



par Georges Tarbouriech
<georges.t/at/linuxfocus.org>

FreeBSD, une véritable alternative



L'auteur:

Georges est un vieil utilisateur d'Unix. Il aime beaucoup les versions libres de ce système extraordinaire.

Résumé:

Oui, vous lisez bien LinuxFocus ! Toutefois, il existe d'autres Unices libres dignes d'intérêt et pleins de caractéristiques marquantes. Cet article est un tour d'horizon d'un autre Unix libre : FreeBSD.

Traduit en Français par:
Georges Tarbouriech
<georges.t/at/linuxfocus.org>

Pourquoi FreeBSD ?

Lorsqu'on parle d'Unices libres, tout le monde comprend "Linux". Certes, Linux est très répandu et chacun (ou presque) en a entendu parler. Le nombre de nouveaux utilisateurs va croissant et les médias ("spécialisés" ou non) font beaucoup de "bruit" autour de ce système. Une des raisons principales de ce succès vient de ce que Linux fonctionne surtout sur des machines Intel, autrement dit, les plus répandues. Bien sûr, il peut fonctionner avec beaucoup d'autres processeurs, puisque l'une des idées derrière la conception d'Unix était "nous nous moquons du processeur". Pourtant Linux n'aurait sans doute pas eu autant de succès s'il avait été développé pour un processeur plus "confidentiel", ou tout au moins, ça aurait pris plus longtemps.

Mais l'Unix libre ne se limite pas à Linux : la "famille" BSD est là aussi. Il en existe trois "saveurs" principales : FreeBSD, NetBSD et OpenBSD. BSD signifie Berkeley Software Distribution. Si vous voulez plus ample information sur l'histoire d'Unix, vous pouvez jeter un coup d'oeil ici (Sachez que cet article ne parle pas des différences entre BSD et System V : un livre y suffirait à peine.) Chaque "saveur" a quelque chose de particulier :

- NetBSD fonctionne sur toutes les plates-formes au monde (ou presque), y compris sur de très vieilles machines : disons entre 30 et 40. NetBSD apporte la preuve de ce qui a été dit plus haut : Unix se moque des processeurs. C'est donc le bon choix si vous possédez un sympathique "rossignol" chez vous ou au travail (mais pas seulement, bien sûr !). Personnellement je l'ai utilisé pendant quelques années sur

Amiga et sur machines Intel avec beaucoup de satisfaction.

- OpenBSD est connu pour être l'un des Unix les plus sécurisés. Il est idéal pour des serveurs et, bien évidemment, fonctionne avec différents processeurs. Il est certainement le système le plus utilisé pour les serveurs sur Internet. L'utiliser chez soi est plutôt déconseillé, mais chacun fait ce qui lui plaît...

- FreeBSD est un amalgame des deux précédents. Il est particulièrement dédié aux processeurs Intel et Alpha et "hérite" d'une partie de la philosophie d'OpenBSD pour ce qui concerne la sécurité. Comme pour Linux, le fait qu'il fonctionne avec des processeurs Intel en fait un des BSD parmi les plus répandus. Pour un ordinateur personnel, c'est un très bon choix. Ce qui ne veut pas dire qu'il ne doit pas être utilisé au travail !

Il y a autre chose concernant FreeBSD : ces gens-là "partagent" leur travail avec Apple, pour la partie logiciel libre de Mac OS X. Et comme j'aime beaucoup OS X... Au fait, pour éviter toute confusion : même si OS X est basé sur BSD, son noyau n'a rien à voir avec celui de FreeBSD. Il s'agit d'un noyau Mach, déjà utilisé avec NeXTstep, "père" d'OS X : évidemment, NeXTstep étant né à la fin des années 80, ce noyau a un peu évolué.

Maintenant que nous avons fait connaissance... allons un peu plus loin.

Obtenir et installer FreeBSD

Comme la plupart du logiciel libre, FreeBSD peut être téléchargé sur le site maison. De toute évidence, cela réclame une connexion rapide. Rassurez-vous : il est facile de trouver des CD partout dans le monde.

