

ITRAINONLINE MMTK

INFRASTRUCTURES BASÉES SUR LINUX

Préparé par : Tomas Krag, wire.less.dk

ITRAINONLINE MMTK.....	1
INFRASTRUCTURES BASÉES SUR LINUX.....	1
Au sujet de ce document.....	1
Renseignements sur le droit d'auteur.....	1
Le rôle de Linux dans les infrastructures de réseaux.....	2
Introduction.....	3
Prérequis.....	3
Hypothèses.....	3
Scénario 1: Une passerelle pour le service sans fil sur le point d'accès « masquerading ».....	4
Étape 0: Configuration initiale.....	4
Étape 1: Configurer les interfaces.....	4
Étape 2: configurer la passerelle dans le noyau « masquerading in the kernel ».....	5
Étape 3: Configurer le serveur DHCP.....	5
Étape 4: Ajouter de la sécurité: configurer un pare-feu.....	6
Scénario 2: Points d'accès à passerelles transparentes	7
Étape 0: Configuration initiale.....	7
Étape 1: Configurer les interfaces.....	7
Étape : Commencer la passerelle.....	8
Scénario 1 et 2 la façon facile.....	9
Scénario 3: un pare-feu central avec identification.....	9
Sommaire de la configuration.....	9
Exigences d'équipement.....	10
Étape 0: Installation du logiciel.....	10
Étape 1: Configurer Apache2 pour SSL.....	11
Étape 2 : Configurer ChilliSpot.....	12
Étape 3 : Configurer le logiciel.....	12

Au sujet de ce document

Ces documents font partie du ItrainOnline MMTK. Le MMTK est un ensemble intégré de documents et de ressources de formation multimédia destiné à aider les médias communautaires, les centres multimédia communautaires, les télécentres et autres initiatives qui utilisent les technologies de l'information et des communications (TIC) à renforcer les communautés et soutenir le travail de développement.

Renseignements sur le droit d'auteur

Cette unité est présentée sous licence Creative Commons **Paternité - Partage des Conditions Initiales à l'Identique 2.5**. Pour savoir comment utiliser ces documents, veuillez lire la déclaration sur le droit d'auteur accompagnant cette unité ou consulter

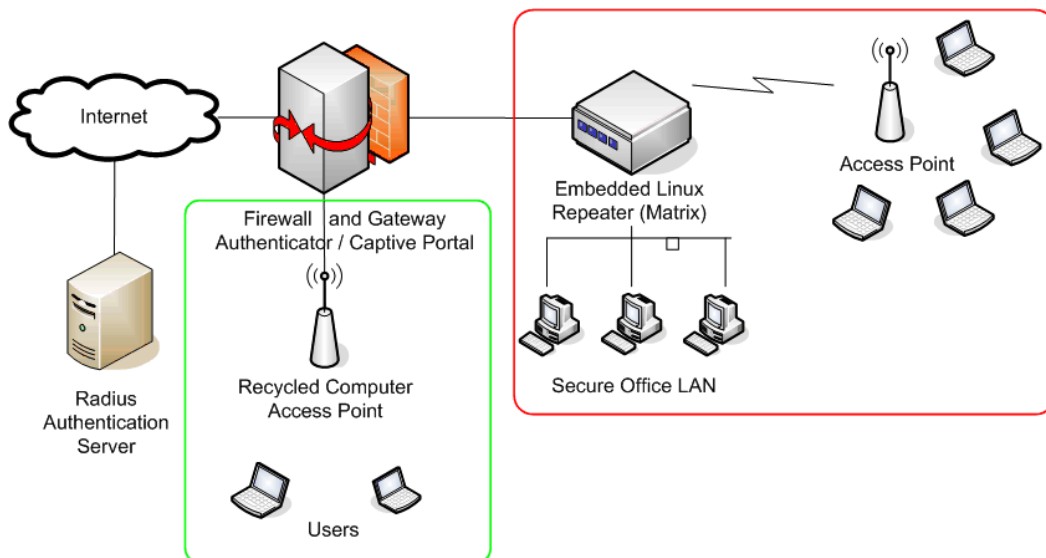
<http://creativecommons.org/licenses/by-sa/2.5/deed.fr>.

Le rôle de Linux dans les infrastructures de réseaux

En général, le système d'opération GNU/Linux (à la différence de Windows) offre à l'administrateur de réseau tout le potentiel d'un accès total à la pile de protocoles réseaux. On peut accéder autant à la couche données-lien « data-link », à la couche réseau et bien sûr à la couche logiciel d'applications.

Jumelé à la flexibilité reconnue de GNU/Linux sur toute une variété d'équipements sur lequel il peut fonctionner, ces avantages font de GNU/Linux un outil très puissant qui peut jouer différents rôles dans l'infrastructure réseau.

