



BlackArch Linux Rehberi

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

İçindekiler

1 Giriş	3
1.1 Ön Bakış	3
1.2 BlackArch Linux Nedir?	3
1.3 BlackArch Linux Rehberi	3
1.4 Supported platforms	3
1.5 Dahil Olun	4
2 Kullanıcı Rehberi	5
2.1 Kurulum	5
2.1.1 Var olan Arch Linux üzerine kurulum	5
2.1.2 Paket kurulumu	5
2.1.3 Kaynak koddan paket kurulumu	6
2.1.4 Temel BlackMan kullanımı	6
2.1.5 Doğrudan çalıştırılabilir ISO ile kurulum yapma	7
3 Geliştirici Rehberi	8
3.1 Arch Linux Paket Derleme ve Paket Depoları	8
3.2 Blackarch PKGBUILD standartları	8
3.2.1 Gruplar	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	Repository structure	14
3.3.1	Betikler	14
3.4	Depoya katkıda bulunmak	16
3.4.1	Yardımcı kaynaklar	16
3.4.2	Katkı aşamaları	16
3.4.3	Örnek	16
3.4.3.1	PKGBUILD elde etme	16
3.4.3.2	PKGBUILD temizleme	17
3.4.3.3	PKGBUILD ayarlama	17
3.4.3.4	Paket derleme	17
3.4.3.5	Paketin kurulumu ve test edilmesi	17
3.4.3.6	Git'e gönderme	18
3.4.3.7	İstek gönderme	18
3.4.3.8	Güncellemeleri takip etmek	18
3.4.4	İstekler	18
3.4.5	Genel ipuçları	18
4	Araç Rehberi	19
4.1	Yakında	19

Bölüm 1

Giriş

1.1 Ön Bakış

Bu rehber 4 ana parçaya bölünmüştür:

- Giriş - Projeye genel bir bakış, giriş ve çeşitli yararlı ilk bilgiler
- Kullanıcı Rehberi - Normal bir kullanıcının BlackArch'ı verimli kullanması için gerekli bilgiler
- Geliştirici Rehberi - BlackArch'ın geliştirilmesine yardımcı veya destek olmak isteyenler için gerekli bilgiler
- Araç Rehberi - Kurulu araçların kullanım yöntemleri ve ipuçları

1.2 BlackArch Linux Nedir?

BlackArch, siber güvenlik araştırmacıları ve sızma testi uzmanları için geliştirilmekte olan bir GNU/Linux dağıtımdır. **ArchLinux** işletim sistemini taban olarak kabul etmiştir ve kullanıcılara BlackArch paketlerini kurma imkanı sağlamaktadır. Kullanıcılar bu paketleri grup olarak yükleyebileceği gibi, doğrudan da yükleyebilir.

Araçların yer aldığı paket deposu, Arch Linux işletim sisteminde olduğu gibi **resmi olmayan kullanıcı deposu** olarak kullanılabilir. Doğal olarak hali hazırda kurulu olan Arch Linux işletim sistemine kurulum yapılabilir. Paket deposu eklemesi yapıldıktan sonra araçlar doğrudan veya gruplar halinde kurulabilir.

Şu anda depo içerisinde **1300** araç bulunmaktadır. Depo kalitesinin korunması için tüm araçlar eklenmeden önce test edilmektedir.

1.3 BlackArch Linux Rehberi

Yakında...

1.4 Supported platforms

Yakında...



1.5 Dahil Olun

BlackArch ekibi ile aşağıdaki yöntemlerden istediğiniz birisi ile iletişime geçebilirsiniz:

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

E-Posta adresi: blackarchlinux@gmail.com

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

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

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

Bölüm 2

Kullanıcı Rehberi

2.1 Kurulum

Bu kısım altında BlackArch deposunun nasıl sisteme dahil edileceği ve paketlerin nasıl kurulacağı yer almaktadır. BlackArch depolarda yer alan derlenmiş paketleri desteklediği gibi kaynak kod ile paket kurulumunu da desteklemektedir.

