GUIDE DE MISE EN PLACE D'UNE SOLUTION D'AUTHENTIFICATION NIVEAU 2

Freeradius + Openldap

Le thème de ce guide est la mise en place d'une solution d'authentification niveau 2 sur un réseau filaire dans le but de contourner les risques de l'accès physique au réseau. Cette authentification nécessite un serveur RADIUS couplé à un annuaire LDAP.

Agence Nationale de la Sécurité Informatique

Gestion des versions du document

Auteur	Version	Date	Modification apportée
K.J	1.0	28/11/2010	Première version

Document Publique

Document Interne

PLAN

I. PRESENTATION

Avant toute chose, il est important de savoir de quoi on parle lorsqu'on traite d'authentification dans le domaine de l'informatique. L'authentification est la procédure qui consiste, pour un système informatique, à vérifier l'identité d'une entité (personne, ordinateur...), afin d'autoriser l'accès de cette entité à des ressources (systèmes, réseaux, applications...). L'authentification permet donc de valider l'authenticité de l'entité en question.

Dans le cas d'un individu, l'authentification consiste, en général, à vérifier que celui-ci possède une preuve de son identité ou de son statut, sous l'une des formes (éventuellement combinées) suivantes :

- Ce qu'il sait (nom d'utilisateur/mot de passe).
- Ce qu'il possède (carte à puce, certificat électronique, clé-USB, etc.).
- Ce qu'il est (caractéristique physique, biométrie).

II. L'AUTHENTIFICATION POUR QUOI FAIRE?

L'objectif principal de l'authentification est la traçabilité des transactions. Le contrôle permanent de l'intégrité et de l'accès (usage, identité du destinataire, émetteur, propriétaire) à un contenu ou à un service constitue le fondement de la traçabilité des transactions et permet :

- La protection du patrimoine informatique de l'entreprise: réduire les dégâts qui résultent d'attaques, de la perte de temps, de la perte d'informations ou de l'espionnage...
- La protection de la vie privée. Les données personnelles véhiculées dans les systèmes d'information sont des données sensibles à protéger.

Les objectifs de ce guide ont été de prendre en compte les éléments suivants dans une solution globale :

- La nécessité désormais d'identifier les utilisateurs d'un réseau.
- Les besoins de sécurité en matière d'accès au réseau (attribution de VLAN).

- Les besoins de traçabilité des utilisateurs du SI : le contexte réglementaire et les bonnes pratiques impliquent que l'administration du SI soit traçable. On parlera d'accounting avec RADIUS.
- La mobilité de certains utilisateurs (mobilité interne et externe). Quel que soit l'accès au SI, les conditions doivent être les mêmes.
- La possibilité d'avoir un réseau invité où les personnes extérieures pourront se connecter avec des droits restreints.
- La localisation géographique des utilisateurs grâce à l'attribution des VLAN.

III. AUTHENTIFICATION PAR ADRESSE MAC

Pour de nombreux réseaux d'entreprise, le risque le plus important lié au déploiement de 802.1X et/ou NAC (Network Access Control) est la notion de « tout ou rien » qui rend un potentiel retour en arrière délicat si quelque chose venait à ne pas se dérouler correctement. Pour réduire ce risque, de nombreux administrateurs réseau et responsables sécurité, ont commencé à déployer « l'authentification par adresse MAC » comme première étape afin de sécuriser un peu plus la couche d'accès réseau. L'authentification MAC est particulièrement utile car elle adresse plusieurs objectifs menant à l'exploration de NAC et 802.1X, comme sécuriser la périphérie du réseau d'entreprise, identifier tous les équipements attachés au réseau, fournir un accès réseau aux invités, maintenir un historique de la localisation et de l'adressage de chaque équipement, et autres. De plus, le déploiement de l'authentification par adresse MAC s'appuie sur les systèmes et les protocoles directement impliqués dans le déploiement de 802.1X. Indépendamment de la durée d'implémentation ou du nombre de phases définies, l'authentification par adresse MAC se justifie pleinement comme première phase d'un déploiement ayant pour objectif 802.1X et/ou NAC.

Les avantages d'une telle approche incluent:

- L'utilisation des éléments de contrôle de 802.1X pour l'authentification réseau, sans toutefois avoir à considérer la configuration des clients
- L'implémentation sans risque d'isoler les utilisateurs ou les équipements
- L'offre de la visibilité sur la localisation des équipements, l'adressage, le comportement…

- Le déploiement rapide : pas de configuration des clients, peut utiliser les infrastructures RADIUS et réseau existantes
- L'approche par phases qui est utilisée à 100% dans les phases suivantes
- Plusieurs exigences relatives aux audits internes et externes
- L'implémentation dans des environnements peu propices à la prise de risque.

IV. AUTHENTIFICATION 802.1X

Le protocole 802.1X définit un contrôle d'accès réseau par port ainsi qu'une méthode d'authentification qui permet de restreindre l'accès aux clients non authentifiés qui se connecteraient à des ports « libres » du réseau local. On définit comme « port » le moyen par lequel un équipement client accède à des ressources partagées via un réseau (MAN ou LAN). Un port peut tout aussi bien être physique (port d'un équipement de commutation) ou virtuel (port sur une borne d'accès WIFI).

Le 802.1X se base sur trois types d'entité :

- « Supplicant » : le client demandant à s'authentifier avant de pouvoir accéder aux ressources du réseau.
- « Authenticator » : l'authentificateur est l'équipement réseau (commutateur, point d'accès, …) auquel le client se connecte. Suivant la réponse du serveur d'authentification, le commutateur laissera passer ou non le trafic du client.
- « Authentication server » : le serveur d'authentification vérifie sur demande du commutateur si le client peut ou non accéder aux ressources réseaux.

Le serveur d'authentification (*free*RADIUS par exemple) va authentifier chaque client qui se connecte au réseau sur un port géré. Ce port connaît deux modes fonctionnels : contrôlé (controlled) et non contrôlé (uncontrolled). Avant authentification du demandeur, seul le mode non contrôlé permet des échanges d'information spécifiques. Ces flux spécifiques sont appelés flux EAPOL pour « EAP Over Lan ». Une fois la phase d'authentification achevée, le port contrôlé est basculé et les flux autorisés peuvent être émis à destination du réseau en lui attribuant un VLAN.

