

Дисциплина «Защита в операционных системах»

Лабораторная работа № 11

Тема: Настройка механизмов организации замкнутой программной среды. Контроль целостности комплекса средств защиты.

Цель: Изучить принципы и технологии контроля целостности данных (в том числе комплекса средств защиты – КСЗ), реализованных в ОССН. Освоить умения, необходимые для решения задач подсчёта носителей контрольных сумм файлов и оптических носителей, контроля соответствия дистрибутиву, регламентного контроля целостности и создания замкнутой программной среды..

Порядок выполнения лабораторной работы: работа выполняется самостоятельно под руководством преподавателя.

Время выполнения лабораторной работы (аудиторные часы) - 4 часа.

Оборудование и программное обеспечение: работа выполняется на ПЭВМ типа IBM PC с использованием стандартных функций ОССН Astra Linux.

1. Теоретические сведения

Автоматизированные системы в защищённом исполнении (АСЗИ) на базе ОССН должны обеспечивать функции как аудита доступа к сущностям файловой системы, так и контроля целостности (integrity) данных и содержимого исполняемых файлов. Подобный контроль позволяет с достаточной уверенностью констатировать факт отсутствия в данных, обрабатываемых системными процессами ОССН, недекларируемых для АСЗИ возможностей.

Для решения задачи контроля целостности в состав КСЗ ОССН включены средства, реализующие частные функции управления целостностью данных:

- вычисления и проверки контрольных сумм файлов и оптических дисков;
- контроля соответствия дистрибутиву;
- регламентного контроля целостности;
- создания замкнутой программной среды.

Базовым методом контроля целостности сущностей файловой системы ОССН является контроль их модификации путём вычисления контрольных сумм.

Контрольная сумма – значение, рассчитанное по набору данных путём применения определённого алгоритма и используемое для проверки целостности данных при их передаче или хранении. Она используется для быстрого сравнения двух наборов данных на эквивалентность: с очень большой вероятностью отличающиеся наборы данных будут иметь разные контрольные суммы.

Алгоритмы вычисления контрольной суммы, как правило, делятся на два вида:

✓ Алгоритмы общего назначения.

К таким алгоритмам, в первую очередь, относится циклический избыточный код (Cyclic Redundancy Check, CRC), реализацией которого являются алгоритмы

CRC8, CRC16, CRC32, применяющиеся для проверки целостности цифровых данных при их передаче по каналам связи.

✓ **Криптографические алгоритмы.**

Эти алгоритмы основаны на процедуре хэширования — преобразования входного массива данных произвольной длины в выходную битовую строку фиксированной длины. К таким алгоритмам относятся, например, семейства алгоритмов MD (Message Digest Algorithm – MD2- MD6), SHA (Secure Hash Algorithm — SHA-1, SHA-2), ГОСТ Р 34. 11 (ГОСТ Р 34. 11-94, снятый с эксплуатации с 1 января 2013 г. , ГОСТ Р 34. 11-2012 «Стрибог») и другие. Областью применения этих алгоритмов является подтверждение целостности и подлинности передаваемых и хранимых данных.

В составе КСЗ ОССН включены следующие средства контроля целостности:

1. Команды, реализующие криптографические алгоритмы:

- `md5sum` (реализация алгоритма МДБ);
- `shasum` (реализация семейства алгоритмов SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/256 и SHA-512/224).

2. Средства проверки соответствия файловых сущностей ОССН её дистрибутиву:

- команда `gostsum` (реализация алгоритма ГОСТ Р 34. 11-94, ГОСТ Р 34. 11-2012 256 и 512 битов);
- графическая утилита `fly-admin-int-check`.

Указанные команды и утилиты реализуют статический контроль целостности файловых сущностей ОССН, включающий следующие компоненты:

1. Система мониторинга целостности файлов (FIM — File integrity monitoring) AFICK (Another File Integrity ChecKer), реализующая регламентный (периодический) контроль целостности файловых сущностей ОССН — вариант динамического контроля целостности.

2. Механизм контроля целостности исполняемых файлов и разделяемых библиотек формата ELF при запуске приложений на выполнение. Он реализован в выгружаемом модуле ядра ОССН `dig-sig.verif` и обеспечивает:

- контроль целостности исполняемых файлов и разделяемых библиотек на основе их контрольных сумм, вычисляемых в соответствии с ГОСТ Р 34. 11-94, ГОСТ Р 34. 11-2012 и электронной подписи, реализованной в соответствии с ГОСТ Р 34. 10-2001 и ГОСТ Р 34. 10-2012. Контрольная сумма и электронная подпись внедрены в файлы формата ELF в процессе сборки ОССН;

- внедрение электронной подписи в исполняемые файлы формата ELF, входящие в состав устанавливаемого ПО.

Команды и утилиты статического контроля целостности функционируют в режимах вычисления (`compute`) и проверки (`check`) контрольных сумм файловых сущностей ОССН.

Например, команда `shasum` в режиме вычисления контрольной суммы файловой сущности с именем `/root/file` с использованием алгоритма SHA-256 имеет следующий синтаксис:

```
shasum -a 256 /root/file
```

В режиме проверки контрольных сумм файловых сущностей команды статического контроля целостности вычисляют их для сущностей, полный путь которых указан в текстовом файле с эталонными контрольными суммами, и сравнивают их с эталонными контрольными суммами из этого файла. Результатом их выполнения в режиме проверки контрольных сумм является передача на стандартный вывод строки формата (в случае совпадения контрольных сумм):

- полный_путь_к_файловой_сущности: ОК

или (в случае их несовпадения):

