Guide pratique des tunnels cryptés utilisant SSH et MindTerm

Duane Dunston

<duane@duane.yi.org>

Revision History

Revision 1.01	2001-06-13	Revised by: PDD
Format de date modifié (YYYY-MM-DD)		

Ce document décrit comment utiliser SSH et le programme Java MindTerm pour créer des tunnels de type VPN rapides et sécurisés sur des réseaux non sécurisés.

Table of Contents

1. Introduction

Pour diverses raisons, cette toute nouvelle version a pour nom de code la version *release*.

De nouveaux noms de code feront leur apparition selon les directives des standards de l'industrie, pour mettre en valeur l'aspect "état de l'art" du document.

J'ai écrit ce document suite à une suggestion qui m'avait été faite de fournir un modèle à remplir pour faire de nouveaux guides pratiques. Ce modèle a été fait à l'origine en extrayant la structure du guide pratique "Multi Disk HOWTO" qui est un guide pratique plutôt volumineux. Ce modèle a ensuite été largement révisé.

Ce petit rappel du contexte me permet de démarrer mon introduction.

Tout d'abord, un peu de jargon juridique. L'évolution récente montre que c'est assez important.

1.1. Informations sur le copyright

1.2. Avis de non responsabilité

Nous ne pouvons être tenus responsables du contenu de ce document. Vous utilisez les concepts, exemples et autres contenus à vos risques et périls. Comme il s'agit d'une nouvelle révision de ce document, il se peut qu'il y ait des erreurs et des inexactitudes qui peuvent bien sûr endommager votre système. Agissez avec prudence, et bien que cela soit tout à fait improbable que vous le fassiez, l'auteur décline toute responsabilité quant à cela.

Sauf mention contraire, tous les copyrights sont détenus par leurs auteurs respectifs. L'utilisation d'un terme dans ce document ne doit pas être considérée comme portant sur la validité d'une quelconque marque commerciale ou de prestataire de service.

La mention de produits ou de marques particuliers ne doit pas être considérée comme une publicité pour ce produit ou cette marque.

Il vous est fortement conseillé d'effectuer une sauvegarde de votre système avant toute installation d'envergure ainsi que des sauvegardes à intervalles réguliers.

1.3. Nouvelles versions

Ce document a subi de nombreuses révisions puique je l'ai entamé comme projet final pour la certification SANS GIAC.

Le numéro de version le plus récent de ce document peut être trouvé sur la page principale du Linux Documentation Project ou sur la page de l'auteur.

Si vous êtes compétent en la matière, ce serait bien de rendre le guide pratique disponible dans un certain *nombre de formats.*

1.4. Remerciements

Dans cette version, j'ai le plaisir de remercier :

Patti Pitz pour sa révision et son aide dans l'organisation de l'article. Doug Eymand pour le côté technique de la révision.

1.5. Vos impressions

Vos réactions sur ce document sont attendues avec le plus grand intérêt. Sans vos propositions et vos contributions, ce document n'aurait jamais existé. Je vous prie d'envoyer les points que vous voudriez ajouter, vos commentaires et vos critiques à l'adresse suivante : `<duane@duane.yi.org>`.

2. Avant de commencer

2.1. Introduction à MindTerm et SSH

De nos jours, les entreprises, les écoles, et les particuliers ont plus que jamais besoin de services réseau sécurisés. Le commerce en ligne s' accroissant, il y a plus de gens qui continuent à avoir accès à des données sensibles d'entreprise au travers de réseaux non sécurisés. Les entreprises utilisent Internet comme moyen principal pour communiquer avec leurs employés en déplacement dans le pays et à l'étranger, envoient des documents et des courriels non cryptés à leurs agences et partenaires de par le monde ; ces communications peuvent être riches en informations que n'importe quelle personne mal intentionnée peut potentiellement intercepter et vendre ou donner à une entreprise rivale. De bonnes politiques de sécurité à la fois pour les utilisateurs et les administrateurs réseau peuvent aider à minimiser les problèmes liés à l'interception ou au vol d'informations à l'intérieur de leur organisation par des personnes mal intentionnées. Dans cet article, nous allons aborder l'utilisation de Secure Shell (SSH) et MindTerm pour sécuriser la communication par Internet au sein d'une organisation.