Si un client ne supportant pas 802.1X se connecte sur un port attendant une authentification 802.1X, l'équipement réseau va demander au client de s'identifier. Le client ne sachant pas

répondre à la requête, le port reste bloqué dans l'état non contrôlé et le client ne peut pas accéder au réseau.

À contrario, quand un client configuré avec le protocole 802.1X se connecte à un commutateur qui n'a pas d'authentification activée, le client va tenter de s'authentifier en envoyant une première requête EAPOL. Ne recevant pas de réponse, il va retenter plusieurs fois avant « d'abandonner » et de considérer le port comme étant dans l'état « contrôlé » et ainsi envoyer tout le trafic réseau.

Une fois que le client est authentifié, le port change d'état et devient « contrôlé » et tout le trafic réseau passe par ce port. Si la réponse du serveur d'authentification est négative, le port reste bloqué dans son état initial (non contrôlé). Mais le processus d'authentification peut être retenté plusieurs fois. Au bout d'un nombre d'essais défini, l'authentification échoue et l'accès au réseau est bloqué.

Lorsqu'un client se déconnecte, il envoie un message de fin (LOGOFF) qui va permettre au commutateur de passer le port en mode non contrôlé et ainsi attendre l'authentification d'un autre client. Lorsque le lien réseau est coupé, le port va de nouveau demander au client de s'authentifier.

V. PROTOCOLE EAP :

La communication entre l'équipement réseau (authenticator) et le serveur d'authentification est assurée par le protocole EAP (Extensible Authentication Protocol), le 802.1X ne fournissant qu'un cadre fonctionnel à l'interaction entre les équipements. Ce protocole EAP est un protocole de transport des informations d'authentification et permet d'utiliser différentes méthodes d'authentification d'où le terme « Extensible ».

Le domaine d'application de ce protocole correspond donc à tous les modes de connexion pouvant être considérés comme des connexions dites point à point telles que : connexion réseau sans fil entre un poste utilisateur et une borne d'accès, connexion filaire entre un poste utilisateur et un commutateur.

Méthodes EAP :

Les méthodes d'authentification les plus communes sur EAP sont :

- **EAP-TLS** (Transport Layer Security): a été créé par Microsoft et acceptée par l'IETF comme RFC 2716. Cette méthode se base sur les certificats numériques. Le serveur et le client s'authentifie mutuellement tout en cryptant les données échangées dans cette phase d'authentification. L'utilisation de clés publiques et privées des deux côtés va permettre de créer un tunnel sécurisé entre les deux parties, ce qui garantit notamment l'intégrité des données. Avec ce principe, le client ne fournit pas de mot de passe, le certificat permettant l'authentification.
- **EAP-TTLS** (Tunneled Transport Layer Security): est un protocole propriétaire développé par Microsoft, Cisco et RSA Security. Ce protocole assure une authentification mixte par certificat et mot de passe, le tout dans un tunnel sécurisé. Cette méthode combine l'avantage de l'authentification du client via le couple login/mot de passe ainsi que la sécurité des données via l'encapsulation cryptée des données et l'authentification du serveur par un certificat.
- **PEAP:** est un protocole propriétaire développé par Microsoft, Cisco et RSA Security. ici seul le serveur d'authentification dispose d'un certificat numérique. Il le transmet au client qui va pouvoir l'authentifier. Un tunnel sécurisé TLS est alors établit entre les deux parties. Le client va s'authentifier via une méthode EAP quelconque mais bénéficiera de l'encapsulation sécurisée des données dans le tunnel TLS.
- **EAP-MD5 Challenge:** l'utilisateur va être authentifié via son login et son mot de passe mais ce dernier ne sera pas transmis en clair sur le réseau. Grâce au mécanisme de challenge / réponse, le serveur envoie un challenge au client, celui renvoie son mot de passe associé au challenge, le serveur compare le résultat avec le mot de passe qu'il détient dans sa base plus le challenge envoyé. Si le résultat est identique alors l'accès est autorisé, sinon il est refusé.
- **LEAP** (Lightweight EAP): est une implémentation propriétaire d'EAP conçu par Cisco Systems assurant une authentification simple par mot de passe via une encapsulation sécurisée, ce protocole est vulnérable aux attaques (cryptage md5) sauf si l'utilisateur utilise des mots de passe complexes.
- **EAP-FAST** (EAP-Flexible Authentication via Secure Tunneling): est une proposition de Cisco Systems pour fixer les faiblesses de LEAP en garantissant une flexibilité d'authentification via une encapsulation sécurisée.

Trame EAP :

Un message EAP peut être de quatre types/code : *Request, Response, Success ou Failure*. Une trame EAP se compose de 5 champs de longueurs différentes détaillés ci-dessous.

Code	Identifiant	Longueur	Type	Données
1	1	2	1	n

Figure1 : une trame EAP

Les quatre types de messages EAP les plus fréquents sont les suivants :

- *Identity* : identité de l'utilisateur souhaitant accéder au réseau
- *Notification* : chaine de caractère envoyé au client (supplicant)
- *Nak* : type proposé uniquement lors des réponses indiquant un refus de la méthode d'authentification et en propose une autre.
- *MD5-Challenge* : utiliser lors du « challenge ».

Les trames EAP sont encapsulées dans des trames EAP over LAN (EAPOL) pour pouvoir être transmises sur les réseaux locaux (Ethernet, Token Ring, FDDI).

Echange de messages EAP :

Le schéma ci-dessous explique le mécanisme classique d'authentification EAP. Il représente une demande d'authentification d'un client au commutateur jouant le rôle de l'authentificateur. Ce dernier va demander l'accès au serveur d'authentification qui acceptera ou non l'authentification du client.

