Московский физико-технический институт
(государственный университет)
Факультет радиотехники и кибернетики
Кафедра информатики и вычислительной техники

Реализация замкнутой программной среды в ОС «Эльбрус» на основе механизма цифровой подписи

Выпускная квалификационная работа бакалавра

Студент: Константинов Н.О.
Научный руководитель: к.т.н Морозов Ю.В.

Москва, 2017 г.

Введение

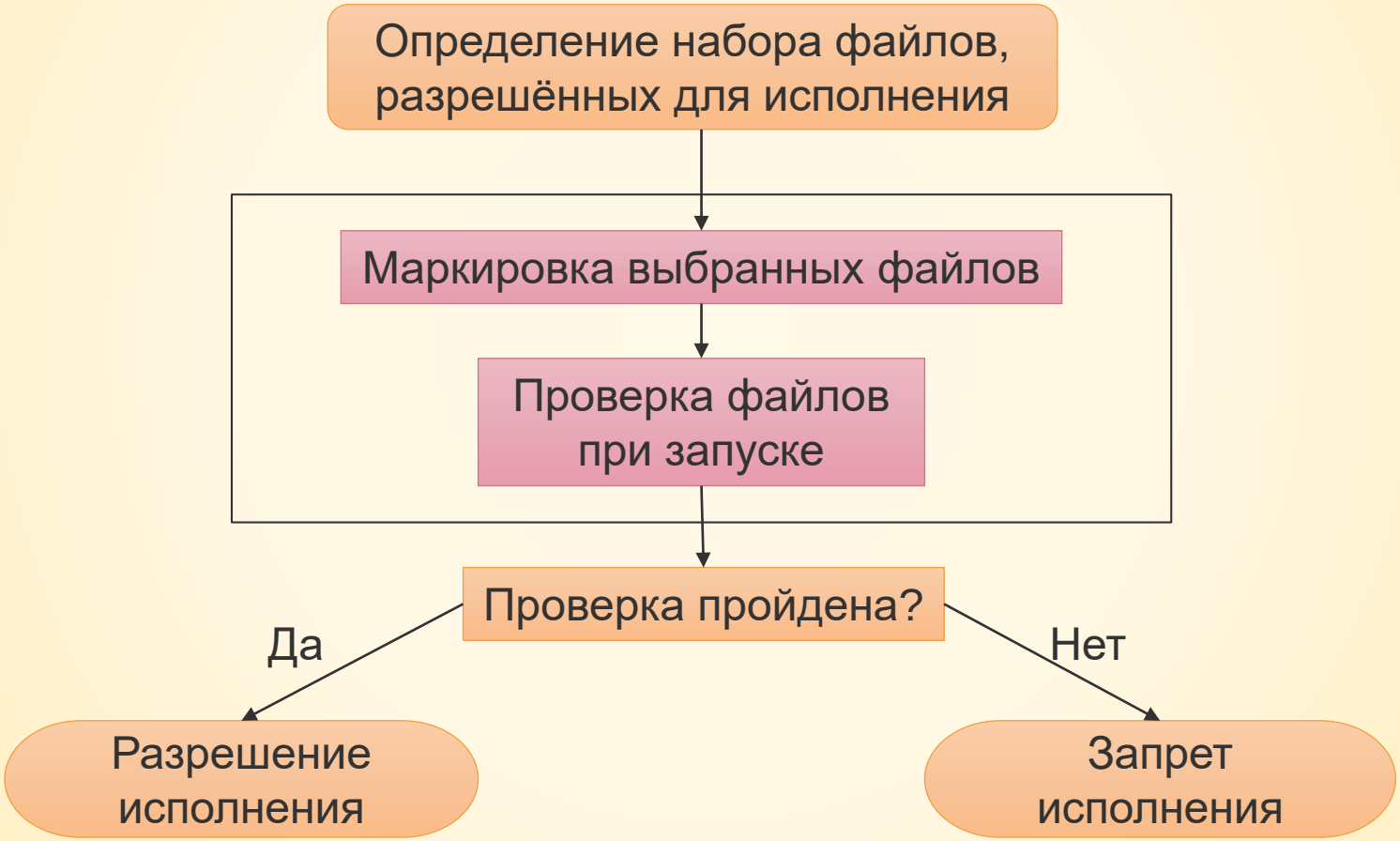
Замкнутая программная среда позволяет определить для вычислительного комплекса набор файлов, разрешённых для исполнения, обеспечивает защиту от исполнения стороннего ПО и внедрённого вредоносного кода.

