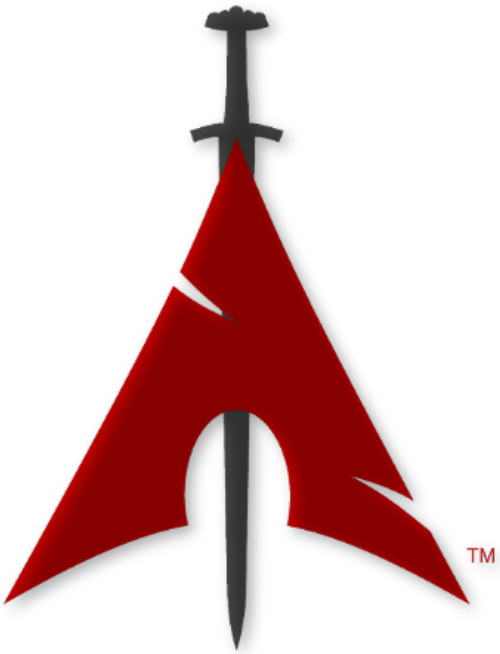

The BlackArch Linux Guide

https://www.blackarch.org/

Оглавление

Глава 1

Введение

1.1 Обзор

Руководство BlackArch Linux разделено на несколько частей:

- Введение - Предоставляет широкий обзор, введение и дополнительную полезную информацию о проекте
- Руководство Пользователя - Все, что обычный пользователь должен знать, чтобы эффективно использовать BlackArch
- Руководство Разработчика - Как начать разработку и внесение вклада в BlackArch
- Руководство по инструментам - Подробные сведения об инструменте по примеру использования (WIP)

1.2 Что такое BlackArch Linux?

BlackArch представляет собой полный Linux дистрибутив для тестеров на проникоовение и исследователей безопасности. Он основан на ArchLinux и пользователи могут установить компоненты BlackArch лично или группами.

Набор инструментов распраняется как неофициальный пользовательский репозиторий Arch Linux, поэтому вы можете установить BlackArch поверх существующего Arch Linux. Пакеты могут устанавливаться отдельно или категориями.

Постоянно расширяющийся репозиторий в настоящее время включает в себя 1925 инструментов. Все инструменты тщательно тестируются перед добавлением в кодовую базу для поддержания качества репозитория.

1.3 История BlackArch Linux

Coming soon...

1.4 Поддерживаемые платформы

Coming soon...

1.5 Принять участие

Вы можете связаться с командой BlackArch, используя следующие возможности:

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/

Глава 2

Руководство Пользователя

2.1 Установка

В следующих разделах рассказывается, как настроить репозиторий BlackArch и установить пакеты. BlackArch поддерживает оба варианта, установка из репозитория с использованием бинарных пакетов и их компиляция и установка из источников.

BlackArch совместим с обычной установкой Arch. Он выступает в качестве неофициального пользовательского репозитория. Если вместо этого вы хотите ISO, см. Раздел Live ISO.

2.1.1 Установка поверх ArchLinux

Запустите strap.sh с правами админа(root) и следуйте инструкциям. Смотрите следующий пример.

```
curl -O https://blackarch.org/strap.sh
sha1sum strap.sh # should match: 6f152b79419491db92c1fdde3fad2d445f09aae3
sudo ./strap.sh
```

Теперь загрузите свежую копию master package list и выполните синхронизацию пакетов:

```
sudo pacman -Syyu
```

2.1.2 Установка пакетов

Теперь вы можете установить инструменты из репозитория blackarch.

1. Чтобы просмотреть все доступные инструменты, выполните

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

2. Чтобы установить все инструменты, выполните

   ```
   pacman -S blackarch
   ```

3. Чтобы установить категорию инструментов, выполните

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

4. Чтобы посмотреть категории blackarch, выполните

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

2.1.3 Установка пакетов из исходников

В рамках альтернативного метода установки вы можете собрать BlackArch пакеты из исходников. Вы можете найти PKGBUILDы на github. Для сборки всего репозитория, вы можете использовать инструмент Blackman.

- Во-первых, вам нужно установить Blackman. Если на вашем компьютере настроен репозиторий пакетов BlackArch, вы можете установить Blackman:

```
pacman -S blackman
```

- Вы можете собрать и установить Blackman из исходников:

```
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
```

- Или вы можете установить Blackman из AUR:

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

2.1.4 Основное использование Blackman

Blackman очень прост в использовании, хотя флаги отличаются от того, чего вы обычно ожидаете от pacman. Основное использование приведено ниже.

