

OpenBSD

Имидж — ничто, безопасность — все!



Максим Гришков

Тот факт, что в мире нет совершенного программного обеспечения, не является большим секретом. На момент релиза программный код любой операционной системы содержит ошибки, которые влияют на ее производительность и надежность. Вследствии этого одной из важных задач системного администратора является поддержание операционной системы в актуальном, с точки зрения безопасности и надежности, состоянии путем наложения патчей и обновлений.

Основная часть

Несмотря на то что целью известного проекта OpenBSD является разработка операционной системы (ОС), которая характеризуется высокими показателями чистоты кода, безопасности, стандартизации и портируемости и, как свидетельствует статисти-

ка, в этом направлении команде разработчиков удалось достигнуть значительных успехов, через некоторое время после выхода очередного релиза данной ОС появляется информация о выявленных потенциальных уязвимостях. Эту информацию можно почерпнуть на страничке «Ошибок»

(<http://www.openbsd.org/errata.html>) официального сайта проекта, где списки выявленных ошибок сгруппированы согласно релизам, в которых они были выявлены: например, [errata43.html](#) — ошибки, выявленные после выхода релиза 4.3. Страничка также содержит краткое описание, характеристику (бе-

зопасность или надежность) и ссылку на заплатку (патч), с помощью которой можно избавиться от уязвимости. На момент написания статьи страничка релиза 4.3 содержала 5 записей.

Согласно терминологии, используемой внутри проекта, ваша система может находиться в одном из 3 возможных состояний: релиз (-release), стабильное (-stable) или текущее (-current). Разница между этими состояниями заключается в том, что релиз – это версия, выпускаемая проектом согласно графику релизов каждые 6 месяцев на CD, стабильная система – это релиз с наложенными на него патчами на текущий момент времени. Патчи представляют собой порции исходного кода, содержащие очень важные исправления, сделанные и опробованные в текущей ветке. Подразумевается, что для того, чтобы обладать наилучшими показателями стабильности и надежности ОС, OpenBSD должна быть приведена к стабильному состоянию. При этом следует учитывать, что поддержка новых устройств и кардинально новые свойства, реализованные в текущей ветке не попадают в стабильную. Текущая ветка в свою очередь предназначена в основном для разработчиков и является полигоном для испытания нововведений. Ни один здравомыслящий системный администратор не станет обновлять ОС ответственного сервера до текущего состояния.

Существуют два способа приведения системы к стабильному состоянию. Первый – это наложение патчей вручную, второй – использование стабильной (патчевой) ветки исходных кодов системы, которая содержит все исправления для обновления системы. Следует учитывать тот факт, что исходный код стабильной ветки кроме обновлений, вышедших в виде патчей, также содержит очевидные и простые исправления, которые были признаны недостойными включения в список патчей на страничке «Ошибок». Разница между способами заключается в том, что в случае наложения патчей будут закрыты только те ошибки, которые были признаны критическими, в следствие этого существенно сокращается время, затрачиваемое на исправление ошибок (даже в случае пересборки ядра). В свою очередь обновление

исходных кодов с помощью стабильной (патчевой) ветки с последующей пересборкой ядра и встроенного программного обеспечения (userland) – более длительный, но и более автоматизированный (если синхронизацию переложить на могучие плечи cron) процесс. Он также позволяет исправить незакрытые патчами ошибки.

Перед тем как приступить непосредственно к наложению патчей или синхронизации исходного кода, необходимо его (код) загрузить. Самый простой способ загрузить его по ftp (ftp://ftp.openbsd.org/pub/OpenBSD/4.3) или скопировать с CD (обычно это третий CD в стандартной поставке). Исходный код, хранящийся на ftp-сервере, разделен на несколько файлов:

- **src.tar.gz** – содержит исходный код набора встроенного ПО;
- **sys.tar.gz** – содержит исходный код ядра;
- **xenocara.tar.gz** – содержит исходный код модифицированной системы X-Window X.org или Xenocara, в интерпретации OpenBSD).