Les utilisateurs à domicile et les voyageurs d'affaires accèdent à des données d'entreprise et envoient des données sensibles au travers de réseaux non sécurisés. *Cela crée toute une catégorie de nouveaux problèmes de sécurité pour les administrateurs système ("Securing the home office sensible and securely")*, particulièrement depuis qu'on estime que le nombre de personnes travaillant à domicile avec un accès haut débit *"va plus que doubler, passant de 24 millions en 2000 à 55 millions en 2005" ("Broadband Access to Increase in Workplace"). Le nombre d'aéroports et d'hôtels offrant un accès Internet, en particulier un accès haut débit, est en croissance et on s'attend à ce qu' il grandisse dans l'avenir ("Broadband Moving On Up").* Ceci peut aussi laisser une porte grande ouverte à une personne mal- intentionnée qui pourrait pirater ou voir le trafic Internet d'autres personnes et accéder à leurs entreprises. Cette personne malintentionnée peut ne pas être intéressée par les travaux de l'employé, mais veut seulement accéder à un serveur très rapide pour lancer des attaques, stocker des fichiers ou pour un autre usage. Les hommes d'affaires courent de grands risques car ils ne savent pas qui surveille leur connexion Internet à l'hôtel, à l'aéroport ou n'importe où pendant leur déplacement. Les utilisateurs des nouvelles connexions haut débit ne sont en général pas formés à des protocoles de sécurité adaptés, et certaines entreprises ne disposent pas du personnel pour aider les utilisateurs à domicile et les voyageurs d'affaires à mettre en place une connexion sécurisée. Les particuliers et, étonnamment, certaines entreprises ont cette mentalité *"Je n'ai rien qui pourrait intéresser les autres"*. Ceci est très inquiétant quand on considère la quantité d'informations sensibles qui voyage à travers Internet depuis le domicile d'un employé ou lors de déplacements. Ce qui est plus inquiétant est la disponibilité de logiciels gratuits pour effectuer ce genre d'attaques et la facilité d' utilisation de ces logiciels. Dsniff (http://www.monkey.org/~dugsong/dsniff/) est un programme disponible gratuitement qui possède des fonctionnalités permettant à n'importe qui avec un ordinateur connecté à un réseau de pirater un réseau local et surveiller ce que les autres font et récupérer des mots de passe et d'autres données sensibles. Dans son livre

"Secrets and Lies : Digital Security in a Networked World", Bruce Schneier affirme que la propagation des techniques et une des principales menaces à la sécurité des réseaux : *"Internet est [...] un média parfait pour propager des outils d'attaque qui marchent. Il suffit que le premier attaquant soit un spécialiste ; tous les autres peuvent utiliser son logiciel" (Schneier).*

L'objectif de cet article n'est pas de savoir comment sécuriser des ordinateurs, mais comment mettre en place des tunnels virtuels pour effectuer des communications sécurisées, que ce soit pour envoyer des documents ou des courriels. Les voyageurs d'affaires devraient lire les articles de Jim Purcell, Frank Reid, et Aaron Weissenfluh sur la sécurité au cours des déplacements. Les utilisateurs à domicile ayant un accès haut débit devraient lire l'article de Ted Tang pour avoir des informations sur la façon de sécuriser un ordinateur avec accès haut débit. Je recommanderais la nombreuse documentation disponible sur www.sans.org, www.securityfocus.com, ou www.securityportal.com avec des tutoriels sur la façon de sécuriser vos ordinateurs et serveurs.

Le moyen pour s'assurer que les données sensibles sont transmises de manière rapide et sécurisée est d'utiliser des méthodes cryptées de transmission de données. Cela peut se faire par le moyen d'un courriel crypté, en utilisant des services web sécurisés de messagerie électronique, ou en établissant des tunnels cryptés entre deux ordinateurs. De plus, un logiciel fiable et facile à installer doit être utilisé pour permettre aux utilisateurs inexpérimentés d'établir rapidement des canaux de communication sécurisés. Secure Shell et MindBright Technology's MindTerm de Taten Ylonen sont une solution rapide, facile d'utilisation et fiable pour sécuriser les communications sur Internet.

2.2. MindTerm et SSH

SSH (Secure Shell) peut remplacer en toute sécurité les programmes de connexion à distance et de transfert de fichiers comme telnet, rsh et ftp qui transmettent les données en texte clair, lisible par toute personne. SSH utilise une méthode d'authentification à clé publique pour établir une connexion cryptée et sécurisée entre la machine de l'utilisateur et la machine distante. Une fois la connexion sécurisée établie, les nom d'utilisateur, mot de passe et toutes les autres informations sont envoyés au travers de cette connexion sécurisée. Vous trouverez plus d'informations sur la façon dont SSH fonctionne, les algorithmes utilisés et les protocoles implémentés pour qu'il reste à un haut niveau de sécurité et de confiance sur le site web de ssh : www.ssh.com. L'équipe OpenBSD a créé un équivalent libre qui s'appelle OpenSSH, disponible sur www.openssh.com. Il conserve les normes élevées de sécurité de l'équipe OpenBSD et des spécifications de l' IETF pour Secure Shell (voir les documents de travail Secure Shell de l'IETF), sauf qu'il utilise des algorithmes libres du domaine public. SSH est en train de devenir un standard pour l'administration de connexions à distance. Il a rencontré un tel succès qu'il y a de nombreux ports SSH pour diverses plateformes, et des clients gratuits disponibles pour se connecter à un serveur SSH sous diverses plateformes également. Voir http://linuxmafia.com/pub/linux/security/ssh-clients pour avoir une liste des clients. Securityportal.com a un excellent article en deux parties sur SSH et sur les liens vers des ports pour différentes plateformes ; il est disponible sur http://www.securityportal.com/research/ssh-part1.html. Il y a des programmes qui utilisent également un utilitaire qui s'appelle Secure Copy (SCP) en fond qui fournit les mêmes fonctionnalités qu'un client FTP complet, tels que WinSCP et le client Java SSH/SCP, qui a une interface SCP modifiée pour MindTerm. Veuillez lire les licences avec précaution pour voir si vous avez l'autorisation légale de télécharger SSH dans votre pays. SSH est libre pour les écoles et les universités. Veuillez lire les licences disponibles sur le site web ssh.com.