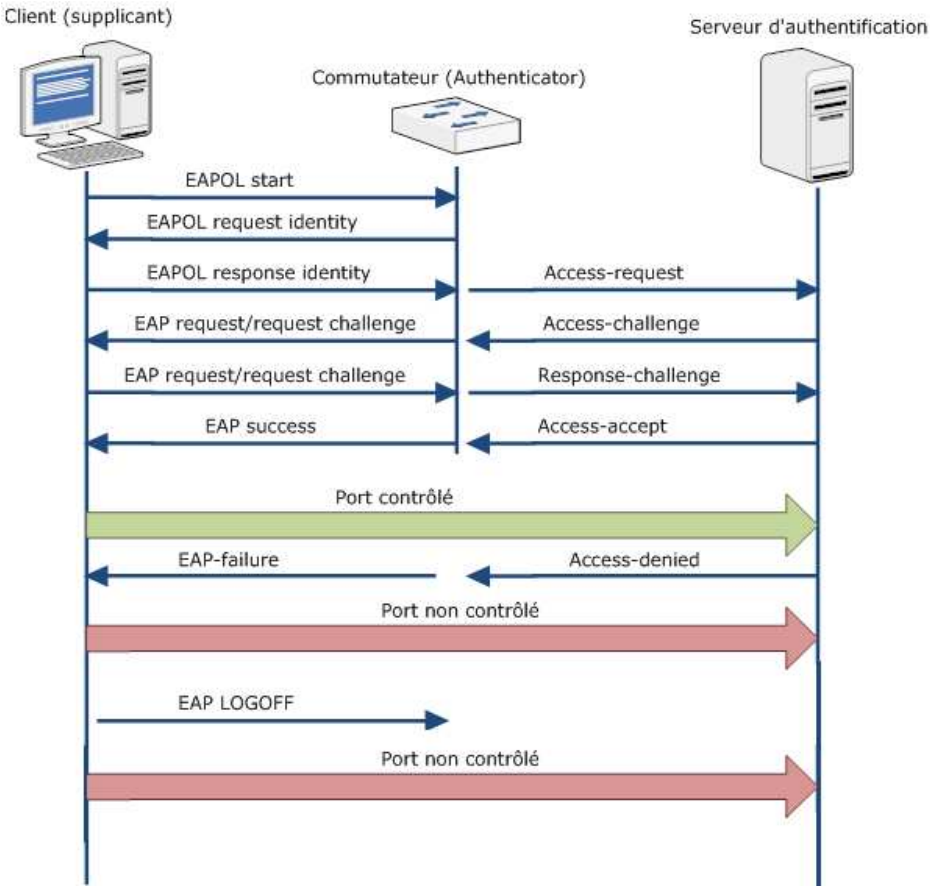


Figure 2: Echange EAP

VI. LE VLAN

Dans un réseau local (LAN), l'utilisation d'un commutateur à la place d'un concentrateur permet de réduire le nombre de messages que reçoivent les machines. En effet, les commutateurs disposent d'une « table de forwarding » qui permet de savoir sur quel port physique il faut envoyer les données. Néanmoins, il reste certains problèmes.

Dans un grand réseau commuté, le trafic multicast et broadcast peut être élevé ce qui pénalise les performances du réseau. Les commutateurs gèrent les domaines de diffusion de façon géographique. Ceci rend difficile le fait de déplacer une station d'un commutateur vers un autre. On observe aussi que les machines connectées sur un même port physique d'un commutateur (par l'intermédiaire d'un concentrateur par exemple) peuvent recevoir du traffic qui ne leur est pas nécessairement destiné. On peut également relever la vitesse de convergence du STP (Spanning Tree Protocol) qui s'avère peu élevée en cas de coupure d'un lien entre deux commutateurs.

Les VLANs peuvent apporter des solutions à ces problèmes. En effet, les VLANs permettent de créer des domaines de diffusion logiques, ce qui facilite le déplacement d'une station ainsi que la gestion du réseau. De plus, toutes les trames destinées à un VLAN particulier ne pourront pas être « écoutées » par des machines ne faisant pas partie du même VLAN. Il existe des solutions pour gérer plusieurs VLAN par le protocole STP.

Il y a trois méthodes principales de gestion pour les VLAN :

- Par port, un VLAN est attribué pour chaque port physique du Switch (niveau 1);
- Par adresse MAC, un VLAN est attribué pour une liste d'adresses MAC (niveau 2);
- Par adresse IP, un VLAN est attribué pour un sous-réseau (niveau 3).

La figure ci-dessous illustre le principe des VLAN. Les machines du VLAN1 ne pourront pas communiquer (au niveau 2) avec les machines du VLAN2 ou du VLAN3 même s'ils se trouvent sur le même commutateur.

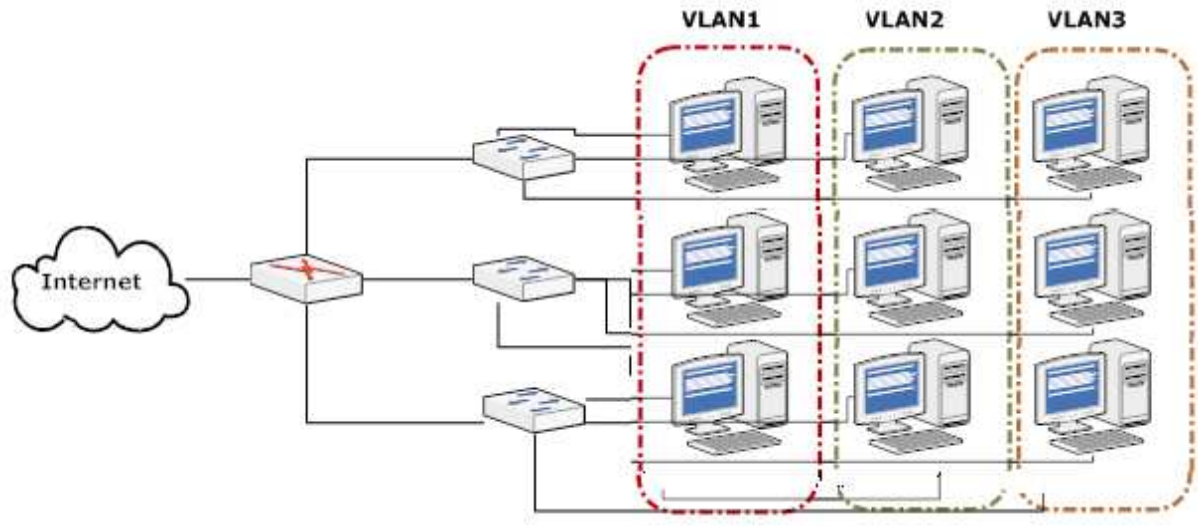


Figure 3: Exemple de vlan

Dans le cas d'un réseau Ethernet, pour savoir à quel VLAN une trame appartient, la norme « 802.1Q » propose de « tagger » l'entête Ethernet. Les trames « taggées » circulent uniquement entre les commutateurs, ce qui rend transparent l'utilisation des VLAN pour les clients. La figure 3 illustre une entête Ethernet standard et la figure 4 l'entête Ethernet « taggée ». La partie en gris correspond à ce que l'on ajoute à la trame Ethernet de base.