BlackArch normal bir Arch Linux kurulumu ile uyumludur. Resmi olmayan kullanıcı deposu olarak düşünülebilir. Paket deposu yerine doğrudan ISO ile kurulum yapmak isterseniz [Live ISO](#) kısmına bakabilirsiniz.

2.1.1 Var olan Arch Linux üzerine kurulum

strap.sh betiğini root yetkileri ile çalıştırınız ve aşağıdaki adımları takip ediniz. Hash değeri kontrolünü yapmayı unutmayınız.

```
curl -O https://blackarch.org/strap.sh
shasum strap.sh # bu degere esit olmalı: 6f152b79419491db92c1fdde3fad2d445f09aae3
sudo ./strap.sh
```

Şimdi, ana paket listesinin güncellenmesi ve paketlerinizin senkron olması için aşağıdaki komutu kullanınız:

```
sudo pacman -Syyu
```

2.1.2 Paket kurulumu

Araçları blackarch deposundan kurabilirsiniz.

1. Kurulabilir tüm araçları listelemek için:

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

2. Tüm araçları kurmak için:

```
pacman -S blackarch
```



3. Belirli bir kategoride yer alan araçları kurmak için:

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

4. Tüm blackarch kategorilerini görmek için:

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

2.1.3 Kaynak koddan paket kurulumu

Kurulumlara alternatif olarak BlackArch paketlerini doğrudan kaynak kod yardımı ile kurabilirsiniz. Bunun için PKGBUILD dosyalarına **paketler** dizini altından erişebilirsiniz. Tüm depoyu derlemek için **Blackman** aracını kullanabilirsiniz.

- BlackArch paket deposu sisteminizde ekliyse doğrudan aşağıdaki komut yardımıyla Blackman aracını kurabilirsiniz.

```
pacman -S blackman
```

- Blackman'ı kaynak koddan kurmak için aşağıdaki adımları takip edebilirsiniz.

```
mkdir blackman
cd blackman
wget https://raw2.github.com/BlackArch/blackarch/master/packages/blackman/PKGBUILD
# PKGBUILD dosyasında herhangi bir sikinti olmadığını kontrol etmeyi unutmayınız!
makepkg -s
```

- Blackman'ı AUR aracılığı ile de kurabilirsiniz.

```
<AUR yardimci araciniz> -S blackman
```

2.1.4 Temel BlackMan kullanımı

Blackman her ne kadar alıştığınız diğer paket yöneticilerinin parametrelerinden farklı bir yapı kullansa da, gayet kolay bir kullanıma sahiptir. En temel kullanımı aşağıda gösterilmiştir.

- Paketi indir, derle ve kur:

```
sudo blackman -i paket
```

- Paket grubunu indir, derle ve içerisinde yer alan tüm paketleri kur:

```
sudo blackman -g grup
```

- Tüm BlackArch araçlarını indir, derle ve kur:

```
sudo blackman -a
```

- BlackArch kategorilerini listeleme:

```
blackman -l
```

- Kategori içerisinde yer alan araçları listeleme:

```
blackman -p category
```



2.1.5 Doğrudan çalıştırılabilir ISO ile kurulum yapma

BlackArch'ı doğrudan çalıştırılabilir ISO aracılığı ile de kurabilirsiniz. Bunun için <https://www.blackarch.org/download.html#iso> sayfasından bulabileceğiniz imaj ile aşağıdaki adımları takip edebilirsiniz. Adımları takip edebilmek için bilgisayarınızı indirdiğiniz imaj ile boot etmeniz gerekmektedir.

- blackarch-installer paketi kurulumu:

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

- Çalıştırma

```
sudo blackarch-install
```


Bölüm 3

Geliştirici Rehberi

3.1 Arch Linux Paket Derleme ve Paket Depoları

PKGBUILD dosyaları derleme betikleridir. Bu betikler makepkg(1) uygulamasına paketin nasıl oluşturulacağı bilgisini sağlamaktadır. PKGBUILD dosyaları Bash betikleri olarak yazılmaktadır.