MindTerm est un client SSH écrit entièrement en Java par MindBright Technology. Une des pratiques clés dans le développement de logiciels de sécurité est une implémentation adéquate des algorithmes sous-jacents et des protocoles utilisés. MindBright Technology a très bien implémenté le protocole SSH dans ce petit fichier d'application. C'est une archive autonome qui nécessite juste d'être décompressée dans un répertoire de votre choix, et qui est prête à être utilisée. Le client peut être utilisé comme programme autonome ou comme applet de page web, ou les deux. Il est disponible sur : http://www.mindbright.se/download/. MindTerm est un client excellent et peu coûteux pour sécuriser les communications vers et depuis un emplacement local et distant. Le programme MindTerm situé à l'adresse de téléchargement précédente est disponible gratuitement pour une utilisation non commerciale et dans l'enseignement, la possibilité d'une utilisation commerciale étant étudiée au cas par cas. Cependant, les modifications apportées par ISNetwork *"sont basées sur le code source de MindTerm 1.21, que MindBright a publié sous GPL [General Public Licence - voir http://www.gnu.org]. Comme notre version a été publiée sous GPL, vous pouvez l'utiliser gratuitement à des fins commerciales" (Eckels)*. L'implémentation d'ISNetworks a toutes les fonctionnalités du MindTerm de MindBright, excepté qu'elle a une interface SCP plus sympathique pour des transferts de fichier plus conviviaux. MindTerm a quand-même l'inconvénient de ne pas prendre en charge les tunnels UDP. Pour sécuriser le trafic UDP, un programme qui s'appelle Zebedee (http://www.winton.org.uk/zebedee/) fera très bien l'affaire. Les programmes serveur et client de Zebedee sont disponibles pour les plateformes Windows et Linux. Il est distribué gratuitement sous licence GPL également. Vous pouvez vous connecter aussi bien aux machines Windows que Linux avec Zebedee. MindTerm ne vérifiera pas si votre système est sécurisé. C'est aux administrateurs et aux utilisateurs de prendre le soin de sécuriser leurs systèmes informatiques. Il est facile à implémenter et il est très efficace pour ce qui est de garder le haut niveau de sécurité implémenté dans le protocole SSH. Nous verrons dans cet article à quel point cela est facile de mettre en place et d'établir des canaux de communication sécurisés pour et par quasiment n'importe quel utilisateur. Les documents, courriels et autres communications de données peuvent être envoyés facilement et en toute sécurité à des utilisateurs à l' autre bout du monde ou à quelques pas de là.

2.3. Comment MindTerm et SSH fonctionnent ensemble

SSH et MindTerm fonctionneront ensemble pour utiliser une technique appelée redirection de port [port forwarding]. La redirection de port consiste à rediriger du trafic d'un hôte et un port donnés, vers un autre hôte et port. En d'autres termes, l'application MindTerm va ouvrir un port sur la machine du client (machine locale) et toute connexion à ce port local est redirigée vers l'hôte distant et son port d'écoute au travers d'une session SSH cryptée. Le fait que la connexion soit acceptée ou pas dépend du type de requête qu'on envoie à l'hôte distant. Par exemple, on ne redirigerait pas des requêtes POP vers un hôte distant écoutant sur le port 21, car le port 21 est réservé aux requêtes FTP. La redirection de port est également utilisée pour permettre la connexion à un serveur situé derrière un pare-feu et/ou qui a une adresse IP privée. Essentiellement, cela s'appelle créer un réseau virtuel privé [Virtual Private Network] (VPN). Un VPN est *"un réseau de données privées faisant usage de l' infrastructure de télécommunications publique, en protégeant la vie privée par l'emploi d'un protocole de tunnel et de procédures de sécurité"* (www.whatis.com). La redirection de port ne peut être effectuée qu'avec des services TCP.

---

3. Installation du logiciel

Pour pouvoir suivre ce tutoriel, vous devrez installer quelques paquetages [packages]. Ce tutoriel part du principe que SSH est déjà installé sur votre serveur ou station de travail. Si ce n'est pas le cas vous pouvez lire la documentation livrée avec le paquetage SSH ou OpenSSH pour avoir des instructions d'installation pour votre plateforme. Pour les exemples suivants, OpenSSH a été installé sur un serveur RedHat 7.0 et sur une station de travail. OpenSSH a été installé sur RedHat 6.0 à 7.0 et fonctionne de la même façon. La machine client utilisée dans ce tutoriel est une machine Windows 2000. Des stations de travail Windows 95/98, NT 4.0, NT 5.0, RedHat 6.0-7.0 ont toutes été testées comme machines client et ont fonctionné de la même façon. Entre parenthèses, exactement la même archive JAR MindTerm a été utilisée sur tous les systèmes client testés.

- SSH ou OpenSSH
- MindTerm
- Client FTP - N'importe quel client FTP devrait marcher pour ce tutoriel. Ws-FTP et Leech-FTP sont les deux plus populaires pour Windows.
- Netscape Communicator - ou n'importe quel autre client de messagerie devrait marcher.
- Optionnel: NTOP
- Optionnel: vlock

---

4. Configurations du serveur et du client

4.1. Configuration du serveur

D'abord, assurez-vous que votre serveur est sécurisé. Bien que le trafic soit crypté pendant qu'il circule sur Internet, il peut être reniflé si quelqu'un a un accès root sur la machine locale et utilise un programme tel que ngrep pour renifler le trafic sur une machine locale. Par exemple, utilisée conjointement avec le programme DSniff mentionné plus haut, la commande suivante pourrait renifler tout le trafic sur le réseau d'interface locale : **ngrep -d lo**. Cependant, la sécurisation du serveur n'entre pas dans le cadre de cet article.