Adresse destination(6)	Adresse source(6)	Etype(2)	Données	FCS(4)

Figure 4: entête Ethernet standard

Adresse destination(6)	Adresse source(6)	Etype(=0x8100)	TPID(2)	Priorité(3bits)	CFI	VLAN ID (12 bits)	Etype(2)	Données	FCS(4)

Figure 5: entête Ethernet 802.1Q

VII. PROTOCOLE RADIUS

Présentation :

RADIUS (Remote Authentication Dial In User Service) est un protocole qui a été créé par Livingston et normalisé par l'IETF (Internet Engineering Task Force) sous la forme des RFCs 2138 et 2139. Le fonctionnement de RADIUS est basé sur un système client/serveur chargé de définir les accès d'utilisateurs distants à un réseau. Il s'agit du protocole de prédilection des fournisseurs d'accès à internet car il est relativement standard et propose des fonctionnalités de comptabilité permettant aux FAI de facturer précisément leurs clients.

RADIUS est un serveur de type AAA, c'est-à-dire qu'il se fait en trois étapes :

- Authentification : savoir qui parle au serveur RADIUS.
- Autorisation : savoir quelles autorisations sont accordées à l'utilisateur.
- Accounting : savoir ce que l'utilisateur fait. Savoir combien de temps l'utilisateur reste connecté au système pour assurer à la fois la journalisation des accès et la facturation.

Le protocole RADIUS repose principalement sur un serveur (le serveur RADIUS), relié à une base d'identification (fichier local, base de données, annuaire LDAP, etc.) et un client RADIUS, appelé NAS (Network Access Server), faisant office d'intermédiaire entre l'utilisateur final et le serveur. Le mot de passe servant à authentifier les transactions entre le client RADIUS et le serveur RADIUS est chiffré et authentifié grâce à un secret partagé.

Il est à noter que le serveur RADIUS peut faire office de proxy, c'est-à-dire transmettre les requêtes du client à d'autres serveurs RADIUS.

Dans ce guide, notre choix s'est porté sur freeradius. Freeradius est un serveur RADIUS libre open source. Il offre une alternative aux autres serveurs d'entreprise RADIUS, et est un des

serveurs RADIUS les plus modulaires et riches en fonctionnalités disponibles aujourd'hui. Freeradius a été lancé en Août 1999 par Alan DeKok et Miquel van Smoorenburg.

Protocole	UDP port 1812 : authentification/autorisation UDP port 1813 : gestion des comptes
Chiffrement	Chiffrement du mot de passe à l'aide du secret partagé
Architecture	Autorisations liées à l'authentification
Emission du profile	Profile global envoyé au NAS à la fin de l'authentification
Protocole non supportés	ARA et NetBEUI (NetBios Extended User Interface)
Challenge/réponse	Unidirectionnelle

Tableau 1 : Protocole RADIUS

Principe de fonctionnement :

Le fonctionnement de RADIUS est basé sur un scénario proche de celui-ci:

- Un utilisateur envoie une requête 802.1x au NAS afin d'autoriser une connexion au réseau.
- Le NAS achemine la demande au serveur RADIUS.
- Le serveur RADIUS consulte la base de données d'identification (un annuaire LDAP a été mis en place pour répondre à ce besoin) afin de connaître le type de scénario d'identification demandé pour l'utilisateur.

Le serveur RADIUS retourne ainsi une des quatre réponses suivantes pour ce qui est de l'authentification:

- ACCEPT : l'identification a réussi
- REJECT : l'identification a échoué
- CHALLENGE : le serveur RADIUS souhaite des informations supplémentaires de la part de l'utilisateur et propose un « challenge »
- CHANGE PASSWORD : le serveur RADIUS demande à l'utilisateur un nouveau mot de passe.

Suite à cette phase dite d'authentification, débute une phase d'autorisation où le serveur retourne les autorisations de l'utilisateur.

Trame RADIUS :

Une trame RADIUS se compose de 5 champs décrits dans le schéma ci-dessous :

Code	Identifiant	Longueur	Autrhentificateur	Attributs
1	1	2	16	n

Figure 6: Schéma d'une trame RADIUS

- Code : identifie le type du message.

Code	**Description**
1. Access-Request	Demande accès à un service
2. Access-Accept	Réponse favorable à la demande du client
3. Access-Reject	Réponse négative au client
4. Accounting-Request	Demande les informations d'authentification
5. Accounting-Response	Informations d'authentification
11. Access-Challenge	Sollicite des informations supplémentaires pour l'autorisation du client

Tableau 2 : Codes RADIUS

- Identifiant : permet de reconnaître les messages (requêtes et réponses) d'une même session d'authentification.
- Longueur : définie la longueur de la trame.
- Authentificateur : permet au client d'authentifier la réponse du serveur RADIUS et de protéger les mots de passe (évite le phénomène de « Man in the middle » par exemple). Il contient également la méthode d'authentification à utiliser avec le client.
- Attributs : ce champ est utilisé pour véhiculer toutes les informations nécessaires, il a pour format :

Type	Longueur	Valeur

Figure7 : Format des attributs RADIUS

Voilà qui conclue ce chapitre sur le fonctionnement des protocoles d'authentification. Dans le prochain chapitre, nous aborderons la mise en production d'un serveur *free*RADIUS sous Debian GNU/Linux.

VIII. ARCHITECTURE DE TEST

Notre maquette mise en place doit nous permettre de tester une authentification sur réseau filaire. Pour cela nous utilisons :

- un serveur d'authentification à base de free RADIUS et d'un annuaire LDAP
- un commutateur Hp Procurve (supportant bien sûre l'authentification 802.1x)
- des stations de travail sous Linux et Windows 7

Le serveur RADIUS devra pouvoir répondre à un certain nombre de contrainte lié aux différents utilisateurs de RADIUS. Après l'analyse de l'environnement, nous avons pu mettre en avant 4 types de profils utilisateurs:

- Profil « dot1x » : Client se connectant en 802.1x PEAP.
- Profil « cli » : NAS administrables avec identification RADIUS.
- Profil « VPN »: NAS distant demandant une connexion VPN via RADIUS.
- Profil « PPP » : Utilisateurs distants se connectant en PPPoE (Point-to-Point Protocol over Ethernet).