- Скачать, скомпилировать и установить пакеты:

```
sudo blackman -i package
```

- Скачать, скомпилировать и установить целую категорию:

```
sudo blackman -g group
```

- Скачать, скомпилировать и установить все инструменты BlackArch:

```
sudo blackman -a
```

- Список blackarch категорий:

```
blackman -l
```

- Список категорий инструментов:

```
blackman -p category
```

2.1.5 Установка из live-, netinstall- ISO или ArchLinux

Вы можете установить BlackArch Linux из одного из наших live- or netinstall- ISOs.
См. https://www.blackarch.org/download.html#iso. После загрузки ISO необходимо выполнить следующие шаги.

- Установка пакета blackarch-installer:

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

- Запуск

```
sudo blackarch-install
```

Глава 3

Руководство разработчика

3.1 Система сборки и Репозиториев Arch

Файлы PKGBUILD - это скрипты сборки. Каждый из них сообщает makepkg (1), как создать пакет. Файлы PKGBUILD написаны на Bash.

Для получения дополнительной информации прочтите следующее:

- Arch Wiki: Creating Packages
- Arch Wiki: makepkg
- Arch Wiki: PKGBUILD
- Arch Wiki: Arch Packaging Standards

3.2 Стандарты Blackarch PKGBUILD

Ради простоты, наши PKGBUILDы аналогичны характеристикам AUR, с несколькими небольшими различиями, описанными ниже. Каждый пакет должен как минимум предналежать к blackarch; многие пакеты могут принадлежать более чем одной группе.

3.2.1 Группы

Чтобы разрешить пользователям устанавливать быстро и легко определенный диапазон пакетов, пакеты были разделены на группы. Группы позволяют пользователям перейти в "pacman -S <group name>"чтобы получить множество пакетов.

3.2.1.1 blackarch

Группа blackarch - это основная группа в которую входят все пакеты. Это позволяет пользователям с легкостью устанавливать каждый пакет.

Что должно быть здесь: Все.

3.2.1.2 blackarch-anti-forensic

Пакеты которые используются для противодействивия судебной деятельности, включая шифрование, стеганографию и все, что изменяет атрибуты файлов/файла. Все это включает в себя инструменты для работы с чем угодно, которые вносят изменения в систему в целях сокрытия информации.

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

3.2.1.3 blackarch-automation

Пакеты, используемые для автоматизации рабочих процессов(workflow automation).

Примеры: blueranger, tiger, wiffy

3.2.1.4 blackarch-backdoor

Пакеты, которые используют уязвимости или бэкдоры на уже уязвимых системах.

Примеры: backdoor-factory, rrs, weevely

3.2.1.5 blackarch-binary

Пакеты, которые обрабатывают бинарные файлы в некоторой форме.

Примеры: binwally, packerid

3.2.1.6 blackarch-bluetooth

Пакеты, которые используют любые уязвимости касаемо стандарта Bluetooth (802.15.1).

Примеры: ubertooth, tbear, redfang

3.2.1.7 blackarch-code-audit

Пакеты, проводящие аудит существующего исходного кода для анализа уязвимости.

Примеры: flawfinder, pscan

3.2.1.8 blackarch-cracker

Пакеты, используемые для взлома криптографических функций, т.е. хешей.

Примеры: hashcat, john, crunch

3.2.1.9 blackarch-crypto

Пакеты, работающие с криптографией, за исключением взлома.

Примеры: ciphertest, xortool, sbd

3.2.1.10 blackarch-database

Пакеты, связанные с эксплуатацией базы данных на любом уровне.

Примеры: metacoretex, blindsql

3.2.1.11 blackarch-debugger

Пакеты, которые позволяют пользователю просматривать то, что определенная программа "делает"в режиме реального времени.

Примеры: radare2, shellnoob

3.2.1.12 blackarch-decompiler

Пакеты, используемые для просмотра исходного кода уже скомпилированных программ.

Примеры: flasm, jd-gui

3.2.1.13 blackarch-defensive

Пакеты, которые используются для защиты пользователя от вредоносных программ и от атак других пользователей.

Примеры: arpon, chkrootkit, sniffjoke

3.2.1.14 blackarch-disassembler

Пакеты, преобразующие программу с машинного кода в текст программы на языке ассемблера.

Примеры: inguma, radare2, smali

3.2.1.15 blackarch-dos

Пакеты, используемые для DoS-атак, цель которых довести систему до отказа.

Примеры: 42zip, nkiller2

3.2.1.16 blackarch-drone

Пакеты, которые используются для управления физически сконструированными дронами.

Примеры: meshdeck, skyjack

3.2.1.17 blackarch-exploitation

Пакеты, которые используют уязвимости в других программах или службах.

Примеры: armitage, metasploit, zarp

3.2.1.18 blackarch-fingerprint

Пакеты, использующие идентификацию по "отпечаткам пальцев".

Примеры: dns-map, p0f, httprint

3.2.1.19 blackarch-firmware

Пакеты, которые используют уязвимости в прошивке.