Nous utiliserons les services POP (port 110), IMAP (port 143), SMTP (port 25), VNC (Virtual Network Computing) (5901+), et NTOP (port 3000 par défaut) pour cet exemple. Tout le trafic sera redirigé vers le port respectif de chaque service sur l'hôte distant où tourne le serveur SSH. Tous les services écoutant sur l'hôte distant écoutent sur toutes les interfaces, à moins que le service soit lié à un port par défaut ou qu'il soit configuré manuellement. Pour montrer à quel point cette technique de tunnel SSH est efficace, nous n'autoriserons que des services précis à écouter sur l'interface locale.

Cependant,vous n'avez pas à changer vos configurations de sécurité courantes. Nous utiliserons tcp_wrappers, qui est installé par défaut sur RedHat 7.0 (et les versions précédentes), pour nous connecter aux services réseau. Dans le fichier `/etc/hosts.deny`, ajoutez la ligne suivante :

```
ALL : ALL
```

Et dans votre fichier `/etc/hosts.allow` ajoutez les lignes suivantes :

```
sshd : ALL
  in.ftpd : 127.0.0.1
  ipop3d : 127.0.0.1
  imapd : 127.0.0.1
```

Avec ce paramétrage, sshd (le serveur SSH) autorisera les connexions depuis n'importe quelle adresse IP. Les autres services n'autorisent les connexions que depuis l'interface locale. On peut vérifier cela tout de suite en configurant un client de messagerie pour qu'il se connecte au serveur POP ou IMAP distant, et/ou un client FTP pour qu'il se connecte au serveur FTP. La connexion ne sera pas autorisée. Il faudra également paramétrer tout compte utilisateur devant avoir accès à ces services.(Note : le paramétrage ci-dessus n'est utile que si les services sont pour un usage interne uniquement, et que les utilisateurs distants ont besoin d'accéder à des services internes pour envoyer et recevoir des courriels ou transférer des fichiers. Les services peuvent être disponibles pour un usage publique et être cryptés avec SSH et MindTerm.) En vue d'une utilisation de MindTerm sur le web pour créer des tunnels ou utiliser les fonctionnalités GUI de Secure Copy, un environnement d'exécution Java (JRE) devra également être installé sur le serveur où tourne SSH.

4.2. Configuration du client

La seule configuration nécessaire pour le client est de s'assurer qu'un JRE est installé sur votre plateforme. Windows et MacOS 8 et plus ont déjà un JRE installé. Il est recommandé d'installer le JRE de Sun sur Windows. IBM a une liste des ports des JRE des différentes plateformes : http://www-105.ibm.com/developerworks/tools.nsf/dw/java-devkits-byname , ainsi que Sun : http://java.sun.com/cgi-bin/java-ports.cgi. (Vous n'avez pas besoin de tout le paquetage Java avec les débogueurs et les compilateurs, juste la machine virtuelle Java (JVM) pour lancer des applications Java.) De plus, pour le tutoriel qui suit, décompressez l'archive MindTerm, implémentation MindBright ou ISNetwork, dans `c:\mindterm` pour Windows.

5. Création des tunnels

MindTerm peut être démarré de plusieurs manières. Si vous avez JRE installé, alors vous pouvez double-cliquer sur le fichier d'application mindtermfull.jar. Une autre manière consiste à ouvrir une invite de commande DOS et à taper la commande :

```
jview -cp c:\mindterm\mindtermfull.jar mindbright.application.MindTerm
```

ou

```
javaw -cp  c:\mindterm\mindtermfull.jar mindbright.application.MindTerm
```

ou

```
java -cp c:\mindterm\mindtermfull.jar mindbright.application.MindTerm
```

(jview s'utilise si vous êtes sous Windows et que vous n'avez pas téléchargé le JRE. Javaw est livré avec le téléchargement de JRE sous Windows et est utilisé car une invite de commande DOS n'est pas nécessaire pour lancer MindTerm, ce qui fait une fenêtre ouverte en moins)

MindTerm 2.0 est maintenant disponible. L'argument pour le lancer a légèrement changé. A la place de la commande ci-dessus :

```
java -cp c:\mindterm\mindtermfull.jar mindbright.application.MindTerm
```

cela démarrera MindTerm en ligne de commande :

```
java -cp c:\mindterm\mindtermfull.jar com.mindbright.application.MindTerm
```

Seul le "com." a été ajouté au paramètre de l'applet.

Cela démarrera le programme MindTerm et ensuite vous pourrez taper le nom du serveur quand vous y serez invité et on vous demandera ensuite "Save as Alias" (enregistrer en tant qu' alias). Vous pouvez entrer un nom de serveur court, comme ça quand vous lancerez l'applet à nouveau vous n'aurez qu'à taper l'**Alias** que vous aurez créé. On vous demandera ensuite votre identifiant de connexion. Après l'avoir tapé, tapez Entrée et une boîte de dialogue apparaîtra pour vous informer que l'hôte n'existe pas et vous demandera d'en créer un. Cliquez sur **Yes**. Une autre boîte apparaîtra pour vous demander si vous voulez ajouter cet hôte à votre fichier `known_host`. Cliquez sur **Yes**. On vous demande ensuite votre mot de passe. Tapez votre mot de passe puis Entrée. Si vous avez entré l'identifiant et le mot de passe adéquats, vous devriez vous trouver en ligne de commande sur le serveur que vous avez spécifié.