- полный_путь_к_файловой_сущности: FAILED

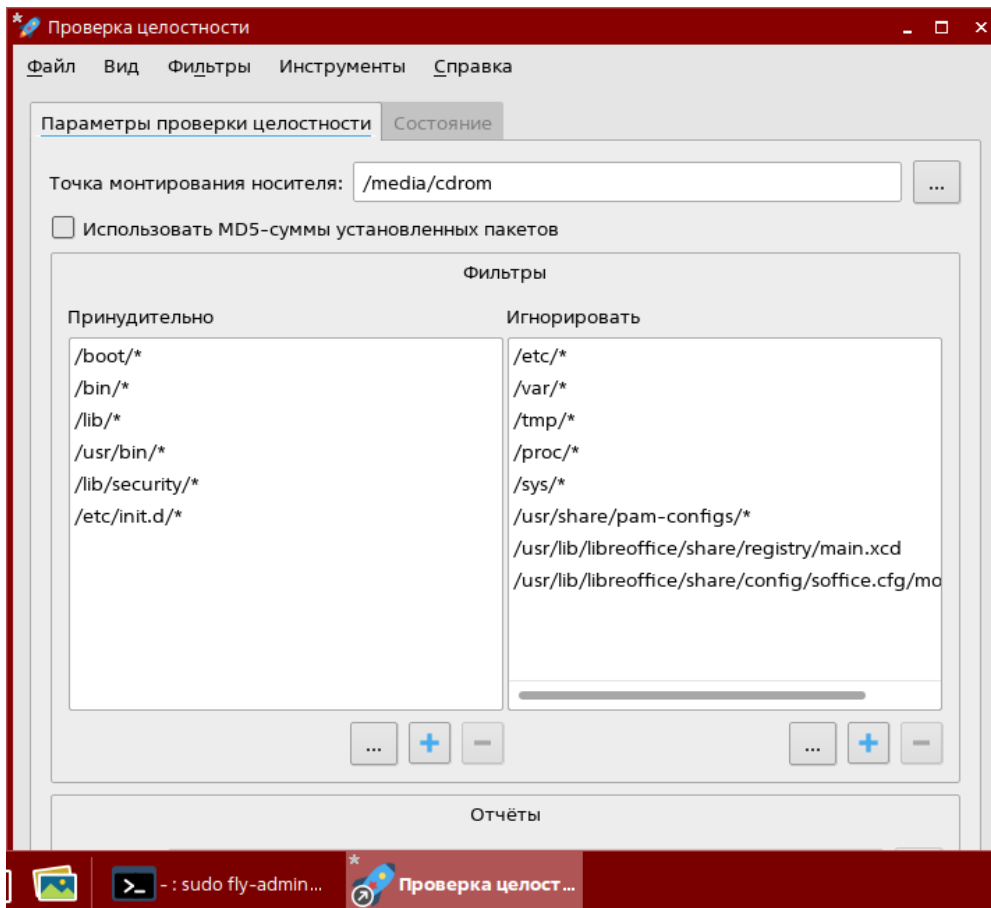
Например, команда `shasum` в режиме проверки контрольных сумм файловых сущностей в каталоге `/root/dir1`, с использованием алгоритма SHA-256 и при наличии текстового файла с эталонными контрольными суммами `/root/dir1.sha`, имеет следующий синтаксис:

```
shasum -a 256 -c /root/dir1.sha
```

Для вычисления контрольных сумм файловых сущностей ОССН с использованием криптографического алгоритма ГОСТ Р 34. 11-2012 256 битов применяется команда `gostsum`, имеющая следующий синтаксис:

```
gostsum    полный_путь_к_файловой_сущности    -    o
полный_путь_к_файловой_сущности_с_контрольными_суммами
```

Для проверки соответствия модулей установленной ОССН модулям, входящим в состав её дистрибутива, используется графическая утилита `fly-admin-int-check`, имеющая следующий интерфейс:



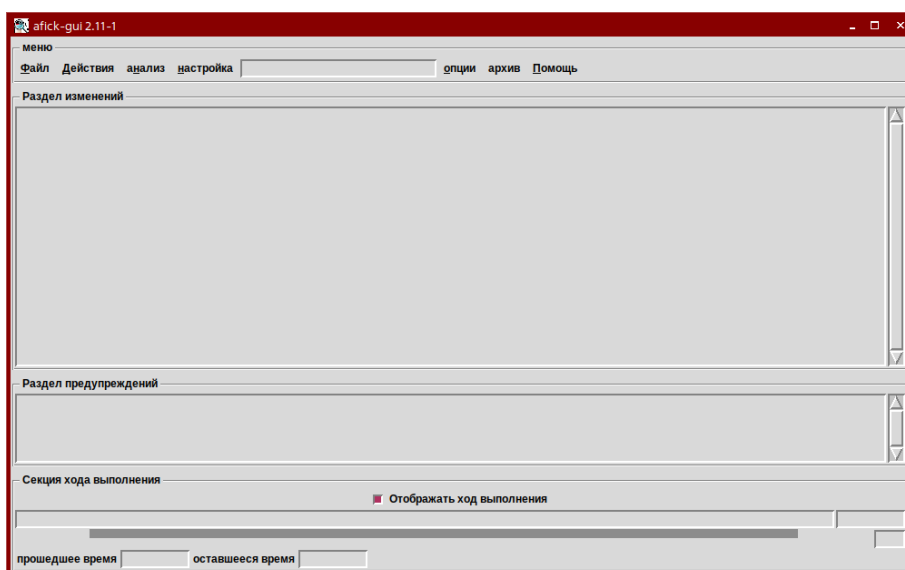
Для выполнения этой проверки в состав дистрибутива ОССН входит файл `gostsums.txt`, созданный командой `gostsum` и содержащий список контрольных сумм всех файлов, входящих в пакеты программ дистрибутива. Получаемый в результате проверки отчёт сохраняется в форматах `*.txt`, `*.htm` и `*.xml`.

Проверка соответствия модулей установленной ОССН модулям, входящим в состав её дистрибутива, является вариантом статического контроля целостности и обеспечивает контроль целостности файловых сущностей, копируемых в корневой раздел ОССН на этапе установки, что позволяет убедиться в отсутствии изменений в файловых сущностях модулей, произошедших на этапе их эксплуатации.