Le serveur RADIUS doit donc pouvoir répondre aux différentes attentes d'identification des profils listés ci-dessus.

❖ Free RADIUS :

L'objectif de cette étape est de mettre en œuvre un serveur freeRADIUS répondant à tous les besoins énoncés précédemment sous un système d'exploitation GNU/Linux Debian. Nous verrons donc dans une première partie comment installer et démarrer freeradius, ensuite sa configuration.

1.1. Installation et test de démarrage

a. Installation :

- Téléchargez la source de freeradius depuis son site officiel http://freeradius.org/download.html

```
tar zxvf freeradius-server-2.1.10.tar.gz

./configure

make

make install
```

- La configuration par défaut est conçue pour fonctionner avec presque toutes les méthodes d'authentification. Donc Une fois le serveur est installé, la première étape consiste à le lancer en mode débogage, en tant que root:

```
radiusd    -X
```

- Cette étape démontre que le serveur a été installé et configuré correctement.

b. Test de démarrage :

- Editez le fichier « users » et ajoutez la ligne suivante en haut du texte, avant toute autre chose

```
testing Cleartext-Password := "password"
```

- Démarrez le serveur en mode débogage (radiusd -X), et à partir d'un autre terminal exécutez radtest.

```
radtest testing password 127.0.0.1 0 testing123
```

- Vous devez voire le serveur vous répond avec « Access-Accept ». sinon vous trouverez l'erreur dans les logs de débogage.
- Une fois les paquets installés, reste à configurer le serveur RADIUS pour répondre à nos besoins. La partie suivante décrit la configuration de freeradius.

1.2. Fichiers de configuration principaux :

Les fichiers de configuration de freeradius se situe dans le répertoire /etc/freeradius, ces fichiers sont très bien commentés et constituent la documentation de freeradius. La section suivante présente les fichiers de configuration principaux:

- « radiusd.conf » pour la configuration globale du serveur. Ce fichier est découpé en deux grandes parties, d'abord les paramètres propres au démon (interfaces d'écoute, port, etc.), puis une partie définition des modules (définition et configuration des modules d'authentification

disponibles hormis ceux du type EAP qui sont traités séparément, des modules de journalisation, de relayage des requêtes, etc.). C'est par exemple dans ce fichier que seront configurés les paramètres (adresses IP, mot de passe, etc.) d'une éventuelle base de données LDAP contenant les noms d'utilisateur et leur mot de passe associé.

- « eap.conf » pour la configuration des méthodes EAP d'authentification. Le contenu de ce fichier était au départ inclus dans la partie module du fichier « radiusd.conf » mais les développeurs ont préféré le séparer pour des raisons de lisibilité car il devenait de plus en plus volumineux du fait du nombre de méthodes d'authentification EAP différentes. En fonction des méthodes EAP que le serveur devra supporter dans son environnement de production il y aura éventuellement certains paramètres à configurer. Par exemple dans le cas d'une authentification via EAP-TLS, il faudra indiquer le répertoire contenant le certificat du serveur (qu'il enverra au supplicant) et la clé privée avec le mot de passe associé, celui contenant le certificat de l'autorité (qui permettra de vérifier le certificat fourni par le supplicant), indiquer si le serveur doit vérifier un fichier contenant les certificats révoqués ou encore s'il faut vérifier que le nom de l'utilisateur correspond au nom du propriétaire du certificat fourni.
- « clients.conf » pour définir et paramétrer le dialogue avec les authentificateurs. Ici sont recensés les authentificateurs via un nom, une adresse IP et un secret partagé. D'autres informations optionnelles peuvent être ajoutées pour éviter les connexions simultanées d'un même utilisateur.
- « users » est le fichier des utilisateurs. Un utilisateur est défini par son nom et sa méthode d'authentification (en fonction des méthodes, ce fichier peut contenir des mots de passe). On peut aussi y mettre des directives post-authentification, comme une attribution de VLAN.

Exemple de déclaration d'un utilisateur :

```
"Med Ben Salah" Auth-Type := EAP

Reply-Message = "Authentification réussie : VLAN ID 1 – permanent",

Tunnel-Type = 13,

Tunnel-Medium-Type = 6,

Tunnel-Private-Group-Id = 3
```

Ici l'utilisateur « Med Ben Salah »doit s'authentifier via une méthode EAP, en fonction de ce que le supplicant lui fournit, le serveur décide de la méthode EAP adéquate ; ce fichier peut contenir le mot de passe associé à l'utilisateur dans le cas où le serveur RADIUS n'est pas interfacé avec une base de données ou avec le fichier de mot de passe Unix (/etc/passwd). Si Med Ben Salah arrive à

s’authentifier alors il recevra le message "Authentification réussie : VLAN ID 1 - permanent" et le port de l’authenticator où sa machine est connectée sera affectée au VLAN n°1.

A noter qu’il est possible de créer un script relançant le démon « radiusd » à chaque modification du fichier « users » afin que le serveur freeRADIUS prenne en compte les ajouts ou suppressions d’utilisateurs.

Le serveur accueillant freeradius doit être dédié à cette tâche, il ne doit pas proposer d’autres services car en répondant aux requêtes d’authentification des utilisateurs il prend une place critique dans l’architecture du réseau. Il devra être redondant afin de ne pas bloquer l’accès au réseau en cas de panne et son accès devra être strictement limité aux authentificateurs. Les commutateurs et points d’accès compatibles 802.1x proposent toujours d’interroger un deuxième serveur RADIUS en cas de non réponse du premier.

1.3. Configuration de freeradius

La configuration de freeradius se fait en quatre étapes :

- clients.conf,

- radiusd.conf,

- eap.conf,

- users.

- ***/usr/local/etc/raddb/clients.conf***

  - Définir l’adresse IP du client. Pour les tests, nous utiliserons d’abord l’adresse locale 127.0.0.0. Puis pour communiquer avec un client de l’extérieur, nous mettrons l’adresse IP de la machine.
  - Choisir un mot de passe. Ce mot de passe sera l’adresse MAC de la machine de l’utilisateur.
  - Prendre l’adresse MAC qui sera son login

```
client 192.168.0.0/16 {

secret =

shortname =
```

```
nastype =

}
```

- ***/usr/local/etc/raddb/radiusd.conf***

Dans ce fichier il faut choisir différentes options possibles :

Nous commencerons par modifier les modules authorize et authenticate.

