Администрирование Linux Физическая безопасность компьютера

Иртегов Д.В.

Новосибирский гос. Университет

2014/2024

Что можно сделать с компьютером при физическом доступе к нему?

- Да что угодно
- Украсть/сломать (DoS)
- Уничтожить данные
- Сбросить пароль администратора
- Скопировать данные
- Модифицировать данные
- Что-то дополнительное установить:
  - Аппаратное: keylogger, перехват трафика
  - Программное: да что угодно

Как от этого защититься

- Украсть, сломать, уничтожить данные
  - только физически
  - Но если есть бэкап, то это только DoS
- Слив данных, сброс пароля, изменение данных, установка ПО
  - Шифрование дисков

Прежде чем вы начнете

- Какой у вас компьютер
  - x86 или x86_64?
  - BIOS или UEFI?

BIOS vs UEFI

- Современные компьютеры x86 поддерживают два типа загрузочных ПЗУ
  - BIOS (IBM PC compatible)
  - UEFI (Unified Extensible Firmware Interface)

BIOS

- Обеспечивает совместимость с оригинальными IBM PC (1981)
- Загрузка в реальном режиме
  - режим эмуляции 8086
- Сам по себе не поддерживает разделы дисков и множественную загрузку
  - сделано через различные костыли – MBR, меню во вторичном загрузчике
- Не поддерживает загрузчики >512 байт
  - Поддержка также реализована через костыли
- Не поддерживает диски >2Тб

UEFI

- Поддержка различных процессоров
  - IA64, x86, ARM
- Защищенный режим x86
- Таблица разделов GPT (диски до 2^{70} байт)
- Поддержка FAT16/32
- SecureBoot

Поддержка EFI/UEFI

- 2000 – Intel Itanium, EFI
- 2005 – спецификация UEFI
- 2005 – Intel XScale (ARM)
- 2006 – Apple iMac (x86)
- 2008 – появление серверов x86 с поддержкой UEFI
- С 2011 года десктопные материнские платы и ноутбуки x86 начинают массово поддерживать UEFI
- 2012 – UEFI включен в требования «готово для Windows 8»

Поддержка ОС

- Windows XP x86 НЕ поддерживает UEFI
- Поддержка началась с XP x64 и Windows Vista
- В Linux/GRUB, поддержка в той или иной форме существовала с 2000 года, но не все дистрибутивы ее включали.
- Сейчас практически все дистрибутивы поддерживают UEFI

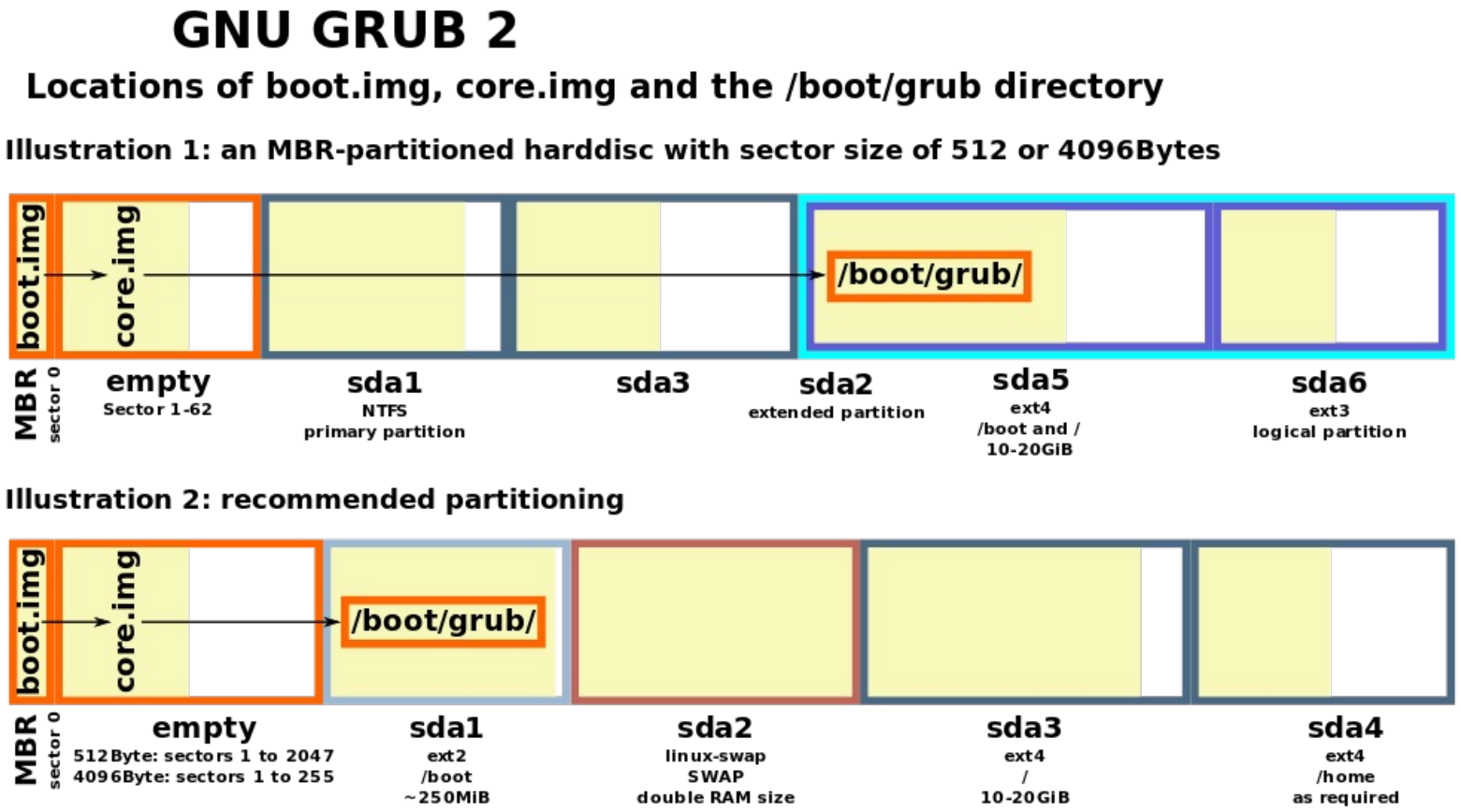
Вторичный загрузчик

- Ядро Linux представляет собой набор модулей (.ko)
- Драйверы загрузочного диска и корневой ФС – тоже модули
- Чтобы их загрузить, нужно уметь читать диск и ФС

