SIGN&GO

RAPPORT DE STAGE :

DEVELOPPEMENT D'UN MODULE D'AUTHENTIFICATION PAR OTP POUR UN LOGICIEL DE GLOBAL SSO

VINCENT DEVILLIERSE

L3 INFORMATIQUE INSTITUT GALILEE 2010

RESPONSABLE DE STAGE : CEDRIC SZANIEC

Contact

ILEX

51 boulevard Voltaire

92600 Asnières-sur-Seine

Téléphone : +33 1 46 88 03 40

Fax : +33 1 46 88 03 41

support@ilex.fr

http://www.ilex.fr

Information légale

Sign&go est une marque déposée de la société Ilex. Toutes les autres marques citées dans ce document sont la propriété de leurs sociétés respectives.

Ce document est fourni uniquement à titre d'information. En aucun cas, le contenu du document, sous quelque forme que ce soit, ne peut engager la responsabilité de la société Ilex. Toutes ces spécifications ou données peuvent être modifiées sans préavis.

En vertu de l'article L. 122-4 du Code de la Propriété Intellectuelle, toute représentation totale ou partielle de ce document, par quelque procédé que ce soit, sans l'autorisation expresse de la société Ilex, est interdite et constituerait une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la Propriété Intellectuelle.

Copyright Ilex 2009. Tous droits réservés.

1 REMERCIEMENTS

Tout d'abord, je souhaite remercier M René Lagrèze, Directeur général, pour m'avoir accordé sa confiance en m'accueillant en tant que stagiaire au sein de la société Ilex.

Je remercie aussi vivement M Jean-Pierre Astruc et Mme Virginie Gueguen pour m'avoir aidé dans ma recherche de stage et dirigé vers la société Ilex.

Naturellement, mes remerciements vont aussi à mon responsable de stage, M Cedric Szaniec, qui a accepté de me prendre en charge et de me confier des missions intéressantes.

Je remercie également toute l'équipe Sign&Go, M Szaniec n'ayant pu être là pendant toute la durée de mon stage, ils ont formidablement pris le relai. Leurs impératifs professionnels respectifs m'ont permis de travailler avec chacun d'eux, ce qui fut vraiment agréable. Je remercie donc M Reza Tahami pour l'immense patience et la pédagogie dont il a fait preuve à mon égard, Mme Stéphanie Martel pour son aide et les idées de réalisations dont elle m'a fait profiter, ainsi que M Vincent Desbiendras pour son assistance et les nombreuses informations qu'il m'a donné sur la vie en entreprise et celle d'un informaticien.

Enfin, j'adresse mes remerciements à l'ensemble des salariés de la société Ilex, qui ont toujours pris le temps de m'aider, de répondre à mes questions et dont l'excellent accueil m'a permis de me sentir tout de suite à l'aise dans l'entreprise.

SOMMAIRE

2 INTRODUCTION

Le cycle de Licence est un cycle que l'on considère souvent comme relativement généraliste, aussi, est-il d'autant plus important de pouvoir compléter sa formation théorique avec des expériences pratiques de vie en entreprise.

Effectuer un stage fut pour moi l'occasion de mettre en application mes connaissances au service d'une entreprise, d'en acquérir de nouvelles, et, au delà de l'aspect technique, d'en apprendre plus sur le monde du travail et de l'informatique en entreprise.

La recherche d'un stage n'a pas toujours été simple, tant par le contexte économique que par le fait que les entreprises soient encore peu préparées à l'accueil des étudiants issus de Licence. Les propositions n'étaient pas très nombreuses et peu adaptées (pas de missions vraiment définies ou pas d'encadrement possible). C'est pourquoi je souhaite adresser mes plus sincères remerciements à Mme Virginie Gueguen et M Astruc qui ont pu me diriger vers la société Ilex, qui, au terme d'un entretien, a accepté de me prendre en tant que stagiaire et de me confier une mission qui fut extrêmement intéressante.

Au cours de cet entretien, le directeur, Monsieur Lagreze, et moi-même avons fait un bilan de mes connaissances pour définir quelle mission pourrait nous convenir à tous deux. Progressivement, nous nous sommes alors dirigés vers une mission de développement en C/C++ au sein de l'équipe responsable du logiciel Sign&Go. L'objectif était de développer un module d'authentification par OTP (One Time Password) pour ce logiciel.

C'est une véritable chance pour moi d'avoir pu effectuer un stage chez un grand éditeur logiciel qui a cru en moi en me confiant une mission passionnante à réaliser.

Dans ce rapport, je vous présenterai en premier lieu la société Ilex, ses produits et plus particulièrement le logiciel Sign&Go sur lequel j'ai travaillé, afin de mieux comprendre les enjeux de ma mission, et je finirai par décrire plus précisément l'ensemble de mes réalisations.

