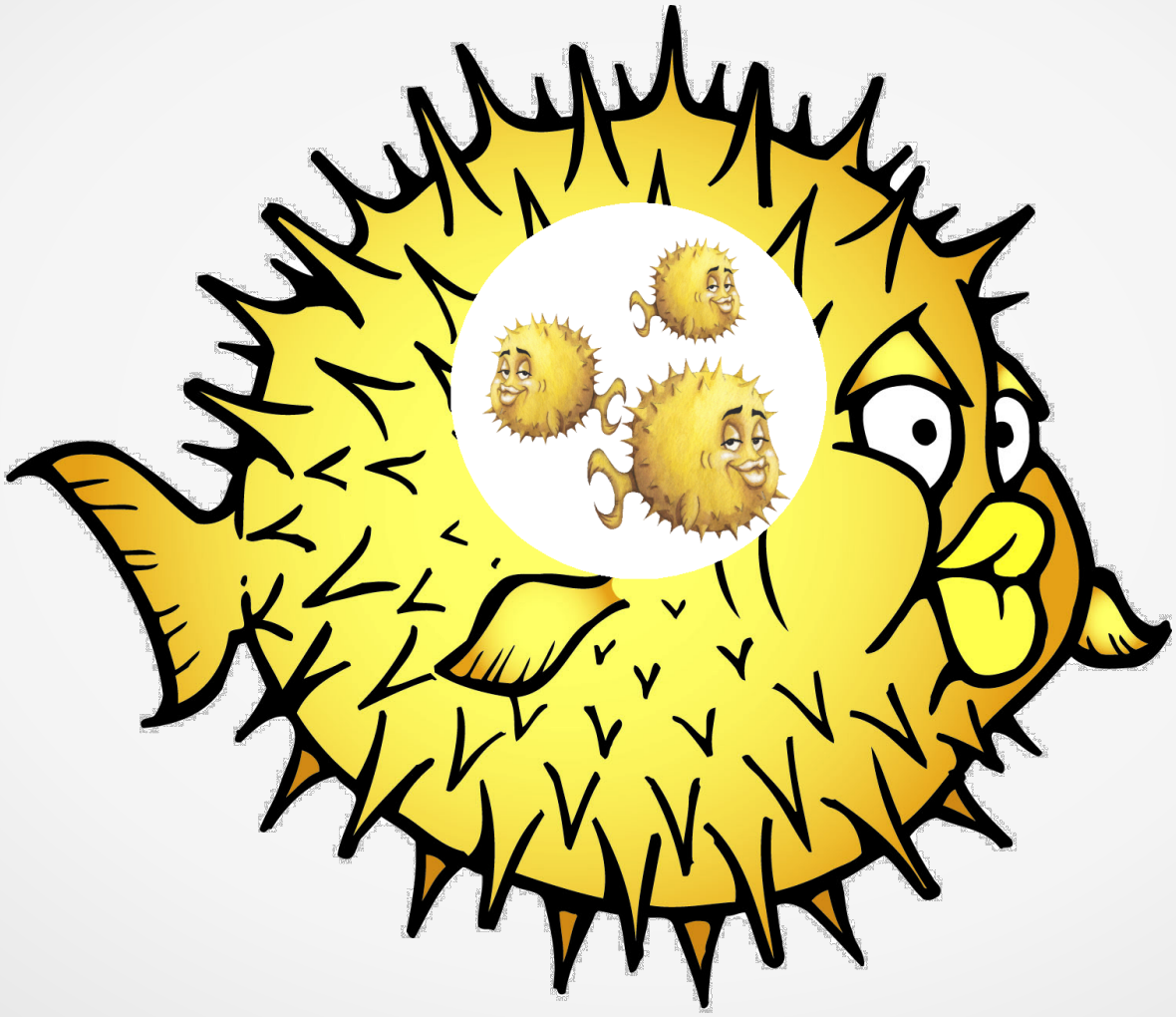
OpenBSD 5.8 & 5.9

То, что вы боялись узнать об OpenBSD, но хотели спросить

Вадим Жуков AKA zhuk@openbsd.org, Москва-Раков, 2016
LVEE Winter 2016

Виртуализация в OpenBSD



Виртуализация в OpenBSD

- Что было:
  - устойчивая работа в KVM, VMware, Hyper-V
  - VirtualBox не совсем адекватен
  - Xen — всё грустно
  - LDOM (sparc64) — полная поддержка

Виртуализация в OpenBSD

- Что появилось:
  - собственный гипервизор: vmm(4) + vmd(8)
    - простой синтаксис:

```
sets="/storage/OpenBSD/snapshots/amd64/"
vm "puffy.lvee.domain" {
        memory 512M
        kernel $sets "bsd.rd"
        disk "/storage/vm/puffy.img"
        disk $sets "install59.fs"
        interfaces 1
}
```

Виртуализация в OpenBSD

- Что появилось:
  - собственный гипервизор: vmm(4) + vmd(8)
    - простой синтаксис
    - архитектуры i386 и amd64
    - в 5.9 умеет грузить только OpenBSD, остальное — в перспективе