После загрузки необходимо разархивировать src.tar.gz и sys.tar.gz в директорию /usr/src, а xenocara.tar.gz в – /usr.

```
$ cd /usr/src
# tar -xvzf /usr/sys.tar.gz
# tar -xvzf /usr/src.tar.gz
$ cd /usr
# tar -xvzf /usr/xenocara.tar.gz
```

После извлечения исходников можно переходить к делу. Начнем с патчей.

Накладываем заплатки

Патчи можно скачать как в виде отдельных файлов (001_openssh.patch и т. д.), так и единым архивом (4.3.tar.gz). Сделать это можно или с упомянутой выше странички errata43.html или непосредственно по ftp (ftp://ftp.openbsd.org/pub/OpenBSD/4.3).

После скачивания необходимо распаковать архив, содержащий патчи в какую-нибудь директорию, например /usr/src/patches.

```
# mkdir /usr/src/patches
$ cd /usr/src/patches
# tar -xvzf /usr/4.3.tar.gz

4.3/common/001_openssh.patch
4.3/common/002_openssh2.patch
```

```
4.3/common/003_xorg.patch
4.3/common/004_bind.patch
4.3/common/005_pcb.patch
```

Как видно из вывода архиватора, в архиве находятся 5 патчей. Все они расположены в каталоге common, что свидетельствует об их платформо-независимости. Назначение патча легко узнать, используя команду head.

```
$ cd /usr/src/patches/4.3/common
$ head -n 11 001_openssh.patch
```

```
Apply by doing:
cd /usr/src
patch -p0 < 001_openssh.patch

And then rebuild and install file:
cd usr.bin/ssh
make obj
make cleandir
make depend
make
make install
```

Из вывода следует, что патч предназначен для исправления исходного кода элемента встроенного ПО (в частности OpenSSH) и потребует пересборки только этого элемента. Тут же приведены инструкции по наложению патча (первый абзац) и последующей пересборке приложения (второй абзац).

```
$ head 003_xorg.patch
```

```
Apply by doing:
cd /usr/xenocara
#Assuming Xenocara is in /usr/xenocara
patch -p0 < 003_xorg.patch

And then rebuild and install the X server:
cd xserver
make -f Makefile.bsd-wrapper build

Index: xserver/Xext/security.c
```

Этот патч предназначен для исправления исходного кода Xenocara, он также потребует пересборки только этого элемента.

```
$ head 005_pcb.patch
```

```
Apply by doing:

cd /usr/src

patch -p0 < 005_pcb.patch

Then build and install a new kernel.
```

Комментарий «Then build and install a new kernel» указывает на то, что патч предназначен для исправления исходного кода ядра и потребует его дальнейшей пересборки (см. далее).

На этом все премудрости работы с патчами, собственно, и заканчиваются. После просмотра заголовка патча (количество строк, выводимых командой `head`, может быть увеличено при помощи параметра `-n`), остается только, следуя полученным инструкциям, пропатчить (команда `patch`) и пересобрать приложение или ядро.

Следуем стабильной (патчевой) ветке исходного кода

Поскольку разработчики OpenBSD используют для управления исходным кодом систему контроля версий CVS, её можно использовать для получения и обновления локальной копии дерева исходных кодов, находящейся в директории `/usr/src`. Для начала необходимо выбрать один из анонимных CVS (AnonCVS) серверов проекта. Это можно сделать на страничке <http://www.openbsd.org/anoncv.html> официального сайта. После выбора сервера можно приступить к обновлению дерева исходных кодов.

```
$ cd /usr/src
# export CVSROOT=anoncv@anoncv.de.openbsd.org:/cvs
# cvs -d$CVSROOT up -rOPENBSD_4_3 -Pd
```

В данном примере был использован анонимный CVS-сервер `anoncv.de.openbsd.org`. Параметр `up` команды `cvs` обеспечивает обновление локальной копии дерева исходных кодов. Флаг `-r` позволяет выбрать версию (ревизию) исходного кода для обновления. Флаги `-Pd` позволяют `cvs` во время обновления удалять пустые директории дерева и создавать новые (без этих флагов обновление закончится неудачей).