Le système actuel (au moment d'écrire ces lignes) est le 4.6. Cet article traite de la version i386. Installer FreeBSD à partir d'un CD-ROM (ou autre) est simple... même s'il n'y a pas d'interface graphique pour ce faire ! A la place, vous utiliserez ncurses. Suivez les instructions et FreeBSD se retrouve sur votre disque dur. Comme d'habitude, vous devrez partitionner votre disque, configurer quelques "bricoles" telles que le réseau ou le noyau (optionnel). Ensuite vous rencontrerez sysinstall, un très bon outil, qui comme son nom l'indique, fait la plus grande partie du travail d'installation. Que vous installiez ou non la totalité des paquetages dépend de vous. Vous pouvez les ajouter n'importe quand, de toute manière.

La gestion de ces paquetages est elle aussi évidente. Vous pouvez vous servir des commandes pkg, comme sous Solaris, par exemple, ou vous pouvez préférer les ports. En résumé, les commandes pkg ressemblent aux rpm ou aux deb de Linux, alors que les ports sont des archives compilées avec make et make install. En d'autres termes, rien de complexe ou de difficile.

Puisque nous comparons à Linux, ajoutons qu'il existe un paquetage de compatibilité qu'il est recommandé d'installer : il vous permettra de compiler et d'exécuter des applications Linux.

Encore une fois, tout est évident, et au cas où, un très bon manuel est fourni qui explique les différentes étapes. De plus, ce dernier existe en plusieurs langues.

En conséquence, il n'est pas nécessaire de s'attarder d'avantage sur le sujet.

Configurer FreeBSD

Encore une fois, nous ne pourrions pas tout dire. Le plus important est de savoir où se trouvent les fichiers essentiels. Comme d'habitude, vous les trouverez dans le répertoire /etc. Ils se nomment

rc.quelquechose et peuvent gérer différentes parties : configuration générale, réseau, sysctl, etc. Le fichier rc.conf est important, puisque, comme le nom l'indique, il est conçu pour personnaliser la configuration. Dans ce fichier, vous pouvez dire au système de démarrer ou non certains démons (sshd, sendmail, etc), le type de pare-feu utilisé, d'activer ou non le niveau de sécurité du noyau, d'activer ou non IPv6, etc. Il existe des tas d'options qui sont visibles dans le fichier par défaut, étonnamment nommé /etc/defaults/rc.conf. Ne vous trompez pas : ce fichier contient les paramètres par défaut, pas des exemples. C'est-à-dire, le fichier /etc/rc.conf remplace les paramètres par défaut. En d'autres termes, ne copiez pas le fichier /etc/defaults/rc.conf vers /etc/rc.conf ou n'essayez pas de le modifier. Les changements doivent être réalisés dans /etc/rc.conf. Pour information, les choix effectués à partir de *sysinstall* sont automatiquement inclus dans /etc/rc.conf (configuration de carte réseau, nom d'hôte, niveau de sécurité, etc). Remarque importante : FreeBSD n'a aucun service actif par défaut dans /etc/inetd.conf. Cela signifie, qu'au démarrage, les démons gérés par inetd sont inactifs. A mon humble avis, ce devrait être obligatoire sur tous les Unixes. En tout cas, c'est très bien ! Les autres fichiers rc permettent de configurer le pare-feu, sysctl et beaucoup d'autres choses. Ceci nous amène, bien sûr, à la personnalisation de FreeBSD.

Personnaliser FreeBSD

L'un des meilleurs outils de personnalisation se nomme sysctl (sous Linux aussi !). Vous pouvez l'utiliser à partir de la ligne de commande pour définir une valeur spécifique ou vous pouvez créer un fichier sysctl.conf (recommandé à moins que vous n'utilisiez la ligne de commande pour tester). Par exemple, si votre machine sert de passerelle, sysctl peut autoriser l'ip forwarding par la commande *sysctl -w net.inet.ip.forwarding=1*. Pour une meilleure sécurité vous pouvez vérifier que les paquets arrivant sur une interface possèdent bien l'adresse IP correspondant à cette interface grâce à la commande *sysctl -w net.inet.ip.check_interface=1*. Vous pouvez contrôler une grande partie du comportement de votre système avec sysctl : voir la page de manuel. Evidemment, si vous voulez écrire ceci dans un fichier sysctl.conf, la commande sysctl doit être supprimée. Le fichier utilise le format *variable = valeur*. Les exemples ci-dessus deviennent alors :

```
net.inet.ip.forwarding=1
net.inet.ip.check_interface=1
```

Certes, votre propre fichier sysctl.conf devrait contenir plus de deux lignes puisqu'il peut tout contrôler, enfin, presque !

