

KyivBSD'13

FreeBSD as FirmWare

Зачем вообще ПО

- ПО как инструмент, а не как предмет культа
- ПО приносит пользу

Виды пользы от ПО

- Уменьшение временных трудозатрат
- Увеличение отдачи
- Повышение качества
- Прогнозируемость возможностей
- Повторяемость результата

Виды вреда от ПО

- Масштабируемость

`delete from very_important_table where id = 1;`

- Неявность зловредности

Много кода может почти всегда делать что ожидается, но не всегда

- Потеря времени

`ipfw flush &&` Такси! Такси!

- Потеря ресурсов

Например, начальство вам показывает формулу

$Loss = downtime_minutes * 50000\$$

- ТВС

Виды оптимизации работы ПО

- Сразу писать “чистый” код
- Тщательная разработка SRS
- Авто-тестирование (желательно на основе SRS)
- Повторно используемые элементы (библиотеки, код, и т.п.)
- Беспереывная интеграция

Чем может похвастать FreeBSD?

- ...

Типичный жизненный цикл ОС

- Взяли свежий установочный носитель (/src тоже “носитель”)
- Установили на жесткий диск
- Наконфигурировали всякое
- Наставили всякий софт
- Опять наконфигурировали, теперь софт
- Молимся на винт (чтоб не умер)

Жизненный цикл после инцидента

- Взяли свежий установочный носитель
- Установили на жесткий диск
- Переносим конфигурацию
- Ставится софт
- Конфигурируется софт

Что такое FreeBSD as FirmWare?

- Это почти nanoBSD. Но!
- Имадж всегда RO.
- У имаджа всегда будет чексум.
- Конфиги выносятся вне, local выносятся вне.
- Забывается навсегда установка как страшный сон
- Всегда можно пощупать, чем отличается ваши конфиги от штатных (/etc-rw)

За счёт чего?

- /boot/loader.conf
- init_script="/etc/find-rwfs.sh"

•/etc/find-rwfs.sh

- Try find `"/dev/gpt/etcfs"`,
mount -> `/etc-rw`
mount union `/etc-rw -> /etc`
- Try find `"/dev/gpt/varfs"`, `"/dev/gpt/localfs"`
- Try find `/usr/local/etc-md.img` (mdconfig it, and if there appeared `"/dev/gpt/etcfs"` — use it.)

Какие ещё нюансы?

- `mount -t unionfs /usr/local/tmp /tmp`
- `mount -t unionfs /usr/local/root-rw /root`
- `/var` тоже должен быть где-то там.
- `/boot/modules = RO`

Типичный сетап в VM

- Disk1 (FreeBSD Firmware) — 800Mb
- Disk2 (localfs + (etc.img)) — 10-20-30...Gb
- Disk3 (datafs) — XXX Gb

Сетап на реальном железе

- Кардридер + карта с ОС (самая старая, на 1Гб достаточно)
- Хранилище данных

Обновление ОС в VM

- VM стоп;
- Удалить disk1;
- Добавить disk1 с новым образом;
- VM старт;
- Проблемы? Вернём старую ОС в таком же режиме.

Можно ещё безопасней?

- Можно!
- Не *.vmdk, а *.iso будет ваша OS
- Как при старте, так и в рабочем режиме можно прочитать и сверить чексумму

Можно ещё классней?

- Можно :)
- Если делать localfs так же обновляемыми, и заменяемыми.
- Из опыта — подготовлены оба диска на замену 10-ке, и заменены на 9-STABLE downtime на перезагрузку.
- Для вас ОС начинает превращаться в абстракцию “revision + набор конфигов”, а софт в “список пакетов + конфиги”

Пару граблей (и плюсов):

- ОС всё-таки на диске, и вытащить его нельзя (но это скорее плюс)
- Желательно иметь запасной диск-копию (мало ли...)
- Mount -t unionfs — дважды выполнить и получаешь DoS — но проектом это не считается как Security issue :)

Вопросы?