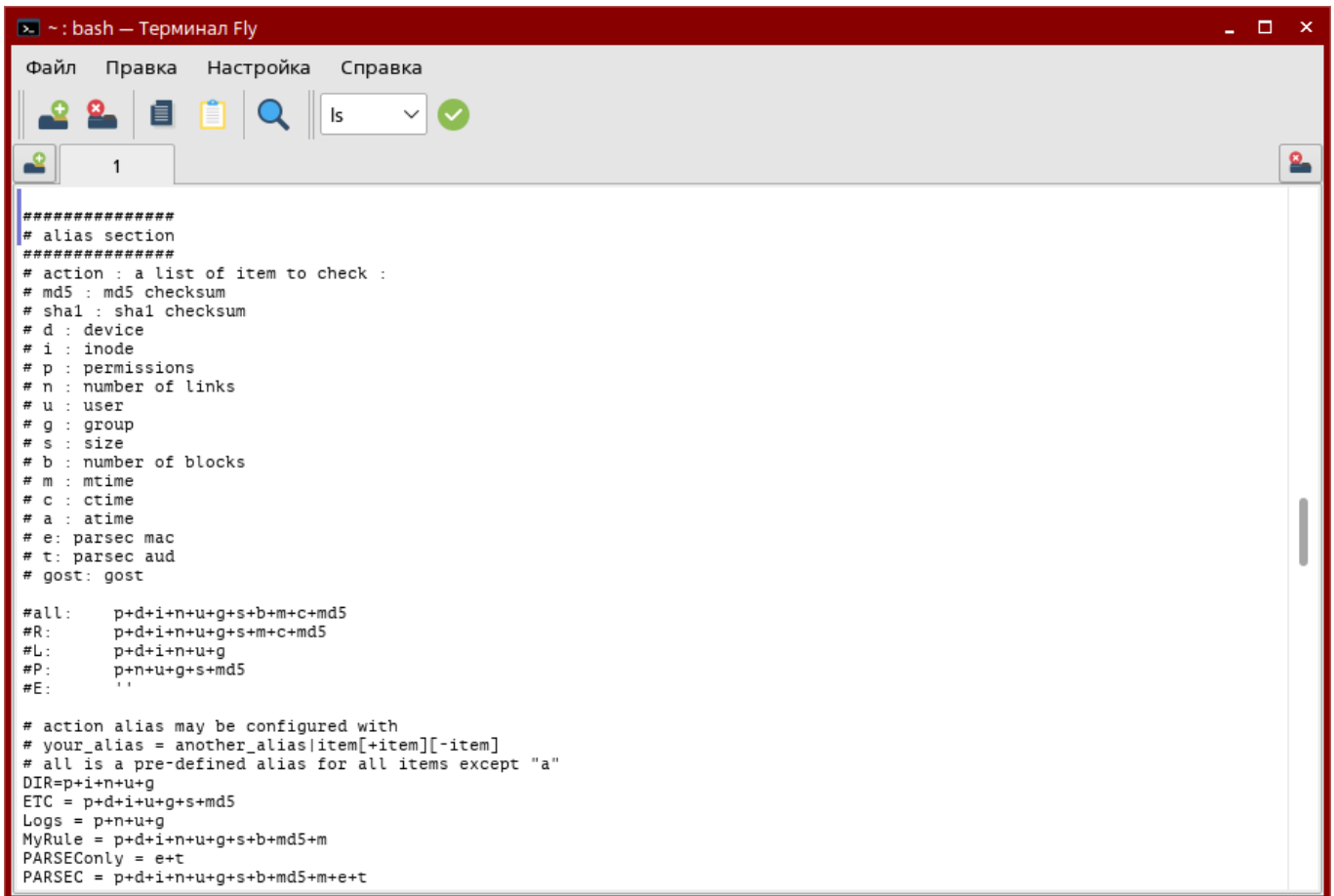
Однако такая проверка является неэффективной для файловых сущностей, содержимое которых многократно изменяется в ходе эксплуатации ОССН, например конфигурационных файлов. Кроме того, контроль целостности на основе только контрольной суммы файла не затрагивает проверку таких атрибутов файла, как временные метки (`timestamps`), дискреционные атрибуты (`Minimal ACL` и `EA ACL`), мандатные метки безопасности. Для выполнения расширенного контроля целостности файлов, обеспечивающего проверку перечисленных атрибутов файлов, в ОССН используется система мониторинга целостности файлов `AFICK`, реализующая функции контроля целостности файлов и их атрибутов с использованием криптографических алгоритмов `MD5` и `SHA-1`. В ОССН применяется модифицированный вариант системы `AFICK`, дополнительно реализующий криптографический алгоритм `ГОСТ Р 34. 11` (для приложения `gostsum` дополнительно поддерживаются алгоритмы `ГОСТ Р 34. 11-94` и `ГОСТ Р 34. 11-2012` с длиной ключа 256 или 512 битов), а также контроль мандатных меток и атрибутов подсистемы аудита безопасности. Дополнительно система `AFICK` имеет возможность настройки правил проверки целостности каталогов.

Благодаря интеграции системы `AFICK` с сервисом запуска приложений по расписанию `cron` имеется возможность выполнения регламентного (периодического) контроля целостности заданных файловых сущностей ОССН.

Система `AFICK` имеет следующий интерфейс (для её запуска можно использовать команду `afick-tk`):



Конфигурационным файлом системы AFICK является `/etc/afick.conf` - текстовый файл, структурированный по секциям. Секция `alias` содержит перечень действий (action) контроля целостности, из которых формируются правила контроля каталогов и файловых сущностей:



```
#####
# alias section
#####
# action : a list of item to check :
# md5 : md5 checksum
# sha1 : sha1 checksum
# d : device
# i : inode
# p : permissions
# n : number of links
# u : user
# g : group
# s : size
# b : number of blocks
# m : mtime
# c : ctime
# a : atime
# e : parsec mac
# t : parsec aud
# gost : gost

#all:  p+d+i+n+u+g+s+b+m+c+md5
#R:    p+d+i+n+u+g+s+m+c+md5
#L:    p+d+i+n+u+g
#P:    p+n+u+g+s+md5
#E:    ''

# action alias may be configured with
# your alias = another_alias|item[+item][-item]
# all is a pre-defined alias for all items except "a"
DIR=p+i+n+u+g
ETC = p+d+i+u+g+s+md5
Logs = p+n+u+g
MyRule = p+d+i+n+u+g+s+b+md5+m
PARSEConly = e+t
PARSEC = p+d+i+n+u+g+s+b+md5+m+e+t
```

Типовыми действиями является проверка:

- контрольных сумм, полученных заданным криптографическим алгоритмом (md5, sha1);
- inode (i) каталога или файловой сущности, её размера (size) и временных меток (mtime, ctime, atime);
- UID и GID (user, group) каталога или файловой сущности, а также прав доступа к ней (permissions).

Применительно к организации подсистемы безопасности ОССН дополнительно определены три действия:

- `e: parsec mac` — контроль целостности мандатных меток безопасности файловых сущностей;
- `t: parsec aud` — контроль целостности данных системы аудита безопасности ОССН;
- `gost:gost` — контроль целостности файловых сущностей с использованием криптографического алгоритма ГОСТ Р 34. 11.

В результате этих действий в секции `alias` формируются типовые правила для каталогов (DIR), файлов конфигурации ОССН (ETC) и файлов журналов системы аудита (Logs).

Например, правило для каталогов вида

`DIR = p + i + n + u + g`

указывает на необходимость выполнения проверки прав доступа, метаданных, количества ссылок и других стандартных атрибутов.

Дополнительно секция `action` включает правила, специфичные для подсистемы безопасности PARSEC — PARSEConly, PARCEC и GOST. Например, правило PARSEC вида `PARSEC = p+d+i+n+u+g+s+b+md5+m+e+t` указывает на необходимость выполнения проверки стандартных атрибутов файловых сущностей с использованием криптографического алгоритма MD5, проверки расширенных атрибутов (меток безопасности и флагов аудита) и списков ACL этих файловых сущностей.

В правиле GOST вида `GOST = p+d+i+n+u+g+s+b+gost+m+e+t` параметр `gost` указывает на необходимость выполнения проверки стандартных атрибутов файловых сущностей с использованием криптографического алгоритма ГОСТ Р 34.11.