Nous allons d'abord créer un tunnel vers le serveur POP et SMTP. Quand vous serez connecté avec succès (et aurez éventuellement activé vlock), cliquez sur Tunnels dans le menu et ensuite sur Basic. Une boîte de dialogue apparaîtra. Ajoutez respectivement les réglages suivants dans chaque boîte :

- Local port: **2010**
- Remote Hosts: *Votre hôte distant (ça devrait être le serveur sur lequel tourne sshd).*
- Remote port: **110**

Maintenant cliquez sur **Add** (ajouter). Il devrait apparaître une boîte de dialogue disant "The tunnel is now open and operational (le tunnel est maintenant ouvert et opérationnel) ". *(Remarque : si vous sélectionnez un port déjà ouvert, un message d' erreur vous dira Could not open tunnel. Error creating tunnel. Error setting up local forward on port XXXX, Address in use - Le tunnel n'as pas pu être ouvert. Erreur lors de la création du tunnel. Erreur lors de la mise en place de la redirection local sur le port XXXX, Adresse en cours d'utilisation.)* Cliquez sur **OK** et la configuration du tunnel devrait maintenant apparaître dans une boîte. Cliquez sur **Close Dialog** (fermer la boîte de dialogue). Ouvrez le menu des options ou préférences de votre client de messagerie. Nous utiliserons Netscape Messenger pour cet exemple.

1. Ouvrez Netscape

2. Cliquez sur **Edit -> Preferences**.
3. Sur la colonne de gauche cliquez sur **Mail " Newsgroups** , si le contenu n'est pas déjà affiché.
4. Cliquez sur **Identity** et entrez vos informations dans chaque boîte.
5. Cliquez sur **Mail Servers** dans la colonne de gauche. L'installation par défaut de Netscape affiche "mail" dans la boîte en-dessous de "Incoming mail servers" (serveurs de réception de mails)
6. Cliquez sur **mail**.
7. Cliquez sur **Edit** à droite de cette boîte et une boîte de dialogue devrait apparaître.
8. Si POP n'est pas déjà sélectionné dans cette boîte déroulante, sélectionnez-le maintenant.
9. Dans la boîte "Server Name" tapez **localhost:2010** *(rappelez-vous qu'on a choisi ce port local dans le menu de création de tunnel de MindTerm pour rediriger vers le port POP (110) des serveurs distants)* et ensuite votre nom d'utilisateur. Réglez toute option qui vous semble convenir.
10. Cliquez sur **OK**.
11. Dans la boîte **Outgoing mail (SMTP) server** (serveur d' envoi de messages) tapez le nom de votre serveur SMTP et en-dessous tapez votre nom d'utilisateur pour le serveur de messages sortants.
12. Cliquez sur **OK**. *(Ne touchez pas à l'option "Use Secure Socket Layer (SSL) or TLS for outgoing messages" - utiliser SSL ou TLS pour les messages sortants).*
13. Maintenant cliquez sur **Communicator** dans le menu.
14. Cliquez sur **Messenger**.
15. Vous devriez ensuite être invité à entrer votre mot de passe. Tapez votre mot de passe puis Entrée. Si vous avez du courrier, vous devriez maintenant être en mesure de le lire.

Tant que vous avez une session SSH MindTerm ouverte, cela devrait marcher avec quasiment tous les clients de messagerie. Rappelez-vous que le nom du serveur distant ou du serveur POP sera "localhost:". Si l'on vous demande le serveur POP et le port séparément, alors ajoutez-les en conséquence. Dans cet exemple, toute connexion au port local 2010 sera redirigée vers le port 110 de l'hôte distant. Si vous configurez un client FTP pour se connecter au port 2010 de l' hôte local, ça ne marcherait pas. Pourquoi? Le protocole POP ne comprend pas le protocole FTP. Pour que le tunnel aboutisse, seuls les clients POP peuvent être redirigés vers le port 2010 de l'hôte local. Un serveur POP ne sert à rien si vous n'avez pas de serveur SMTP. Si vous avez un programme de messagerie comme Postfix (www.postfix.net), Qmail (www.qmail.org), ou Sendmail (www.sendmail.org) un tunnel sécurisé peut être également créé.

Le client MindTerm tournant encore, cliquez sur "Tunnels" à nouveau, puis sur "Basic" et ajouter ces réglages.

- Local Port: **2025***(vous pouvez écraser les réglages d'avant)*
- Remote Host: *Votre serveur SMTP distant.*
- Remote Port: **25**

Cliquez sur **Add**. Cliquez ensuite sur **OK** dans le menu de confirmation. Maintenant SMTP devrait être ajouté à la liste en-dessous des réglages pour POP. Dans les réglages du serveur de messagerie dans Netscape Messenger, ajoutez : **localhost:2025** comme étant votre *Outgoing mail (SMTP) server* (serveur de messages sortants). Tous les courriels que vous enverrez à l'hôte distant seront cryptés. Cependant, si vous envoyer un message à quelqu'un en-dehors du serveur de messagerie de l' hôte distant, votre courriel sera crypté seulement de votre machine locale à votre serveur SMTP distant. Du serveur SMTP distant à n'importe quel autre hôte, il ne sera pas crypté, à moins que vous ayez configuré un tunnel vers les autres hôtes.

