



The BlackArch Linux Guide

[https ://www.blackarch.org/](https://www.blackarch.org/)

Table des matières

1	Introduction	3
1.1	Vue d'ensemble	3
1.2	Qu'est-ce que BlackArch Linux ?	3
1.3	Histoire de BlackArch Linux	3
1.4	Plateformes supportées	3
1.5	S'impliquer	4
2	Guide d'utilisation	5
2.1	Installation	5
2.1.1	Installer le dépôt	5
2.1.2	Installation de paquets	5
2.1.3	Installation des paquets depuis le code source	6
2.1.4	Utilisation de Blackman	6
2.1.5	Installation d'un LiveCD, netinstall, ou d'Arch Linux	6
3	Guide de développement	8
3.1	Le Arch Build System et les dépôts de paquets	8
3.2	Le standard de paquetage de BlackArch	8
3.2.1	Groupe	8
3.2.1.1	blackarch	8
3.2.1.2	blackarch-anti-forensic	9
3.2.1.3	blackarch-automation	9
3.2.1.4	blackarch-backdoor	9
3.2.1.5	blackarch-binary	9
3.2.1.6	blackarch-bluetooth	9
3.2.1.7	blackarch-code-audit	9
3.2.1.8	blackarch-cracker	9
3.2.1.9	blackarch-crypto	10
3.2.1.10	blackarch-database	10
3.2.1.11	blackarch-debugger	10
3.2.1.12	blackarch-decompiler	10
3.2.1.13	blackarch-defensive	10
3.2.1.14	blackarch-disassembler	10
3.2.1.15	blackarch-dos	10
3.2.1.16	blackarch-drone	11
3.2.1.17	blackarch-exploitation	11
3.2.1.18	blackarch-fingerprint	11
3.2.1.19	blackarch-firmware	11
3.2.1.20	blackarch-forensic	11
3.2.1.21	blackarch-fuzzer	11

3.2.1.22	blackarch-hardware	11
3.2.1.23	blackarch-honeypot	11
3.2.1.24	blackarch-keylogger	12
3.2.1.25	blackarch-malware	12
3.2.1.26	blackarch-misc	12
3.2.1.27	blackarch-mobile	12
3.2.1.28	blackarch-networking	12
3.2.1.29	blackarch-nfc	12
3.2.1.30	blackarch-packer	12
3.2.1.31	blackarch-proxy	13
3.2.1.32	blackarch-recon	13
3.2.1.33	blackarch-reversing	13
3.2.1.34	blackarch-scanner	13
3.2.1.35	blackarch-sniffer	13
3.2.1.36	blackarch-social	13
3.2.1.37	blackarch-spoof	13
3.2.1.38	blackarch-threat-model	14
3.2.1.39	blackarch-tunnel	14
3.2.1.40	blackarch-unpacker	14
3.2.1.41	blackarch-voip	14
3.2.1.42	blackarch-webapp	14
3.2.1.43	blackarch-windows	14
3.2.1.44	blackarch-wireless	14
3.3	Structure du dépôt	15
3.3.1	Scripts	15
3.4	Contribuer au dépôt	16
3.4.1	Tutoriels requis.	16
3.4.2	étapes à suivre afin de contribuer	16
3.4.3	Exemple	16
3.4.3.1	Obtenir le PKGBUILD	16
3.4.3.2	Nettoisement du PKGBUILD	17
3.4.3.3	Ajustement du PKGBUILD	17
3.4.3.4	Construire le paquet	17
3.4.3.5	Installation et test du paquet	17
3.4.3.6	Ajout, commit et push du paquet	18
3.4.3.7	Création d'un pull request	18
3.4.3.8	Ajout d'un remote pour l'upstream	18
3.4.4	Requêtes	18
3.4.5	Conseils généraux	18

4 Guide outils

19

Chapitre 1

Introduction

1.1 Vue d'ensemble

Ce guide est divisé en quatre parties majeures, soient :

- Introduction - Offre une vue d'ensemble et quelques informations relatives au projet
- Guide utilisateur - Tous les éléments nécessaires à la bonne utilisation de BlackArch
- Guide développeur - Comment bien débiter afin de contribuer au projet BlackArch
- Guide outils - Description détaillée des outils ainsi que des exemples d'utilisation (En construction)

1.2 Qu'est-ce que BlackArch Linux ?

BlackArch Linux est une distribution GNU/Linux basée sur Arch Linux destinée aux professionnels et chercheurs en sécurité informatique.