Le diagramme suivant présente le scénario qui est à la base de ce guide – mais il est important de noter qu'il ne décrit que des rôles sélectionnés dans le réseau. Premièrement, tous les rôles dans le diagramme auraient pu être intégrés dans un seul ordinateur ou réorganisés différemment. Pour simplifier cependant, un sommaire de chacun de ces rôles a été présenté dans le texte plus bas.



Dans ce diagramme, nous avons illustré un réseau local intégrant 2 segments séparés; un segment ouvert et un segment fermé avec identification. Il y a trois unités différentes dans cet infrastructure sans fil – ils sont différentes dans le diagramme seulement pour illustrer la variété dans les choix. Un point d'accès peut en effet être une unité intégrée comme le Linksys WRT54G, un équipement sans fil fonctionnant sur la base d'un programme basé sur linux comme Matrix Mklou encore un vieux ordinateur recyclé fonctionnant sous Linux.

Dans ce guide, nous verrons comment configurer Linux dans les situations suivantes :

- Un point d'accès sans fil avec « *masquerading/NAT* » (aussi appelé passerelle sans fil).
- Un point d'accès sans fil qui agit comme un pont transparent. Il peut être utilisé comme un point d'accès simple ou comme un répéteur avec deux cartes radio.
- Un routeur central avec un simple système d'identification, permettant l'identification d'utilisateurs provenant de divers points d'accès.

Introduction

Prérequis

De façon à vraiment profiter au maximum de ce guide, les lecteurs devraient être familier avec l'utilisation de Linux et être capable d'installer la version Linux de leur choix. Une connaissance de base des lignes de commandes (interface terminal) est aussi requise.

De façon à suivre la partie sur la configuration des points d'accès inclus dans ce document, un ordinateur fonctionnant sous Linux est requis avec au moins une carte sans fil déjà installée. Les exemples utilisent des cartes et pilotes spécifiques, mais il existe un nombre important de cartes différentes qui devraient fonctionner de la même façon. Voyez le module KFMM- Installation sans fil sur Linux pour des trucs sur comment installer l'interface sans fil sur Linux.

Finalement, une compréhension du réseautage sous TCP/IP (voyez le module mise en réseau avancé si nécessaire).

Hypothèses

Logiciel

Afin de rester simple, ce document assume que vous utilisez un même logiciel sur le même équipement. Les commentaires et idées d'utilisateurs utilisant d'autres versions de Linux ou pilotes sans fil sont les bienvenus (t@wire.less.dk).

Ces exemples devraient aussi fonctionner sous d'autres versions de Linux plus récentes; quoi que nous avons noté que des changements doivent être fait avec Mandrake et Fedora. Cependant, ces changements n'ont pas été testés.

Les exemples sont basés sur Ubuntu, version 5.1 (Breezy Badger), avec une carte sans fil utilisant le pilote madwifi (ou hostap) – voyez les *Ressources additionnelles* pour les liens.

Il est possible de configurer ces ressources avec d'autres pilotes en autant qu'ils sont compatibles avec le Master mode (AP). C'est aussi possible, mais pas toujours désirable, de faire les mêmes configurations en mode direct (Ad hoc) lequel est possible avec l'ensemble des pilotes.

Sous Linux, les programmes suivants sont nécessaires pour compléter l'installation :

- Outils sans fil (iwconfig, iwlist commands)
- iptables firewall
- dnsmasq (serveur de cache DNS et serveur DHCP)

Équipement

Il y a peu d'équipements nécessaire. Un ordinateur de bureau ou un portable, avec un interface Ethernet et un autre avec un ou deux interfaces sans fil (tels que requis pour la partie sur la configuration des passerelles pour les points d'accès « *masquerading* ». Pour ces équipements, aucun processeur spécifique n'est requis. Ce sont des ordinateurs réguliers ou des vieux équipements en voie d'être transformé en point d'accès fonctionnant sous Linux (Linksys WRT54G).

Le document sur les ressources additionnelles présente des liens vers des vendeurs suggérés (pas recommandé) qui peuvent être utile ici.

Le scénario sur le pare-feu avec identification demande plus d'équipements mais fonctionnera sous un ordinateur avec 500 MHz x86 avec 10 Gb de disque dur (ou même une carte compacte USB 2 Gb), et 128 Mb RAM. Tout cela bien sûr dépend de l'utilisation prévue du système,

Scénario 1: Une passerelle pour le service sans fil sur le point d'accès « masquerading »

C'est le plus simple des scénarios, et spécialement utile lorsque vous voulez un seul point d'accès pour un bureau ou autre.