- Dans **« *authorize* »** on définit les autorisations pour :

- preprocess : modifier les attributs dans un format standard.

- chap : accepter le protocole CHAP.

- suffix : utiliser pour ne pas tenir compte de tous les domaines existants.

- eap : placer les attributs d'EAP-TYPE dans la liste de demande au type d'EAP.

- files : lire le fichier users

```
authorize {

preprocess

chap

suffix

eap

files

}
```

- Dans ***« authenticate »***, on définit :

- Auth-Type LDAP permet l'authentification LDAP.

- eap permet l'authentification EAP.

```
authenticate {

Auth-Type LDAP {

ldap

}

eap

}
```

- L'étape suivante dans radiusd.conf consiste à choisir un fonctionnement d'authentification, pour freeradius. Nous utilisons CHAP auquel nous avons fait allusion dans authorize et authenticate.

```
chap {

authtype = CHAP {

}
```

CHAP est un protocole d'authentification basé sur la résolution d'un « challenge ». Celui-ci se déroule en plusieurs étapes, le serveur envoie au client un nombre aléatoire de 16 bits. Puis le client crypte son mot de passe en md5 grâce à ce chiffre, et le renvoie sur le réseau. Le serveur compare le message reçu avec celui d'origine en crypté et s'ils sont égaux la connexion peut avoir lieu.

- ***/usr/local/etc/raddb/eap.conf***

Plusieurs protocoles EAP sont utilisés dans freeRADIUS : EAP-TLS, LEAP, PEAP, EAP-TTLS, etc…

En général, avec freeRADIUS, on utilise LEAP malgré son problème de vulnérabilité aux attaques. Il est rare que freeRADIUS fonctionne seul comme système d'authentification. Nous avons déjà vu son utilisation associée à une authentification CHAP.

```
Default_eap_type = peap

tls {

private_key_password = SecretKeyPass77
```

```
private key file = $ {raddbdir} /certs/cert-serv.pem

CA_file = $ {raddbdir} /certs/demoCA/cacert.pem

dh_file = $ {raddbdir} /certs/dh

random_file = /dev/urandom

}
```

PEAP emploie TLS, donc nous configurons le module tls. Nous devons choisir le mot de passe de la clé privée et ainsi nous obtenons la clé privée. Ensuite un système de confiance avec les fichiers CA_file et dh_file est obtenu.

Puis, nous passons à la configuration du module PEAP pour le renseigner sur le système d'authentification utilisé.

```
peap {

default_eap_type = chap

}
```

Nous précisons à PEAP que l'on va utiliser CHAP.

- ***/usr/local/etc/raddb/users***

Dans ce fichier, on rajoute les utilisateurs inscrits dans le fichier clients.conf comme par exemple :

```
"utilisateur1″ Auth-Type:=EAP, User-Password==”topsecret”

"utilisateur2″ Auth-Type:=Local, User-Password==”monmotdepasse”

"utilisateur3″ Auth-Type:=CHAP, User-Password==”monmotdepasse”
```

À la place de « User-Password » on trouvera le nom d'utilisateur et le mot de passe du client. Dans le premier cas, l'utilisateur emploie le système d'authentification EAP et dans le second l'utilisateur est en local : c'est-à-dire qu'il est connecté à la machine. Dans le troisième cas l'utilisateur se sert du système d'authentification CHAP.

1.4. Démarrage de freeradius

Une fois le serveur freeradius configuré nous pouvons le mettre en marche grâce à la commande suivante :

```
/etc/init.d/freeradius start
```

L'ajout d'un utilisateur se fait sur une base de données donc elle ne peut être réalisée sur freeRADIUS qui n'en possède pas. C'est pour cela qu'on a besoin d'un annuaire afin de gérer les utilisateurs. Dans notre guide, on va implémenter l'annuaire OpenLDAP.

2. OpenLDAP :

L'objectif de cette étape est la mise en place d'un serveur OpenLDAP permettant au serveur freeradius de rechercher ses utilisateurs dans un annuaire. Nous verrons donc, tout d'abord, le fonctionnement d'un annuaire LDAP. Ensuite l'installation et la configuration du serveur OpenLDAP.

2.1. Définitions :

- Un annuaire est un système de stockage de données, dérivé des bases de données hiérarchisées, permettant en particulier de conserver les données pérennes, c'est-à-dire les données n'étant que peu mises à jour (historiquement, sur une base annuelle, d'où le nom), comme les coordonnées des personnes, des partenaires, des clients et des fournisseurs d'une entreprise. C'est pourquoi, grâce à des optimisations, un annuaire est beaucoup plus rapide en consultation qu'en mise à jour.
- LDAP (Lightweight Directory Access Protocol) est un protocole permettant l'interrogation et la modification des services d'annuaire. Ce protocole repose sur TCP/IP. L'intérêt principal de LDAP est la normalisation de l'authentification. C'est l'opération Bind qui permet d'authentifier un utilisateur. De plus en plus possèdent un module d'authentification prenant en charge LDAP. C'est le cas du serveur freeradius.

2.2. Fonctionnement du protocole LDAP :

LDAP est basé sur un fonctionnement client-serveur (comme pour une base de données). Un client débute une session LDAP en se connectant sur le port TCP 389 du serveur. Le client envoie ensuite des requêtes d'opération au serveur.

Le serveur envoie des réponses en retour. À part quelques exceptions, le client n'a pas besoin d'attendre de réponse du serveur pour envoyer de nouvelles requêtes, et le serveur peut envoyer ses réponses dans n'importe quel ordre.

Ainsi LDAP fournit à l'utilisateur des méthodes lui permettant de :

- ✓ se connecter
- ✓ se déconnecter
- ✓ rechercher des informations
- ✓ comparer des informations
- ✓ insérer des entrées
- ✓ modifier des entrées
- ✓ supprimer des entrées

2.3. Installation:

a. Pré requis :

Installation de la base Berkeley DB : vous pouvez le téléchargez la source depuis ce lien http://www.oracle.com/technology/software/products/berkeley-db/index.html ou à l'aide de gestionnaire de paquet de debian comme si dessous :

```
Apt-get install libdb4.6-dev
```

b. Installation:

Dans cette partie, nous allons traiter l'installation d'un serveur OpenLDAP sous debian. Téléchargeons-nous la dernière version stable depuis ce lien http://www.openldap.org/software/download

```
gunzip -c openldap-2.4.23.tgz | tar xvfB -

cd openldap-2.4.23

./configure

Make depend

Make

Make test

Make install
```

2.4. Configuration :

- Editez votre fichier de configuration « slapd.conf » à l'aide de votre éditeur préféré afin de configurer la base.