L'ensemble d'outils est distribué tel un **dépôt non-officiel** d'Arch Linux, vous permettant d'installer les composantes de BlackArch sur une installation d'Arch Linux existante. Les paquets peuvent y être installés individuellement ou par catégorie.

Le dépôt logiciel de Black Arch contient plus de 1300 outils et ce nombre augmente sans cesse. Chacun de ces outils est minutieusement testé avant d'être ajouté aux dépôts, afin d'assurer leur bon fonctionnement.

1.3 Histoire de BlackArch Linux

En construction...

1.4 Plateformes supportées

En construction...



1.5 S'impliquer

Il est possible de contacter l'équipe Black Arch via :

Website : <https://www.blackarch.org/>

Mail : team@blackarch.org

IRC : <irc://irc.freenode.net/blackarch>

Twitter : <https://twitter.com/blackarchlinux>

Github : <https://github.com/Blackarch/>

Chapitre 2

Guide d'utilisation

2.1 Installation

La section suivante illustre comment configurer le dépôt de paquets de Black Arch ainsi que la procédure d'installation de paquets. BlackArch supporte l'installation de fichiers binaires précompilés ainsi que l'installation à l'aide du code source original des paquets.

BlackArch est compatible avec une installation de base d'Arch Linux. Il fait office de dépôt non-officiel. Si vous désirez utiliser une image ISO à la place, visitez la section [Live ISO](#).

2.1.1 Installer le dépôt

Exécuter **strap.sh** en tant que root et suivez les instructions suivantes :

```
curl -O https://blackarch.org/strap.sh
sha1sum strap.sh # doit etre egal a 86eb4efb68918dbfdd1e22862a48fda20a8145ff
sudo ./strap.sh
```

Mettez ensuite à jour la liste des paquets en exécutant :

```
sudo pacman -Syyu
```

2.1.2 Installation de paquets

Vous pouvez désormais installer des outils directement du dépôt de BlackArch

1. Afin d'obtenir la liste de tous les outils disponibles exécutez

```
pacman -Sgg | grep blackarch | cut -d' ' -f2 | sort -u
```

2. Afin d'installer tous les outils, exécutez

```
pacman -S blackarch
```

3. Afin d'installer une catégorie complète d'outils, exécutez

```
pacman -S blackarch-<category>
```

4. Afin d'obtenir la liste des catégories de paquets disponibles, exécutez

```
pacman -Sg | grep blackarch
```



2.1.3 Installation des paquets depuis le code source

Il existe une méthode alternative à l'installation de paquets sous forme binaire. BlackArch vous offre la possibilité d'obtenir le code source directement du dépôt officiel du paquet désiré, et BlackArch le compilera pour vous. Vous trouverez les fichiers PKGBUILD sur [github](#). Afin d'obtenir le dépôt en entier, vous pourrez utiliser l'outil **blackman**.

- En premier lieu, vous devez installer blackman. Si vous avez déjà configuré le dépôt de paquets de BlackArch sur votre système, tel qu'indiqué au début de ce document, vous vous pouvez installer blackman à l'aide de pacman en utilisant :

```
pacman -S blackman
```

- Vous pouvez installer blackman depuis les sources :

```
mkdir blackman
cd blackman
wget https://raw2.github.com/BlackArch/blackarch/master/packages/blackman/PKGBUILD
# S'assurer de l'intégrité du fichier PKGBUILD avec:
makepkg -s
```

- Vous pouvez également installer blackman depuis les dépôts AUR

```
<votre assistant AUR> -S blackman
```

2.1.4 Utilisation de Blackman

Blackman est un outil simple à utiliser. Cependant, les paramètres peuvent être différents de ce qu'offrent des gestionnaires de paquets tel que pacman. Les exemples ci-dessous illustrent les cas d'utilisation de base.