В секции `files to scan` задаются полные пути и правила, применяемые к каталогам и файловым сущностям, для которых выполняется регламентный контроль целостности. Формат записей секции `files to scan` следующий:

`file action` — проверяются каталоги, подкаталоги и файловые сущности с параметром «действия»;

`file` — из проверки каталогов и подкаталогов исключается файловая сущность `file`;

`= directory action` — с параметром «действия» проверяется только каталог, и из проверки исключаются подкаталоги.

Например:

□ `/boot GOST` — проверка в каталоге `/boot` всех подкаталогов и файловых сущностей с помощью правила GOST;

□ `= /DIR` — проверка с помощью правила DIR только корневого каталога, исключая подкаталоги;

□ `/root/.bash_history` — исключение проверки в каталоге `/root` файловой сущности `.bash-history`.

Эталонные значения контрольных сумм и атрибутов файловых сущностей и каталогов хранятся в базе данных системы AFICK в файле с расширением `ndbm`. Эта база данных создаётся в соответствии с параметрами секции «files to scan» файла `/etc/afick.conf`.

Результаты контроля целостности оформляются в виде log-файлов и сохраняются:

- в случае принудительного (инициированного администратором) контроля — в каталоге `/var/lib/afick/archive` в log-файлах с форматом имени `afick.YYYYMMDDHHMMSS`. В аналогичных log-файлах сохраняются результаты обновления (update) базы данных системы AFICK;

- в случае регламентного (периодического, инициированного сервисом стоп) контроля — в каталоге `/var/log/afick` в log-файлах с форматом имени `afick.log.N` (где N принимает значения от 1 до 7).

Средство создания замкнутой программной среды в ОССН - невыгружаемый модуль ядра ОССН `digsig_verif` функционирует в трёх режимах (аналогично применяются режимы проверки подписи в расширенных атрибутах, т. е. не только для ELF-файлов, с использованием параметра `DIGSIG_XATTR_MODE`):

- **штатный режим** — исполняемым файловым сущностям формата ELF и разделяемым библиотекам, не имеющим ЭП или имеющим некорректную ЭП, исполнение запрещается (`DIGSIG_ELF_MODE = 1`);
- **режим проверки ЭП в комплексе средств системного ПО** — исполняемым файловым сущностям формата ELF и разделяемым библиотекам, не имеющим ЭП или имеющим некорректную ЭП, исполнение разрешается, но при этом выводится сообщение об ошибке проверки ЭП (`DIGSIG_ELF_MODE = 2`);
- **отладочный режим для тестирования комплекса средств системного ПО (установлен по умолчанию)** – ЭП исполняемых файловых сущностей формата ELF и разделяемых библиотек не проверяется (`DIGSIG_ELF_MODE = 0`).

Для выбора одного из указанных выше режимов функционирования модуля `digsig_verif` необходимо отредактировать конфигурационный файл `/etc/digsig/digsig_initramfs.conf`.

Управление модулем `digsig_verif` осуществляется через графический интерфейс `fly-admin-smc` либо через интерфейс файловой системы `sysfs` с использованием следующих файлов:

- `/sys/digsig/enforce` — в данном файле задаются указанные выше режимы работы;
- `/sys/digsig/key` — файл загрузки мастер-ключа ЭП;
- `/sys/digsig/additional` — файл загрузки дополнительных ключей ЭП.

Каждый дополнительный ключ для подписи системного ПО должен быть помещён в каталог `/etc/digsig/keys`. Создание дополнительных ключей выполняется с помощью команды `gpg` (GNU Privacy Guard), модифицированной для использования криптографических алгоритмов ГОСТ Р 34.11-94 и ГОСТ Р 34.11-2012.

При администрировании средств контроля целостности данных и средств контроля соответствия дистрибутиву, а также при работе со средствами создания замкнутой программной среды используются следующие команды:

- `afick` — команда управления параметрами системы контроля целостности файловых сущностей;
- `bsign` — команда создания и проверки ЭП в файлах формата ELF;
- `digsig_initramfs` — команда загрузки ключей ЭП и инициализация режима `Enforce` модуля `digsig_verif`;
- `fly-admin-int-check` — графическая утилита администрирования контроля целостности файловых сущностей;
- `gpg` — команда работы с сертификатами пользователей;
- `lsmod` — команда получения списка загруженных модулей ядра;
- `modinfo` — команда получения информации о заданном модуле ядра;
- `md5sum`, `gostsum`, `shasum` — команда вычисления контрольных сумм;

update-initramfs — команда инициализации начального загрузочного образа ОСCH (initrd).

2. Задание

1. Начать работу со входа в ОСCH в графическом режиме с учётной привилегированного пользователя пользователя, например: user (уровень доступа — 0, неиерархические категории — нет, уровень целостности — «Высокий») и запустить терминал Fly в «привилегированном» режиме командой `sudo fly-term`.

2. В домашнем каталоге создать подкаталог `checksum` и скопировать в него все файлы (включая вложенные каталоги) из каталога `/etc`.

3. Используя алгоритм MD5, вычислить контрольные суммы всех файлов в каталоге `/home/user/checksum` и перенаправить результат их вычисления в файл `/home/user/md5.check`, а поток с перечнем ошибок в файл `/home/user/error.md5`, командой

```
md5sum /home/admin-astra/checksum/* > /home/admin-astra/md5check 2> /home/admin-astra/error.md5.
```