1. Il existe un pare-feu dédié. Et une passerelle tournant sous Linux. Vous souhaitez ajouter un interface sans fil.
2. Vous avez sous la main un vieil ordinateur ré-usiné ou un portable que vous pouvez utiliser comme un point d'accès.
3. vous voulez plus de capacité de suivi, de sécurité que ce qu'offre la plupart des points d'accès commercial, mais vous ne souhaitez PAS payer pour un point d'accès d'entreprise (voyez aussi le scénario 3)
4. Vous voulez qu'une seule machine agisse comme 2 points d'accès (et pare-feu) afin de pouvoir offrir un accès Internet sécurité (avec identification) et un accès ouvert pour les invités (voyez aussi le scénario 3).

Étape 0: Configuration initiale

Démarrez votre ordinateur déjà configuré sur GNU/Linux. Cela peut être un serveur sous Ubuntu ou encore Fedora. Cet ordinateur doit avoir au moins 2 interfaces pour ce travail, et au moins une de ces interface doit être sans fil. Le reste de la description assume que votre port Ethernet (eth0) est connecté à l'Internet (et qu'il obtient son adresse IP d'un serveur DHCP) et qu'il existe un interface sans fil (wlan0) qui sera le point d'accès. Tel que précédemment mentionné, le mode « master » (ou mode de point d'accès) a seulement été testé avec les pilotes madwifi (puce Atheros) et hostap (puce Intersil Prism2/2.5/3).

Pour savoir si votre puce offre le mode « master », essayez la commande suivante comme « root » :

```
# iwconfig wlan0 mode Master
```

(remplacez wlan0 par le nom de votre interface).

Si vous ne recevez Pas de message d'erreur, votre carte doit faire l'Affaire (mais il n'y a pas de garantie). Si vous recevez un message d'erreur, vous pouvez essayer la même configuration en mode directe (ad hoc), lequel fonctionne sur toutes les puces. Cela exigera cependant que vous connectiez par la suite à ce point d'accès tous les postes en mode direct aussi. Éventuellement, cela pourra ne pas fonctionner tel que vous l'attendiez.

Avant de continuer, assurez-vous d'installer dnsmasq sur votre machine (utilisez le gestionnaire graphique d'installation de votre version. Sur Ubuntu, vous pouvez faire ce qui suit (après avoir configuré votre répertoire de nouveaux programmes - voyez <http://www.ubuntuguide.org/>):

```
# sudo apt-get install dnsmasq
```

Étape 1: Configurer les interfaces

Configurez le serveur de façon à ce que eth0 soit connecté à l'Internet (utilisez l'outil de gestion de votre version ou essayez la commande suivante en mode « root »):

```
# dhclient eth0
```

Configurez votre interface sans fil en mode « master » et donnez-lui le nom de votre choix :

```
# iwconfig wlan0 essid "mon reseau" mode Master enc off
```

Le «enc off» annule le mode de cryptage WEP. **To enable we add a hex-key sting of the relevant length after the keyword enc instead , i.e. iwconfig wlan0 essid "my network" mode Master enc 1A2B3C4D5E)**

Donnez maintenant une adresse IP à votre interface sans fil, dans un sous-réseau privé. Mais faites attention de ne pas lui donner le même sous-réseau que celui de l'adaptateur Ethernet:

```
# ifconfig wlan0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
```

Étape 2: configurer la passerelle dans le noyau « masquerading in the kernel »

De façon à pouvoir permettre aux adresses d'être traduites entre les deux interfaces, nous devons installer « masquerading (NAT) » au sein même du noyau de Linux. Il faut d'abord ouvrir le module en question :

```
# modprobe ipt_MASQUERADE
```

Maintenant, nous allons éliminer toutes les règles existantes en termes de pare-feu pour être certain que celles-ci n'empêchent pas la circulation des paquets de données entre nos deux interfaces. Si vous avez un pare-feu en activité, soyez certain de savoir comment restaurer les règles par la suite.

```
# iptables -F
```

Permettre les fonctions NAT entre les deux interfaces

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Finalement, nous devons permettre au noyau de transmettre des paquets entre les deux interfaces :

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Sur les version de Linux basé sur Debian (telles que Ubuntu), ce changement peut aussi être fait en modifiant le fichier :

```
/etc/network/options
```

changez la ligne

```
ip_forward=no
```

pour

```
ip_forward=yes
```

puis redémarrez les interfaces de réseaux avec :

```
# /etc/init.d/network restart
```

ou

```
# /etc/init.d/networking restart
```

Étape 3: Configurer le serveur DHCP

À ce stade, vous devriez déjà avec un point d'accès fonctionnel. Cela pourra se vérifier en connectant une autre machine sans fil et en donnant à cette machine une adresse IP contenu dans le même éventail que celle de notre interface sans fil sur le serveur (10.0.0.0/24 si vous avez suivi les exemples).