3 LA SOCIETE ILEX

Ilex est une Société de Services en Ingénierie Informatique (SSII) et un éditeur de logiciels, principalement axé autour de la sécurité et de la gestion des identités. La société a été créée en 1989 par Laurent Gautier et René Lagreze. Son nom est la concaténation d'un I pour Informatique et de lex, la loi en latin, le premier sigle de la société était ainsi une balance. Ilex est aussi le nom latin du houx, un grand houx trône d'ailleurs dans la cour intérieure des locaux ! A noter qu'il fut un temps où l'entreprise disposait d'un journal interne baptisé "La feuille de houx"…

3.1 Produits

La société Ilex édite plusieurs logiciels :

- **Meibo** : un gestionnaire de contenu d'annuaire. Un annuaire d'entreprise, exploité par Meibo, permettra l'authentification des utilisateurs, la gestion des droits et sera également le garant des procédures de l'entreprise. Meibo permet donc de réaliser en toute simplicité des applications de gestion de contenu d'annuaire.
- **Meibo People Pack** est une solution de gestion des identités couvrant le parcours des personnes (arrivées, départs,…) et l'allocation de leurs ressources de travail (badges, ordinateurs portables, messagerie, comptes applicatifs,…). Meibo People Pack, solution de gestion des identités prête à l'emploi, est le produit phare actuel de la société. Il a été bâti avec Meibo.
- **Sign&Go** est une application de SSO "Single Sign-On". Elle permet de gérer l'authentification sur différentes applications et sur des systèmes hétérogènes sans qu'il ne soit nécessaire de saisir son identité pour chacune d'elles. Une version spécifiquement adaptée au monde de la santé, Sign&Go Santé a également été créée en s'appuyant sur les cartes CPS et Vitale pour réaliser les authentifications. C'est sur ce logiciel que j'ai travaillé et la section suivante lui sera consacrée.

3.2 Partenaires

Ilex est très présent sur le marché de la santé avec des solutions adaptées aux professionnels de santé (prise en charge de la carte CPS --Carte de Professionnel de Santé-- comme moyen d'authentification par Sign&Go), mais travaille aussi avec de grands groupes privés ou publics comme le montre le schéma suivant.

3.3 Services

La société compte actuellement 49 personnes (avec 2 stagiaires, moi compris) répartis au travers de 6 services :

- Direction générale
- Administration
- Marketing-Communication
- Commercial
- Avant-vente
- Pôle technique

Le Pôle technique est le service regroupant le plus de personnes, c'est à lui que j'ai été rattaché, au sein de l'équipe responsable de Sign&Go.

J'avais donc pour collaborateurs directs Monsieur Cedric Szaniec, mon responsable de stage et responsable produit sur le projet Sign&Go, Mme Stéphanie Martel et M Réza Tahami travaillant dans l'équipe Sign&Go ainsi que Monsieur Vincent Desbiendras intégrateur Sign&Go.

Tous ont été d'une extrême gentillesse avec moi, toujours prêts à prendre de leur temps pour m'aider et me guider sur mon travail ou pour répondre à mes questions sur la société Ilex, sur la vie d'un informaticien en entreprise et les différentes facettes de leur activité.

J'ai eu la chance de bénéficier d'un espace de travail (bureau, ordinateur et téléphone personnels) dans les mêmes conditions que n'importe quel employé ce qui m'a aidé à me sentir tout de suite à l'aise et à m'intégrer au sein de l'entreprise.

L'entreprise offre des conditions de travail très agréables à ses employés, avec des équipements ménagers partagés, une cour intérieure où déjeuner ou prendre une pause… On sent que tous les employés travaillent avec plaisir et parfois sans compter leur temps. Il en résulte une ambiance tout aussi agréable que dynamique, ce qui a représenté un excellent moteur de motivation pour moi tout au long de ce stage.

4 LE LOGICIEL SIGN&GO

4.1 Identités et sécurité

Dans une entreprise, les employés se doivent souvent d'utiliser différentes applications nécessitant une authentification. En effet, tous les utilisateurs ne bénéficient pas toujours des mêmes droits sur certaines données et il est important de pouvoir identifier les utilisateurs de manière sure, afin de vérifier à qui doivent être donnés certains accès ou droits.

C'est par cette identification qu'un utilisateur va pouvoir prouver son identité et accéder à un service avec les droits qui lui ont été accordés. La sécurité autour des ces identifications est vitale pour la protection des données et la bonne marche de l'entreprise.