4. Вывести в терминал содержимое файлов `/home/user/md5check` и `/home/user/error.md5` цепочкой команд `cat /home/user/md5check; cat /home/user/error.md5` и указать, для каких объектов в каталоге `/home/user/checksum` контрольные суммы не были созданы.

```
admin-astra: bash — Терминал Fly
root@astra:/home/admin-astra# md5sum /home/admin-astra/checksum/*.* > /home/admin-astra/md5check 2> /home/admin-astra/error.md5
root@astra:/home/admin-astra# cat md5check
575f5f671730120cbeb3f391461e34b3 /home/admin-astra/checksum/adduser.conf
6daf827d6d70c8e2be08b81338b8586b /home/admin-astra/checksum/afick.conf
00ecf641c7dba91188ca476fbc3690d5 /home/admin-astra/checksum/astra-safepolicy.conf
89269e1298235f1b12b4c16e4065ad0d /home/admin-astra/checksum/bash.bashrc
4c09213317e4e3dd3c71d74404e503c5 /home/admin-astra/checksum/bindresvport.blacklist
8c0619be413824f1fc7698cee0f23811 /home/admin-astra/checksum/debconf.conf
773fb95e98a27947de4a95abb3d3f2a2 /home/admin-astra/checksum/deluser.conf
d8eee708df0a5fc071752211ed8d9a54 /home/admin-astra/checksum/fly-brightness.conf
b4a42674d1a7a81cc1720e3b84485731 /home/admin-astra/checksum/flygetexe.conf
114da9ab67884a2e0fd800424559d00a /home/admin-astra/checksum/fstab.pdac
298587592c8444196833f317def414f2 /home/admin-astra/checksum/fuse.conf
c7a6335961195bae6099e77ef9d7d63b /home/admin-astra/checksum/gai.conf
4eb63731c9f5e30903ac4fc07a7fe3d6 /home/admin-astra/checksum/host.conf
d0cfb796d371b0182cd39d589b1c1ce3 /home/admin-astra/checksum/hosts.allow
208d7a7756d6daa7111bb0296f4ce6a /home/admin-astra/checksum/hosts.deny
7e0f70ffa406df2d5de03c81b5c96346 /home/admin-astra/checksum/issue.net
f1ed9c3e91816337aa7351bd5f58a442 /home/admin-astra/checksum/kernel-img.conf
035138d9c7dd6a922a96b047fabd9c22 /home/admin-astra/checksum/ld.so.cache
4317c6de8564b68d628c21efa96b37e4 /home/admin-astra/checksum/ld.so.conf
6612b6780909964f611266a780ff8a /home/admin-astra/checksum/libao.conf
cdc703f9d27f0d980271a9e95d0f18b2 /home/admin-astra/checksum/libaudit.conf
10316f363e674717d76ce5537a936987 /home/admin-astra/checksum/locale.alias
d7a66c8ca60c85abc6f2db2b43209105 /home/admin-astra/checksum/locale.alias_old
```



```

admin-astra: bash — Терминал Fly
root@astra:/home/admin-astra# cat error.md5
md5sum: /home/admin-astra/checksum/apparmor.d: Это каталог
md5sum: /home/admin-astra/checksum/binfmt.d: Это каталог
md5sum: /home/admin-astra/checksum/cron.d: Это каталог
md5sum: /home/admin-astra/checksum/cron.daily: Это каталог
md5sum: /home/admin-astra/checksum/cron.hourly: Это каталог
md5sum: /home/admin-astra/checksum/cron.monthly: Это каталог
md5sum: /home/admin-astra/checksum/cron.weekly: Это каталог
md5sum: /home/admin-astra/checksum/depmod.d: Это каталог
md5sum: /home/admin-astra/checksum/environment.d: Это каталог
md5sum: /home/admin-astra/checksum/fstab.d: Это каталог
md5sum: /home/admin-astra/checksum/grub.d: Это каталог
md5sum: /home/admin-astra/checksum/gtk-2.0: Это каталог
md5sum: /home/admin-astra/checksum/gtk-3.0: Это каталог
md5sum: /home/admin-astra/checksum/init.d: Это каталог
md5sum: /home/admin-astra/checksum/inserv.conf.d: Это каталог
md5sum: /home/admin-astra/checksum/ld.so.conf.d: Это каталог
md5sum: /home/admin-astra/checksum/libpaper.d: Это каталог
md5sum: /home/admin-astra/checksum/logrotate.d: Это каталог
md5sum: /home/admin-astra/checksum/modprobe.d: Это каталог
md5sum: /home/admin-astra/checksum/modules-load.d: Это каталог
md5sum: /home/admin-astra/checksum/pam.d: Это каталог
md5sum: /home/admin-astra/checksum/profile.d: Это каталог
md5sum: /home/admin-astra/checksum/python2.7: Это каталог
md5sum: /home/admin-astra/checksum/python3.7: Это каталог

```

5. Используя алгоритм SHA-512/256, вычислить контрольные суммы всех файлов в каталоге /home/user/checksum и перенаправить результат вычислений в файл /home/user/sha512256check командой

```
shasum -a 512256 /home/admin-astra/checksum/* > /home/admin-astra/sha512256check.
```

Вывести на экран содержимое файла /home/user/sha512256check командой `less /home/user/sha512256check`.

6. Используя редактор `vim`, изменить содержимое файла /home/user/checksum/passwd, удалив из него учётную запись суперпользователя (строку `root: x: 0:0:root: /root: /bin/bash`).

7. Используя алгоритм MD5, проверить контрольные суммы всех файлов в каталоге /home/user/checksum и перенаправить результат проверки в файл /home/user/fullcheck командой `md5sum -c ./md5check > /home/user/fullcheck`.

8. Используя алгоритм SHA512/256, проверить контрольные суммы всех файлов в каталоге /home/user/checksum и перенаправить результат проверки (с добавлением) в файл /home/user/full-check командой `shasum -a 512256 -c ./sha512256check >> /home/user/fullcheck`.

9. Найти в файле /home/user/fullcheck строки, указывающие на файлы с нарушением целостности (содержащие слова ПОВРЕЖДЁН и FAILED), вывести в терминал их содержимое и число цепочкой команд

```
grep 'ПОВРЕЖДЁН' /home/admin-astra/fullcheck > /home/admin-astra/tmpcheck; grep 'FAILED' /home/admin-astra/fullcheck >> /home/admin-astra/tmpcheck; wc -l /home/admin-astra/tmpcheck; less /home/admin-astra/tmpcheck.
```