De façon à rendre l'accès plus facile à ceux qui ne connaissent pas l'éventail IP, nous créerons un serveur DHCP qui distribuera les adresse IP automatiquement aux clients sans fil.

Nous utilisons pour cela le programme **dnsmasq**. Comme son nom l'indique, ce programme a été développé spécialement pour la pare-feu NAT et en plus du DHCP, il fournit une cache au serveur. Disposer d'une cache pour le serveur peut être particulièrement utile si votre connexion Internet est faible ou impose une latence trop élevée comme c'est le cas par exemple des VSAT ou des connexions par modem. Cela signifie que les demandes de DNS pourront être résolues localement, épargnant beaucoup de bande passante sur l'Internet et faisant paraître la connexion passablement plus rapide.

Installez dnsmasq à partir de votre gestionnaire de programmes ou téléchargez le code source directement du Web (voyez les Ressources additionnelles).

Tout ce qui est nécessaire pour faire fonctionner dnsmasq est de modifier quelques lignes dans son fichier de configuration. Ces lignes sont:

```
/etc/dnsmasq.conf
```

Le fichier de configuration est bien écrits et offre plusieurs types de configuration. Pour avoir le serveur DHCP élémentaire fonctionnel, il faut seulement modifier 2 lignes.

Trouvez les lignes qui commencent par:

```
interface=
```

soyez certain de voir :

```
interface=wlan0
```

Puis trouvez les lignes qui commencent par:

```
#dhcp-range=
```

réécrivez la ligne pour y inclure l'adresse IP utilisée :

```
dhcp-range=10.0.0.10,10.0.0.110,255.255.255.0,6h
```

sauvez le fichier et redémarrez dnsmasq:

```
# /etc/init.d/dnsmasq start
```

Voilà, vous devriez être en mesure de vous connecter au serveur en tant que point d'accès et recevoir une adresse IP du serveur DHCP. Vous devriez par la suite pouvoir vous connecter à l'Internet par ce serveur.

Étape 4: Ajouter de la sécurité: configurer un pare-feu

Lorsque que les serveurs sont configurés et testés, vous pouvez ajouter un pare-feu en utilisant n'importe quel outil pare-feu inclut dans votre distribution.

Voici quelques outils pare-feu :

- firestarter – un programme client pour Gnome qui exige que le serveur tourne avec GNOME
- KNetfilter – un programme client pour KDE qui exige que le serveur tourne avec KDE
- Shorewall – un ensemble de scripts et fichiers de configuration qui facilitent la mise en place d'un pare-feu « *iptables* » **There are also front ends for shorewall such as webmin-shorewall**

- fwbuilder - un outil puissant mais plutôt complexe qui vous permet de créer des scripts iptables sur une machine séparée du serveur et de les transférer sur le serveur par la suite. Il ne requiert pas que vous installiez un système d'opération graphique sur le serveur. C'est une option sérieuse pour ceux qui sont les plus conscients de la sécurité.

Pour plus d'information pour mettre en place davantage de système de sécurité, voyez aussi le **scénario 3**.

Scénario 2: Points d'accès à passerelles transparentes

Ce scénario peut être utilisé pour un répéteur à deux cartes radio, ou pour un point d'accès connecté à un réseau Ethernet pour lequel nous souhaitons que les deux côtés du point d'accès soient sur le même sous-réseau. Cela peut être particulièrement utile lorsque vous souhaitez configurer plusieurs points d'accès tout en maintenant un seul pare-feu central avec peut-être un serveur d'identification. Puisque tous les clients sont sur le même sous-réseau, ils peuvent tous être gérés par les mêmes serveur DHCP et pare-feu.

Par exemple, vous pouvez configurer un serveur comme dans le scénario no 1, mais en utilisant 2 interfaces câblée Ethernet plutôt qu'une câblée et l'autre sans fil. Une interface sera votre connexion Internet et l'autre sera branchée à un commutateur. Puis, connectez autant de point d'accès que vous le souhaitez à ce même commutateur. Configurez ces points d'accès comme des passerelles transparentes et tout le monde accédera au réseau par le même pare-feu et utilisant le même serveur DHCP. Par la suite, vous pourrez suivre le scénario 3 et ajouter l'identification au serveur central.

Étape 0: Configuration initiale

Mis à part le fait qu'il ne nécessite pas dnsmasq, la configuration initiale d'un point d'accès à passerelle transparente est similaire est celle d'un point d'accès à passerelle « masquering » (voyez **étape 0** sous **Scénario 1**).

En plus de cette configuration, un programme passerelle est nécessaire pour faire ce travail. Le programme existe pour Ubuntu et les autres versions basées sur Debian tout comme Fedora Core. Soyez certain que le programme est installé et que la commande *brctl* est disponible avant de procéder.