Daha fazla bilgi için aşağıdaki kaynaklara bakabilirsiniz:

- [Arch Wiki: Paket Oluşturulması](#)
- [Arch Wiki: makepkg](#)
- [Arch Wiki: PKGBUILD](#)
- [Arch Wiki: Arch Paketleri Standartları](#)

3.2 Blackarch PKGBUILD standartları

AUR deposundaki paketler ile benzerliği korumak için PKGBUILD dosyaları ufak farklar haricinde benzer şekilde oluşturulmaktadır. Her bir paket en az blackarch grubuna dahil olmalıdır. Benzer şekilde bir paket birden fazla gruba dahil olabileceği gibi bir grup içerisinde de birden fazla paket bulunabilir.

3.2.1 Gruplar

Kullanıcılar kolay ve hızlı bir şekilde benzer paketleri kurabilmesi için çeşitli gruplamalar yapılmıştır. Grublamanın getirdiği avantaj sayesinde "pacman -S jgrup_adi" şeklinde bir komut ile tüm paketler kurulabilir.

3.2.1.1 blackarch

"blackarch" grubu diğer tüm grupların dahil olmak zorunda olduğu temel gruptur. Bu, tüm kullanıcıların tüm paketleri rahatlıkla kurmasını sağlar.

Bu grupta ne bulunabilir? Herşey.



3.2.1.2 blackarch-anti-forensic

Şifreleme, steganografi ve dosya özelliklerini düzenleme işlemlerini içeren adli bilişim işlemleri için kullanılan paketlerdir. İçerdiği tüm araçların amacı sistemde bilgi gizlemek için değişiklik yapmaktır.

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

3.2.1.3 blackarch-automation

Araçların veya iş akışının düzenlenmesi için kullanılan paketlerdir.

Örnek: blueranger, tiger, wiffy

3.2.1.4 blackarch-backdoor

Zafiyetli sistemlerdeki açık backdoorlar veya exploitlerdir.

Örnek: backdoor-factory, rrs, weevely

3.2.1.5 blackarch-binary

Bazı formlardaki binary dosya işlemleri için kullanılan paketlerdir.

Örnek: binwally, packerid

3.2.1.6 blackarch-bluetooth

Bluetooth standardı(802.15.1) ile alakalı exploitlerdir.

Örnek: ubertooh, tbear, redfang

3.2.1.7 blackarch-code-audit

Zafiyet analizi için kaynak kod denetimi yapan paketlerdir.

Örnek: flawfinder, pscan

3.2.1.8 blackarch-cracker

Hash gibi kriptografik fonksiyonların cracklenmesinde kullanılan paketlerdir.

Örnek: hashcat, john, crunch

3.2.1.9 blackarch-crypto

Crack işlemleri haricindeki kriptografik işlemlerde kullanılan paketlerdir.

Örnek: ciphertest, xortool, sbd



3.2.1.10 blackarch-database

Herhangi bir seviyedeki veritabanı exploitlerini içeren paketlerdir.

Örnek: metacoretex, blindsqli

3.2.1.11 blackarch-debugger

Belirli bir programın realtime'da yaptığı işi incelemeye yarayan paketlerdir.

Örnek: radare2, shellnoob

3.2.1.12 blackarch-decompiler

Derlenmiş programları kaynak koduna çevirmeye yarayan paketlerdir.

Örnek: flasm, jd-gui

3.2.1.13 blackarch-defensive

Kullanıcıyı zararlı yazılımlardan ve saldırılardan koruyan paketlerdir.

Örnek: arpon, chkrootkit, sniffjoke

3.2.1.14 blackarch-disassembler

Bu paketler blackarch-decompiler paketlerine benzer ve bir çok paket her iki grupta da bulunabilir. Bunların farkı ise binary dosyayı reverse ederek kaynak kodu yerine assembly çıktısı verirler.

Örnek: inguma, radare2

3.2.1.15 blackarch-dos

DoS (Denial of Service) saldırıları için kullanılan paketlerdir.

Örnek: 42zip, nkiller2

3.2.1.16 blackarch-drone

Fiziksel olarak drone'ları yönetmek için kullanılan paketlerdir.