Команду `cvs` также можно использовать для получения дерева исходных кодов и дерева портов, указав опцию `checkout` и ветвь, которую нужно скачать (`src`, `hepokeya`, `ports`). Пример для скачивания исходных кодов ядра и встроенного ПО.

```
$ cd /usr
# export CVSROOT=anoncv@anoncv.de.openbsd.org:/cvs
# cvs -d$CVSROOT checkout -rOPENBSD_4_3 -P src
```

После того как наложены заплатки, либо дерево исходных кодов синхронизировано со стабильной (патчевой) веткой, можно пересобирать ядро и встроенное ПО (`userland`).

Пересобираем ядро и встроенное ПО

Сборка ядра происходит с параметрами, которые хранятся в файле конфигурации ядра. Обычно этот файл находится в директории `conf` по адресу `/usr/src/sys/arch/<платформа>/conf`. Файл конфигурации ядра, поставляемый с дистрибутивом, по умолчанию носит название `GENERIC`. Разработчики OpenBSD уверяют, что этот конфигурационный файл подойдет для большинства случаев, за исключением систем с нестандартным оборудованием, очень маленьким объемом оперативной памяти, а также в случае, если вас интересует активация экспериментальных или выключенных по умолчанию функций.

Перед сборкой ядра необходимо сделать резервную копию работающего в данный момент ядра:

```
# cp /bsd /bsd.old
```

Непосредственно процесс пересборки ядра предельно прост:

```
$ cd /usr/src/sys/arch/i386/conf
# config GENERIC
# cd ../compile/GENERIC
# make clean && make depend && make
# make install
```

Правда, стоит отметить, что он длится довольно долго: от 30-40 минут на современной технике до суток на раритетном железе (которое частенько используется для установки OpenBSD). После завершения сборки ядра необходимо перезагрузить компьютер для загрузки нового ядра. Если по какой-то причине собранное ядро не может быть загружено, процесс загрузки будет остановлен и появится приглашение загрузчика: `<boot>`.

В этом случае необходимо загрузить копию старого рабочего ядра, исправить ошибки в конфигурационном файле и повторить процедуру.

```
boot> bsd.old
```


Если пересборка ядра завершилась успешно, можно приступить к пересборке встроенного ПО. Следует быть особо внимательным при выполнении команды `rm` с флагами `-rf` от пользователя `root`, ошибочно указав директорию (не `/usr/obj/*`), можно в лучшем случае лишиться копии дерева исходных кодов, в худшем – системы.

```
# rm -rf /usr/obj/*
$ cd /usr/src
# make obj
# cd /usr/src/etc && env DESTDIR=/ make distrib-dirs
# cd /usr/src
# make build
```

По сравнению со сборкой ядра процесс сборки встроенного ПО еще более длительный (на слабой машине может длиться более суток). После завершения пересборки встроенного ПО систему можно считать обновленной до стабильного состояния.

Заключение

Команда разработчиков ОС OpenBSD прилагает много усилий для своевременного и оперативного исправления выявленных в процессе эксплуатации ПО ошибок. Разрабатываемая ими система включает достаточный и подробно документированный инструментарий для поддержания безопасности на высоком уровне.

Обновлять или патчить, каждый решает сам, но с тем, что безопасности системы необходимо уделять особое внимание, я думаю, согласится каждый. 

1. OpenBSD 4.3 release errata & patch list – <http://www.openbsd.org/errata43.html>.
2. Documentation and Frequently Asked Questions – <http://www.openbsd.org/faq>.
3. Anonymous CVS – <http://www.openbsd.org/anoncv.html>.
4. Following -stable (the «Patch branch») – <http://www.openbsd.org/stable.html>.
5. Patching and Kernel Building – <http://www.openbsd101.com/patching.html>.