Étape 1: Configurer les interfaces

Sur Ubuntu ou Debian, on peut configurer les interfaces en modifiant le fichier

```
/etc/network/interfaces
```

Ajoutez une section comme ce qui suit, mais changez en conséquence les noms des interface et adresse IP. L'adresse IP et le sous-réseau doivent être assorti au réseau. Cet exemple assume que vous construisez un répéteur sans fil avec 2 interfaces (wlan0 et wlan1), mais cela peut être aussi fait avec un interface câblé et un interface sans fil (eth0 et wlan0).

Ajoutez ce qui suit au fichier:

```
auto br0
iface br0 inet static
    pre-up ifconfig wlan 0 0.0.0.0 up
    pre-up ifconfig wlan1 0.0.0.0 up
    pre-up iwconfig wlan0 essid "client"
    pre-up iwconfig wlan1 essid "AP" mode Master
    address 192.168.1.2
    network 192.168.1.0
    netmask 255.255.255.0
```

```
broadcast 192.168.1.255
gateway 192.168.1.1
bridge_ports wlan0 wlan1
post-down ifconfig wlan1 down
post-down ifconfig wlan0 down
```

regardez bien toutes les autres sections du fichier qui réfèrent à wlan0 or wlan1 pour être certain que rien n'interfère avec vos nouvelles configurations .

Cette syntaxe pour créer des passerelles via /etc/network/interfaces est spécifique aux versions basées sur Debian et les détails pour configurer ces passerelles sont gérées par plusieurs scripts, notamment :

```
/etc/network/if-pre-up.d/bridge
/etc/network/if-post-down.d/bridge
```

Et le document pour configurer les passerelles sont à:

```
/usr/share/doc/bridge-utils/
```

Si ces scripts n'existent pas sur votre version (comme Fedora Core) voici une alternative de configuration /etc/network/interfaces qui parviendra au même résultat avec seulement quelques petits problèmes marginaux supplémentaires :

```
iface br0 inet static
pre-up ifconfig wlan 0 0.0.0.0 up
pre-up ifconfig wlan1 0.0.0.0 up
pre-up iwconfig wlan0 essid "client"
pre-up iwconfig wlan1 essid "AP" mode Master
pre-up brctl addbr br0
pre-up brctl addif br0 wlan0
pre-up brctl addif br0 wlan1
bridge_ports wlan0 wlan1
post-down ifconfig wlan1 down
post-down ifconfig wlan0 down
post-down brctl delif br0 wlan0
post-down brctl delif br0 wlan1
post-down brctl delbr br0
```

Étape : Commencer la passerelle

Quand la passerelle est définie comme interface, lancez-la en tapant :

```
# ifup -v br0
```

le "-v" signifie « *verbose output* » et vous donnera de l'information sur ce qui se passe.

Sur Fedora Core (une version non basée sur Debian) vous aurez besoin de donner à votre interface-passerelle une adresse IP et ajouter un routage par défaut au reste du réseau :

```
# ifconfig br0 192.168.1.2 netmask 255.255.255.0 broadcast 192.168.1.255
# route add default gw 192.168.1.1
```

Vous devriez maintenant pouvoir connecter un ordinateur portable au réseau sans fil pour accéder à l'Internet (ou minimalement au reste du réseau).

Si vous souhaitez avoir plus d'information au sujet de votre passerelle, regardez la commande *brctl* . par exemple:

```
# brctl show br0
```


devrait vous montrer des informations sur ce que fait votre passerelle.

Scénario 1 et 2 la façon facile

Plutôt que d'installer une version Linux et par la suite configurer votre ordinateur pour qu'il soit une passerelle pour votre point d'accès, vous préférerez peut-être jeter un coup d'œil aux versions de Linux spécifiquement conçues pour ce travail et qui peuvent parfois être aussi simple d'utilisation que d'insérer un CD sur un équipement équipé d'une interface sans fil.

Par exemple, il existe une version sur un seul CD appelé Peeble. Elle est basé sur Debian développé par NYC Wireless. Il existe aussi une version améliorée, développée par Metrix Communications et adaptée pour un ordinateur

m0n0wall est une version de pare-feu qui rend facile la configuration de passerelles « masquerading ». Elle permet aussi de rouler certains logiciels liés comme WRAP.

Linux, et en conséquence ces scénarios peuvent aussi fonctionner sur des équipements dédiés aux points d'accès tels que Linksys WRT54G sur lesquels il est possible d'installer des éléments supplémentaires à ceux d'origine.

Voyez les *ressources additionnelles* pour les liens dédiés à ces versions.

