



Guida a BlackArch Linux

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

Contents

1	Introduzione	3
1.1	Panoramica	3
1.2	Che Cosa è BlackArch Linux?	3
1.3	Storia di BlackArch Linux	3
1.4	Piattaforme Supportate	3
1.5	Partecipa	4
2	Guida dell'utente	5
2.1	Installazione	5
2.1.1	Installazione su ArchLinux	5
2.1.2	Installare i Pacchetti	5
2.1.3	Installare i Pacchetti da Sorgente	6
2.1.4	Utilizzo Base di Blackman	6
2.1.5	Installazione da live-, netinstall- ISO or ArchLinux	7
3	Guida Sviluppatore	8
3.1	Build System e Repositories di Arch	8
3.2	Blackarch PKGBUILD standards	8
3.2.1	Gruppi	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	9
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	10
3.2.1.17	blackarch-exploitation	10
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	11
3.2.1.25	blackarch-malware	11
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	12
3.2.1.32	blackarch-recon	12
3.2.1.33	blackarch-reversing	12
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	13
3.2.1.39	blackarch-tunnel	13
3.2.1.40	blackarch-unpacker	13
3.2.1.41	blackarch-voip	13
3.2.1.42	blackarch-webapp	14
3.2.1.43	blackarch-windows	14
3.2.1.44	blackarch-wireless	14
3.3	Struttura del Repository	14
3.3.1	Scripts	14
3.4	Contribuire ad un Repository	15
3.4.1	Tutorials Richiesti	16
3.4.2	Passaggi per Contribuire	16
3.4.3	Esempio	16
3.4.3.1	Ottenere il PKGBUILD	16
3.4.3.2	Ripulire il PKGBUILD	16
3.4.3.3	Adjust PKGBUILD	17
3.4.3.4	Genera il Pacchetto	17
3.4.3.5	Installa e testa il pacchetto	17
3.4.3.6	Aggiungi, esegui il commit e il push del pacchetto	17
3.4.3.7	Crea una pull request	18
3.4.3.8	Aggiungere un Remoto per Upstream	18
3.4.4	Richieste	18
3.4.5	Consigli Generali	18
4	Guida Strumenti	19
4.1	Coming Soon	19

Chapter 1

Introduzione

1.1 Panoramica

La guida a BlackArch Linux è divisa in alcune parti:

- Introduzione - Fornisce un'ampia panoramica, presentazione ed altre informazioni utili sul progetto
- Guida utente - Tutto ciò di cui un utente tipico necessita per utilizzare al meglio BlackArch
- Guida sviluppatore - Come iniziare a sviluppare e contribuire a BlackArch
- Guida strumenti - Dettagli approfonditi con esempi di utilizzo degli strumenti

1.2 Che Cosa è BlackArch Linux?

BlackArch è una distribuzione Linux completa per pentesters e ricercatori di sicurezza. Deriva da [ArchLinux](#) e gli utenti possono installare i componenti di BlackArch individualmente o in gruppo direttamente su di essa.

L'insieme degli strumenti è distribuito come un [repository utente non ufficiale](#) per ArchLinux, quindi puoi installare BlackArch sopra ad un'installazione di ArchLinux già esistente. I pacchetti possono essere installati individualmente o per categoria.

Il repository in costante espansione include al momento oltre [1300](#) strumenti. Tutti gli strumenti vengono testati a fondo prima di essere aggiunti, in maniera da mantenere la qualità del repository.

1.3 Storia di BlackArch Linux

Prossimamente...

1.4 Piattaforme Supportate

Prossimamente...



1.5 Partecipa

Puoi metterti in contatto con il team di BlackArch attraverso i seguenti metodi:

Sito web: <https://www.blackarch.org/>

Mail: team@blackarch.org

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

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

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

Chapter 2

Guida dell'utente

2.1 Installazione

Le sezioni seguenti ti mostreranno come impostare il repository BlackArch ed installare i pacchetti. BlackArch supporta l'installazione sia dal repository utilizzando i pacchetti binari, sia compilando ed installando i sorgenti.

BlackArch è compatibile con le normali installazioni Arch. Si comporta come un repository utente non ufficiale. Se invece vuoi un'ISO, guarda la sezione [Live ISO](#).