- Télécharger, compiler et installer un paquet :

```
sudo blackman -i paquet
```

- Télécharger, compiler et installer un groupe entier :

```
sudo blackman -g groupe
```

- Télécharger, compiler et installer tous les outils BlackArch :

```
sudo blackman -a
```

- Lister les catégories disponibles :

```
blackman -l
```

- Lister les outils d'une catégorie :

```
blackman -p category
```

2.1.5 Installation d'un LiveCD, netinstall, ou d'Arch Linux

Vous pouvez installer BlackArch depuis l'une des images ISO fournies (live ou netinstall. Voir <https://www.blackarch.org/download.html#iso>). Voici les étapes à réaliser une fois le système démarré depuis une ISO.



— Installer le paquet blackarch-installer :

```
sudo pacman -S blackarch-installer
```

— Exécuter :

```
sudo blackarch-install
```


Chapitre 3

Guide de développement

3.1 Le Arch Build System et les dépôts de paquets

Les fichiers PKGBUILD sont de simples scripts permettant d'installer et de compiler et installer un programme sur plusieurs environnements différents. Chaque PKGBUILD indique à la commande `makepkg(1)` comment créer un paquet valide qui pourra ensuite être installé. Les fichiers PKGBUILD sont écrits en bash.

Pour plus d'information, il est suggéré de lire ces pages :

- [Arch Wiki FR : Standard paquetage](#)
- [Arch Wiki FR : Makepkg](#)
- [Arch Wiki FR : PKGBUILD](#)

3.2 Le standard de paquetage de BlackArch

Par souci de simplicité, nos PKGBUILDs sont similaires à ceux du Arch User Repository, à quelques différences près. Tous les paquets doivent être membre du groupe *blackarch*. Il est possible qu'un paquet soit membre de plusieurs groupes.

3.2.1 Groupes

Afin de permettre aux utilisateurs d'installer certaines catégories de paquets , ceux-ci ont été répertoriés en plusieurs groupes. Ces derniers permettent aux utilisateurs d'exécuter la commande `pacman -S <nom_du_groupe>` afin d'installer tous les paquets compris dans ce groupe.

3.2.1.1 blackarch

En théorie, le groupe *blackarch* doit contenir tous les paquets qu'offre BlackArch. Tous les paquets doivent être membre de ce groupe, comme expliqué précédemment.



3.2.1.2 blackarch-anti-forensic

Groupe contenant les paquets utilisés pour contrer les tentatives d'investigation informatique de systèmes. Vous y trouverez des paquets offrant des solutions de chiffrement de données, stéganographie, modification d'attributs de fichiers, et tout autre type de logiciels permettant la modification d'un système afin de masquer de l'information.

Exemples : luks, TrueCrypt, Timestomp, dd, ropeadope, secure-delete

3.2.1.3 blackarch-automation

Groupe contenant les paquets utilisés pour l'automation de plusieurs tâches. Puisque les paquets présents dans ce groupe varient énormément en fonctionnalités, il vous est suggéré d'y consulter directement les paquets afin de vous informer sur leurs capacités.

Exemples : blueranger, tiger, wiffy

3.2.1.4 blackarch-backdoor

Groupe contenant les paquets reliés à l'exploitation ou à l'ouverture de backdoors sur des systèmes vulnérables.

Exemples : backdoor-factory, rrs, weevely

3.2.1.5 blackarch-binary

Groupe contenant les paquets reliés aux fichiers binaires. Vous y trouverez entre autres des paquets permettant l'extraction d'archives d'un firmware, des logiciels de rétro-ingénierie, logiciels aidant à l'exploitation de fichiers binaires.

Exemples : binwally, packerid, hex2bin, binwalk

3.2.1.6 blackarch-bluetooth

Groupe contenant les paquets relatifs à l'exploitation du standard Bluetooth (802.15.1).