Örnek: meshdeck, skyjack

3.2.1.17 blackarch-exploitation

Diğer program ve servisleri exploit ederken avantaj sağlayan paketlerdir.

Örnek: armitage, metasploit, zarp



3.2.1.18 blackarch-fingerprint

Parmak izi biyometrik okuyucuların exploit edilmesinde kullanılan paketlerdir.

Örnek: dns-map, p0f, httpprint

3.2.1.19 blackarch-firmware

Firmware'deki zafiyetlerin exploit edilmesinde kullanılan paketlerdir.

Örnek: Henüz yok, en kısa sürede değiştirilecek.

3.2.1.20 blackarch-forensic

Fiziksel disk ve hafızadaki verilerin bulunmasında kullanılan paketlerdir.

Örnek: aesfix, nfex, wyd

3.2.1.21 blackarch-fuzzer

Fuzzing işlemlerinde kullanılan paketlerdir. Örneğin bu paketlerle hedefe random değerler göndererek sonucunda ne olduğunu inceleyebilirsiniz.

Örnek: msf, mdk3, wfuzz

3.2.1.22 blackarch-hardware

Fiziksel donanım ile yapılabilecek şeyleri yönetmek veya exploit etmek için kullanılan paketlerdir.

Örnek: arduino, smali

3.2.1.23 blackarch-honeypot

"Honeypot" görevi gören araçlardır. Honeypot'lar saldırganları tuzağa düşürmek için kullanılan programlardır.

Örnek: artillery, bluepot, wifi-honey

3.2.1.24 blackarch-keylogger

Bir sistemdeki klavye girdilerini saklayan paketlerdir.

Örnek: Henüz yok, en kısa sürede değiştirilecek.

3.2.1.25 blackarch-malware

Malware'lerin veya zararlı olabilecek yazılımların tespitinde kullanılan paketlerdir.

Örnek: malwaredetect, peepdf, yara



3.2.1.26 **blackarch-misc**

Başka herhangi bir gruba ait olmayan paketlerdir.

Örnek: oh-my-zsh-git, winexe, stompy

3.2.1.27 **blackarch-mobile**

Mobil platformların manipüle edilmesinde kullanılan paketlerdir.

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

3.2.1.28 **blackarch-networking**

Network işlemlerinde kullanılan paketlerdir.

Örnek: Hemen hemen herşey

3.2.1.29 **blackarch-nfc**

NFC (near-field communications) işlemlerinde kullanılan paketlerdir.

Örnek: nfcutils

3.2.1.30 **blackarch-packer**

Packerları içeren paketlerdir.

/textifPackerlar diğer yazılımlara zararlı yazılım gömebilen paketlerdir.

Örnek: packerid

3.2.1.31 **blackarch-proxy**

Proxy görevi görevi gören uygulamalardır. Proxy'ler trafiği internetteki başka node'lar üzerinden geçiren yapılardır.

Örnek: burpsuite, ratproxy, sslnuke

3.2.1.32 **blackarch-recon**

Aktif zafiyet taraması yapan paketlerdir. Daha çok benzer paketlerin bir araya toplandığı şemsiye görevi görür.

Örnek: canri, dnsrecon, netmask

3.2.1.33 **blackarch-reversing**

Decompiler, disassembler gruplarının veya benzer programların bir araya toplandığı gruptur.

Örnek: capstone, radare2, zerowine



3.2.1.34 **blackarch-scanner**

Hedef sistemde zafiyet taraması yapan paketlerdir.

Örnek: scanssh, tiger, zmap

3.2.1.35 **blackarch-sniffer**

Ağ trafiğini analiz etmeye yarayan paketlerdir.

Örnek: hexinject, pytactile, xspy

3.2.1.36 **blackarch-social**

Öncelikli olarak sosyal ağ sitelerine saldırmaya yarayan paketlerdir.

Örnek: jigsaw, websploit

3.2.1.37 **blackarch-spoof**

Saldırmanın kendisini gizleyerek kurbanı aldatma girişimlerinde kullandığı paketlerdir.

Examples: arpoison, lans, netcommander

3.2.1.38 **blackarch-threat-model**

Belirli bir senaryoda verilen tehdit modelini kaydetmek/raporlamak için kullanılan araçlardır.