Цифровая подпись – реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи.
Позволяет проверить:

- целостность – отсутствие искажения информации в электронном документе с момента формирования подписи
- авторство – принадлежность подписи владельцу сертификата ключа

В случае успешной проверки позволяет подтвердить факт подписания электронного документа (неотказуемость).

Принципиальная схема функционирования замкнутой программной среды



Цель

Создание замкнутой программной среды в ОС «Эльбрус» с помощью цифровой подписи

Задачи

- Изучение существующих средств
- Сравнительный анализ
- Реализация замкнутой программной среды в ОС «Эльбрус» на основе выбранного средства

Существующие средства создания замкнутой программной среды

	Проверка подписи в реальном времени	Типы файлов	Настройка политики проверки	Хранение подписи
DigSig	+	Бинарные ELF-файлы и библиотеки	-	В файле
IMA/EVM	+	Все файлы	+	В расширенных атрибутах

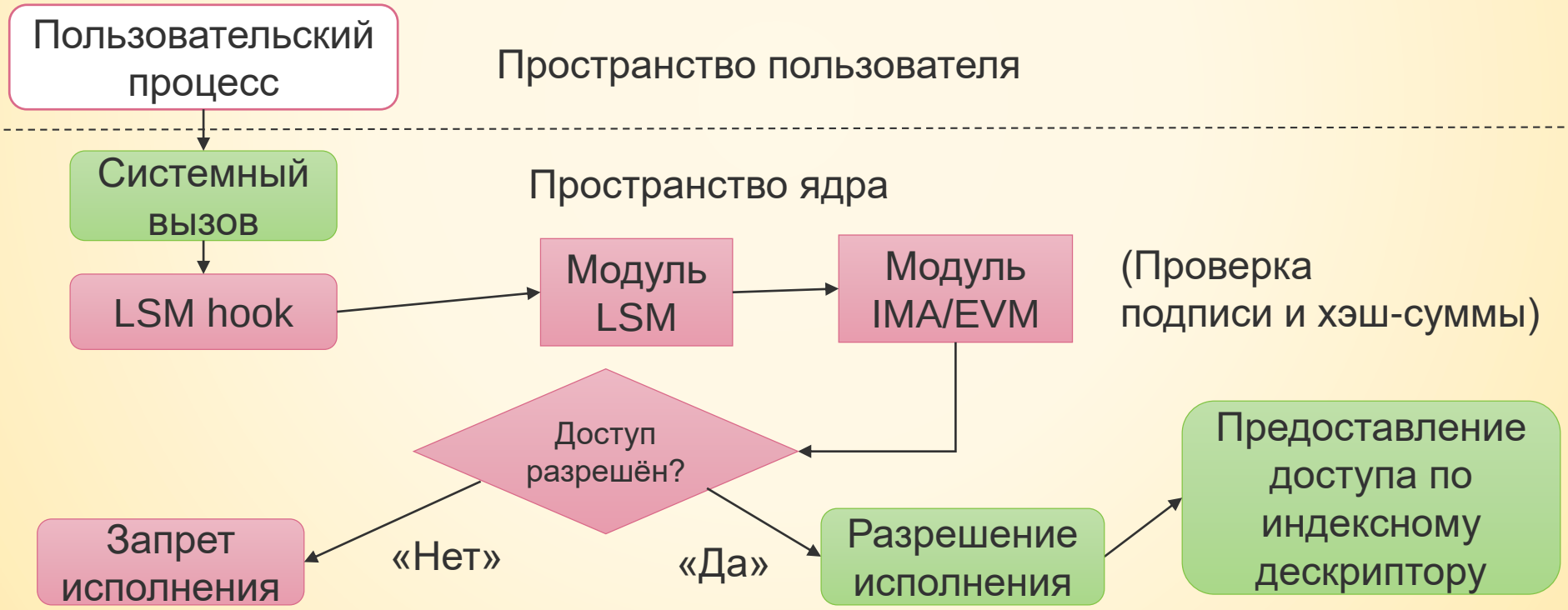
- *DigSig (Digital Signature)* – модуль ядра, позволяющий производить аутентификацию исполняемых файлов формата ELF.
- *IMA/EVM* (Integrity measurement architecture and Extended verification module) – подсистема Linux, позволяющая осуществлять контроль целостности файловой системы.

В результате проведённого анализа выбрана IMA/EVM.

Схема работы IMA/EVM

IMA/EVM основывается на работе LSM модуля.