Vi /usr/local/etc/openldap/slapd.conf

```
Database   bdb
suffix     "dc=<MY-DOMAIN>,dc=<COM>"
rootdn     "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>"
rootpw     secret
directory  /usr/local/var/openldap-data
```

Remplacez alors <MY-DOMAIN> et <COM> par le nom de votre domaine comme suit dans notre guide :

```
Database   bdb
suffix     "dc=<ansi>,dc=<tn>"
rootdn     "cn=Manager,dc=<ansi>,dc=<tn>"
rootpw     secret
directory  /usr/local/var/openldap-data
```

2.5. Test et Démarrage du serveur :

a. Test de configuration : Tester en interrogeant l'annuaire

« -x » : sans utiliser SASL

« -b » : avec la base abdomain.org

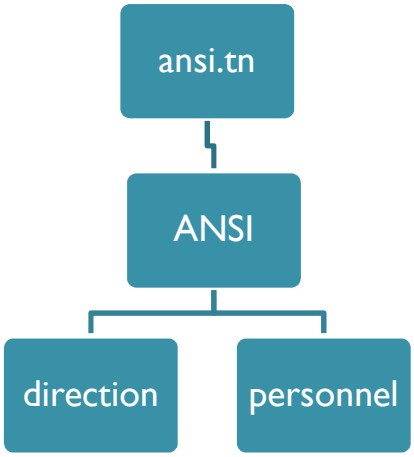
```
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

b. Démarrage du serveur:

- Le lancement du démon « slapd » s'effectue grâce à cette commande :

```
/usr/local/libexec/slapd
```

2.6. Organiser l'annuaire



Afin d'ajouter des Objets (organisation, groupe, utilisateur) dans votre annuaire LDAP, vous pouvez utiliser la commande « ldapadd ». Mais cette commande demande en paramètre un fichier avec extension « LDIF ».

Voici la syntaxe de la commande ldapadd :

-x : ne pas utiliser SASL

-D : avec le login manager@test.com

-W : s'authentifier par mot de passe

-f : utiliser le fichier test.ldif

a. Ajout d'un objet dans l'annuaire :

- Insérez les lignes suivantes « /usr/local/etc/openldap/test.ldif » en remplaçant <MY-DOMAIN>, <COM> et <MY ORGANIZATION> par le votre :

```
dn: dc=<MY-DOMAIN>,dc=<COM>
objectclass: dcObject
objectclass: organization
o: <MY ORGANIZATION>
dc: <MY-DOMAIN>

dn: cn=Manager,dc=<MY-DOMAIN>,dc=<COM>
objectclass: organizationalRole
cn: Manager
```

- Exécutez la commande ldapadd :

```
ldapadd -x -D "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>" -W -f  /usr/local/etc/openldap/test.ldif
```

- Vérifiez maintenant si cette entrée est bien ajoutée à l'annuaire :

```
ldapsearch -x -b 'dc=ansi,dc=tn' '(objectclass=*)'
```

b. Ajout de groupe dans l'annuaire LDAP:

- Insérer un groupe de classe posixGroup dans personnel.ansi.test.com. Editer un fichier gr.ldif

```
dn: cn=mongroupe, ou=personnel, ou=ANSI, dc=ansi, dc=tn

objectClass: top

objectClass: posixGroup

cn: mongroupe

gidNumber: 1111
```

- Ajouter le groupe

```
ldapadd -x -D "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>" -W -f  /usr/local/etc/openldap/gr.ldif
```

c. Ajouter un utilisateur dans l'annuaire

- Insérer un utilisateur nommé « Ben Salah » de classe posixAccount dans etudiants.people.test.com avec les attributs uid, uidNumber, gidNumber, homeDirectory, shell ... Editer un fichier ut.ldif

```
dn: uid=Ben Salah, ou=personnel, ou=ANSI, dc=ansi, dc=tn

objectClass: top

objectClass: posixAccount

objectClass: person

objectClass: organizationalPerson

objectClass: inetOrgPerson

uid: Ben Salah

cn: Ben Salah Med

sn: Ben Salah

givenName: Med

uidNumber: 1100

gidNumber: 1111

homeDirectory: /home/user/Ben Salah

loginShell: /bin/bash

userPassword: poiuyt

mail: Ben Salah@abdomain.org

l: Tunisie

ou: mongroupe
```

- Ajouter l'utilisateur Ben Salah

```
ldapadd -x -D "cn=Manager,dc=<MY-DOMAIN>,dc=<COM>" -W -f  /usr/local/etc/openldap/ut.ldif
```

- Redémarrer le serveur LDAP pour prendre en compte les nouvelles définitions

```
Killall slapd

/usr/local/libexec/slapd
```

3. Association de RADIUS et LDAP

Rappelons que la structure des données d'un annuaire est donnée par des schémas. Nous souhaitons utiliser notre annuaire LDAP pour l'identification avec un serveur RADIUS. Il nous faut donc prendre en compte un schéma qui contiendra tous les attributs nécessaires à RADIUS. Ce schéma est un fichier nommé RADIUS-LDAPv3.schema, il se trouve dans le répertoire suivant: /usr/share/doc/freeradius/examples et doit être copié dans le répertoire /etc/ldap/schema.

```
user@client:~$ sudo cp openldap.schema /etc/ldap/schema/RADIUS-LDAPv3.schema
```

Une fois le schéma copié, il nous faut modifier le fichier de configuration slapd.conf pour prendre en compte ce schéma et paramétrer les droits d'administration de l'annuaire LDAP.

```
user@client:~$ sudo nano /etc/ldap/slapd.conf
```

4. Configuration du commutateur:

Les principaux paramètres à prendre en compte dans la configuration des équipements de commutation sont :

- Les paramètres d'accès au serveur Radius
- Les paramètres de configuration des ports

Il est à noter que la configuration du 802.1X sur les équipements de commutation peut s'effectuer de façon globale. Il est alors primordial de prendre en compte les interactions entre les divers commutateurs interconnectés (paramètre « dot1x system-auth-control »).

Dans ce guide on a utilisé un commutateur « hp procurve ». Tout d'abord il faut définir le serveur RADIUS sur le commutateur, puis vous spécifiez le protocole d'authentification à utiliser. Ensuite, vous définissez les ports à authentifier, et enfin vous activez les ports.