Scénario 3: un pare-feu central avec identification

L'objectif est de créer une passerelle qui forcera les usagers à s'identifier via une page Web obligé. Après configuration, la machine aura deux interfaces réseau (nous utiliserons eth0 et eth1 dans cet exemple). La première interface est connectée à l'Internet (eth0), la seconde est une interface interne par laquelle nous nous connectons aux autres machines (eth1). Cela peut être un port Ethernet avec un commutateur auquel sont attachés un nombre d'autres machines ou de points d'accès (2^e couche passerelle transparente) ou cela peut être une interface sans fil, transformant automatiquement la passerelle en point d'accès (voir scénario 1)

Dans ce scénario, nous ajouterons un système d'identification qui requiert des usagers un nom d'identifiant et mot de passe avant qu'ils puissent accéder aux ressources du réseau (Internet, imprimantes, etc.).

Sommaire de la configuration

ChilliSpot contrôle l'interface interne (eth1) utilisant un module « *vtun kernel* » qui fournit une interface virtuelle (tun0). Dans les faits, le module vtun Kernel est utilisé pour déplacer les paquets de données du mode kernel vers le mode usager de façon à ce que ChilliSpot puisse fonctionner avec toutes sortes de modules hors norme. ChilliSpot installe un serveur DHCP (cela peut être défait via le fichier *conf* de ChilliSpot) sur l'interface tun0.

Un client se connectant sur cette interface voit toutes ses données rejetées jusqu'à ce qu'il soit identifié à partir de la page d'identification. Quand un client non identifié essaie de se connecter à une page Web (par les ports 80 ou 443), la requête est interceptée par ChilliSpot et redirigée à un script Perl appelé *hotspotlogin.cgi* (fourni par Apache sur https).

Hotspotlogin.cgi renvoi une page à l'utilisateur une page avec des champs *nom* et *mot de passe*. L'information soumise dans ces champs est par la suite transférée au serveur FreeRADIUS, qui vérifie dans l'information disponible à son niveau (utilisant PAP or CHAP) Le résultat dans ce cas sera initialement un fichier texte, mais peut aussi être un nombre de services comme LDAP, Kerberos, Unix passwd files pour même Active Directory (probablement).

L'utilisateur est à ce moment rejeté ou identifié par FreeRADIUS qui demande à hotspotlogin.cgi de renvoyer soit un message de rejet ou une page d'entrée avec un lien de déconnexion pour l'utilisateur.

Dès que nous avons installé et testé cela avec un fichier texte, nous remplacerons FreeRADIUS par une base de données MySQL qui nous permettra d'ajouter des fonctions comme les cartes de prépayées pour l'accès Internet.

Exigences d'équipement

N'importe quel PC muni de deux interfaces de réseau devrait fonctionner. Mais contrairement aux scénarios 1 et 2, cette machine devrait avoir un disque dur ou minimalement une carte compact « flash » pour l'espace de rangement puisque MySQL a besoin d'espace.

Étape 0: Installation du logiciel

Essentiellement une répétition des premières étapes du scénario 1; nous avons besoin d'un ordinateur avec 2 interfaces fonctionnant sous Linux. Mais nous nécessiterons aussi de différents programmes qui devront être installés pour ce travail.

Nous avons débuté par l'installation de Ubuntu Linux. Nous avons utilisé la version Hoary mais cela devrait fonctionner aussi sur d'autres versions de GNU/Linux distributions telles que Fedora Core, Mandriva etc.

Ceci a été testé autant avec les versions serveurs que client de Ubuntu. L'installation de base va au-delà de la portée de cette unité, mais le site de Ubuntu présente toute la documentation nécessaire pour installer Ubuntu à partir de zéro.

Quand Ubuntu est installé, nous devons ajouter quelques programmes supplémentaires qui ne sont pas présents par défaut. Si vous ignorez comment installer des programmes sous Ubuntu, lisez ceci AVANT de procéder. Si vous êtes familier avec les lignes de commandes Linux, lisez la page pour la commande apt-get command. Il existe un « How to » - (comment?) pour l'installation du répertoire *Universe* sur Ubuntu et celui-ci est requis pour certains des programmes qui doivent être installés (voir Ressource Additionnelles: Unofficial Ubuntu Guide).

Vous devez installer les programmes suivant via synaptic ou la commande apt-get avant de procéder. Certains de ces programmes peuvent être déjà installés par défaut.

- mysql-server
- apache2
- freeradius
- freeradius-mysql
- phpmyadmin

Finalement, vous devez posséder le programme ChilliSpot (qui ne fait pas partie de la famille Ubuntu) en demandant à l'administrateur de la page Web de ChilliSpot (voir Ressource Additionnelles: ChilliSpot)

Quand vous avez téléchargé le fichier, ouvrez le terminal, aller au répertoire où il se retrouve et tapez ::

```
$ sudo dpkg -i chillispot_1.0RC3-1_i386.deb
```

Utilisant le nom de la version ChilliSpot package que vous avez téléchargé.