Au début de cet article, nous avons parlé des niveaux de sécurité. Il en existe 4 différents de -1 à 3. -1 représente le niveau sans sécurité et 3 représente le plus haut niveau de sécurité. Il est conseillé de bien comprendre ce que génère chaque niveau avant d'en choisir un. Utiliser des niveaux autres que -1 ou 0, peut amener votre système à ne plus fonctionner comme prévu. Le premier inconvénient apparaît au niveau 1 puisqu'il peut empêcher d'utiliser correctement le serveur X. De même, vous ne pourrez plus, par exemple, charger ou décharger des modules du noyau. Vous voici prévenus !

Toutefois, ces niveaux élevés peuvent être utiles pour un serveur spécifique réclamant une forte protection. En tant qu'utilisateur particulier, le niveau 0 est idéal, mais ça vous regarde...

La commande *chflags* est en relation avec les niveaux de sécurité. Pour en savoir plus, lisez la page de manuel : c'est une commande très puissante.

Continuer là-dessus ferait un article "comment sécuriser FreeBSD", par conséquent, passons au chapitre suivant.

Améliorer FreeBSD

Supposons que vous veniez d'acheter un coffret de CD de FreeBSD 4.5. Quelques jours plus tard arrive la version 4.6. Pas de chance !

Ne vous inquiétez pas : FreeBSD propose CVS pour améliorer le système. Vous avez le choix entre CVS et CVSup. Le premier est plus facile à utiliser alors que le second est plus efficace. Avec ces outils, le "jeu" consiste à récupérer le nouvel arbre source et à synchroniser le tout.

Ensuite, vous devrez compiler l'ensemble avec *make buildworld*. Comment le faire est parfaitement expliqué dans le manuel. Le point important, c'est que ça vous permet d'avoir un système tout neuf et à jour.

Toutefois, ça ne s'arrête pas là. Comme toujours avec le logiciel libre, des vulnérabilités sont trouvées et très rapidement corrigées. A la différence de certains systèmes propriétaires dans lesquels une vulnérabilité peut rester sans solution pour l'éternité (et un jour), les personnes investies dans le logiciel libre réagissent à la vitesse de la lumière et fournissent des correctifs. Bien sûr, c'est votre rôle de suivre ce qui se passe. Le site de FreeBSD a une section sécurité contenant les alertes par version. Vous pouvez y télécharger des fichiers .asc (fichiers texte) vous disant ce que vous devez savoir : une description du problème et sa solution. D'autres sites proposent ce type d'information. Mentionnons par exemple, linuxsecurity, CERT, etc.

Ceci implique évidemment que votre système doit être mis à jour.

Mettre à jour FreeBSD

Vous ne souhaitez pas conserver des failles connues dans votre système, n'est-ce-pas ? FreeBSD, comme les autres Unixes libres fournit des correctifs. Récupérez ces correctifs et appliquez-les. Pour les utilisateurs de Linux, la différence principale vient du fait que vous n'obtenez pas un paquetage corrigé mais un correctif du code source. Cela signifie que vous devrez recompiler le noyau après avoir appliqué le correctif à votre code source, et bien sûr, ceci implique que le code source est présent sur votre machine. C'est vrai aussi pour Linux, sinon que les utilisateurs se contentent souvent de télécharger la rpm (par exemple) modifiée, au moins pour ce qui concerne les applications, mais aussi dans le cas des corrections de noyau.

L'une des différences les plus importantes entre Linux et FreeBSD vient du noyau. FreeBSD est basé sur BSD 4.4 et n'a pas plusieurs versions de noyau comme c'est le cas dans le monde Linux. Cela signifie que c'est un noyau stable mais qui grossit avec le temps, étant plutôt monolithique même s'il peut bénéficier de modules. L'inconvénient, c'est que vous devrez recompiler le noyau plus souvent que sous Linux si vous voulez conserver un système aussi sécurisé que possible.

Je préfère de loin cette philosophie que celle de Linux, mais ce n'est que mon opinion. Je ne pense pas que d'avoir, disons, 3 différents noyaux "en cours" puisse amener à la stabilité. Ne parlons pas des applications devant être "portées" d'une version à l'autre, particulièrement celles consacrées à la sécurité. Un autre exemple pourrait être les systèmes de filtrage de paquets : 3 versions du noyau, 3 systèmes de filtrage de paquets différents ! Et ainsi de suite.

J'ai vraiment beaucoup de respect pour les gens qui accomplissent cet excellent travail, mais est-ce vraiment la marche à suivre ? N'en parlons plus !