Примеры: None yet, amend asap.

3.2.1.20 blackarch-forensic

Пакеты, которые используются для поиска данных на физических дисках или встроенной памяти.

Примеры: aesfix, nfex, wyd

3.2.1.21 blackarch-fuzzer

Пакеты, в которых используется принцип fuzz-тестирования (фаззинга), заключающийся в передаче приложению на вход неправильных, неожиданных или случайных данных, чтобы проверить надёжность ПО и компьютерных систем.

Примеры: msf, mdk3, wfuzz

3.2.1.22 blackarch-hardware

Пакеты, которые используют или управляют чем-либо, связанным с физическим оборудованием.

Примеры: arduino, smali

3.2.1.23 blackarch-honeypot

Пакеты, которые действуют как "приманки т.е. программы, которые оказались уязвимыми службами, используемыми для привлечение хакеров в ловушку.

Примеры: artillery, bluepot, wifi-honey

3.2.1.24 blackarch-keylogger

Пакеты, которые записывают и сохраняют нажатия клавиш на другой системе.

Примеры: None yet, amend asap.

3.2.1.25 blackarch-malware

Пакеты, обнаруживающие любые типы вредоносного ПО.

Примеры: malwaredetect, peepdf, yara

3.2.1.26 blackarch-misc

Пакеты, которые не вписываются ни в какие категории; разнообразные пакеты.

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

3.2.1.27 blackarch-mobile

Пакеты, которые работают с мобильными платформами.

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

3.2.1.28 blackarch-networking

Пакет, который включает в себя IP-networking.

Примеры: Anything pretty much

3.2.1.29 blackarch-nfc

пакеты, которые используют nfc (near-field communications).

Примеры: nfcutils

3.2.1.30 blackarch-packer

Пакеты, которые оперируют с упаковщиками или связаны с ними.

Упаковщики - это программы, которые внедряют вредоносное ПО в другие исполняемые файлы.

Примеры: packerid

3.2.1.31 blackarch-proxy

Пакеты, которые действуют как прокси-сервер, т.е. перенаправляют трафик через другой узел в Интернете.

Примеры: burpsuite, ratproxy, sslnuke

3.2.1.32 blackarch-recon

Packages that actively seeks vulnerable exploits in the wild. More of an umbrella group for similar packages.

Примеры: canri, dnsrecon, netmask

3.2.1.33 blackarch-reversing

Это группа umbrella для любого декомпилятора, дизассемблера или любой подобной программы.

Примеры: capstone, radare2, zerowine

3.2.1.34 blackarch-scanner

Пакеты, которые сканируют выбранные системы на наличие уязвимостей.

Примеры: scanssh, tiger, zmap

3.2.1.35 blackarch-sniffer

Пакеты, которые включают в себя анализ сетевого трафика.

Примеры: hexinject, pytactle, xspy

3.2.1.36 blackarch-social

Пакеты, которые в первую очередь атакуют сайты социальных сетей.

Примеры: jigsaw, websploit

3.2.1.37 blackarch-spoof

Пакеты, которые пытаются обмануть атакующего таким образом, в котором атакующий не появляется жертве в качестве атакующего.

Примеры: arpoison, lans, netcommander

3.2.1.38 blackarch-threat-model

Пакеты, которые будут использоваться для отчетов/записи модели угрозы, изложенной в конкретном сценарии.

Примеры: magictree

3.2.1.39 blackarch-tunnel

Пакеты, которые используются для туннелирования сетевого трафика в данной сети.

Примеры: ctunnel, iodine, ptunnel

3.2.1.40 blackarch-unpacker

Пакеты, которые используются для извлечения предварительно упакованных вредоносных программ из исполняемого файла.

Примеры: js-beautify

3.2.1.41 blackarch-voip

Пакеты, которые оперируют voip программами и протоколами.

Примеры: iaxflood, rtp-flood, teardown

3.2.1.42 blackarch-webapp

Пакеты, которые оперируют internet-facing приложениями.

Примеры: metoscan, whatweb, zaproxy

3.2.1.43 blackarch-windows

Эта группа предназначена для любого родного пакета Windows, который работает через wine.

Примеры: 3proxy-win32, pwdump, winexe

3.2.1.44 blackarch-wireless

Пакеты, которые оперируют беспроводными сетями на любом уровне.

Примеры: airpwn, mdk3, wiffy

3.3 Струкрура репозитория

Вы можете найти главный git репозиторий BlackArch тут:
https://github.com/BlackArch/blackarch.

Также имеется несколько вторичных репозиториев:
https://github.com/BlackArch.

В основном git репозитории есть 3 важных каталога:

- docs - Документация.
- packages - PKGBUILD файлы.
- scripts - Полезные небольшие скрипты.