Exemples : ubertooth, tbear, redfang

3.2.1.7 blackarch-code-audit

Groupe contenant les paquets à fin d'audit de code source. Ces paquets analysent statiquement le code source d'un projet afin d'y déceler des vulnérabilités.

Exemples : flawfinder, pscan

3.2.1.8 blackarch-cracker

Groupe contenant les paquets utilisés afin de tester la robustesse de multiples fonctions cryptographiques.

Exemples : hashcat, john, crunch



3.2.1.9 blackarch-crypto

Groupe contenant tous types de paquets relatifs à la cryptographie.

Exemples : ciphertest, xortool, sbd

3.2.1.10 blackarch-database

Groupe contenant les paquets relatifs à l'exploitation de bases de données.

Exemples : metacoretex, blindsql

3.2.1.11 blackarch-debugger

Groupe contenant les paquets permettant à l'utilisateur d'investiguer le fonctionnement interne d'un programme lors de son utilisation.

Exemples : radare2, shellnoob

3.2.1.12 blackarch-decompiler

Groupe contenant les paquets qui tentent d'effectuer de la rétro-ingénierie sur un programme compilé.

Exemples : flasm, jd-gui

3.2.1.13 blackarch-defensive

Groupe contenant les paquets utilisés afin de protéger un système des logiciels (malwares) et utilisateurs malveillants.

Exemples : arpon, chkrootkit, sniffjoke

3.2.1.14 blackarch-disassembler

Ce groupe est très semblable à *blackarch-decompiler* dans le sens que ces deux groupes tentent d'effectuer une rétro-ingénierie générique de fichiers exécutables, mais les paquets compris dans *blackarch-disassembler* en extrairont le code assembleur et non un code source sous forme de pseudocode par exemple.

Exemples : inguma, radare2

3.2.1.15 blackarch-dos

Groupe contenant les paquets relatifs aux attaques de type *déni de service*.

Exemples : 42zip, nkiller2



3.2.1.16 blackarch-drone

Groupe contenant les paquets aidant à l'entretien et l'utilisation de drones.

Exemples : meshdeck, skyjack

3.2.1.17 blackarch-exploitation

Groupe contenant les paquets utilisés lors de l'exploitation de multiples programmes et services.

Exemples : armitage, metasploit, zarp

3.2.1.18 blackarch-fingerprint

Groupe contenant les paquets utilisés à des fins de reconnaissance et prise d'empreinte.

Exemples : dns-map, p0f, httpprint

3.2.1.19 blackarch-firmware

Groupe contenant les paquets utilisés lors de l'exploitation de firmware.

Exemples : None yet, amend asap.

3.2.1.20 blackarch-forensic

Groupe contenant les paquets utilisés lors d'investigation informatique portant sur le recouvrement de données potentiellement effacés.

Exemples : aesfix, nfex, wyd

3.2.1.21 blackarch-fuzzer

Groupe contenant les paquets permettant de fuzzer des logiciels.

Exemples : msf, mdk3, wfuzz

3.2.1.22 blackarch-hardware

Groupe contenant les paquets relatifs à l'exploitation de matériel physique.

Exemples : arduino, smali

3.2.1.23 blackarch-honeypot

Groupe contenant les paquets agissant comme des *honeypots*, c'est-à-dire des services sécuritaires, mais se faisant passer pour vulnérables afin d'observer son comportement.

Exemples : artillery, bluepot, wifi-honey



3.2.1.24 blackarch-keylogger

Groupe contenant les paquets utilisés lors d'installation de *keylogging* , soit l'enregistrement de touches clavier sur un système qui pourrait être utilisé par un utilisateur tiers afin d'en soutirer des informations confidentielles.

Exemples : klogger, logkeys, xspy

3.2.1.25 blackarch-malware

Groupe contenant tous types de paquets ayant un lien au *malware*, que ce soit à l'utilisation de ceux-ci ou à leur détection.

Exemples : malwaredetect, peepdf, yara

3.2.1.26 blackarch-misc

Groupe contenant les paquets ne semblant pas prendre part dans aucunes autres catégories.