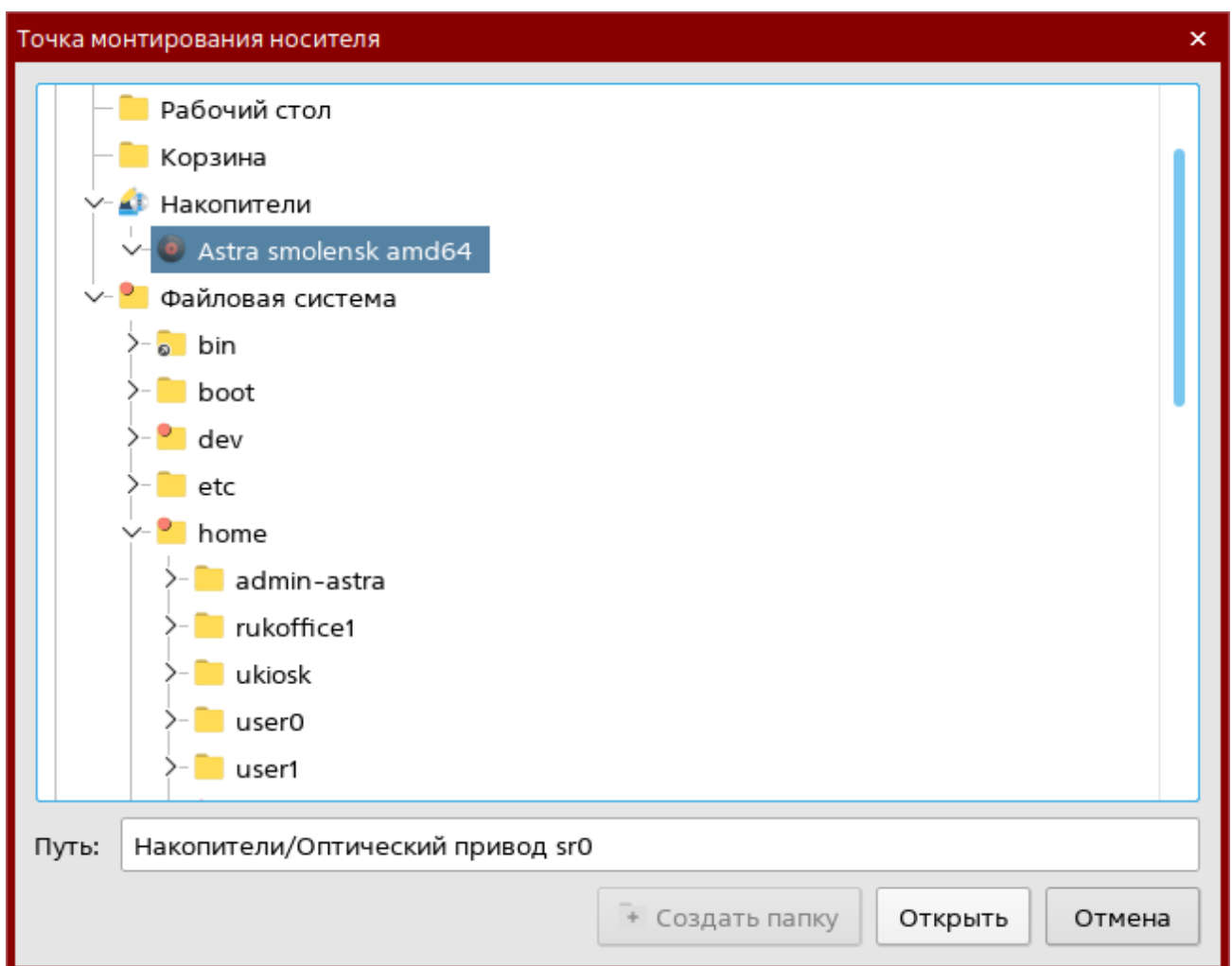
10. Используя алгоритм ГОСТ Р 34. 11-2012 (256 битов), вычислить контрольную сумму файла /home/user/checksum/shadow, перенаправить результат проверки в файл /home/user/gostcheck и вывести в терминал содержимое файла /home/user/gostcheck цепочкой команд `gostsum`

```
/home/user/checksum/shadow -o ./gostcheck; less
/home/user/gostcheck.
```

11. Установить оптический диск с дистрибутивом ОССН и, используя алгоритм ГОСТ Р 34. 11-2012 (256 битов), вычислить его контрольную сумму (по умолчанию файл устройства оптического диска /dev/sr0) и перенаправить результат вычисления в файл /home/user/isocheck командой `gostsum -d /dev/sr0 > /home/user/isocheck` (выполнение команды занимает длительное время).

12. Запустить графическую утилиту `fly-admin-int-check` и во вкладке «Параметры проверки целостности»:

- выбрать точку монтирования устройства «Astra Smolensk amd64» (по умолчанию это, чаще всего, каталоги /media/cdrom или /media/cdrom0) и выполнить монтирование;



- настроить фильтр проверки целостности в разделе «Принудительно», добавив регулярное выражение, содержащее абсолютный путь ко всем файлам каталога /usr/lib: `/usr/lib/*`;

- настроить фильтр проверки целостности в разделе «Игнорировать», удалив регулярное выражение, содержащее абсолютный путь к каталогу /tmp;

- в разделе «Отчёты» задать только текстовый формат файла отчёта, определив путь размещения файла `report.txt` в каталоге `/home/user/report`;

- изменить содержимое файла /usr/share/doc/libcap2/copyright командой `vim /usr/share/doc/libcap2/copyright`, удалив в нем две первые строки:

Upstream-Contact: Andrew G. Horgan morgan@kernel.org
 Source: <https://www.kernel.org/pub/linux/libs/security/linux-privs/libcap2/:d2>

- начать проверку и зафиксировать предполагаемое время проверки, перейти во вкладку «Состояние» и проконтролировать статус проверки, после окончания проверки завершить работу графической утилиты;

- в файле /home/user/report/report.txt найти строки, содержащие текст: «Файлы, целостность которых нарушена», «Контр. сумма» и «/usr/share/doc/libcap2/copyright», сохранить результаты поиска в файл /home/user/report-2 цепочкой команд: `grep 'Файлы, целостность которых нарушена' /home/user/report/report.txt -A 4 > /home/user/report-2; grep 'Контр. сумма' /home/user/report/report.txt -A 4 >> /home/user/report-2; grep '/usr/share/doc/libcap2/copyright' /home/user/report/report.txt >> /home/user/report-2.`

13. Отредактировать секцию `directives` конфигурационного файла /etc/afick.conf системы AFICK, отменив проверку выполняющихся приложений: исходный вариант секции:

`directives: running_files: = yes,`

отредактированный вариант секции:

`directives: running_files: = 0.`

14. Отредактировать секцию `alias` конфигурационного файла /etc/afick.conf системы AFICK:

- изменить правило ETC, удалив из него проверку размера файловых сущностей и добавив проверку времени их модификации:

исходный вариант правила:

`ETC = p+d+i+u+g+s+md5,`

отредактированный вариант правила:

`ETC = p+d+i+u+m+c+a+md5;`

- отредактировать правило MyRule, удалив из него проверку для файловых сущностей количества ссылок на них и добавив проверку контроля целостности мандатных меток безопасности, контроля целостности данных системы аудита безопасности и контроля целостности с использованием криптографического алгоритма ГОСТ Р 34. 11-2012 вместо алгоритма MD5:

исходный вариант правила:

`MyRule = p+d+i+n+u+g+s+b+md5+m,`

отредактированный вариант правила:

`MyRule = p+d+i+u+g+s+b+gost+m+e+t.`

15. Отредактировать секцию `file section` конфигурационного файла `/etc/afick.conf`.

- заменить для каталога `/boot` правило проверки GOST на правило проверки PARSEC:

исходный вариант:

`/boot GOST,`

отредактированный вариант:

`/boot PARSEC`

- добавить для файловой сущности `/etc/fstab` правило проверки MyRule:

отредактированный вариант:

`/etc/fstab MyRule,`

- активировать правило проверки по умолчанию для каталога `/lib`:

исходный вариант:

`#/lib MyRule,`

отредактированный вариант:

`/lib MyRule.`

16. Обновить базу данных системы AFICK с учётом выполненных изменений в секции `file section` командой `afick -u`.

17. Изменить содержимое файла `/etc/fstab`, удалив в нем две первые строки.

18. Запустить графическую утилиту «Контроль целостности файлов» (`afick-tk`) управления AFICK из меню «Системные» главного пользовательского меню и выполнить принудительную проверку целостности, выбрав действие — сравнение с базой.