3.3.1 Скрипты

Вот ссылка на скрипты в `scripts/` каталоге:

- baaur - Soon, this will upload packages to the AUR.
- babuild - Сборка пакета.
- bachroot - Управление chroot для тестирования.

- baclean - Очистить старые .pkg.tar.xz файлы из репозитория пакетов.
- baconflict - Вскорее, это заменит `scripts/conflicts`.
- bad-files - Поиск плохих файлов в встроенных пакетах.
- balock - Obtain or release the package repo lock.
- banotify - Notify IRC about package pushes.
- barelease - Выпуск пакетов в релиз к репозиторию пакетов.
- baright - Отобразить информацию об авторских правах BlackArch.
- basign - Подпись пакетов.
- basign-key - Подпись ключа.
- blackman - Ведет себя как pacman но собирается из git (не путать с nrz's Blackman).
- check-groups - Проверка групп.
- checkpkgs - Проверить пакеты на наличие ошибок.
- conflicts - Проверить наличие конфликтов файлов.
- dbmod - Изменение базы данных пакета.
- depth-list - Создать список, отсортированный по глубине зависимостей.
- deptree - Создать дерево зависимостей, перечисляя пакеты предоставляемые только blackarch.
- get-blackarch-deps - Получить список зависимостей для пакета.
- get-official - Получить официальные пакеты для релиза.
- list-loose-packages - Список пакетов, которые не входят в группы, и не зависят от других пакетов.
- list-needed - Список недостающих зависимостей.
- list-removed - Список пакетов, которые находятся в репозитории пакетов, но не в git.
- list-tools - Список инструментов.
- outdated - Поиск пакетов, которые устарели в репозитории пакетов относительно git репозитория.
- pkgmod - Изменение сборки пакета.
- pkgrel - Инкремент pkgrel в пакете.
- prep - Очистка стиля PKGBUILD файла и поиск ошибок.
- sitesync - Синхронизация между локальной копией репозитория пакетов и удаленной копией.
- size-hunt - Поиски больших пакетов.
- source-backup - Резервные копии исходных файлов.

3.4 Вклад в репозиторий

В этом разделе показано, как внести вклад в проект BlackArch Linux. Мы принимаем pull requests всех размеров, от крошечных изменений до новых пакетов.
За помощью, предложениями или вопросами вы можете связаться с нами.

Все желающие могут внести свой вклад. Все вклады приветствуются.

3.4.1 Необходимые туториалы

Прочтите следующие туториалы, прежде чем вносить свой вклад:

- Arch Packaging Standards
- Creating Packages
- PKGBUILD
- Makepkg

3.4.2 Шаги по содействию

Чтобы внести изменения в проект BlackArch Linux, выполните следующие действия:

1. Сделать форк репозитория https://github.com/BlackArch/blackarch
2. Hack the necessary files, (e.g. PKGBUILD, .patch files, etc).
3. Закомитьте свои зменения.
4. Отправьте свои изменения в главный репозиторий.
5. Ask us to merge in your changes, предпочтительно через pull request.

3.4.3 Пример

Следующий пример демонстрирует отправку нового пакета в проект BlackArch. Мы используем yaourt (вы также можете использовать pacaur) для извлечения уже существующего PKGBUILD файла для **nfsshell** из AUR и настройте его в соответствии с нашими потребностями.

3.4.3.1 Извлечение PKGBUILD

Получить *PKGBUILD* файл используя yaourt или pacaur:

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

3.4.3.2 Очистка PKGBUILD

Очистка *PKGBUILD* файла и сохранение времени:

```
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 Настройка PKGBUILD

Настройка *PKGBUILD* файл:

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

3.4.3.4 Сборка пакета

Сборка пакета:

```
==> 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 Установка и тестирование пакета

Установка и тестирование пакета

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

3.4.3.6 Add, commit и push пакета

Add, commit и push пакета

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

3.4.3.7 Создать pull request

Создайте pull request на github.com

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

3.4.3.8 Adding a remote for upstream

A smart thing to do if you're working upstream and on a fork is to pull your own fork and add the main ba repo as a remote

```
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
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)
```

By default, git should push straight to origin, but make sure your git config is configured correctly. This won't be an issue unless you have commit rights as you won't be able to push upstream without them.

If you do have the ability to commit, you might have more success using *git@github.com:blackarch/blackarch.git* but it's up to you.

3.4.4 Requests

1. Don't add **Maintainer** or **Contributor** comments to *PKGBUILD* files. Add maintainer and contributor names to the AUTHORS section of BlackArch guide.
2. For the sake of consistency, please follow the general style of the other *PKGBUILD* files in the repo and use two-space indentation.

3.4.5 Общие советы

namcap может проверять пакеты на наличие ошибок.

Глава 4

Руководство по инструментам

Coming soon...

4.1 Coming Soon

Coming soon...