Exemples : oh-my-zsh-git, winexe, stompy

3.2.1.27 blackarch-mobile

Groupe contenant les paquets relatifs aux activités d'entretien et d'audits d'applications, de développement d'application, et de matériel mobile.

Exemples : android-sdk-platform-tools, android-udev-rules

3.2.1.28 blackarch-networking

Groupe contenant les paquets relatifs au réseau.

Exemples : Anything pretty much

3.2.1.29 blackarch-nfc

Groupe contenant les paquets utilisant la technologie *NFC* (near-field communications).

Exemples : nfcutils

3.2.1.30 blackarch-packer

Groupe contenant les paquets relatifs aux *packers*, soit des logiciels qui créent des applications embarquant un programme malveillant.

Exemples : packerid



3.2.1.31 blackarch-proxy

Groupe contenant les paquets agissant tel un proxy, redirigeant le trafic internet afin de l'analyser ou de le modifier.

Exemples : burpsuite, ratproxy, sslnuke

3.2.1.32 blackarch-recon

Groupe contenant les paquets servant à la recherche de services exploitable.

Exemples : canri, dnsrecon, netmask

3.2.1.33 blackarch-reversing

Groupe contenant les paquets de décompilation, de désassemblage, et tout autres types de programmes relatifs à la rétro-ingénierie.

Exemples : capstone, radare2, zerowine

3.2.1.34 blackarch-scanner

Groupe contenant les paquets utilisés afin de scanner multiples systèmes permettant d'y déceler des vulnérabilités actives.

Exemples : scanssh, tiger, zmap

3.2.1.35 blackarch-sniffer

Groupes contenant les paquets aidant à l'analyse de trafic réseau.

Exemples : hexinject, pytactile, xspy

3.2.1.36 blackarch-social

Groupe contenant les paquets relatifs aux attaques sur les sites de réseaux sociaux.

Exemples : jigsaw, websploit

3.2.1.37 blackarch-spoof

Groupe contenant les paquets qui tentent de dissimuler un attaquant, de tel que la victime ne puisse le percevoir comme un acteur malicieux.

Exemples : arpoison, lans, netcommander



3.2.1.38 blackarch-threat-model

Groupe contenant les paquets utilisés pour enregistrer et créer des rapports sur plusieurs modèles de menaces pouvant se produire lors d'un scénario spécifique.

Exemples : magictree

3.2.1.39 blackarch-tunnel

Groupe contenant les paquets utilisés afin d'encapsuler le contenu réseau via un tiers parti, afin de dissimuler la provenance initiale de données.

Exemples : ctunnel, iodine, ptunnel

3.2.1.40 blackarch-unpacker

Groupe contenant les paquets utilisés lors de l'extraction de logiciel malveillants de logiciels *packés* ou obfusqués.

Exemples : js-beautify

3.2.1.41 blackarch-voip

Groupe contenant les paquets permettant de mener des opérations sur les protocoles et programmes VoIP.

Exemples : iaxflood, rtp-flood, teardown

3.2.1.42 blackarch-webapp

Groupe contenant les paquets permettant de tester les applications web.

Exemples : metoscan, whatweb, zaproxy

3.2.1.43 blackarch-windows

Groupe contenant les paquets Windows s'exécutant via Wine.

Exemples : 3proxy-win32, pwdump, winexe

3.2.1.44 blackarch-wireless

Groupe contenant les paquets utilisés lors d'audits de sécurité sur une infrastructure sans-fil.

Exemples : airpwn, mdk3, wiffy



3.3 Structure du dépôt

Vous pouvez trouver le dépôt principal de BlackArch à <https://github.com/BlackArch/blackarch>. Il existe aussi plusieurs dépôts de code à <https://github.com/BlackArch>.