2.1.1 Installazione su ArchLinux

Esegui `strap.sh` come root e segui le istruzioni. Vedi l'esempio seguente.

```
curl -O https://blackarch.org/strap.sh
sha1sum strap.sh # should match: 86eb4efb68918dbfdd1e22862a48fda20a8145ff
sudo ./strap.sh
```

Ora scarica una copia aggiornata della lista principale dei pacchetti e sincronizza i pacchetti:

```
sudo pacman -Syyu
```

2.1.2 Installare i Pacchetti

Ora puoi installare gli strumenti dal repository blackarch.

1. Per elencare tutti gli strumenti disponibili, esegui

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

2. Per installare tutti gli strumenti, esegui

```
pacman -S blackarch
```

3. Per installare una categoria di strumenti, esegui



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

4. Per elencare le categorie di blackarch, esegui

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

2.1.3 Installare i Pacchetti da Sorgente

Come parte di un metodo alternativo di installazione, puoi generare i pacchetti BlackArch da sorgente. Puoi trovare i PKGBUILDs su [github](#). Per generare l'intero repo, puoi utilizzare lo strumento [Blackman](#).

- Come prima cosa, devi installare Blackman. Se il repository BlackArch è impostato sulla tua macchina, puoi installare Blackman:

```
pacman -S blackman
```

- Puoi compilare ed installare Blackman dal sorgente:

```
mkdir blackman
cd blackman
wget https://raw2.github.com/BlackArch/blackarch/master/packages/blackman/PKGBUILD
# Make sure the PKGBUILD has not been maliciously tampered with.
makepkg -s
```

- Oppure puoi installare Blackman da AUR:

```
<whatever AUR helper you use> -S blackman
```

2.1.4 Utilizzo Base di Blackman

Blackman è molto semplice da utilizzare, sebbene i flags siano differenti da quello che ti aspetteresti da qualcosa simile a pacman. L'utilizzo base viene descritto nel seguito.

- Scarica, compila e installa i pacchetti:

```
sudo blackman -i package
```

- Scarica, compila e installa un'intera categoria:

```
sudo blackman -g group
```

- Scarica, compila e installa tutti gli strumenti di BlackArch:

```
sudo blackman -a
```

- Per elencare le categorie di BlackArch:

```
blackman -l
```

- Per elencare gli strumenti di una categoria:

```
blackman -p category
```



2.1.5 Installazione da live-, netinstall- ISO or ArchLinux

Puoi installare BlackArch Linux da una delle nostre live- o netinstall-ISOs. Vedi <https://www.blackarch.org/download.html#iso>. Dopo l'avvio dell'ISO sono richiesti i seguenti passaggi.

- Installa il pacchetto blackarch-installer:

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

- Esegui

```
sudo blackarch-install
```


Chapter 3

Guida Sviluppatore

3.1 Build System e Repositories di Arch

I files PKGBUILD sono dei build scripts. Ognuno dice a `makepkg(1)` come creare un pacchetto. I files PKGBUILD sono scritti in Bash.

Per altre informazioni, leggi (o sfoglia) le seguenti pagine:

- [Arch Wiki: Creating Packages](#)
- [Arch Wiki: makepkg](#)
- [Arch Wiki: PKGBUILD](#)
- [Arch Wiki: Arch Packaging Standards](#)

3.2 Blackarch PKGBUILD standards

Per semplicità, i nostri PKGBUILDs sono simili a quelli di AUR, con qualche piccola differenza evidenziata di seguito. Ogni pacchetto deve appartenere almeno a blackarch, ci saranno anche molte sovrapposizioni tra vari pacchetti appartenenti a più gruppi.

3.2.1 Gruppi

Per permettere agli utenti di installare velocemente e facilmente uno specifico insieme di pacchetti, questi sono stati separati in gruppi. I gruppi permettono agli utenti di dare semplicemente `"pacman -S <group name>"` per installare tanti pacchetti.

3.2.1.1 blackarch

Il gruppo blackarch è il gruppo base a cui tutti i pacchetti devono appartenere. Questo consente agli utenti di installare tutti i pacchetti con facilità.

Cosa dovrebbe stare qui: tutto.



3.2.1.2 **blackarch-anti-forensic**

Pacchetti che sono usati per contrastare le attività forensi, includono crittografia, steganografia, e qualsiasi cosa che modifichi i files o i loro attributi. Questo vale per tutti gli strumenti che lavorano con qualsiasi cosa in generale che modifichi un sistema per nascondere informazioni.

Ad esempio: luks, TrueCrypt, Timestomp, dd, ropeadope, secure-delete

3.2.1.3 **blackarch-automation**

Pacchetti che sono usati per l'automazione del flusso di lavoro.

