayant physiquement accès à la borne puisse changer sa configuration ou tout simplement la remplacer par une autre.
- Prévenir ou au moins détecter l'installation « sauvage » de bornes Wi-Fi.
 - Définir et faire respecter une politique de sécurité³⁴.
 - Répondre à la demande en fournissant un service de qualité .
 - Établir des contrôles au niveau des ports des commutateurs pour éviter la connexion de matériel inconnu (filtres, 802.1x, etc.).
 - Utiliser des logiciels de détection de bornes Wi-Fi clandestines³⁵.
- Se méfier des matériels ayant à la fois une interface Ethernet filaire et une interface Wi-Fi.
 - Si les 2 interfaces sont simultanément actives et que le routage est autorisé (une case à cocher³⁶) on établit un chemin entre deux réseaux qui court-circuite toutes les politiques de sécurité.
 - Même sans routage ou connexion simultanée, il est toujours possible de récupérer un virus sur l'interface Wi-Fi et le renvoyer ultérieurement sur l'interface Ethernet.
 - Une machine connectée au réseau Ethernet avec une interface Wi-Fi en mode « ad hoc³⁷ » est l'équivalent d'une borne d'accès. Avec les interfaces Bluetooth on a la même problématique avec le facteur aggravant qu'ils sont souvent activés par défaut et que l'utilisateur n'est forcément au courant qu'un trafic peut s'établir avec les machines situées dans le voisinage.
- Un seul matériel doit être connecté à un port d'un commutateur. Éviter de partager un port à l'aide d'un « hub » ou d'un commutateur sans fonction de sécurité.
- La configuration livrée avec un commutateur est destinée à permettre un fonctionnement immédiat mais au détriment de tout ce qui concerne la sécurité (VLAN 1 sur tous les ports, tous les ports sont activés, adaptation automatique à un trafic avec ou sans «tag » 802.1q, etc.). Il est donc impératif de la modifier dès l'installation.
- Minimiser l'utilisation du VLAN 1³⁸.
- Définir un réseau spécifique pour l'administration à travers IP. Utiliser à cette fin et réserver uniquement à cet usage un VLAN différent de 1. Choisir pour les commutateurs des adresses IP privées (10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16 comme décrit dans le RFC 1918) ce qui constitue une sécurité supplémentaire et évite de gaspiller des adresses IP pour des machines qui ne doivent pas être visibles de l'extérieur. Pour les machines servant à l'administration il existe différentes possibilités :
 - Réserver une ou plusieurs machines spécifiques connectées uniquement sur le réseau des commutateurs
 - Établir un routage avec un filtrage strict entre le réseau des commutateurs et la (les) machines servant à administrer.
 - Utiliser une machine avec 2 interfaces, une pour le réseau normal et une autre pour le réseau des commutateurs. Évidemment il ne doit pas y avoir de routage sur cette

³² S'il s'agit de construire un réseau local totalement isolé pour relier entre elles les différentes machines d'une ferme de calcul, la sécurité ne sera pas un souci. Par contre les performances et le coût seront des critères déterminants.

³³ En ayant physiquement accès à un commutateur et avec un ordinateur portable il est relativement facile de définir une nouvelle configuration

³⁴ La technique ne peut avoir réponse à tout, il faut agir sur le comportement des individus.

³⁵ On trouve des références à différents produits de détection à l'URL suivante : <http://www.cru.fr/wl/>

³⁶ Avec les interfaces graphiques pour gérer les systèmes, on n'est qu'à un petit nombre de clics d'une catastrophe.

³⁷ Le mode « ad hoc » permet d'établir un réseau entre plusieurs machines sans passer par une borne d'accès.

³⁸ L'expérience a montré que bien des problèmes constatés de non-étanchéité entre les VLAN résultaient d'une utilisation inappropriée du VLAN 1.

machine entre les deux réseaux.

- Pour les ports assurant la liaison entre 2 commutateurs («trunk») et véhiculant des VLAN établir des filtres autorisant uniquement les VLAN nécessaires. Généralement sur un commutateur il existe un moyen de spécifier la liste des VLAN autorisés (le VLAN 1 est lui toujours transporté, c'est pourquoi il faut en limiter l'usage).
- Toujours pour ces mêmes ports «trunk», définir judicieusement le VLAN natif qui sera utilisé pour attribuer un numéro de VLAN aux trames qui ne sont pas au format 802.1q. Il s'agit bien évidemment de toujours éviter le VLAN 1.
- Définir un VLAN spécifique uniquement utilisé à cette fin pour les ports inutilisés. Affecter ce numéro de VLAN à tous les ports inutilisés. En outre, comme on n'est jamais trop prudent, il faut aussi fermer («shutdown») tous les ports inutilisés.
- Se méfier des machines qui génèrent des trames 802.1q/802.1p. Il faut impérativement contrôler, au niveau du commutateur, les numéros de VLAN utilisés et éventuellement la priorité. La politique en matière de sécurité ou de qualité de service ne doit pas être laissée à la seule initiative du poste de travail, il faut la contrôler au niveau du commutateur.
- La confiance que l'on accorde aux VLAN ne doit pas être totale. Il n'est absolument pas raisonnable d'utiliser des VLAN pour regrouper sur un même commutateur le réseau externe et le réseau interne. Ils doivent appartenir à des réseaux physiques séparés avec un élément filtrant routeur ou coupe-feu entre les deux.
- Si l'affectation des numéros de VLAN aux différents ports est statique, contrôler les adresses MAC pour éviter qu'il suffise de déplacer une machine d'un port à un autre pour accéder à un autre VLAN.
- Être à l'écoute des avis de sécurité concernant les commutateurs utilisés et appliquer les correctifs éventuels.