Plusieurs moyens d'authentification existent, le plus connu étant le système du login/mot de passe. Il est extrêmement important de protéger son mot de passe, s'il est récupéré par un individu malveillant, celui-ci pourra bénéficier des mêmes accès que l'employé et potentiellement nuire à l'entreprise (destruction, modification ou vol de données ou d'informations critiques).

Si il existe des recommandations usuelles dans le choix du mot de passe (il ne doit pas exister dans un dictionnaire, comporter au moins un chiffre et un caractère spécial, être suffisamment long et changé régulièrement, être différent d'une application à l'autre…), il est parfois difficile de trouver un compromis entre un mot de passe respectant ces conditions… mais dont on se rappelle sans peine ! Et ce d'autant plus si l'utilisateur doit définir plusieurs mots de passe pour plusieurs applications et en changer régulièrement.

La situation devient alors souvent difficile à gérer pour l'employé, qui peut être tenté d'abaisser le niveau de sécurité de ses mots de passe (utiliser le même sur plusieurs applications, en choisir des plus simples, voire même les écrire "sous le clavier") compromettant ainsi la sécurité de la société. Cette gestion des mots de passe présente également un coût important pour une entreprise, estimé entre 60 et 90 euros par utilisateurs selon les analyses.

4.2 La solution apportée par Sign&Go

Le logiciel Sign&Go s'inscrit dans le cadre de la réponse à cette problématique. C'est un logiciel de SSO "Single Sign-On" (authentification unique). Comme son nom l'indique, il va permettre à l'utilisateur de s'identifier une seule fois pour toutes les différentes applications auxquelles il voudrait avoir accès.

L'administrateur en charge va associer manuellement le compte de l'utilisateur à une série de login/mot de passe correspondant aux applications susceptibles d'être utilisées par l'employé. Le logiciel reconnait les fenêtres d'invite d'authentification et injecte directement les login/mot de passe associés au compte avant de valider. Ce mécanisme se fait de manière totalement transparente pour l'utilisateur. Le logiciel peut également faire de l'auto apprentissage pour retenir de nouveaux identifiants ou changer les anciens. Dans tous les cas, l'utilisateur n'aura plus qu'un seul mot de passe à retenir et la sécurité, centralisée, sera alors plus simple et plus performante.

Cependant, pour que ce système soit fonctionnel, il convient évidemment d'apporter toute son attention sur l'authentification première de l'utilisateur. Cette "super authentification" doit se faire selon ce que l'on appelle une authentification "forte", extrêmement sécurisée. Sign&Go propose plusieurs modes opératoires pour cette authentification.

La méthode par login/mot de passe reste disponible, même si ce n'est pas la plus sécurisée. Il est possible de rajouter certaines contraintes, comme une obligation de changer de mot de passe après un certain temps ou un certain nombre d'utilisations. Il est conseillé d'utiliser une "politique de mot de passe" assez stricte.

Un autre moyen est de s'authentifier à l'aide d'une carte à puce. Les informations concernant l'utilisateur sont stockées sur la carte et accessibles si le bon code PIN est rentré. C'est un des moyens d'authentification les plus surs car il impose un objet physique personnel ainsi qu'un code dont seul l'employé a normalement connaissance. La carte CPS est aussi une carte à puce et est utilisée par les professionnels de santé (CPS = Carte de Professionnel de Santé) dans la version de Sign&Go dédiée à la santé.

Il existe encore d'autres moyens, comme des clefs USB, des badges… C'est au client de déterminer quels moyens il souhaite utiliser dans sa version de Sign&Go.

J'ai travaillé pour ma part sur une authentification par OTP. OTP signifie "One Time Password". L'administrateur associe chaque utilisateur à un boitier qui va générer un mot de passe (ici sous forme de nombre pseudo aléatoire) dont la validité sera limitée dans le temps et à un seul essai. Ainsi, si l'utilisateur attend trop avant d'utiliser son OTP, celui ne sera plus valide ; de même après que le code OTP ai été envoyé au serveur gérant l'authentification. L'avantage apporté par cette technique est plus que conséquent, en effet, même si un individu arrivait à intercepter l'OTP lors de son envoi, il lui serait inutile. Contrairement à une authentification par login/mot de passe classique, le vol du mot de passe sur le réseau ne présente aucun intérêt pour un pirate. L'authentification par OTP combine la nécessité de posséder un objet physique à un mot de passe qui n'a pas besoin d'être retenu, dont la perte ne présente aucun risque et qui est changé à chaque fois : il s'agit donc bien d'une authentification forte !

4.3 Fonctionnement de l’authentification par OTP

On peut séparer les différents éléments de SNG en 3 parties : les composants se trouvant sur le poste de l’utilisateur, le Serveur de Sécurité de Sign&Go et le ou les serveurs faisant l'authentification (voir le schéma en Annexe page 21).