Ad esempio: blueranger, tiger, wiffy

3.2.1.4 **blackarch-backdoor**

Pacchetti che sfruttano o aprono backdoors su sistemi già vulnerabili.

Ad esempio: backdoor-factory, rrs, weevely

3.2.1.5 **blackarch-binary**

Pacchetti che in qualche modo operano su file binari.

Ad esempio: binwally, packerid

3.2.1.6 **blackarch-bluetooth**

Pacchetti che sfruttano qualsiasi cosa riguardi lo standard Bluetooth (802.15.1).

Ad esempio: ubertooth, tbear, redfang

3.2.1.7 **blackarch-code-audit**

Pacchetti che effettuano la revisione di codice sorgente per analisi di vulnerabilità.

Ad esempio: flawfinder, pscan

3.2.1.8 **blackarch-cracker**

Pacchetti che sono usati per il cracking di funzioni crittografiche, ie hashes.

Ad esempio: hashcat, john, crunch

3.2.1.9 **blackarch-crypto**

Pacchetti che lavorano con la crittografia, esclusi quelli per il cracking.

Ad esempio: ciphertest, xortool, sbd



3.2.1.10 blackarch-database

Pacchetti che comportano l'exploitation a qualsiasi livello di databases.

Ad esempio: metacoretex, blindsqli

3.2.1.11 blackarch-debugger

Pacchetti che permettono all'utente di vedere in realtime cosa sta facendo un particolare programma.

Ad esempio: radare2, shellnoob

3.2.1.12 blackarch-decompiler

Pacchetti che cercano di ricavare il codice sorgente da un programma compilato.

Ad esempio: flasm, jd-gui

3.2.1.13 blackarch-defensive

Pacchetti che sono utilizzati per proteggere un utente da attacchi di malware da parte di altri utenti.

Ad esempio: arpon, chkrootkit, sniffjoke

3.2.1.14 blackarch-disassembler

Questo è simile a blackarch-decompiler, e probabilmente ci saranno molti programmi che ricadranno in entrambe le categorie, tuttavia questi pacchetti producono l'output in assembly anziché in codice sorgente di alto livello.

Ad esempio: inguma, radare2

3.2.1.15 blackarch-dos

Pacchetti che utilizzano attacchi DoS (Denial of Service).

Ad esempio: 42zip, nkiller2

3.2.1.16 blackarch-drone

Pacchetti che sono utilizzati per gestire droni.

Ad esempio: meshdeck, skyjack

3.2.1.17 blackarch-exploitation

Pacchetti che approfittano di exploits in altri programmi o servizi.

Ad esempio: armitage, metasploit, zarp



3.2.1.18 blackarch-fingerprint

Pacchetti per fare fingerprinting.

Ad esempio: dns-map, p0f, httpprint

3.2.1.19 blackarch-firmware

Pacchetti che sfruttano vulnerabilità nel firmware.

Ad esempio: None yet, amend asap.

3.2.1.20 blackarch-forensic

Pacchetti che sono utilizzati per trovare dati su dischi fisici o memorie embedded.

Ad esempio: aesfix, nfex, wyd

3.2.1.21 blackarch-fuzzer

Pacchetti che utilizzano il principio di testing del fuzzing, ie inviano degli inputs casuali per vedere cosa succede.

Ad esempio: msf, mdk3, wfuzz

3.2.1.22 blackarch-hardware

Pacchetti che sfruttano o gestiscono qualsiasi cosa che abbia a che fare con l'hardware fisico.

Ad esempio: arduino, smali

3.2.1.23 blackarch-honeypot

Pacchetti che agiscono da "honeypots", ie programmi che sembrano servizi vulnerabili, utilizzati per trarre in trappola gli hackers.

Ad esempio: artillery, bluepot, wifi-honey

3.2.1.24 blackarch-keylogger

Pacchetti che registrano e conservano le battiture su un altro sistema.

Ad esempio: None yet, amend asap.

3.2.1.25 blackarch-malware

Pacchetti che contano come qualsiasi tipo di software malevolo o malware detection.

Ad esempio: malwaredetect, peepdf, yara



3.2.1.26 **blackarch-misc**

Pacchetti che non rientrano specificamente in nessuna categoria.

Ad esempio: oh-my-zsh-git, winexe, stompy

3.2.1.27 **blackarch-mobile**

Pacchetti che manipolano piattaforme mobile.

Ad esempio: android-sdk-platform-tools, android-udev-rules

3.2.1.28 **blackarch-networking**

Pacchetti che coinvolgono l'IP networking.