```
ProCurve Switch 2810-24G(config)# radius-server host 10.1.10.10 key procurve
```

```
ProCurve Switch 2810-24G(config)# aaa authentication port-access eap-radius

ProCurve Switch 2810-24G(config)# aaa port-access authenticator 1-24

ProCurve Switch 2810-24G(config)# aaa port-access authenticator active

ProCurve Switch 2810-24G(config)# write mem
```

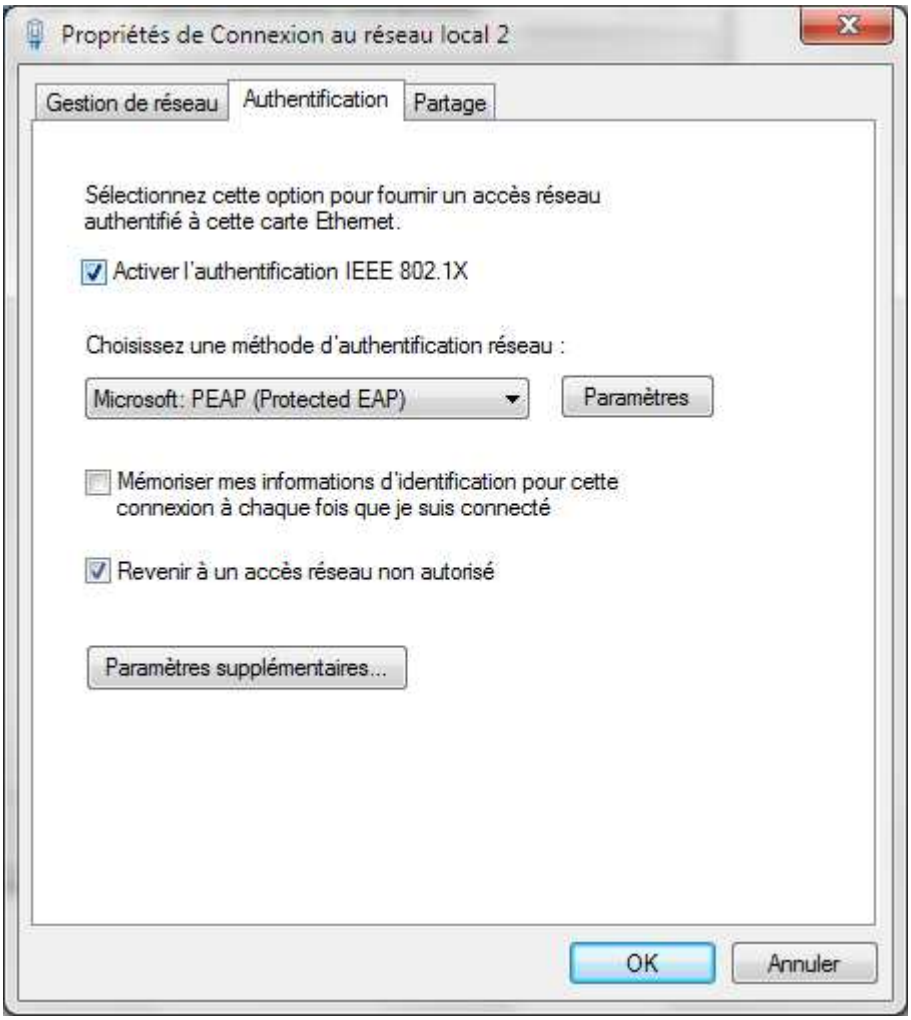
5. Configuration des postes clients:

De façon à pouvoir se connecter à travers un port utilisant du 802.1X, il est nécessaire que le poste client supporte le protocole. Le support peut-être soit natif, soit obtenu grâce à un logiciel client.

5.1. Client Windows :

Sous Windows xp, vista et 7 le client d'authentification 802.1x est intégré par défaut mais il faut juste l'activer comme suit :

- Cliquez sur le bouton Démarrer. Dans la zone de recherche, tapez « **services.msc** », puis appuyez sur Entrée. Si vous êtes invité à fournir un mot de passe administrateur ou une confirmation, fournissez le mot de passe ou la confirmation.
- Dans la boîte de dialogue Services, cliquez sur l'onglet Standard dans le bas du volet principal, cliquez avec le bouton droit sur « **Configuration automatique de réseau câblé** » puis sur Démarrer.
- Ouvrez les Connexions réseau.
- Cliquez avec le bouton droit sur la connexion pour laquelle vous souhaitez activer l'authentification 802.1X, puis cliquez sur Propriétés. Si vous êtes invité à fournir un mot de passe administrateur ou une confirmation, fournissez le mot de passe ou la confirmation.
- Cliquez sur l'onglet Authentification, puis activez la case à cocher Activer l'authentification IEEE 802.1X.



- Dans la liste Choisissez une méthode d'authentification réseau, cliquez sur la méthode à utiliser.

5.2. Client Linux :

Sous les systèmes d'exploitation GNU/linux, il faut utiliser un client d'authentification 802.1x tel que le client « Xsupplicant » qui est un outil open source et vous pouvez le téléchargé depuis son lien suivant : http://open1x.sourceforge.net/. Ce client supporte tous les types d'authentification EAP actuellement existants (EAP-PEAP inclus).

Après une installation classique, le fichier exécutable Xsupplicant se trouve dans« /usr/local/sbin », le fichier de configuration xsupplicant.conf dans « /etc/ ». C'est dans ce fichier de configuration qu'il faut indiquer le répertoire contenant les certificats (ceux de l'utilisateur et de l'autorité), indiquer son nom d'utilisateur et son mot de passe en fonction de la méthode EAP choisie. Dans sa version actuelle Xsupplicant ne propose pas d'interface graphique pour sa configuration, tout doit se faire « à la main». Par défaut le fichier de configuration doit contenir les mots de passe, le processus d'authentification se fait alors sans l'intervention de l'utilisateur, mais il est possible d'utiliser l'exécutable « xsup_monitor

» fourni avec Xsupplicant, qui demande de saisir le mot de passe au cours de l'authentification dans un terminal, ceci permet d'éviter de laisser son mot de passe dans un fichier non crypté.