à l'intérieur du dépôt principal, vous trouverez les trois dossiers principaux, tels que :

- docs - Contient la documentation du projet
- packages - Contient les fichiers PKGBUILD
- scripts - Contient de multiples scripts d'automation

3.3.1 Scripts

Voici une référence des scripts que vous trouverez dans le dossier scripts/ :

- baaur - Bientôt, ce script enverra nos paquets au dépôt AUR.
- babuild - Construit un paquet.
- bachroot - Prépare un chroot afin d'effectuer des tests.
- baclean - Nettoie les anciens fichiers .pkg.tar.xz du dépôt.
- baconflict - Bientôt, ce script remplacera scripts/conflicts.
- bad-files - Trouve des fichiers erronés dans les paquets construits.
- balock - Crée ou termine le lock sur le dépôt de paquets.
- banotify - Indique IRC des push de paquets.
- barelease - Envoie les paquets au dépôt.
- baright - Affiche les informations de copyright de BlackArch.
- basign - Signe les paquets.
- basign-key - Signe une clé.
- blackman - Opère sensiblement comme pacman mais construit les paquets à partir de git. (à ne pas confondre avec le programme Blackman, écrit par nzt)
- check-groups - Valide les groupes.
- checkpkgs - Valide les paquets.
- conflicts - Valide les paquets pour des conflits.
- dbmod - Modifie la base de donnée d'un paquet.
- depth-list - Crée une liste de dépendances ordonnée par profondeur.
- deptree - Crée une arbre de dépendance, ne listant que les paquets obtenus via BlackArch.
- get-blackarch-deps - Affiche une liste des dépendances de paquets obtenus via BlackArch pour un paquet donné.
- get-official - Obtient les paquets officiels.
- list-loose-packages - Liste les paquets qui ne se trouvent dans aucun groupe et ne sont pas des dépendances pour d'autres paquets.
- list-needed - Liste les dépendances manquantes.
- list-removed - Liste les paquets qui sont dans le dépôt mais pas dans le répertoire git.
- list-tools - Liste les outils.
- outdated - Cherche les paquets qui ne sont plus à jour lorsque comparés avec le répertoire git.
- pkgmod - Modifie un paquet de construction.
- pkgrel - Incrémente le pkgrel d'un paquet.
- prep - Nettoie un fichier PKGBUILD de ses erreurs de style et autre.
- sitesync - Synchronise entre une copie locale et sa version distante.
- size-hunt - Recherche de gros paquets.
- source-backup - Sauvegarde les fichiers sources d'un paquet.



3.4 Contribuer au dépôt

Cette section indique comment contribuer au projet BlackArch Linux. Nous acceptons les pull requests de toutes tailles, que ce soit de petites modifications ou un ajout de paquet. Pour recevoir de l'aide, n'hésitez pas à nous contacter.

Tout le monde est bienvenu à contribuer. Toute contribution est appréciée.

3.4.1 Tutoriels requis.

Veuillez lire les documents suivants avant de contribuer :

- [Arch Wiki FR : Standard paquetage](#)
- [Arch Wiki FR : Makepkg](#)
- [Arch Wiki FR : PKGBUILD](#)

3.4.2 étapes à suivre afin de contribuer

Afin de contribuer au projet BlackArch Linux, suivez ces étapes :

1. Forkez le dépôt sur github à partir de <https://github.com/BlackArchLinux/blackarchlinux>.
2. Effectuez vos modifications sur les fichiers nécessaires (e.g. PKGBUILD, .patch files, etc).
3. Commitez vos changements.
4. Poussez vos changements.
5. Créez un pull request vers le dépôt.

3.4.3 Exemple

L'exemple suivant démontre la soumissions d'un nouveau paquets au projet BlackArch. Nous utilisons **yaourt** (vous pouvez aussi utiliser **pacaur**) afin d'obtenir le PKGBUILD existant de **nfsshell**, et nous pouvons ensuite l'éditer, au besoin.