Bilan

En matière de sécurité il faut s'intéresser à différents aspects :

- Confidentialité
- Contrôle d'intégrité
- Authentification

La confidentialité d'un réseau Ethernet câblé ne repose pas sur le chiffrement des échanges. Une infrastructure à base de commutateurs adaptés, une configuration correcte de ceux-ci, comme décrit ci-dessus, rendent les écoutes difficiles. En effet il faudrait pouvoir accéder physiquement à des points critiques comme les commutateurs qui sont supposés être dans des locaux protégés ou au câblage mais une fibre optique reste difficile à intercepter. De toute façon il ne faut pas se leurrer, il est largement plus facile, si on est capable de pénétrer les locaux d'agir directement sur le poste de travail³⁹. Le chiffrement des échanges n'apporterait qu'un accroissement illusoire de sécurité. Si on a de réels soucis de confidentialité, il faut agir à la source et ne stocker que des données chiffrées.

La confidentialité du Wi-Fi est assurée par le chiffrement des données échangées par voie hertzienne entre le poste de travail et la borne d'accès. La mise en œuvre du protocole 802.1x est obligatoire pour assurer un renouvellement suffisamment fréquents des clés de chiffrement afin d'en interdire le passage.

Sur un réseau Ethernet câblé, il n'y a pas de contrôle d'intégrité, hormis un simple CRC qui n'a pas d'autre prétention que de détecter des erreurs de transmission. Mais la modification ou l'insertion de données est au moins aussi difficile que l'écoute.

Pour le Wi-Fi les mécanismes employés pour la transmission de données chiffrées impliquent un contrôle d'intégrité.

³⁹ Il suffit de voir ce qu'il est possible de faire avec un système démarré à partir d'un CD comme Knoppix (<http://www.knoppix.org>). Et encore il ne s'agit pas d'un outil à vocation de piratage !

La norme 802.1x apporte désormais une authentification réellement fiable de qui est connecté au réseau.

Les commutateurs intègrent désormais des fonctionnalités de filtrage qui permettent de réaliser un cloisonnement du réseau et un contrôle fin du trafic. Il est devenu possible de définir une architecture logique du réseau indépendamment de l'infrastructure physique. De plus l'appartenance à un sous-réseau n'est plus nécessairement lié à une prise mais directement à la machine ce qui favorise la mobilité.

Dans un environnement où vouloir contrôler les différents postes de travail connectés relève de l'utopie, le réseau local reste un des rares points où il est encore possible d'agir efficacement.

Du fait de l'absence de chiffrement, et de l'utilisation de circuits spécialisés sur les commutateurs ont obtenu d'excellentes performances en terme de débit.

Avant d'envisager de déployer des solutions à base d'IPSec, de SSH qui conservent toute leur utilité, il faut étudier s'il n'est pas possible d'agir au niveau du réseau local et ce à moindre coût.

Il ne faut pas croire que tous les risques proviennent de l'extérieur⁴⁰, un contrôle en entrée de réseau par un coupe-feu ne suffit pas. Cloisonner le réseau interne, contrôler les machines qui s'y connectent devient une nécessité. De fait nous sommes amenés à utiliser les techniques employées pour sécuriser le réseau interne vis à vis de l'extérieur pour protéger entre elles les machines de ce même réseau interne. Il ne faut donc pas s'étonner que l'on définissent, au sein même d'un réseau local, des VPN et des tunnels ou qu'on lui applique des règles de filtrage.

Aujourd'hui nous avons des outils, comme 802.1x, pour faire de la sécurité sur un réseau local Ethernet. Utilisons les ! C'est probablement la seule chose que nous puissions réellement faire, alors que le poste de travail échappe à tout contrôle⁴¹.

Références

- [1] FreeRADIUS <http://www.freeradius.org>
- [2] Open1x Open source implementation of IEEE802.1X <http://www.open1x.org>
- [3] Nancy Cam-Winget, Russ Housley, David Wagner, Jesse Walker. Security Flaws in 802.11 Data Link Protocols. Communications of the ACM May 2003/Vol. 46, N° 5
- [4] OpenVMPS <http://vmips.sourceforge.net/>

⁴⁰ « Dans de nombreux cas, nous avons aussi pu constater que les utilisateurs d'ordinateurs portables constituaient aussi une menace supplémentaire : des systèmes souvent peu maintenus et amenés à se déplacer sur des réseaux dont les mesures de sécurité sont variables. Malgré les filtres en bordure, ou des mesures de mise à jour de parc informatique interne, ceux-ci peuvent ainsi introduire le chaos (ou ajouter au chaos) dans un environnement qu'on pensait à peu près maîtrisé. » *Avis CERT RENATER 2003/STAT036*.

⁴¹ Le paradigme a complètement changé. Nous n'avons plus affaire à un utilisateur derrière un poste de travail que nous administrons mais désormais à un individu connectant son ordinateur portable sur le réseau.