Bien sûr, vous pouvez aussi compiler un nouveau noyau pour le faire correspondre à vos besoins... et

cette fois, c'est bien comme sous Linux. La façon de le faire est très bien expliquée dans le manuel, passons donc à autre chose.

Applications

Comme déjà mentionné, une grande majorité d'applications libres (sinon toutes) fonctionne sous FreeBSD. Vous n'avez pas forcément besoin des ports ou des paquetages cités plus haut. Vous pouvez compiler la plupart des archives disponibles, quel que soit le type de logiciel.

Pour les "accros" d'environnements de bureau, Gnome et KDE sont disponibles... et sont inclus dans la distribution. Même chose pour les gestionnaires de fenêtres. Le génial environnement de développement, GNUstep, fonctionne aussi très bien sous FreeBSD. Pour le compiler, il vous faut un gcc plus récent que celui de la distribution : pas de problème ! Récupérez l'archive correspondante et compilez le nouveau compilateur. Bien évidemment les applications GNUstep telles que GNUMail.app, Gorm.app ou ProjectBuider.app fonctionnent aussi parfaitement.

Vraiment, tous les logiciels que vous utilisez sous Linux sont disponibles pour FreeBSD... et il y en a beaucoup d'autres !

Par exemple, FreeBSD dispose d'un paquet d'outils concernant la sécurité. Des outils de surveillance, de gestion, etc, font partie de la distribution.

Pare-feu, proxies, scanners, IDS, etc, sont également disponibles. Par exemple, vous avez le choix entre IPsec ou ipfw. Vous pouvez utiliser nessus, nmap, portsentry, etc. Encore une fois, la plupart font partie de la distribution mais rien ne vous empêche de récupérer et de compiler les dernières versions.

Nous avons déjà abordé les environnements de bureau ou les gestionnaires de fenêtres mais nous n'avons pas dit un mot de ce qui leur est nécessaire pour fonctionner : le X Window System. XFree 4.* est maintenant le système par défaut. Pour information, jusqu'à FreeBSD 4.5, XFree 3.* était celui par défaut. Vous aviez bien XFree 4.* dans la distribution mais le processus d'installation choisissait d'installer XFree 3.*. Maintenant, XFree 4.* est installé par défaut, mais vous pouvez toujours utiliser la version 3.* si vous préférez.

Matériel

Puisque nous parlons de la version i386, disons que la plupart du matériel disponible fonctionne sous FreeBSD, y compris, s'il date un peu. Par exemple, une vieille carte réseau combo de 3Com donnera les résultats espérés si vous précisez le type de media dans la commande ifconfig. Si vous utilisez une connexion BNC (coaxial), il suffit d'ajouter *media 10base2/BNC* à la fin de la ligne ifconfig dans */etc/rc.conf*

Même chose pour quelques cartes graphiques un peu anciennes : XFree 4.* résoudra la plupart des problèmes. C'est particulièrement vrai pour les cartes ATI plutôt âgées.

Pour confirmation, n'hésitez pas à consulter les notes sur le matériel sur le site de FreeBSD.

Sécurité

Nous avons déjà mentionné quelques outils tels que pare-feu, scanners, etc. FreeBSD propose bien d'autres outils de sécurité. OpenSSH, OpenSSL, Kerberos, MD5, etc, sont également inclus dans la distribution. Comme d'habitude, vous ne pouvez pas vivre sans ces outils, mais bien sûr, c'est à vous de surveiller la découverte d'éventuelles failles.

Un exemple récent montre l'importance de la prudence. Imaginons que vous veniez d'être informé d'une faille découverte dans OpenSSH, par exemple. Le comportement adapté consiste à aller sur le site maison et à télécharger la version corrigée.

Vrai, mais pas suffisant ! Contrôlez le checksum du paquetage.

L'exemple d'OpenSSH n'a pas été pris par hasard : quelques semaines plus tôt, quelqu'un a remplacé l'archive originale par une archive corrompue. Comme d'habitude, les gens d'OpenSSH ont réagi très vite. Toutefois, les utilisateurs qui ont téléchargé le paquetage dans ce bref laps de temps sans vérifier le checksum ont installé une version "vérolée". D'où la nécessité d'utiliser des checksums... et de les vérifier.

Même si ça n'entre pas dans le cadre de cet article, insistons sur le fait que la sécurité est avant tout un comportement.