Étape 1: Configurer Apache2 pour SSL

Pour des raisons de sécurité, nous voulons présenter la page d'identification via une connexion cryptée (https). Nous avons donc besoin de configurer apache2 pour la fourniture de pages cryptées SSL. Cette section est adaptée d'un courrier sur un forum Ubuntu.

<http://ubuntuforums.org/showpost.php?p=19832&postcount=4>

Premièrement, nous devons générer un certificat SSL pour apache 2:

```
$ sudo apache2-ssl-certificate
```

La commande vous retournera une série de questions sur le nom de votre organisation, le courriel qui sera inclus dans le certificat montré aux personnes se connectant au site.

Pour permettre les extensions SSL pour apache2, tapez :

```
$ sudo a2enmod ssl
```

Configurez maintenant SSL en ajoutant un fichier « config » pour les sites SSL. Créez ce nouveau fichier en mode « root ».

```
$ sudo pico /etc/apache2/sites-available/ssl
```

Ce qui suit est un exemple de fichier qui permet le ssl pour les répertoires par défaut de apache2 et cgi-bin :

```
NameVirtualHost *:443
<VirtualHost *:443>
    ServerAdmin admin@domain.com
    DocumentRoot /var/www/
    SSLEngine On
    SSLCertificateFile /etc/apache2/ssl/apache.pem

    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options FollowSymLinks
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog /var/log/apache2/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn

    CustomLog /var/log/apache2/access.log combined
    ServerSignature On
```

```
</VirtualHost>
```

Maintenant, nous devons dire à Apache d'écouter le port 443 (https) autant que le port 80 (http)
Modifiez le fichier `/etc/apache2/ports.conf` et ajoutez une ligne comme celle-ci :

```
Listen 443
```

Pour confirmer cette nouvelle configuration, nous devons d'abord confirmer la nouvelle configuration du site ssl puis relancer Apache (ou redémarrer le serveur Web).

```
$ sudo a2ensite ssl
$ sudo /etc/init.d/apache2 force-reload
```

c'est fait!

Étape 2 : Configurer ChilliSpot

Finalement, copiez le script cgi de ChilliSpot dans le répertoire cgi-bin de apache2 cgi-bin :

```
$ sudo cp /usr/share/doc/chillispot/hotspotlogin.cgi /usr/lib/cgi-bin/
$ sudo chmod +x /usr/lib/cgi-bin/hotspotlogin.cgi
```

Étape 3 : Configurer le logiciel

Les sections qui suivent sont pour la plupart adaptées de l'installation de ChilliSpot pour Debian Sarge (et en partie des instructions de Fedora Core).

Configuration du réseau et du pare-feu

Nous assumons 2 interfaces réseau,

- eth0 est connecté à l'Internet et devrait être configuré à cette fin (utilisez `ifconfig`, `/etc/network/interfaces`, ou l'outil de configuration sous *System->Administration->Networking*).
- eth1 est l'interface permettant aux autres ordinateurs de se connecter. Cette interface ne devrait pas être initialement configurée.

```
$ sudo ifconfig 0.0.0.0 up
```

de façon à permettre aux paquets d'être transféré, vous devrez changer la ligne qui suit dans `/etc/network/options`:

```
ip_forward=yes
```

et relancer le réseau:

```
$ sudo /etc/init.d/network restart
```

Pour configurer le pare-feu et NAT, vous pouvez utiliser le script de pare-feu dans `"/usr/share/doc/chillispot/firewall.iptables"` comme point de départ. Au minimum, vous devez modifier le fichier pour vous assurer que les deux noms de l'interface sont identiques à ceux de votre système. Trouvez les lignes qui débutent par `INTIF=` et `EXTIF=` et modifiez-les pour qu'ils correspondent à votre système. `INTIF` est l'interface à laquelle les gens se connectent (`eth1`) et `EXTIF` est l'interface avec la connexion Internet (`eth0`). Quand vous aurez revu les règles du pare-feu, vous exécuterez le script par la commande :

```
$ sudo sh /usr/share/doc/chillispot/firewall.iptables
```

Le Script du pare-feu devra être exécuté chaque fois que l'ordinateur est relancé. Une façon simple de s'assurer que cela est fait est de copier le fichier /etc/init.d/

```
$ sudo cp /usr/share/doc/chillispot/firewall.iptables /etc/init.d/chili.iptables
$ sudo chmod u+x /etc/init.d/chilli.iptables
$ ln -s /etc/init.d/chilli.iptables /etc/rcS.d/S40chilli.iptables
```