Ad esempio: più o meno tutto

3.2.1.29 **blackarch-nfc**

Pacchetti che usano nfc (near-field communications).

Ad esempio: nfcutils

3.2.1.30 **blackarch-packer**

Pacchetti che operano su o coinvolgono packers.

i packers sono programmi che incorporano del malware dentro altri eseguibili

Ad esempio: packerid

3.2.1.31 **blackarch-proxy**

Pacchetti che agiscono come proxy, ie redirigendo il traffico attraverso un altro nodo su internet

Ad esempio: burpsuite, ratproxy, sslnuke

3.2.1.32 **blackarch-recon**

Pacchetti che cercano attivamente vulnerabilità da exploitare in the wild. Più di un gruppo ombrello per pacchetti simili.

Ad esempio: canri, dnsrecon, netmask

3.2.1.33 **blackarch-reversing**

Questo è un gruppo ombrello per qualsiasi decompilatore, disassemblatore o ogni programma simile.

Ad esempio: capstone, radare2, zerowine



3.2.1.34 blackarch-scanner

Pacchetti che analizzano alla ricerca di vulnerabilità i sistemi selezionati.

Ad esempio: scanssh, tiger, zmap

3.2.1.35 blackarch-sniffer

Pacchetti che coinvolgono l'analisi del traffico di rete.

Ad esempio: hexinject, pytactle, xspy

3.2.1.36 blackarch-social

Pacchetti che attaccano principalmente siti di social networking.

Ad esempio: jigsaw, websploit

3.2.1.37 blackarch-spoof

Pacchetti che provano a falsificare l'attaccante, in modo da non apparire come tale alla vittima.

Ad esempio: arpoison, lans, netcommander

3.2.1.38 blackarch-threat-model

Pacchetti che vengono usati per scrivere report sul threat model delineatosi in un particolare scenario.

Ad esempio: magictree

3.2.1.39 blackarch-tunnel

Pacchetti che sono utilizzati per fare tunneling del traffico di rete.

Ad esempio: ctunnel, iodine, ptunnel

3.2.1.40 blackarch-unpacker

Pacchetti che sono utilizzati per estrarre da un eseguibile del malware precedentemente incorporato.

Ad esempio: js-beautify

3.2.1.41 blackarch-voip

Pacchetti che operano su programmi e protocolli voip.

Ad esempio: iaxflood, rtp-flood, teardown



3.2.1.42 blackarch-webapp

Pacchetti che operano su applicazioni internet-facing.

Ad esempio: metoscan, whatweb, zaproxy

3.2.1.43 blackarch-windows

Questo gruppo è per qualsiasi pacchetto nativo Windows che viene eseguito via wine.

Ad esempio: 3proxy-win32, pwdump, winexe

3.2.1.44 blackarch-wireless

Pacchetti che operano su reti wireless a qualsiasi livello.

Ad esempio: airpwn, mdk3, wiffy

3.3 Struttura del Repository

Puoi trovare il principale git repo di BlackArch qui: <https://github.com/BlackArch/blackarch>. Ci sono anche alcuni repos secondari qui: <https://github.com/BlackArch>.

All'interno del principale git repo, ci sono tre importanti direttori:

- docs - Documentazione.
- packages - PKGBUILD files.
- scripts - Piccoli utili scripts.

3.3.1 Scripts

Qui c'è un riferimento per gli scripts nel direttorio scripts/ :

- baaaur - Presto questo caricherà i pacchetti su AUR.
- babuild - Genera un pacchetto.
- bachroot - Gestisce un chroot per testing.
- baclean - Pulisce i vecchi files .pkg.tar.xz dal repo dei pacchetti.
- baconflict - Presto questo rimpiazzerà scripts/conflicts.
- bad-files - Trova files corrotti nei pacchetti generati.
- balock - Ottiene o rilascia il lock del repo dei pacchetti.
- banotify - Notifica IRC riguardo ai push dei pacchetti.



