Portupgrade - корректное обновление установленого софта

Опубликовано muff в Пнд, 2012-01-09 18:38

Если воспользоваться возможностями утилиты portaudit **[1]**, то возникает вопрос о том, как корректно обновить установленный софт, чтобы устанить уязвимости. Надеюсь, что за актуальностью системы портов вы следите, иначе необходимо обновить порты вручную.

Итак, выполним установку **portupgrade** из системы портов:

```
# cd /usr/ports/ports-mgmt/portupgrade && make install clean && rehash
```

Поскольку на серверах не использую графический интерфейс, то следует последовать совету и добавить в **make.conf** инструкцию сборки - без поддержки X-сервера:

```
# echo 'WITHOUT_X11=yes' >> /etc/make.conf
```

Обновлять софт не так и сложно. Синтаксис запуска **portupgrade** следующий - **portupgrade <имя_порта>**. Однако, не стоит забывать о том, что во время обновления сервис будет остановлен, к тому же могут обнаружиться и другие "неожиданости", поэтому обновление рекомендую проводить в часы, когда простой сервиса будет "незаметен". Также рекомендую перед каждым обновлением сохранять копию конфигурационных файлов.

Ну и не стоит забывать о золотом правиле: "Работает - не трогай". На наиболее критичных серверах обновление софта выполняю только согласно вывода **portaudit**, устраняя уязвимости. Постоянно обновляться - себе дороже, поскольку службы на этих серверах должны работать 24 часа в сутки, 7 дней в неделю.

Ну а для некритичных серверов рекомендую обновлять весь софт до последней версии. Посмотрим, какие установленные порты успели "устареть":

```
# pkg_version -v | grep "need"


nmap-5.61.t2                        <   needs updating (port has 5.61.t4)pcre-8.21
                        <   needs updating (port has 8.21_1)ru-freebsd-doc-20110110
             <   needs updating (port has 20111014)
```

Выполним обновление на примере **nmap**.

Проверим, какая версия **nmap** установлена в даный момент:

```
# pkg_info | grep nmap
nmap-5.61.t2       Port scanning utility for large networks
```

Выполним процедуру обновления с помощью **portupgrade**:

```
# portupgrade nmap-5.61.t2
```

По завершению обновления проверяем текущую версию:

```
# pkg_info | grep nmap
 nmap-5.61.t4       Port scanning utility for large networks
```

Что ж... Успешно обновились до новой версии. Аналогично необходимо поступить с **pcre-8.21** и **ru-freebsd-doc-20110110**.

Также можно выполнить "ход конем", запустить обновление всех установленных портов:

```
# portupgrade -aR
```

Используемые ключи:

- **a** - обновлять все порты
- **R** - проверять и обновлять приложения, от которых зависит порт (обновлять рекурсивно)

Также, во время обновлений можно увидеть сообщение системы, в котором предлагается выполнить команду **pkgdb -F**. Эта команда выполнит обновление базы данных пакетов, которая хранится в **/var/db/pkg/pkgdb.db**. Запуск команды проводит коррекцию в интерактивном режиме, запрашивая у пользователя, как необходимо поступать в определенных ситуациях.

Однако стоит иметь ввиду, что **portupgrade** - это целый набор утилит:

- **portupgrade** – одноименная утилита, средство автоматического обновления установленных программ
- **portinstall** – утилита установки новых программ, присутствующих в системе портов **FreeBSD**
- **portversion** – утилита показывающая установленный софт и наличие обновлений для него.
- **portsclean** – утилита очистки мусора остающигося после сборки портов
- **portsdb** – утилита создающая базу программ, присутствующих в портах **FreeBSD**
- **ports_glob** – предназначена для поиска по базе портов
- **portcvsweb** – браузер для **CVSWeb**
- **pkgdu** – утилита, предназначенная для подсчета дискового пространства занятого установленными программами
- **pkgdb** – еще одна утилита для управления базой портов
- **pkg_which** – предназначен для поиска по базе портов
- **pkg_sort** – утилита сортирует порты в соответствии с зависимостями
- **pkg_fetch** – утилита для скачки собранных(бинарных) пакетов
- **pkg_deinstall** – средство для массового удаления установленного софта

Теперь более подробно о некоторых инструментах.

Portupgrade

Служит для обновления установленного ПО до более новой версии. Способы использования **portupgrade** уже описывались, поэтому не будем останавливаться на этом.

Portinstall

С помощью **portinstall** можно устанавливать новый софт из портов.Пример использования:

```
# portinstall /usr/ports/misc/mc-light
```

Portversion

Утилита **portversion** служит для сравнения версий установленных портов с версиями, которые находятся в портах. Пример использования:

```
# portversion | grep ipcalc
ipcalc                    =
```

Результат "**=**", означает, что пакет в обновлении не нуждается. Результат "**>**" означает что установлен пакет новее, чем присутствует в базе портов, а "**<**" – означает что пакет устарел и нуждается в обновлении.

Исходя из этого, проверить, какие порты нуждаются в обновлении, можно такой командой:

```
# portversion | grep \<
```

Portsclean

Утилита **portsclean** служит для чистки дерева портов от "мусора", который остается после неудачных сборок.

Для очистки каталогов, которые используется для сборки, выполняем такую команду:

```
# portsclean -C
```

Для очистки **/usr/ports/distfiles** от устаревших исходников необходимо выполнить:

```
# portsclean -D
```

Portsdb

Для создания файла базы данных **INDEX.db** в **/usr/ports** запускаем **portsdb** с такими ключами:

```
# portsdb -Uu
```

Просмотреть содержимое **INDEX.db** обычным просмотрщиком не получится, поскольку это не текстовый ASCII файл. Однако некоторые утилиты portupgrade используют эту базу данных. **Btree** - алгоритм базы данных, специально предназначенный для просмотра большого количества данных. Многие утилиты **portupgrade** быстрее и эффективнее "родных", благодаря **Btree**.

Ports_glob

Утилиту **ports_glob** можно использовать для поиска месторасположения порта относительно **/usr/ports**. Пример:

```
# ports_glob unzip
 archivers/unzip
```

Portcvsweb

Portcvsweb - псевдобраузер. Для запуска воспользуйтесь:

```
# portcvsweb -F
```

Pkgdu

Утилита **pkgdu**, как уже оговаривалось, служит для подсчета места, занимаемого програмным обеспечением. Например, посмотрим, сколько занимает утилита **iperf** (в килобайтах):

```
# pkgdu -k iperf
 112 iperf-2.0.5
```

Для обзора возможностей необходимо воспользоваться ключем **-h**.

Pkgdb

Утилита **pkgdb** служит для исправления зависимостей. Для запуска необходимо использовать ключ **-F**:

```
# pkgdb -F
```

Pkg_deinstall

Утилита **pkg_deinstall** служит для удаления установленных портов. Используя ключ **-a** можно удалить все установленные порты, оставив систему "чистой".

Источник (получено *2026-11-23 05:52*):
http://muff.kiev.ua/content/portupgrade-korrektnoe-obnovlenie-ustanovlenogo-softa

Ссылки:
[1] http://muff.kiev.ua/content/portaudit-monitoring-uyazvimostei-v-sisteme