3.4.3.1 Obtenir le PKGBUILD

En utilisant yaourt ou pacaur, obtenir le fichier *PKGBUILD* :

```
user@blackarchlinux $ yaourt -G nfsshell
==> Download nfsshell sources
x LICENSE
x PKGBUILD
x gcc.patch
user@blackarchlinux $ cd nfsshell/
```



3.4.3.2 Nettoyement du PKGBUILD

Nettoyer le fichier *PKGBUILD* à l'aide de *prep*, afin de sauver du temps :

```
user@blackarchlinux nfsshell $ ./blackarch/scripts/prep PKGBUILD
cleaning 'PKGBUILD'...
expanding tabs...
removing vim modeline...
removing id comment...
removing contributor and maintainer comments...
squeezing extra blank lines...
removing '|| return'...
removing leading blank line...
removing $pkgname...
removing trailing whitespace...
```

3.4.3.3 Ajustement du PKGBUILD

Ajustez le fichier *PKGBUILD* :

```
user@blackarchlinux nfsshell $ vi PKGBUILD
```

3.4.3.4 Construire le paquet

Construire le paquet :

```
==> Making package: nfsshell 19980519-1 (Mon Dec  2 17:23:51 CET 2013)
==> Checking runtime dependencies...
==> Checking buildtime dependencies...
==> Retrieving sources...
-> Downloading nfsshell.tar.gz...
% Total      % Received % Xferd  Average Speed   Time    Time     Time
CurrentDload  Upload    Total   Spent    Left  Speed100 29213  100 29213    0
0 48150      0 --:--:-- --:--:-- --:--:-- 48206
-> Found gcc.patch
-> Found LICENSE
...
<lots of build process and compiler output here>
...
==> Leaving fakeroot environment.
==> Finished making: nfsshell 19980519-1 (Mon Dec  2 17:23:53 CET 2013)
```

3.4.3.5 Installation et test du paquet

Installation et test du paquet :

```
user@blackarchlinux nfsshell $ pacman -U nfsshell-19980519-1-x86_64.pkg.tar.xz
user@blackarchlinux nfsshell $ nfsshell # test it
```



3.4.3.6 Ajout, commit et push du paquet

Ajoute, commit et push du paquet :

```
user@blackarchlinux ~/blackarchlinux/packages $ mv ~/nfsshell .
user@blackarchlinux ~/blackarchlinux/packages $ git commit -am nfsshell && git push
```

3.4.3.7 Création d'un pull request

Créez un pull request de votre fork vers [le projet officiel](#) :

```
firefox https://github.com/<contributor>/blackarchlinux
```

3.4.3.8 Ajout d'un remote pour l'upstream

Une fois votre modification faite, il faut ajouter le projet officiel comme remote, afin d'ajouter les modifications au projet à votre fork. Cela vous permettra de synchroniser votre fork avec BlackArch.

```
user@blackarchlinux ~/blackarchlinux $ git remote -v
origin <the url of your fork> (fetch)
origin <the url of your fork> (push)
user@blackarchlinux ~/blackarchlinux $ git remote add upstream https://github.com/blackarch/blackarch
user@blackarchlinux ~/blackarchlinux $ git remote -v
origin <the url of your fork> (fetch)
origin <the url of your fork> (push)
upstream https://github.com/blackarch/blackarch (fetch)
upstream https://github.com/blackarch/blackarch (push)
```

Par défaut, git devrait push directement à l'origine, mais assurez-vous que votre configuration git est correcte. Cela ne sera pas un problème à moins d'avoir les droits de commit puisque vous ne pourrez pas push upstream sans cela.

Si vous avez les droits de commit, vous aurez probablement plus de succès en utilisant `git@github.com :blackarch/blackarch.git`. A vous de voir.

3.4.4 Requêtes

1. N'ajoutez pas de commentaire **Maintainer** ou **Contributor** au fichiers *PKGBUILD*. Ajoutez ces commentaires à la section **AUTHORS** du guide BlackArch.
2. Afin de conserver la consistance du projet, veuillez suivre les règles générales de style des autres fichiers *PKGBUILD* en utilisant une indentation de deux espaces.

3.4.5 Conseils généraux

namcap peut trouver des erreurs dans vos paquets. Il est recommandé de l'utiliser avant de soumettre vos modifications.

Chapitre 4

Guide outils

En construction...