L'authentification sera, dans notre cas, réalisée par un serveur d'authentification Radius communiquant avec un serveur de la société Gemalto (éditrice des boitiers OTP) afin de vérifier la bonne combinaison du code OTP et du nom d'utilisateur.

La plupart des entreprises utilise le système d'exploitation Windows, et le logiciel s'y adapte particulièrement bien en remplaçant la GINA Windows (*Graphical Identification and Authentication*, la fenêtre d'authentification Windows) par une GINA Sign&GO. Parfois, certains clients préfèrent conserver la GINA Windows et il est alors possible de s'authentifier auprès de Sign&GO par des composants présents sur le poste après ouverture de sa session par l'utilisateur.

Dans tous les cas, le cheminement restera le même. L'utilisateur va d'abord sélectionner un mode d'authentification, une requête est alors envoyée au Serveur de Sécurité afin de connaître la liste des schémas d'authentification associés à ce mode (par exemple pour une authentification par login/mot de passe, il y a un schéma différent pour les identifications en tant qu'utilisateur ou administrateur). Dans notre cas, il n'y a qu'un seul schéma d'authentification disponible : une authentification par serveur Radius, aussi la question n'est elle pas posée à l'utilisateur et ce schéma est choisi par défaut de manière invisible pour l'utilisateur.

Une demande va être envoyée au Serveur de Sécurité, comprenant le type d’authentification que l’on désire effectuer (avec le schéma choisi ou le schéma par défaut si il n'y en a qu'un) et les paramètres associés. Dans notre cas, on demandera donc au Serveur de Sécurité de faire une authentification OTP en utilisant un serveur Radius et on passera en paramètres le login de l'utilisateur ainsi que son code OTP.

Le Serveur de Sécurité aura été configuré pour savoir répondre à cette requête et la transmettra donc au serveur Radius afin qu'il procède à l'authentification. Ce dernier enverra une requête au serveur Gemalto qui déterminera si le code OTP est valide et associé au bon login et répondra au Serveur de Sécurité. Si la réponse est positive, le Serveur de Sécurité va chercher si l'utilisateur existe dans sa base de données (LDAP dans notre cas) et si oui, il renverra un " jeton "qu'un composant poste utilisera pour placer l’utilisateur en tant qu’utilisateur courant sur la machine. Si non (le serveur Radius a fait une réponse négative, ou le Serveur de Sécurité n'a pas trouvé l'utilisateur dans sa base), un message d’erreur explicatif sera affiché à l'utilisateur.

L'authentification par Sign&Go peut fonctionner en mode déconnecté en utilisant un cache pour récupérer les informations récentes (identifiants de connexion, liste des schémas d'authentification...). Dans ce cas, la synchronisation se fait à intervalles réguliers afin de ne pas trop solliciter le Serveur de Sécurité.

4.4 Sécurité dans les communications

Les communications entre les différents composants et serveurs doivent se faire de manière cryptée afin de garantir un maximum de sécurité. On utilisera donc des communications TCP/IP cryptées avec un jeu de clefs asymétriques. Pour cela, on dispose d'une clef publique, connue de tous et une clef privée gardée secrète, formant ainsi un couple de clefs. Dans le cas de deux utilisateurs par exemple, le premier utilisateur génère un couple de clefs et va envoyer sa clef publique au second utilisateur et garder secrète sa clef privée. Le second utilisateur va maintenant pouvoir envoyer un message au premier. Pour cela il crypte son message avec la clef publique qu'il a reçu de celui-ci et le lui envoie. Le premier utilisateur est alors capable de décrypter le message avec sa clef privée. Personne d'autres que lui ne peut décrypter ce message.

Cette technique est particulièrement sure, si tant est que les clefs sont gardées secrètes. Elle est néanmoins plus couteuse en temps de calcul qu'une méthode par clefs symétriques (chacun dispose d'une même clef secrète pour crypter et décrypter les messages), aussi utilise-t-on parfois un système à cheval entre les deux. Les deux utilisateurs utilisent une clef privée symétrique, une "clef de session" qu'ils vont se communiquer via un jeu de clefs asymétriques.

5 TRAVAIL EFFECTUE - - PART 1

5.1 Objectif

Mon objectif était de développer un module d'authentification OTP. Après avoir étudié le fonctionnement du logiciel Sign&Go en général, j'ai focalisé mon attention sur les composants poste (raccourci pour désigner les composants présents sur le poste de l'utilisateur). Afin de réaliser l'authentification, je devais créer un système permettant de recevoir les informations de l'utilisateur, les faire remonter au serveur de sécurité, recevoir sa réponse et la faire revenir à l'utilisateur. Toute la partie se situant au niveau du Serveur de Sécurité n'était pas considérée dans cette approche, je devais travailler comme si elle avait déjà été réalisée et était fonctionnelle.