Виртуализация в OpenBSD

- Что появилось:
  - собственный гипервизор: vmm(4) + vmd(8)
  - поддержка Xen
    - (почти) полный набор виртуальных устройств
    - Xen очень требователен к порядку инициализации

Новое старое

Новое старое

- file(1)
  - реализован Nicholas Marriott, автором tmux и fdm
  - реализовано разделение привилегий по процессам
  - конечный код ~3700 строк

Новое старое

- file(1)
- doas(1)
  - sudo — это хорошо!
  - но для базовой системы он слишком жирный
  - Ted Unangst…
  - Todd Miller, сопровождающий sudo, не обиделся :)

UTF-8

1100 0011

UTF-8

- Что такое UTF-8?
  - `char` vs. `wchar_t`
  - `010011010`
  - `101001010 01001010`
  - `100011010 10010110 10101010 00101010`

UTF-8

- Что такое UTF-8?
  - `100011010 10010110 10101010 00101010`
- Проблемы:
  - ломается логика «один байт — один символ на экране»
  - невидимые символы
  - суррогатные пары

UTF-8

- Что такое UTF-8?
  - `100011010 10010110 10101010 00101010`
- Проблемы:
  - ломается логика «один байт — один символ на экране»
  - невидимые символы
  - суррогатные пары
- Требуется переделка всех программ, считающих колонки на терминале и символы в файлах

UTF-8

- Что такое UTF-8?
  - `100011010 10010110 10101010 00101010`
- Проблемы
- Требуется переделка всех программ, считающих колонки на терминале и символы в файлах
- Возможно скрытие или искажение данных на терминалах
  - особенно на семибитных
- Программы для X обычно уже адаптированы

UTF-8

- Решение:
  - для каждой программы все новые функции добавляются в отдельный файл utf8.c
  - вместо превращения строк полностью в `wchar_t*`, работаем по возможности с пробеганием по `char*`
  - типовые функции:
    - int mbsprint(const char *mbs, int print)
    - int mbsavis(char** outp, const char *mbs)

UTF-8

- Решение:
  - для каждой программы все новые функции добавляются в отдельный файл utf8.c
- Главный толчок произошёл на u2k15, организованном stsp@ и uwe@
- Большая часть текущей работы проделывается schwarze@ (автором mandoc) при поддержке OpenBSD Foundation

pledge(2)

pledge(2)

Disclaimer:

1) Ваш покорный слуга не принимал участия в разработке
2) Картинка с предыдущего слайда честно стырена из презентации Teo де Раадта: http://www.openbsd.org/papers/hackfest2015-pledge/

pledge(2)

- Старая цель — ограничиваем системные вызовы:
  - systrace: по списку вызовов и их аргументов
    https://www.usenix.org/legacy/event/sec03/tech/full_papers/provos/provos_html/
  - SELinux: посредством MAC

pledge(2)

- Старая цель — ограничиваем системные вызовы
- Новая идея: вместо составления подробного списка системных вызовов формируем семантические группы:
  - `rpath`
  - `stdio`
  - `dns`
  - и так далее

pledge(2)

- Старая цель — ограничиваем системные вызовы
- Новая идея: вместо составления подробного списка системных вызовов формируем семантические группы
- Семантика не диктуется сверху:
  - По мере внедрения pledge(2) в базовую систему пополнялось понимание потребностей ПО
  - Было больно

pledge(2)

- Старая цель — ограничиваем системные вызовы
- Новая идея: вместо составления подробного списка системных вызовов формируем семантические группы
- Семантика не диктуется сверху
- Нельзя отключить:
  - `echo 0 >/selinux/enforce`
  - «Use the SetProcessMitigationPolicy function to enable or disable security mitigation programmatically»: https://msdn.microsoft.com/ru-ru/library/windows/desktop/hh769088%28v=vs.85%29.aspx

pledge(2)

- Около 90% программ в базовой системе использует pledge(2)
- Пропатчен ряд портов:
  - архиваторы
  - mutt
  - Chromium (в том числе Iridium)

Спасибо!

- Дмитрию Костюку, Павлу Чеботарёву, Надежде Шарпио и другим вовлечённым лицам — за организацию LVEE
- Свете, Мише и Кириллу — за отличную компанию
- Тео, Марку, Стюарту, Лэндри, Антуану и многим другим — за свет в конце тоннеля
- ParaType за шрифты PT Sans
- Всем хорошим людям — за то, что они есть!

Спасибо!

- Дмитрию Костюку, Павлу Чеботарёву, Надежде Шарпио и другим вовлечённым лицам — за организацию LVEE
- Свете, Мише и Кириллу — за отличную компанию
- Тео, Марку, Стюарту, Лэндри, Антуану и многим другим — за свет в конце тоннеля
- ParaType за шрифты PT Sans
- Всем хорошим людям — за то, что они есть!
- Тому, кого нельзя называть — за то, что (пока) ещё не разбомбил Воронеж