Pour permettre des sessions FTP cryptées, ajoutez ces informations à un nouveau tunnel.

- Local Port: **2021** *(vous pouvez écraser les réglages d'avant)*
- Remote Host: *Votre serveur FTP distant.*
- Remote Port: **21**

Cliquez sur **Add**. Ensuite cliquez sur **OK** dans le menu de confirmation. Maintenant FTP (voir l'exemple leech ftp et wsftp - image 1 et image 2) devrait être ajouté à la liste en-dessous des réglages SMTP.

Réglages Imap :

- Local Port: **2043** *(vous pouvez écraser les réglages d'avant)*
- Remote Host: *Votre serveur IMAP distant*.
- Remote Port: **143**

Cliquez sur **Add**. Ensuite cliquez sur **OK** dans le menu de configuration. Maintenant IMAP devrait être ajouté à la liste en-dessous des réglages SMTP.

Tous ces réglages peuvent être automatisés dans un fichier batch. Ajoutez simplement ce qui suit dans un script de démarrage pour créer automatiquement un tunnel vers votre serveur POP après authentification :

```
jview (ou java ou javaw) -cp c:\mindterm\mindtermfull.jar mindbright.application.MindTerm
  -server -local0 2010:localhost:110
```

Voici un exemple basé sur ce que nous avons fait plus haut. Ajoutez ce qui suit à un fichier dans un éditeur :

```
jview (ou java ou javaw) -cp c:\mindterm\mindtermfull.jar mindbright.application.MindTerm
   -server -local0 2010:localhost:110 -local1 2025:localhost:25 -local2 /ftp/2021:localhost:21
  -local3 2043:localhost:143
```

Maintenant sauvegardez-le avec une extension `.bat`. Double-cliquez dessus. On devrait vous demander votre identifiant de connexion lorsque MindTerm démarre, ensuite entrez votre mot de passe. Après vous être authentifié, cliquez sur le menu **Tunnels** puis sur **Basic**. Vous devriez voir les tunnels dans la boîte qui s' ouvre. C'est une manière simple de permettre à des utilisateurs distants de démarrer les tunnels sans trop de configuration de leur part. Ils n'ont qu'à double-cliquer sur le fichier `.bat` et entrer leur nom d' utilisateur et mot de passe, et accessoirement lancer vlock. Leur logiciel client peut être pré-configuré pour les profils distants qui se connectent aux tunnels automatiquement.

Quand vous avez fini d'utiliser MindTerm, veillez à bien fermer toutes les applications utilisant un tunnel. Si vous oubliez de fermer les programmes utilisant un tunnel, MindTerm affichera un message lorsque vous essaierez de quitter depuis la console ou de quitter le programme.

Qu'en est-il de VNC et NTOP? Ces services fonctionnent de la même façon. Ici, le serveur VNC tournait sur une station de travail RedHat 7.0. Quand vous démarrez le serveur VNC, il commence par écouter sur le port 5901, puis chaque serveur suivant incrémente d'un port, de telle façon que la 2ème instance de VNC écoutera sur le port 5902, la 3ème sur 5903, etc. Sous Linux, vous pouvez lancer plusieurs serveurs VNC et les gens peuvent se connecter indifféremment à chaque serveur VNC. Dans MindTerm, vous pouvez ajouter un tunnel VNC simplement, avec les réglages suivants :

- Local Port: **2001**
- Remote Host: *Le nom de votre serveur VNC distant*.
- Remote Port: **5901** *(s'il s'agit de la 1ère instance de serveur qui tourne)*

Cliquez sur **Add**. Cliquez ensuite sur **OK** dans le menu de confirmation.

Lancez l'application vncviewer sur votre machine locale et tapez : **localhost:2001**, puis, quand cela est demandé, le mot de passe pour le bureau VNC, et vous avez une session VNC cryptée.

NTOP fonctionne de la même manière. Si vous voulez exécuter NTOP en mode web en tant que moniteur réseau, vous pouvez faire emprunter aux connexions un tunnel vers votre machine locale et visualiser les statistiques dans votre navigateur local, sans avoir à installer un serveur web ou ouvrir le port 3000 sur votre serveur distant. Par défaut, NTOP en mode web écoute sur le port 3000 et attend une connexion HTTP pour afficher des statistiques réseau. Créez simplement un tunnel vers le serveur exécutant le serveur SSH et NTOP. Exécutez d'abord NTOP en mode web : ntop -d -w 3000 . Puis ajoutez ces réglages au tunnel MindTerm :

- Local Port: **2080**
- Host: *Serveur exécutant NTOP*.
- Remote Port: **3000**

Cliquez sur **Add**. Cliquez ensuite sur **OK** dans le menu de confirmation.