19. После завершения контроля целостности:

- в меню утилиты `afick-tk` «Файл — история» определить дату и время последнего принудительного контроля целостности;
- найти в каталоге `/var/lib/afick/archive` log-файл, соответствующий выполненной принудительной проверке (значение `YYYYMMDDHHMMSS` в имени log-файла должно совпадать с найденными в предыдущем пункте датой и временем проверки);
- просмотреть найденный log-файл с помощью команды `less` и в его секции `#detailed changes` найти запись о нарушении целостности файловой сущности `/etc/fstab` (раздел `changed file: /etc/fstab`);
- проанализировать найденную запись о нарушении целостности и определить параметры, соответствующие действиям (action) нарушения целостности, и их текущие значения.

20. Запустить терминал Fly в «привилегированном» режиме командой `sudo fly-term`.

21. Просмотреть загруженные модули ядра ОССН и вывести в терминал данные о невыгружаемом модуле `digsig_verif` конвейером команд `lsmod | grep «digsig_verif»`.

Ответить на вопрос: связан ли модуль `digsig_verif` с другими загружаемыми (невыгружаемыми) модулями?

```
root@astra:/home/admin-astra# sudo lsmod | grep digsig_verif
digsig_verif          495616    1 parsec
```

- Просмотреть информацию о модуле `digsig_verif` командой `modinfo digsig_verif`. Определить расположение модуля `digsig_verif` и информацию о разработчике.

- Выполнить импорт открытых ключей, используемых для проверки ЭП файлов. Для этого выполнить следующие действия:

- инициализировать каталог `/root/.gnupg` при просмотре текущих ключей командой `gpg --list-sigs`;
- импортировать открытый мастер-ключ «JSC RPA RusBITech (PRIMARY RBT ROOT KEY 2018)» командой `gpg --import /etc/digsig/primary_key-2018.gpg`;
- импортировать открытые ключи `partners_rbt_root_key_2018.gpg` и `build_system_rbt_root_key_2018.gpg` (данный ключи используется для подписи файлов ОССН), командой `gpg --import /etc/digsig/имя_файла_ключа`.
- Вывести перечень используемых ключей командой `gpg --list-keys`

```
root@astra:/home/admin-astra# gpg --list-keys
/root/.gnupg/pubring.kbx
-----
pub   gP256 2018-06-17 [SC]
      8066E9BD2201D9783E2D842BD2B6689A37DB8024
uid   [ неизвестно ] JSC RPA RusBITech (BUILD-SYSTEM RBT ROOT KEY 2018) <mail@rusbitech.ru>

pub   gP256 2018-06-17 [SC]
      A12D7B5BAFCE40D87FD41DAFC82D49FC3675B6FA
uid   [ неизвестно ] JSC RPA RusBITech (PARTNERS RBT ROOT KEY 2018) <mail@rusbitech.ru>

pub   gP256 2018-06-17 [SC]
      8E839E2F389F882A259F47997285E858DB069FF5
uid   [ неизвестно ] JSC RPA RusBITech (PRIMARY RBT ROOT KEY 2018) <mail@rusbitech.ru>
```

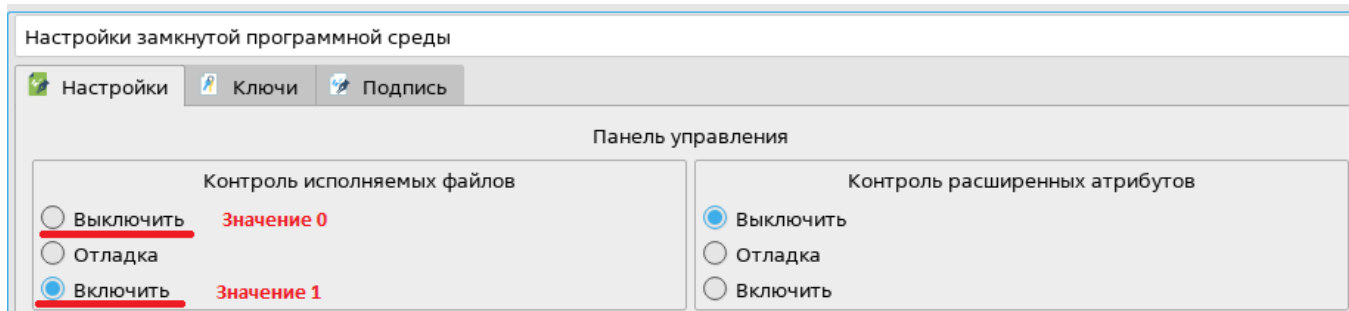
• Вывести текущие ключи командой `gpg --list-sigs`. Определить идентификатор мастер-ключа «JSC RPA RusBITech (PRIMARY RBT ROOT KEY 2018)». Используется ли он для подписи других загруженных ранее ключей?

• Проверить корректность ЭП файла `/bin/dash` командой `bsign -w $(which dash)`. Определить, каким ключом был подписан данный файл по его идентификатору в строке «`signer:` ».

• Переписать открытый ключ `/etc/digsig/build_system_rbt-root-key_2018.gpg` в каталог `/etc/digsig/keys` командой `cp /etc/digsig/build_system_rbt_root_key_2018.gpg /etc/digsig/keys`.

• Перейти в каталог `/etc/digsig` и изменить файл `digsig_initramfs.conf` (значение `DIGSIG_ELF_MODE` установить равным 1).

• Проверить корректность установки данного параметра путём открытия настройки «Замкнутой программной среды» в «Панели управления».



- Создать дополнительный ключ ЭП командой `gpg -full-generate-key`. В диалоге команды `gpg`:

- ☐ выбрать пункт 15 «GOST R 34. 10-2012»;

- ☐ указать длину ключа – 1024;

- ☐ указать неограниченный срок действия дополнительного ключа ЭП, выбрав значение 0

- ☐ указать параметры:

- ☐ указать полное имя: `rootserver`, адрес электронной почты: `root@server.test` и получить User ID: "`rootserver root@server.test`".

- Вывести текущие ключи командой `gpg --list-sigs` и определить идентификатор ключа "`rootserver root@server.test`".

- Скопировать файл `/bin/dash` в каталог `/root`, указав при этом новое имя файла `l.elf`.

- Подписать файл `l.elf` новым ключом «`rootserver <root@server.test>`» командой `bsign --sign /root/l.elf`.

- Вывести новую подпись файла командой `bsign -w /root/l.elf` и проверить соответствие идентификатора ключа ЭП в строке «`signer.`» данным ключа «`rootserver <root@server.test>`».

- Включить штатный режим проверки ЭП с использованием модуля `digsig_verif`, установив значение ключа `DIGSIG_ELF_MODE = 1` в конфигурационном файле `/etc/digsig/digsig_initramfs.conf`.