Configurer le fichier conf de ChilliSpot

Vous devez dire à ChilliSpot où se retrouve le serveur d'identification (qui dans ce scénario est sur la même machine que ChilliSpot). Cela sera fait en modifiant la ligne "/etc/chilli.conf":

```
uamserver https://192.168.182.1/cgi-bin/hotspotlogin.cgi
```

192.168.182.1 est l'adresse par défaut que ChilliSpot donne à l'interface virtuelle tun0, vous pouvez sans risque le laisser comme tel. Pour plus de sécurité, nous devons ajouter un secret partagé entre hotspotlogin.cgi et chilli. Trouvez la ligne "/etc/chilli.conf"

```
#uamsecret ht2eb8ej6s4et3rg1ulp
```

Modifiez cette ligne (enlevez le #) et CHANGEZ le secret pour quelque chose d'aussi bizarre, mais différent. Rappelez-vous de ce secret puisqu'il doit aussi se retrouver dans le script hotspotlogin.cgi script (nous ferons cela plus tard).

Puisque nous utilisons aussi le serveur radius (FreeRADIUS) sur la même machine, nous devons trouver et modifier les lignes qui pointent vers le serveur radius "/etc/chilli.conf". Il faut retrouver :

```
radiusserver1 127.0.0.1
radiusserver2 127.0.0.1
```

Vous devriez aussi changer la ligne dans "/etc/chilli.conf" qui commence par radiussecret, de façon à ce qu'elles n'utilisent pas le secret par défaut qui crypte le trafic entre chilli et radius.

```
radiussecretsquelquechosedetresdifficileadeviner
```

Rappelez-vous de ce mot secret puisqu'il devra être ajouté aussi à la configuration de FreeRADIUS.

Configurer FreeRADIUS

Les fichiers de configuration de FreeRADIUS se retrouvent tous dans le répertoire /etc/freeradius/. Pour commencer (et pour tester), nous utiliserons le fichier texte "/etc/freeradius/users" qui permet un seul usager pour les tests (steve). Par la suite, nous pouvons changer la configuration pour utiliser MySQL pour classer les noms et les mots de passe. Mais premièrement, nous voulons être certain que tout fonctionne de la façon la plus simple.

Modifier "/etc/freeradius/clients.conf".

Trouver la section qui contient

```
client 127.0.0.1 {
```

Soyez certain que cela n'a pas été modifié et, dans la section entre { et ce qui suit }, changez les lignes suivantes :

```
secret = testing123
```

changez testing123 pour qu'il corresponde au secret radius choisi pour "/etc/chilli.conf" (quelquechosedetresdifficileadeviner)

Changez "/etc/freeradius/users"

Modifiez les lignes suivantes dans le fichier :

```
#steve Auth-Type := Local, User-Password == "testing"
```

Ce seront les usager et mot de passe « test » pour être certain que tout fonctionne.

Ajustez le script d'identification « Hotspot »

Pour améliorer la sécurité, nous devons ajouter le uamsecret" de "/etc/chilli.conf" au script d'identification hotspot. Modifiez "/usr/lib/cgi-bin/hotspotlogin.cgi".

Trouver la section qui contient

```
#$uamsecret = "ht2eb8ej6s4et3rg1ulp";
```

Modifiez le secret pour qu'il corresponde à celui dans "/etc/chilli.conf" (le uamsecret, PAS le radiussecret).

Retirez les ajouts de la ligne qui commence par:

```
#$userpassword=1;
```

Maintenant, pour être certain que tous ces changements prendront effet, redémarrez apache2, freeradius et chilli

```
$ /etc/init.d/apache2 force-reload  
$ /etc/init.d/freeradius restart  
$ /etc/init.d/chilli restart
```

Utiliser ChilliSpot

Vous devriez maintenant avoir un serveur d'identification permettant à un ordinateur de se connecter et d'avoir accès au réseau. Connectez un ordinateur dans l'interface eth1 sur la machine avec ChilliSpot, soit avec un commutateur ou un Hub, ou utilisant un câble UTP (ou en connectant une passerelle transparente dans l'interface eth1). Nous appellerons cette machine le client.

Sur la machine client, faites venir l'interface réseau avec DHCP. ChilliSpot devrait vous donner une adresse IP dans le réseau 192.168.182.0/24

Ouvrez votre fureteur et essayez d'aller sur n'importe quelle page Internet.

Vous devriez être redirigé sur une page d'identification avec les champs nom et mot de passe « username - password ». connectez-vous comme « steve » avec le mot de passe « testing » et vous devriez recevoir un message disant que vous vous êtes connecté avec succès. Vous devriez maintenant avoir un accès total à l'Internet à moins que vous ne cliquiez sur le bouton de déconnexion sur la page fournie par ChilliSpot.