5.2 Environnement matériel

Pour réaliser mon travail, il me fallait intégrer du code en C/C++ au sein du projet du logiciel Sign&Go. Beaucoup de personnes travaillent en même temps sur le logiciel, et toutes ont besoin du code pour vérifier que leur travail s'intègre bien. Il est cependant important de ne pas se perdre entre les différentes versions sur lesquelles chacun travaille et c'est pour cela que l'on utilise un SVN (ou Subversion). Le code du logiciel est entreposé sur un serveur principal, et chacun a la possibilité de descendre tous les fichiers du projet, dont il a besoin, en local sur sa machine, de les modifier et de les faire remonter, en remplaçant les anciens (on parle de commit) ou en les ajoutant. Cela permet ainsi à tous de travailler sur des versions fonctionnelles et à jour, vérifier si d'autres utilisateurs ont fait des modifications et d'éviter les conflits.

Quand on remonte un fichier dans le projet, on est parfois prévenu de la création d'éventuels conflits, aussi faut-il comparer les anciennes et nouvelles versions pour étudier la ou les différences. Comme il n'est pas facile de repérer d'un coup d'œil des différences entre deux fichiers de code parfois très longs, on a utilisé un utilitaire très pratique "Examdiff" permettant de rendre compte des différences de manière très lisible et graphique.

Enfin j'ai pu travailler avec un IDE (VisualStudio), chose que je n'avais que très peu expérimentée auparavant. L'utilisation de VisualStudio apporte énormément d'avantages, comme un Makefile automatisé ou encore un debugger. L'utilisation du debugger permet de suivre les instructions dans le code au fur et à mesure que le programme s'exécute, à l'aide de break point ; on peut suivre l'état des variables, structures et ainsi mieux retrouver une éventuelle erreur. Cet outil est un énorme gain de temps et me fut extrêmement utile !

5.3 Réalisation

Mon travail consistait donc à rajouter un module d'authentification par OTP sur l'agent poste. Les premiers jours, je me suis surtout familiarisé avec le logiciel Sign&Go, ses composants, comment ils étaient reliés entre eux, comment se déroulait la communication… En effet, il y avait de fortes similitudes avec certains types d'authentification et il était intéressant d'essayer de les comprendre.

J'ai également étudié mon nouvel environnement de travail, j'étais sous Windows avec un IDE que je ne connaissais pas du tout.

Je devais donc créer un nouveau "chemin" du poste vers le Serveur de Sécurité, navigant au travers des différents composants les reliant.

En premier lieu, il a fallu travailler sur WS Agent. Celui-ci faisait apparaître un petit utilitaire dans la barre des tâches proposant un menu avec différents modes d'authentification. Il nous a fallu rajouter un onglet OTP dans la liste des modes d'authentification de ce menu (à partir d'un fichier de configuration). Cliquer sur ce bouton lançait le début du processus d'authentification. Il a alors été rajouté un cas pour l'authentification OTP (une case dans le switch gérant le choix de l'utilisateur). La liste des schémas d'authentification par OTP était alors récupérée auprès du Serveur de Sécurité, en l'occurrence il n'y en avait qu'un seul, le schéma Radius. Cette première partie a consisté à ajouter le code nécessaire dans des fichiers existant, mais pour la suite j'ai créé de nouveaux fichiers qui ont été inclus au projet.

Le fait de cliquer sur le bouton OTP devait ouvrir une fenêtre demandant à l'utilisateur de rentrer les informations nécessaires. Deux informations sont requises, un login et un mot de passe. Ce dernier est la concaténation d'un OTP et d'un mot de passe associé au compte de l'utilisateur. Le processus d'identification ne pouvait être lancé que si toutes les informations avaient été remplies. Aussi, un test était réalisé sur les deux champs et, en cas de champ vide, un message d'erreur était spécifié, le curseur de la souris placé dans le champ laissé vide et aucun processus d'authentification n'était lancé.

Réaliser une fenêtre sous Visual Studio et Windows n'a pas été simple, je n'avais réalisé d'interface graphique en C qu'en utilisant la bibliothèque GTK et là, le fonctionnement était un peu différent. Si les outils de création de fenêtre de Visual Studio étaient relativement simples, les API Windows le sont un peu moins et j'ai du rechercher beaucoup d'informations sur un site pour le développement d'applications Windows (http://msdn.microsoft.com/).