GRUB

- GRand Unified Bootloader
- Используется для загрузки Linux, *BSD, Solaris/x86
- Может работать как бутменеджер и загружать другие ОС (chainloader)
- Читает диск через сервисы BIOS или UEFI
- Имеет собственные подгружаемые драйверы ФС (stage1.5, stage2)

Размещение GRUB



GRUB

- Поддерживает файловые системы
  - UFS, ISO9660,
  - FAT16/32
  - UFS/UFS2/FFS
  - ext2/3, ext4
  - ReiserFS, XFS, JFS
- На загрузочной ФС размещаются
  - файл конфигурации menu.lst или grub.cfg (в GRUB2 это командный файл)
  - ядро ОС (kernel)
  - образ загрузочного виртуального диска (initrd)
- Пар kernel/initrd может быть много, они задаются в grub.cfg
- Ядру можно передать параметры
- Initrd (init RAM disk) – образ специальной ФС, которая содержит модули и конфигурационные файлы, необходимые ядру на первых этапах загрузки, в т.ч. драйверы загрузочного диска и корневой ФС

Разделы и LVM

- Разделы – виртуальные диски в пределах физического диска
- Раздел занимает непрерывное пространство на диске
- BIOS/MBR показывает каждый раздел как отдельный диск
- Linux видит разделы как отдельные блочные устройства
  - /dev/sda – весь диск
  - /dev/sda1 – первый раздел
  - /dev/sda4 – четвертый раздел

Разделы MBR

- BIOS/MBR позволяют 4 primary раздела или 3 primary + extended.
- Windows может загружаться только с Primary раздела
- GRUB – откуда угодно
- Разделы MBR имеют тип (двузначное 16ричное значение)
- Желательно чтобы тип соответствовал типу ФС

Разделы UEFI (gpt)

- До 256 разделов
- Теоретически, можно создавать диски с обоими типами таблиц разделов (gpt+mbr)
- На практике, ни одна современная ОС такие таблицы не понимает
- Поэтому диск может быть разбит только одним способом
- ПЗУ материнской платы выбирает тип загрузки (BIOS или UEFI) по типу таблицы разделов

LVM

- Logical Volume Manager
- Прослойка между драйверами диска и ФС
- Позволяет объединять диски и разделы в логические диски
  - RAID0 и JBOD (объединение нескольких дисков в один)
  - RAID1 (зеркало)
  - RAID5
  - Моментальные снимки (snapshot)
  - Миграция данных между дисками
- Логический диск не обязан занимать непрерывное пространство
- Логические диски можно увеличивать и уменьшать на ходу, если ФС это поддерживает
- GRUB не умеет работать с LVM, поэтому необходимо создать обычный раздел, где будут размещаться GRUB и ядра. Корневая ФС может находиться на LVM

Типовая установка

- Небольшой раздел для grub
  - Похоже на скрытый раздел у современных Windows
- Остальной диск под LVM, разбитый на тома в соответствии с потребностями

LUKS

- Linux Unified Key Setup
- Шифрует блочные устройства 512-битным ключом.
- Ключ зашифрован паролем, предусмотрены слоты для 32 паролей (в LUKS1 только для 8)
- Пароль надо вводить при загрузке

Установка LUKS

- Удобнее всего включать при начальной установке
- В Rocky Linux есть такая кнопка в инсталляторе
- Упражнение: установить систему с включенным LUKS

Как быть с вводом пароля?

- Если это ноутбук или персональная рабочая станция, проблемы вроде бы нет
- Если это разделяемая рабочая станция, и не слишком много пользователей, можно сделать 32 пароля
- А как быть с сервером?

Как быть с сервером?

- Если у вас BIOS, никак
  - На самом деле, можно использовать сетевые сервера key escrow/KMS
  - Этот подход годится и для виртуальных машин
- UEFI+Secure boot+TPM
  - Можно записать пароль в TPM
  - и настроить его отдавать только если загружен определенный образ
  - Для VM годится только для демонстрации (содержимое «TPM» лежит в конфиге VM)

Процедура для RHeL 8

- https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/8/html/security_hardening/configuring-automated-unlocking-of-encrypted-volumes-using-policy-based-decryption_security-hardening#configuring-manual-enrollment-of-volumes-using-tpm2_configuring-automated-unlocking-of-encrypted-volumes-using-policy-based-decryption
- Лучше оставить возможность расшифровать диск по паролю
  - Если при апдейте ОС или firmware что-то пойдет не так

Что происходит под капотом

- В secure boot добавляется ваш ключ подписи, чтобы вы могли подписать бандл из бутлоадера, ядра и initrd
- В TPM добавляется пароль LUKS и правило отдавать этот пароль только если загружен подписанный вашим ключом образ
- В initrd добавляется примочка, которая достает пароль из TPM и отдает его LUKS
- Собирается бандл с новым initrd, подписывается вашим ключом и регистрируется в EFI nvram
- В процедуру обновления ядра добавляется примочка делать все это при каждом обновлении ядра или initrd