- Активировать настройки командой `sudo update-initramfs -u -k all`, затем выполнить перезагрузку и повторный вход в ОСНН.

- Запустить терминал `Fly` в «привилегированном» режиме командой `sudo fly-term`.

- Проверить включение штатного режим функционирования модуля `digsig_verif` (в файле `/sys/digsig/elf_mode` должно быть установлено значение «1») командой `cat /sys/digsig/elf_mode`.

- Выполнить попытку запуска файла `/root/l.elf`, который был подписан с использованием ключа «`rootserver <root@server.test>`» (данный ключ не был подписан мастер-ключом «`JSC RPA RusBITech (PRIMARY RBT ROOT KEY 2018)`»), и проанализировать выводимые ошибки.

- Установить значение ключа `DIGSIG_ELF_MODE=0` в конфигурационном файле `/etc/digsig/digsig_initramfs.conf`, активировать настройки

командой `sudo update-initramfs -u -k all`, затем выполнить перезагрузку и повторный вход в ОССН.

- В «привилегированном» режиме терминала Fly выполнить команду `/root/1.elf` и проанализировать вывод.

- Выйти из запущенного интерпретатора «dash» (файл `1.elf`) командой `exit`. Активировать настройки командой `sudo update-initramfs -u -k all`, затем выполнить перезагрузку и повторный вход в ОССН.

- Создать ключи и выполнить подпись файла конфигурации:

- запустить терминал Fly от имени учётной записи привелигированного пользователя командой `fly-term`;

- скопировать файлы `/etc/passwd` и `/bin/dash` в каталог `~` и сменить владельца на `user1`;

- выполнить команду генерации мастер-ключа для подписи в `xattr` командой `gpg --full-generate-key`, выбрать алгоритм (15) и установить имя: `xattr-key`;

- выполнить команду генерации ключа для подписи в `xattr` командой `gpg --full-generate-key`, выбрать алгоритм (15) и установить имя: `xattr-key-sign`;

- выполнить подпись ключа «`xattr-key-sign`» командой `gpg --sign-key "xattr-key-sign" > xattr-key-sign.gpg`;

- экспортировать ключ «`xattr-key`» командой `gpg --export "xattr-key" xattr-key.gpg`;

- проверить наличие подписанного ключа «`xattr-key-sign`» командой `gpg --list-sigs` (при этом ключ «`xattr-key-sign`» должен быть подписан ключом «`xattr-key`»);

- запомнить идентификаторы ключей «`xattr-key`» и «`xattr-key-sign`» (8 байтов в шестнадцатеричном формате - 16 символов);

- создать хэш файла `~/passwd` и записать его в расширенные атрибуты командой `sudo bsign --hash ~/passwd` (обратить внимание, что никаких ключей разблокировки секретного ключа при этом не запрашивается у пользователя);

- создать файл `~/.gnupg/gpg.conf` с содержимым: `default-key идентификатор_ключа_xattr-key-sign`;

- выполнить подпись файла `passwd` командой `bsign --sign ~/passwd`;

- выполнить проверку подписи файла `passwd` командой `bsign -w ~/passwd`;

- скопировать ключи (`xattr-key-sign.gpg` и `xattr-key.gpg`) для работы с подписями файлов в каталог `/etc/digsig/xattr_keys`;

- в графическом файловом менеджере `fly-fm` перейти в каталог «Домашний» и открыть в контекстном меню «Свойства», «Подпись» файла `passwd`;

- нажать кнопки «Загрузить ключи» и «Информация», при этом проверить корректность созданного хэш и наличие подписи.

3. Контрольные вопросы

1. В чем заключается отличие команд `md5sum`, `shasum` и `gostsum` с точки зрения вычисления контрольной суммы файлов?
2. В каком формате организован вывод команд `md5sum`, `shasum` и `gostsum` при вычислении контрольной суммы файлов?
3. В какой из команд `md5sum`, `shasum` или `gostsum` возможно изменение алгоритма хэширования?
4. Какие правила в конфигурационном файле системы регламентного контроля целостности AFICK сформированы по умолчанию, а какие являются специфическими для подсистемы безопасности PARSEC?
5. Как инициализировать базу данных системы регламентного контроля целостности AFICK после внесения изменений в её конфигурационный файл?
6. Каким видом модулей ядра ОССН является модуль `digsig_verif`?
7. Какой формат файлов ключей ЭП СПО использует модуль `digsig.verif`?
8. Какой файл сценария командного интерпретатора `bash` применяется при добавлении дополнительных ключей ЭП для модуля `digsig_verif`?

4. Требования к отчёту

Отчёт выполняется каждым студентом индивидуально. Работа должна быть оформлена в электронном виде в формате `.doc` и распечатана на листах формата А4. На титульном листе указываются: наименование учебного учреждения, наименование дисциплины, название и номер работы, вариант, выполнил: фамилия, имя, отчество, группа, проверил: преподаватель ФИО (образец титульного листа представлен в приложении 1).

Отчет должен содержать:

- титульный лист;
- цель работы;
- краткие теоретические сведения, ответы на контрольные вопросы;
- описание хода выполнения работы со скриншотами:
 - Полный перечень использованных команд с описанием их назначения.
 - Примеры выполнения команд, которые были использованы в ходе работы, с описанием результатов их выполнения.
 - Описание порядка работы с графическим интерфейсом при выполнении следующих операций:
 - ☐ конфигурирование обязательных для проверки и игнорируемых путей в графической утилите `fly-admin-int-check`;
 - ☐ конфигурирование пути размещения файла отчёта в графической утилите `fly-admin-int-check`;
 - ☐ просмотр текущих правил проверки в графической утилите «Контроль целостности файлов» (`afick-tk`) системы регламентного контроля целостности AFICK;
 - ☐ запуск проверки целостности данных в графической утилите `afick-tk`.
 - Описание порядка работы с командами при выполнении следующих операций:
 - ☐ вычисление контрольной суммы файла с использованием команды `md5sum`;

- ☐ вычисление контрольной суммы файла с использованием команды shasum;
- ☐ вычисление контрольной суммы файла с использованием команды gostsum;
- ☐ проверка целостности файлов с использованием команды md5sum;
- ☐ проверка целостности файлов с различными алгоритмами вычисления контрольной суммы с использованием утилиты shasum.

➤ Описание особенностей конфигурирования и режимов функционирования модуля `digsig_verif`.

— выводы



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»
(ДГТУ)**

Факультет Информатика и вычислительная техника
Кафедра Кибербезопасность информационных систем

Лабораторная работа № _____
на тему « _____ »

Выполнил обучающийся гр. _____

(Фамилия, Имя, Отчество)

Проверил:

(должность, Фамилия, Имя, Отчество)

Ростов-на-Дону

20 _____