- barelease - Rilascia i pacchetti al repo.
- baright - Stampa le info sul copyright di BlackArch.
- basign - Firma i pacchetti.
- basign-key - Firma una chiave.
- blackman - Questo si comporta tipo pacman ma compila da git (da non confondere con Blackman di nrz).
- check-groups - Controlla i gruppi.
- checkpkgs - Controlla gli errori nei pacchetti.
- conflicts - Controlla i conflitti tra file.
- dbmod - Modifica il database di un pacchetto.
- depth-list - Crea una lista ordinata per profondità delle dipendenze.
- deptree - Crea un albero delle dipendenze, elencando solo i pacchetti forniti da blackarch.
- get-blackarch-deps - Ottiene una lista di dipendenze blackarch per un pacchetto.
- get-official - Ottiene i pacchetti ufficiali per il rilascio.
- list-loose-packages - Elenca i pacchetti che non sono nei gruppi e non sono dipendenze per altri pacchetti.
- list-needed - Elenca le dipendenze mancanti.
- list-removed - Elenca i pacchetti che sono nel repo dei pacchetti ma non in git.
- list-tools - Elenca gli strumenti.
- outdated - Cerca i pacchetti out-dated nel repo dei pacchetti rispetto al git repo.
- pkgmod - Modifica un build package.
- pkgrel - Incrementa pkgrel in un pacchetto.
- prep - Ripulisce lo stile di un file PKGBUILD e trova gli errori.
- sitesync - Effettua la sincronizzazione tra una copia locale del repo dei pacchetti e una copia remota.
- size-hunt - Cerca i pacchetti grandi.
- source-backup - Effettua il backup dei files sorgente di un pacchetto.

3.4 Contribuire ad un Repository

Questa sezione ti mostra come contribuire al progetto BlackArch Linux. Accettiamo pull requests di qualsiasi dimensione, da piccole correzioni di battitura a nuovi pacchetti. Per aiuto, suggerimenti, o domande sentiti libero di contattarci.

Tutti sono invitati a contribuire. Tutti i contributi sono apprezzati.



3.4.1 Tutorials Richiesti

Per favore leggete i seguenti tutorials prima di contribuire:

- [Arch Packaging Standards](#)
- [Creating Packages](#)
- [PKGBUILD](#)
- [Makepkg](#)

3.4.2 Passaggi per Contribuire

Per inviare i tuoi cambiamenti al progetto BlackArchLinux segui i seguenti passaggi:

1. Forka il repository da <https://github.com/BlackArch/blackarch>
2. Modifica i files necessari, (e.g. PKGBUILD, .patch files, etc).
3. Effettua il commit delle tue modifiche.
4. Effettua il push delle tue modifiche.
5. Richiedici di includere le tue modifiche, preferibilmente attraverso una pull request.

3.4.3 Esempio

L'esempio successivo dimostra come proporre un nuovo pacchetto al progetto BlackArch. Utilizziamo [yaourt](#) (you can use [pacaur](#) as well) per ottenere un file PKGBUILD pre-esistente per [nfsshell](#) da [AUR](#) e modifichiamolo secondo i nostri bisogni.

3.4.3.1 Ottenere il PKGBUILD

Ottieni il file *PKGBUILD* utilizzando yaourt or pacaur:

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

3.4.3.2 Ripulire il PKGBUILD

Ripulisci il file *PKGBUILD* e risparmia un po' di tempo:



```
user@blackarchlinux nfsshell $ ./blarckarch/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 Adjust PKGBUILD

Modifica il file *PKGBUILD*:

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

3.4.3.4 Genera il Pacchetto

Genera il pacchetto:

```
==> 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 Installa e testa il pacchetto

Installa e testa il pacchetto:

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

3.4.3.6 Aggiungi, esegui il commit e il push del pacchetto

Aggiungi, esegui il commit e il push del pacchetto

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



3.4.3.7 Crea una pull request

Crea una pull request su github.com

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

3.4.3.8 Aggiungere un Remoto per Upstream

Una cosa utile da fare se stai lavorando upstream e su un fork è di fare il pull del tuo fork e aggiungerlo al repo ba principale come remoto.

```
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/blackarchlinux
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)
```

Per default, git dovrebbe fare il push direttamente all'origine, ma assicurati che la tua configurazione di git sia corretta. Questo non sarà un problema a meno che tu abbia i diritti di commit dato che non sarai in grado di effettuare il push upstream senza di essi.

Se hai la capacità di fare commit, potresti avere più successo usando `git@github.com:blackarch/blackarch.git` ma dipende da te.

3.4.4 Richieste

1. Non aggiungere commenti **Maintainer** o **Contributor** ai files *PKGBUILD*. Aggiungi i nomi di maintainer e contributor alla sezione AUTHORS della guida BlackArch.
2. Per consistenza, per favore segui lo stile generale degli altri file *PKGBUILD* nel repo e usa l'indentazione a due spazi.

3.4.5 Consigli Generali

namcap può controllare i pacchetti alla ricerca di errori.

Chapter 4

Guida Strumenti

Prossimamente...

4.1 Coming Soon

Prossimamente...