Si l'utilisateur avait bien rempli les deux champs, le processus pouvait commencer. La personne qui m'accompagnait me répétait souvent de voir chaque fonction, chaque module comme un système indépendant avec une entrée et une sortie. C'est avec cette façon de voir que j'ai alors poursuivi.

WS Agent devait alors transmettre la question à sngCliApi (en passant par cliApiDyn, que l'on peut considérer comme une passerelle mais sur laquelle nous ne nous attarderons pas). sngCliApi est une dll (l'équivalent d'une bibliothèque sous Linux), elle reçoit la question et la transforme pour l'envoyer à WSService et attend sa réponse.

WSservice la reçoit et demande à sngAgApi (nouvelle bibliothèque jouant le rôle de passerelle) de transmettre la question au serveur de sécurité de Sign&Go. Si l'authentification

réussit (on rentrera dans les détails de l'authentification dans la prochaine partie), sngAgApi reçoit une réponse positive et un jeton Sign&Go, que WSService utilisera pour placer l'utilisateur en tant qu'utilisateur courant sur la machine. Un code de retour sera retransmis jusqu'à WSAgent.

Dans tous les cas, ce code de retour indiquera quelle message spécifier à l'utilisateur, soit qu'il est bien authentifié, soit qu'il n'a pu l'être et pourquoi.

5.4 Phase de tests

J'ai alors pris soin d'écrire une "recette" un document de tests comportant des tests à effectuer et les résultats attendus pour vérifier la conformité du module. Il est notamment utilisé pour vérifier le bon fonctionnement d'un composant après des changements en amont/aval du projet.

J'ai rédigé ce document en procédant aux tests au fur et à mesure. J'ai ainsi pu trouver des choses à modifier, par exemple au niveau de la gestion des messages d'erreur. En effet, les messages actuels étaient "login incorrect" ou "password incorrect", des messages qui offrent trop d'informations. Certes ils peuvent être utiles à l'utilisateur pour diagnostiquer son erreur mais ils représentent un risque si quelqu'un essaie d'accéder au compte de manière indue, ce genre de message lui permet d'avancer beaucoup plus vite. On a donc utilisé un message d'erreur générique "Mauvais login ou mot de passe". Deuxième soucis, les messages doivent être possiblement affichés dans plusieurs langues (français ou anglais). Or, la fonction d'affichage des messages d'erreur qui était employée, utilisait une liste de messages n'existant qu'en français. J'ai alors traduit cette liste de messages en anglais et fait valider cette liste par le traducteur anglais officiel (qui m'a quand même corrigé quelques erreurs).

Ceci fait il n'y avait plus de problèmes et la recette a pu être validée !

6 TRAVAIL EFFECTUE - - PART 2

6.1 Objectif

Cette seconde mission (dans le prolongement de la première) m'a été suggérée par Mme Martel. Il s'agissait de créer une Machine Virtuelle (VM) de référence pour l'authentification par OTP. En effet, à ce moment du projet, une VM contenait le serveur Gemalto et une seconde VM le serveur Radius et le Serveur de Sécurité de Sign&Go. Or, ce serveur subit de fréquentes modifications et il est important de vérifier que celles-ci ne créent pas de conflit avec les différents moyens d'authentification ni n'invalident certaines fonctionnalités.

Aussi était-il alors important de bénéficier d'une VM contenant le serveur Radius et le serveur Gemalto, une VM de référence, que l'on pourrait "connecter" à n'importe quelle autre contenant le Serveur de Sécurité de Sign&Go. Ce travail, qui me permettait de prolonger mon étude de l'authentification OTP de Sign&Go, m'a immédiatement séduit, aussi m'y suis-je lancé dés la première mission terminée.

6.2 Environnement matériel

J'ai utilisé VMWARE, un logiciel de virtualisation, permettant de simuler l'exécution de systèmes d'exploitation comme s'ils étaient installés sur des machines différentes. Il est ainsi possible de disposer d'une interface graphique sur ces machines rendant l'émulation plus crédible.

Ainsi j'ai pu me familiariser avec plusieurs systèmes d'exploitation, tels que Windows Server 2003, Ubuntu, OpenSuse et leur mise en réseaux.

La société éditrice des boitiers (Gemalto) possède son propre serveur et fournissait ainsi un CD d'installation complet (documentations, plug-in etc) qui m'a été remis et m'a été très utile!

J'ai pu utiliser un petit utilitaire de transfert de fichier (WinSCP, très pratique pour transférer des fichiers directement de Windows à Linux par exemple) ainsi que l'éditeur VIM, complexe, mais beaucoup plus complet que les éditeurs pour terminal que j'utilisais auparavant.