LSM(Linux Security Module) – фреймворк, позволяющий изменить стандартное поведение программы посредством перехвата системного вызова и передачи управления модулям безопасности системы.



Создание замкнутой программной среды на основе IMA/EVM

Для создания замкнутой программной среды было реализовано:

1. Сконфигурировано и собрано ядро для поддержки IMA/EVM
2. Настроен автоматический поиск и загрузка ключей в ядро (1)
3. Настроена ОС для активации IMA/EVM в «мягком» режиме (1)
4. Запущена ОС с активированным IMA/EVM в «мягком» режиме (запуск в «мягком» режиме применяется для осуществления первичной маркировки системы)
5. Сгенерированы ключи и осуществлена загрузка ключей в ядро (2)
6. Определён перечень разрешённых файлов и произведена маркировку (3)
7. Сформирована политика проверки
8. Проверена работоспособность системы в «мягком» режиме
9. Настроена ОС для активации IMA/EVM в «жёстком» режиме (1)
10. Запущена IMA/EVM в «жёстком» режиме

Реализовано в скриптах: (1) – post-install.sh
(2) – create_key.sh
(3) – label.sh

Настройка IMA/EVM в составе ОС

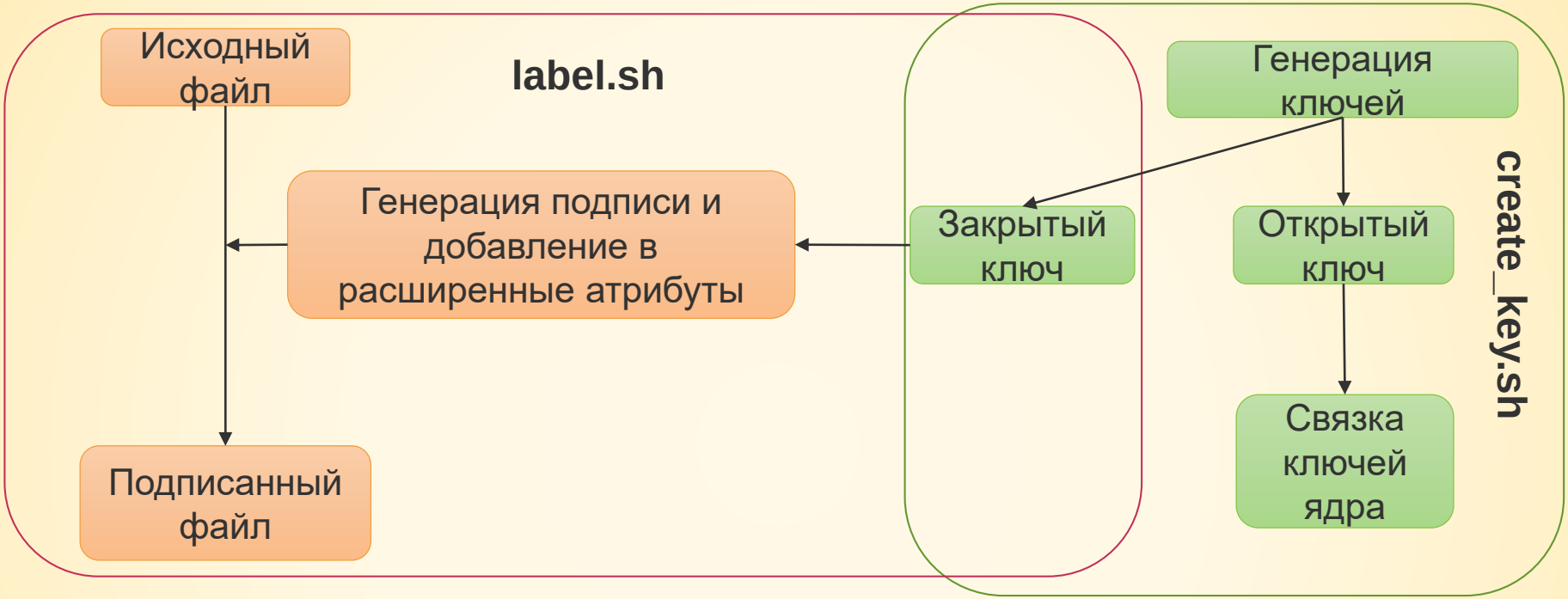
Разработан скрипт post-install.sh { fix | appraise | init }, который:

- монтирует файловую систему IMA и настраивает её автоматическое монтирование при последующих запусках

- настраивает параметры командной строки в конфигурационном файле для нужного режима работы:
  - fix – «мягкий» режим, используется на этапе настройке системы и разрешает запуск любых программ, регистрируя события несанкционированного доступа в журнале
  - appraise – «жёсткий» режим, разрешает запуск программ только при прохождении проверки

- init – настраивает автоматический поиск и загрузку ключей в ядро при старте системы при помощи пользовательской утилиты evmctl и утилиты keyctl

Генерация ключей и маркировка файлов



Скрипт create_key.sh evm { rsa | rsa_encrypted| LMK_encrypted } осуществляет создание закрытого и открытого ключей RSA и добавление открытого ключа в связку ключей ядра для осуществления последующей проверки вычисленных подписей.