Örnek: magictree

3.2.1.39 **blackarch-tunnel**

Verilen ağa tünel ap trafiği oluşturmak için kullanılan paketlerdir.

Örnek: ctunnel, iodine, ptunnel

3.2.1.40 **blackarch-unpacker**

Çalıştırılabilir dosyalara yerleştirilmiş zararlı yazılımları çıkarmak için kullanılan paketlerdir.

Örnek: js-beautify

3.2.1.41 **blackarch-voip**

Voip programları ve protokolleri üzerinde işlem yapan paketlerdir.

Örnek: iaxflood, rtp-flood, teardown



3.2.1.42 blackarch-webapp

İnternet arayüzü olan uygulamaların işlemlerinde kullanılan paketlerdir.

Örnek: metoscan, whatweb, zaproxy

3.2.1.43 blackarch-windows

"wine" ile çalışan windows uygulamalarının bulunduğu paketlerdir.

Örnek: 3proxy-win32, pwdump, winexe

3.2.1.44 blackarch-wireless

Kablosuz ağ işlemleri yapan paketlerdir.

Örnek: airpwn, mdk3, wiffy

3.3 Repository structure

Ana BlackArch git deposuna buradan erişebilirsiniz: <https://github.com/BlackArch/blackarch>. Bir kaç tane ikincil depoya da erişmek için aşağıdaki linki kullanabilirsiniz: <https://github.com/BlackArch>.

Ana git deposunda üç önemli klasör bulunmaktadır:

- docs - Dökümantasyon.
- packages - PKGBUILD dosyaları.
- scripts - Basit kullanışlı scriptler.

3.3.1 Betikler

Aşağıda scripts/ klasöründe bulunan betikler için referanslar listelenmiştir:

- baaur - Yakında. Paketleri AUR'a yüklemek için kullanılacak.
- babuild - Paketleri derler.
- bachroot - Test için chroot'u yönetir.
- baclean - Eski .pkg.tar.xz dosyalarını paket deposundan temizler.
- baconflict - Yakında. scripts/conflicts'leri düzeltecek.
- bad-files - Derlenmiş paketlerdeki hatalı dosyaları bulur.
- balock - Paket deposunu kilitler/açar.
- banotify - Paket değişikliklerinden IRC'yi haberdar eder.



- barelease - Paketleri paket deposuna gönderir.
- baright - BlackArch copyright bilgisini bastırır.
- basign - Paketleri imzalar.
- basign-key - Anahtarları imzalar.
- blackman - Bir nevi pacman gibi davranır, fakat paketleri git'ten çeker. (nrz'nin Blackman'inden farklıdır.).
- check-groups - Grupları kontrol eder.
- checkpkgs - Paketlerdeki hataları kontrol eder.
- conflicts - Dosya çakışmalarını kontrol eder.
- dbmod - Paket veritabanını düzenler.
- depth-list - Bağımlılık(dependency) derinliğine göre liste oluşturur.
- deptree - Yalnızca blackarch tarafından sunulan paketleri listeleterek bağımlılık ağacını oluşturur.
- get-blackarch-deps - Bir paket için blackarch bağımlılıklarını listeler.
- get-official - Yayınlanacak resmi paketleri listeler.
- list-loose-packages - Herhangi bir grupta yer almayan veya diğer paketlerle bağımlılığı olmayan paketleri listeler.
- list-needed - Eksik bağımlılıkları listeler.
- list-removed - Paket deposunda bulunan, git'te bulunmayan paketleri listeler.
- list-tools - Araçları listeler.
- outdated - Paket deposunda bulunan paketlerin versiyonlarının git deposuna göre eski olanları bulur.
- pkgmod - Derlenmiş bir pakette değişiklik yapar.
- pkgrel - Bir paketdeki pkgrel'i artırır.
- prep - PKGBUILD dosyalarını temizler ve hataları bulur.
- sitesync - Bir paketin paket deposundaki local kopyası ile uzak sunucudaki kopyası arasında senkronizasyon sağlar.
- size-hunt - Büyük paketleri arar.
- source-backup - Paketlerin kaynak dosyalarını yedekler.