6.3 Réalisation

Dans la construction de cette VM, j'ai décidé de procéder par étapes, changeant les différents "composants" un à un, afin de mieux identifier la marche à suivre et d'identifier les éventuelles erreurs avec plus de facilité.

En premier lieu, j'ai demandé si il était possible que l'on me remette une VM avec un Serveur de Sécurité Sign&Go seul, afin de désactiver celui de la première VM et d'utiliser celui-ci à la place. J'ai alors pu me familiariser avec le fonctionnement du Serveur de Sécurité. Pour sa configuration, on utilise un panel administrateur accessible depuis une certaine url dans un navigateur web. On peut alors détailler le schéma d'authentification pour l'OTP (ici Radius) et l'adresse de ce serveur Radius. Ceci fait, j'ai rajouté un nouvel un utilisateur dans l'annuaire (LDAP) installé sur la VM du Serveur de Sécurité qui allait être mon utilisateur test.

Dans les fichiers de configuration de l'agent poste j'ai alors désigné ce nouveau Serveur de Sécurité comme le Serveur de Sécurité à utiliser dorénavant (en spécifiant la nouvelle adresse du serveur). J'ai fait un premier test d'authentification… qui n'a pas fonctionné. Je me suis donc penché du côté des fichiers de logs à chaque étape pour voir où s'interrompait la chaine de messages. Visiblement, le message partait de l'agent poste, allait au Serveur de Sécurité puis au serveur Radius mais celui-ci le rejetait sans l'envoyer au serveur Gemalto car il ne reconnaissait pas le Serveur de Sécurité comme client autorisé.

J'ai alors cherché dans les fichiers de configuration du serveur Radius et ai trouvé le fichier permettant de lister les clients autorisés à se connecter. Il y avait la machine locale (l'ancien serveur de sécurité) il fallait donc que je rajoute la nouvelle. Ceci fait, j'ai modifié le fichier de DNS (résolution de noms) en conséquence (le nom de la machine devait être utilisé également). Après redémarrage… le serveur de sécurité était toujours rejeté ! Cet épisode fut une perte de temps assez conséquente pour moi, ne comprenant pas pourquoi il était rejeté alors que tout semblait indiquer que c'était le bon fichier que j'avais modifié et de la bonne manière.

Après avoir perdu beaucoup de temps dans d'infructueux tests, je me suis rendu compte que les fichiers de configuration que je modifiais, situés dans le répertoire d'installation du serveur Radius, n'étaient en fait pas ceux qui étaient utilisés! Les mêmes fichiers (même noms, même contenus) se trouvaient ailleurs et c'était ceux-ci qui étaient lus. J'ai donc procédé à de nouvelles modifications et l'authentification a fonctionné.

Il était alors temps de s'intéresser au serveur Gemalto. M Szaniec m'avait prévenu que je devrais l'installer sous Linux (car le serveur Radius que nous utilisions, Freeradius, un serveur Radius libre, ne fonctionnait que sous cet OS). Je devais donc choisir une distribution. Mon premier choix s'est porté sur Ubuntu (la 10.4 était sortie depuis quelques temps et je n'avais pas eu l'occasion de la tester).

J'ai donc créé une VM (étape rendue relativement simple par le logiciel VMWARE) en utilisant une image CD d'Ubuntu préalablement téléchargée . J'ai alors lancé l'installation du serveur Gemalto depuis le CD fourni. Tout allait bien, lorsqu'un problème est survenu au milieu de l'installation. Un logiciel, firebird (serveur de base de données), n'arrivait pas à être installé. Et le message d'erreur souffrait de problèmes d'affichage empêchant de le fermer ou d'annuler l'installation… J'ai recherché des informations dans les documentations de Gemalto et suis remonté à un passage qui m'avait échappé traitant… des distributions Linux supportant le serveur (dont Ubuntu ne faisait évidemment pas partie).

J'ai donc consulté la liste des distributions supportées et mon choix s'est porté sur openSuse, les versions Red Hat conseillées étant payantes (j'ai appris seulement après qu'il existait des versions de Red Hat recompilées à l'identique à l'exception des logos propriétaires Red Hat gratuites que j'aurais sans doute pu utiliser). Je recrée une nouvelle VM avec OpenSuse, procède à l'installation du Serveur Gemalto, qui se déroule sans problème!