Communications

Le réseau sous FreeBSD est très évolué. La gestion d'IPv6, par exemple, est très bonne. Les moyens de communication actuels ne sont pas non plus oubliés. Si vous utilisez des modems ADSL ou câble, leur configuration est très facile.

De nombreux FAI utilisent PPPoE aujourd'hui. FreeBSD gère cela de manière simple : quelques lignes dans le fichier ppp.conf suffiront. Pour ce qui concerne PPPoA (ATM), pour l'instant, seul le modem Alcatel est supporté. Ceci devrait changer très rapidement (il existe peut-être déjà un support pour d'autres modems).

Documentation

Le manuel fourni n'est pas la seule documentation disponible. Vous disposez aussi d'une FAQ, d'un guide d'installation, etc, accessibles sur le site de FreeBSD. Une autre source d'information importante se trouve sur <http://www.freebsd-howto.com>.

Et après ?

C'est un très bref tour d'horizon. Ce système, comme les autres Unixes libres est très intéressant. Il possède des caractéristiques marquantes et mérite vraiment un essai. Malheureusement, cet article ne peut pas être une véritable visite guidée et des tas de particularités d'importance n'ont même pas été abordées.

FreeBSD demandera un peu plus de connaissance d'Unix que certaines distributions Linux très typées "Windos", mais il reste très facile à utiliser.

Certes, quelqu'un n'ayant aucune connaissance d'Unix peut rencontrer des problèmes puisqu'il devra

souvent recompiler son système après mise à jour. Toutefois, c'est un excellent moyen d'apprendre, même si ça peut prendre un "certain" temps. On semble avoir oublié aujourd'hui, qu'à l'origine, Unix a été conçu par des développeurs pour des développeurs, et non pas pour l'utilisateur normal ou le particulier.

C'est une chance pour ces derniers que les Unixes libres ou propriétaires aient transformé l'approche originale. Merci à tous les développeurs qui ont permis cela. Dans les années 80, la plupart des utilisateurs de Linux d'aujourd'hui, seraient partis en courant loin d'Unix, rien qu'à cause du processus d'installation. Les distributions étaient fournies sur bandes magnétiques et la phase d'installation pouvait représenter une journée de travail, le processus de configuration réclamant au moins une journée supplémentaire. Ne parlons pas de la configuration des applications ! Et aujourd'hui, vous pouvez utiliser Unix sans même le savoir, avec des systèmes comme Mac OS X. C'était déjà vrai au début des années 90 grâce au "père" d'OS X, NeXTstep.

Cela peut sembler hors-sujet, pourtant NeXTstep a ouvert la voie aux Unixes libres et propriétaires... et à quelques autres ! Amener Unix chez le particulier était un véritable défi, et enfin, c'est arrivé.

Félicitations à tous ceux qui se sont investis dans ce défi, que ce soit moyennant finances ou gratuitement. Ceux travaillant gratuitement méritent une mention spéciale et encore plus de "merci ;-)

Le fait que FreeBSD "partage" avec Apple me semble une excellente chose. Chacun profite de l'autre dans le bon sens du terme. Les systèmes propriétaires ont toujours utilisé le logiciel libre... sans le "crier" trop fort. Apple et SGI n'hésitent pourtant pas à le dire. D'autres semblent faire de même... mais avec une arrière-pensée. Malgré tout, je continue à penser qu'il y a de la place pour tout le monde... enfin, sauf pour un qui prend déjà toute la place, et là, ça n'a plus rien à voir avec Unix !

Le "bruit" fait autour de Linux maintient les autres Unixes libres dans l'ombre : jusque là, ça va. Il peut y avoir un côté positif... et c'est la raison principale pour laquelle j'utilise FreeBSD. Bien sûr, je continue à utiliser Linux (et beaucoup d'autres) : j'adore la diversité !

Si cet article vous donne envie de tester FreeBSD, il n'aura pas été inutile. Essayez-le pour découvrir tout ce qui n'a pas été abordé ici.

Quand je vous dis qu'on vit une époque formidable !

Site Web maintenu par l'équipe d'édition

LinuxFocus

© Georges Tarbouriech

"some rights reserved" see linuxfocus.org/license/

<http://www.LinuxFocus.org>

Translation information:

en --> -- : Georges Tarbouriech <georges.t@linuxfocus.org>

en --> fr: Georges Tarbouriech
<georges.t@linuxfocus.org>