Скрипт label.sh directory { sign | hash } осуществляет маркировку файлов хэш-суммой или цифровой подписью на основе закрытого ключа при помощи пользовательской утилиты evmctl и осуществляет запись вычисленной подписи в расширенные атрибуты файла.

Политика проверки

В подсистеме IMA/EVM поддерживается возможность изменения политики, позволяющая настроить условия измерения и проверки подписи файлов.

```
default policy:
        # PROC_SUPER_MAGIC
        dont_measure fsmagic=0x9fa0
        # SYSFS_MAGIC
        dont_measure fsmagic=0x62656572
        # DEBUGFS_MAGIC
        dont_measure fsmagic=0x64626720
        # TMPFS_MAGIC
        dont_measure fsmagic=0x01021994
        # SECURITYFS_MAGIC
        dont_measure fsmagic=0x73636673

        measure func=BPRM_CHECK
        measure func=FILE_MMAP mask=MAY_EXEC
        measure func=FILE_CHECK mask=MAY_READ uid=0
```

Стандартная политика IMA вычисляет подпись:

- всех исполняемых файлов при вызове bprm_check
- библиотек и файлов, отображаемых в память, при вызове file_mmap
- файлов, открытых на чтение пользователем root.

Тестирование замкнутой программной среды

Осуществляется в «жёстком» режиме работы.
Для проверки работоспособности системы производится обращение к следующим файлам:

1. с актуальной цифровой подписью в расширенных атрибутах security.ima
2. без цифровой подписи security.ima
3. с неактуальной цифровой подписью security.ima (файл был модифицирован после подписания или подписан несоответствующим ключом)

Порядок действий:

1) Файл с актуальной цифровой подписью. Разрешение исполнения.

```
root # evmctl sign --imasig ./test.sh /root/rsa_private.pem
root # ./test.sh
Hello World
```

Тестирование замкнутой программной среды
Порядок действий

2) Файл без цифровой подписи. Запрет доступа

```
root # getfattr -m . -d /etc/mtab
getfattr: Removing leading '/' from absolute path names
# file: etc/mtab
security.selinux="system_u:object_r:etc_runtime_t"
```

```
root # cat /etc/mtab
cat: /etc/mtab: Permission denied
```

```
root # dmesg | tail -1
[  256.756465] type=1800 audit(1356637858.947:53): pid=3852 uid=0 auid=0 ses=2
subj=root:sysadm_r:sysadm_t op="appraise_data" cause="missing-hash" comm="cat"
name="/etc/mtab" dev="dm-2" ino=394144 res=0
```

3) Файл с неактуальной подписью. Запрет исполнения и запрет доступа

```
root # echo "echo \"And now...\"" >> test.sh
```

```
root # ./test.sh
bash: ./test.sh: Permission denied
```

```
root # cat test.sh
cat: test.sh: Permission denied
```

```
root # dmesg | tail -2
[  643.211490] type=1800 audit(1356639603.315:37): pid=3956 uid=0 auid=0 ses=3
subj=root:sysadm_r:sysadm_t op="appraise_data" cause="invalid-signature"
comm="bash" name="/bin/test.sh" dev="dm-2" ino=131466 res=0
[  649.123917] type=1800 audit(1356639609.227:38): pid=3958 uid=0 auid=0 ses=3
subj=root:sysadm_r:sysadm_t op="appraise_data" cause="invalid-signature"
comm="cat" name="/bin/test.sh" dev="dm-2" ino=131466 res=0
```

Результаты

Реализована и протестирована замкнутая программная среда в ОС «Эльбрус» на основе механизма IMA/EVM

- Активирована подсистема IMA/EVM
- Осуществлён перенос пользовательских утилит в дистрибутив
- Разработан набор скриптов, осуществляющий:
  - настройку системы после установки IMA/EVM
  - генерацию ключей и загрузку в связку ключей ядра
  - маркировку файловой системы на основе цифровой подписи