3.4 Depoya katkıda bulunmak

Bu bölüm BlackArch Linux projesine nasıl katkıda bulunabileceğinizi anlatmaktadır. Küçük harf hataları düzeltmelerinden yeni paketlere kadar her türlü pull request'ler kabul edilmektedir. Yardımcı olmak, öneride bulunmak veya soru sormak için bizimle iletişime geçebilirsiniz.

Herkes katkıda bulunabilir. Tüm destekler değerlendirilecektir.

3.4.1 Yardımcı kaynaklar

Lütfen katkıda bulunmadan önce aşağıdaki kısmı okuyunuz:

- [Arch Paketleme Standartları](#)
- [Paket Oluşturma](#)
- [PKGBUILD](#)
- [Makepkg](#)

3.4.2 Katkı aşamaları

BlackArch Linux projesine değişikliklerinizi göndermek için aşağıdaki adımları takip edebilirsiniz:

1. Depoyu <https://github.com/BlackArchLinux/blackarchlinux> adresinden forklayın.
2. Gerekli dosyaları düzenleyin, (e.g. PKGBUILD, .patch dosyaları, vb).
3. Değişikliklerinizi commitleyin.
4. Değişikliklerinizi pushlayın.
5. Tercihen pull request ile değişikliklerinizi birleştirmemizi isteyin.

3.4.3 Örnek

Aşağıdaki örnek yeni bir paketi BlackArch projesine göndermeyi göstermektedir. **nfsshell** için önceden var olan PKGBUILD dosyasını almak için **yaourt** kullanıyoruz (isterseniz **pacaur**'da kullanabilirsiniz.) ve ihtiyaçlarımıza göre ayarlıyoruz.

3.4.3.1 PKGBUILD elde etme

Yaourt veya pacaur kullanarak *PKGBUILD* dosyasını alıyoruz:

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



3.4.3.2 PKGBUILD temizleme

PKGBUILD dosyasını temizleyerek biraz zaman kazanıyoruz:

```
user@blackarchlinux nfsshell $ ./blackarch/scripts/prepare 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 PKGBUILD ayarlama

PKGBUILD dosyasını ayarlıyoruz:

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

3.4.3.4 Paket derleme

Paketi derliyoruz:

```
==> 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 Paketin kurulumu ve test edilmesi

Paketi kurup test ediyoruz: Install and test the package:

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



3.4.3.6 Git'e gönderme

Paketi commitleyip git'e gönderiyoruz:

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

3.4.3.7 İstek gönderme

github.com üzerinden pull request oluşturuyoruz:

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

3.4.3.8 Güncellemeleri takip etmek

Eğer kendi forkladığınız depo üzerinde çalışıyorsanız ve ana ba deposunu uzak depo olarak belirlediyseniz güncellemeleri takip etmek için yapılacaklar:

```
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/blackarchlinux/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)
```

Git varsayılan olarak origin'e gönderir ama git config dosyanızın doğru ayarlandığından emin olmanız gerekir. Normal commitler sırasında bu problem olmaz fakat eğer uzak sunucuya göndermek isterseniz bu mümkün olmayacaktır.

Eğer yapabiliyorsanız `git@github.com:blackarch/blackarch.git` kullanarak commitlerinizi yapmanız daha başarılı olacaktır. Fakat bu konuda tercih size kalmış.

3.4.4 İstekler

1. **Maintainer** ya da **Contributor** isimlerini yorum olarak *PKGBUILD* dosyalarına eklemeyin. Maintainer ve contributor isimlerini BlackArch rehberindeki AUTHORS bölümüne ekleyebilirsiniz.
2. Tutarlılığa uyması adına depoda bulunan başka bir *PKGBUILD* dosyasının genel stilini kullanınız. Ayrıca girintilerde iki boşluk kullanınız.

3.4.5 Genel ipuçları

`namcap` ile paket hatalarını kontrol edebilirsiniz.

Bölüm 4

Araç Rehberi

Yakında...

4.1 Yakında

Yakında...