J'ai alors pu commencer la configuration du serveur Gemalto. Cette étape fut relativement difficile, la documentation n'étant pas toujours très détaillée. Dans un premier temps, il a fallu relier les boitiers OTP au serveur. Il existe pour cela deux méthodes, soit en chargeant un fichier contenant les informations sur ces boitiers, soit en rentrant des informations directement depuis le panel administrateur. Pour plus de simplicité, je voulus passer par le panel mais me suis néanmoins heurté à un problème, certaines informations demandées (un "secret partagé" crypté correspondant au boitier de 48 caractères hexadécimaux...) étaient introuvables. Après moult pérégrinations je finis par trouver un fichier sur le disque d'installation contenant ces informations, que j'avais déjà parcouru mais ne sachant pas à quoi il correspondait (comme beaucoup de fichiers malheureusement...) je l'avais oublié. J'ai donc changé de méthode et chargé le fichier ; tous les boitiers furent ajoutés sur le serveur (sauf un, il y avait une petite coquille dans le fichier que j'ai normalement pu résoudre !).

Toujours sur le panel, j'ai alors créé un nouvel utilisateur (login + mot de passe) et lui ai assigné un des boitiers précédemment entrés. J'ai alors pu activer et synchroniser le boitier en rentrant deux codes OTP successifs, générés à partir dudit boitier dans le panel. Cet utilisateur correspondait à mon utilisateur test créé dans l'annuaire relié au Serveur de Sécrurité (même nom).

Un panel utilisateur permettait de tester la configuration, il offrait la possibilité de réaliser une authentification à partir du nom d'utilisateur, du mot de passe et d'un code OTP généré à partir du boitier lié à l'utilisateur. J'ai alors testé la configuration en l'état et l'authentification par le panel utilisateur fut un succès !

Il restait alors à installer et configurer le serveur Radius. J'ai récupéré l'archive d'installation depuis l'ancienne VM et l'ai extraite sur la nouvelle. L'installation était automatisée à l'aide d'un Makefile et s'est déroulée sans aucun problème.

Pour la configuration j'ai suivi pas à pas la documentation fournie par Gemalto et les commentaires des fichiers de configuration freeradius, tous deux très détaillés pour cette fois. Cette configuration permettait d'expliquer au serveur Radius comment contacter le serveur Gemalto et quels paramètres lui envoyer. Après avoir rajouté un plug-in sous forme de fichier rpm, j'ai relancé le serveur Radius et fait un test d'authentification, depuis les composants poste, qui fut concluant !

6.4 Phase de tests

Il avait été prévu dès le commencement que cette tâche devrait déboucher sur la création d'une documentation. Cette documentation devait expliquer quelles manipulations avaient été effectuées pour la création de la VM et l'installation et la configuration des serveurs Gemalto et Radius. Elle devait permettre de reprendre le travail en cas d'éventuelle perte ou d'aider à l'installation/configuration sur d'autres plates-formes, en étant la plus claire possible.

Je me suis donc attaché à noter depuis le début tout ce que j'avais fait. Le document final rédigé, j'ai repris le document depuis le début et ai tout recommencé en partant de VM vierges. J'ai pu constater qu'à quelques ajustements prêts, la documentation était complète, claire et solide.

J'ai mis un point d'honneur à ce que n'importe qui lisant ce document soit capable de reproduire les manipulations, même sans connaître le fonctionnement détaillé. Ce document vous a normalement été remis en annexe, à titre consultatif.

7 Conclusion

Je tire de ce stage un bilan extrêmement positif. Je pense que pour mon premier stage, j'ai eu beaucoup de chance. La chance de travailler chez un éditeur logiciel, avec une mission passionnante, développant des problématiques actuelles et des enjeux de sécurité importants. La chance de bénéficier d'un cadre et d'une ambiance de travail chaleureuse et dynamique, véritables sources de motivation.

J'espère que mon travail a pu satisfaire mes responsables, et qu'il leur est, et sera, utile. Pour ma part, je peux retirer beaucoup d'enseignements de ce stage.

Du point de vue technique déjà, j'ai eu la chance de travailler dans plusieurs domaines, comme le développement ou les réseaux. J'ai utilisé de nouveaux systèmes d’exploitation, appris de nouveaux outils, découvert de nouveaux logiciels...

J'ai également énormément appris sur le plan de la méthode, sur la manière rigoureuse et efficace de penser, de concevoir le développement, et ceci, je le dois à la grande pédagogie et aux nombreux conseils qui m'ont été prodigués.

Enfin, j'ai découvert beaucoup de choses sur la vie en entreprise, sur son organisation, mais aussi sur la vie de l'informaticien, que je cerne un peu mieux maintenant. Je sais qu'il est possible de faire des formations, de l'intégration, différentes facettes du métier que je n'appréhendais pas.

Si je ne sais pas encore exactement ce que je souhaite faire plus tard, je possède du moins maintenant les clefs pour me poser les bonnes questions pour mon futur professionnel.

8 ANNEXE : SCHEMA D'ARCHITECTURE GENERALE SIGN&GO