Ouvrez ensuite votre navigateur web et tapez dans la barre d'URL : **http://localhost:2080** Vous devriez maintenant voir les pages de statistiques réseau pour NTOP (voir le manuel NTOP pour ajouter un accès protégé par mot de passe à l'affichage NTOP). Idem, si vous voulez installer un serveur web pour utiliser les applications web pour pouvoir contrôler votre serveur ou pare-feu, créez simplement un tunnel vers le port 80. Vous n'avez pas à ouvrir un port sur l'interface publique. Il faut simplement relier le serveur web à l'interface locale et créer un tunnel vers le port 80 de l'hôte distant. Pour Apache, éditez le fichier `httpd.conf` et remplacez l'option *BindAddress* * par **BindAddress 127.0.0.1**. Ajoutez ensuite **localhost** à la directive *ServerName* : **ServerName localhost**. Enfin, remplacez la directive *Listen* par : **Listen 127.0.0.1:80** Comme vous pouvez le voir à présent, MindTerm peut sécuriser à peu près n'importe quel service TCP. Il peut être utilisé sur un serveur distant pour exécuter Webmin, qui est une excellente application web pour administrer vos serveurs. Celle-ci est livrée avec ses propres serveurs web basés sur du perl, et écoute par défaut sur le port 10000. Créez simplement un tunnel vers celui-ci en utilisant MindTerm et ça devrait marcher sans modifier votre application Webmin ou votre navigateur web local. Le fichier zip MindTerm en téléchargement contient de nombreux exemples utiles, comme son utilisation depuis une ligne de commande ou une explication de toutes les options du menu. MindTerm possède plus de fonctionnalités que ce qui est décrit dans ce tutoriel, mais l'option concernant le tunnel vaut vraiment la peine d'y passer du temps.

---

6. MindTerm sur le web

MindTerm peut être également utilisé sur le web. Les utilisateurs n'ont pas à télécharger l'application. Copiez simplement le fichier `mindtermfull.jar` dans un répertoire du répertoire web et les utilisateur pourront l'utiliser simplement comme une application intégrée ou comme une applet Java autonome. Par exemple, créez un dossier nommé `mindterm` dans votre répertoire web. Copiez le fichier `mindtermfull.jar` utilisé plus haut, dans le dossier `mindterm` du répertoire web. Ajoutez ensuite le fichier `index.html` dans le répertoire avec le contenu suivant (repris du `README`) :

<html> <head></head> <body> <applet archive="mindtermfull.jar" code=mindbright.application.MindTerm width=700 height=400> <param name=server value="<nom de votre serveur>"> <param name=port value="22"> <param name=cipher value="blowfish"> <param name=te value="xterm-color"> </applet> </body> </html>

MindTerm 2.0 est maintenant disponible. L'argument pour démarrer l'applet web a légèrement changé. A la place des paramètres d'applet ci-dessus, et de l' exemple de code ci-dessous, modifiez la ligne :

```
<applet archive="mindtermfull.jar"
  code=mindbright.application.MindTerm width=700 height=400>
```

comme ceci :

```
<applet archive="mindtermfull.jar"
  code=com.mindbright.application.MindTerm width=700 height=400>
```

Seul le *com.* doit être ajouté au paramètre *code*= de l'applet. Le code ci-dessous sera donc modifié comme ceci :

```
<applet archive="mindterm_ns.jar" code=com.mindbright.application.MindTerm.class width=1
  height=1>
```

Naviguez jusqu'à l'emplacement du répertoire dans votre navigateur web *(http://<nom de votre serveur>/mindterm/index.html)*, assurez-vous que Java est activé dans votre navigateur et vous devriez pouvoir vous connecter au serveur maintenant.

Pour pouvoir créer des tunnels, la version la plus récente de MindTerm doit être téléchargée depuis le site web de MindBright, version 1.99. Cette archive contient une applet signée par MindBright qui peut être utilisée dans votre page web pour créer des tunnels comme expliqué plus haut. Après avoir téléchargé la version la plus récente, ajoutez le fichier `mindterm_ns.jar` au répertoire `mindterm` dans votre serveur web. Maintenant ajoutez un fichier qui s'appelle `standapplet.html` au répertoire `mindterm` et ajoutez le code suivant pour démarrer MindTerm comme client à part pour créer des tunnels. (*Remarque : l'archive contient une applet pour à la fois Netscape et Explorer*)

<html> <head></head> <body> <applet archive="mindterm_ns.jar" code=mindbright.application.MindTerm.class width=1 height=1> <param name=server value="<nom de votre serveur>"> <param name=port value="22"> <param name=cipher value="blowfish"> <param name=sepframe value="true"><!-- exécuter dans un cadre séparé ou pas --> <param name=autoprops value="both"><!-- activer/désactiver sauvegarde/chargement automatique des réglages --> </applet> </body> </html>

Maintenant naviguez jusqu'à l'emplacement du répertoire dans votre navigateur web *(http://<lt;nom de votre serveur>/mindterm/standapplet.html)* . Cela démarrera MindTerm en tant qu'applet Java autonome, comme si elle était lancée depuis une ligne de commande. Des tunnels peuvent être créés en utilisant les balises de l'applet de façon à ce que les utilisateurs n'aient rien d'autre à faire que naviguer jusqu'à la page et se connecter. Ensuite il auraient accès à leurs services comme expliqué dans les exemples précédents. Ils peuvent cependant créer leurs propres tunnels ou de nouveaux tunnels depuis le menu *Tunnels* comme expliqué plus haut. Le `README` livré avec l'archive zip MindTerm possède de nombreux autres paramètres d'applet qui peuvent être ajoutés. Quand vous créez des tunnels, vous pouvez alors cliquer sur **File** puis sur **Save** pour que les tunnels que vous avez créés soient conservés quand vous vous identifiez à nouveau.

Quelques consignes de sécurité : vous ne pouvez pas vous connecter à un autre serveur en utilisant l'applet d'identification initiale. Vous ne pouvez vous connecter que sur le serveur où l'applet est située. Cependant, après vous être connecté avec succès, vous pouvez alors vous connecter sur un autre serveur en ligne de commande. De plus, cette applet MindTerm est signée par MindBright, donc il faut contacter le département de ventes chez MindBright pour obtenir une signature cryptographique pour votre organisation. Cela dit, si c'est nécessaire.

---

7. Remarques sur la sécurité

Lorsqu'une session SSH démarre, les clés publiques sont envoyées au travers d'une connexion non sécurisée jusqu'à ce que le procédé d'authentification soit établi. Cela peut permettre à une personne d'intercepter une session SSH et de placer sa propre clé publique dans le processus de connexion. SSH est conçu pour avertir l'utilisateur si une clé publique a changé par rapport à ce qui existe dans son propre fichier known_host. L'avertissement donné ne passe pas du tout inaperçu et SSH annulera la connexion si les clés publiques sont différentes, mais l'utilisateur peut quand même faire confiance au certificat car il peut penser que son entreprise a changé les clés publiques du serveur. Ce type d'attaque n'est pas difficile car le paquetage dsniff mentionné plus tôt contient les outils pour l'effectuer. Cette attaque est plus communément appelée *"attaque de l'homme du milieu" [man-in-the-middle attack] (The End of SSL and SSH).*

Un correctif temporaire et facile pour cela consiste à d'abord apprendre à l'utilisateur comment reconnaître les signes indiquant que la clé de l'hôte a changé, et comment faire pour récupérer la bonne clé publique. Deuxièmement, il faudrait poster la clé publique du serveur SSH sur un site web, serveur FTP ou la distribuer d'une autre façon pour que les utilisateurs puissent y avoir accès n'importe quand.

---

8. Conclusion

SSH et MindTerm ensemble peuvent fournir aux utilisateurs locaux et distants un haut niveau de sécurité à l'aide d'une simple petite application qui se lance sans installation préalable. Ils peuvent également être utilisés depuis à peu près n'importe quelle plateforme disponible. Java a été choisi pour son interopérabilité entre plateformes. S'il existe un JRE disponible pour une plateforme utilisée par une personne, cette personne peut utiliser l' application MindTerm pour communiquer de manière sûre sur de longues distances. Comme SSH est en train de devenir le standard pour l'administration et l'identification à distance, bientôt quasiment toutes les plateformes seront capables de faire tourner un serveur SSH. MindBright travaille actuellement sur un serveur SSH Java.

Ce tutoriel montre également comment quelqu'un peut faire un tunnel au travers d'un pare-feu. Ce n'est pas du tout le but de cet article. On espère que les gens l'utiliseront comme un remplaçant de type VPN sûr, rapide et gratuit pour l'administration à distance, pour les gens en voyage d'affaire ; on espère aussi que d'autres secteurs verront l'utilité de cet excellent programme. Tant que vous serez autorisé à effectuer des connexions SSH vous pourrez faire passer des services vers une machine distante au travers d'un tunnel. Les administrateurs système et sécurité devraient établir une politique contre la création de tunnels au travers des pare-feu car cela peut causer des brèches de sécurité internes en cas de mauvaise utilisation. Rappelez-vous que la communication est sécurisée, mais que les commandes et fichiers auxquels vous accédez et/ou téléchargez sont quand-même exécutés sur vos machines locale et distante. De plus, toute commande que vous tapez est également journalisée sur la plupart des serveurs. SSH protège les données sur le réseau ou l'Internet, mais ce qui est fait sur les machines distantes peut être journalisé. SSH et MindTerm ne protégeront pas contre quelqu'un essayant d'obtenir l'accès à l'ordinateur d'un utilisateur distant, et installant des programmes enregistreurs de frappe ou d'autres outils de surveillance.

Il est très simple et rapide d'établir une communication sécurisée, mais le seul moyen pour les utilisateurs d'accroître l'utilisation de communications sécurisées est d'encourager leurs entreprises, les institutions financières, les services médicaux et autres entreprises à offrir des services sécurisés.

---

9. Références

Broadband Access to Increase in Workplace. 25 Jan. 2001. CyberAtlas. 12 Mar. 2001 <http://cyberatlas.internet.com/markets/broadband/article/0,,10099_570571,00.html>.

Broadband Moving On Up. 10 Jan. 2001. CyberAtlas. 12 Mar. 2001. <. http://cyberatlas.internet.com/markets/broadband/article/0,,10099_556391,00.html>.

Connolly, P.J. "Secure the home office sensible and easily" Infoworld. 8 Mar. 2001. 22 Mar. 2001. <http://www.infoworld.com/articles/tc/xml/01/03/12/010312tcsoho.xml>.

Eckels, Josh. "Commercial Use" E-mail to Josh Eckels. 13 Mar. 2001

MindTerm: README. MindBright Technology. 3 March 2001 <. http://www.mindbright.se/documentation/README>. Schneier, Bruce. Secrets and Lies: Digital Security in a Networked World. New York:Wiley & Sons, 2000.

Seifried, Kurt. "The End of SSL and SSH" 18 Dec. 2000. SecurityPortal. 12 March 2001 <http://www.securityportal.com/cover/coverstory20001218.html>.

virtual private network: [Definition]. 6 Oct. 2000. Whatis.com. 15 Mar. 2001. <http://whatis.techtarget.com/definitionsSearchResults/1,289878,sid9,00.html?query=virtual+private+network>.

---

10. Foire Aux Questions

Rien pour l'instant.