АННОТАЦИЯ

Настоящий документ содержит сведения о настройке механизмов контроля целостности программной платформы «СКАДА А-СОФТ».

СОДЕРЖАНИЕ

1 Назначение и условия применения программы

1.1 Назначение ПО механизмов контроля целостности

1.1.1 ПО предназначено для:

– регламентного контроля целостности базового ПО (ПП «СКАДА А-СОФТ»);

– контроля целостности ППО.

1.2 Условия применения

1.2.1 Состав аппаратных средств

ПО контроля целостности функционирует на аппаратных средствах, описанных в документе «Руководство программиста.».

1.2.2 Состав программных средств

Для настройки контроля целостности необходимым является наличие следующих программных средств:

– операционная система Astra Linux SE Smolensk 1.6 (x64);

– изделие программное ПП «СКАДА А-СОФТ».

1.2.3 Требования к квалификации персонала

Специалисты, занимающиеся настройкой программного обеспечения, должны иметь знания для работы с системой Linux на уровне пользователя и знания для работы с ПП «СКАДА А-СОФТ» (приведено в документации по ПП «СКАДА А-СОФТ»).

2 Характеристика программы

2.1 Регламентный контроль целостности ПП «СКАДА А-СОФТ»

Регламентный контроль целостности ПП «СКАДА А-СОФТ» обеспечивается набором программных средств на основе «Another File Integrity Checker», входящим в состав Asrta Linux SE. В указанном наборе программных средств реализована возможность для проведения периодического (с использованием системного планировщика заданий cron) вычисления контрольных сумм файлов и соответствующих им атрибутов расширенной подсистемы безопасности PARSEC (мандатных атрибутов и атрибутов расширенной подсистемы протоколирования) с последующим сравнением вычисленных значений с эталонными. В указанном наборе программных средств реализовано использование библиотеки libgost, обеспечивающей подсчет контрольных сумм в соответствии с ГОСТ Р 34.11-94.

Эталонные значения контрольных сумм и атрибутов файлов хранятся в БД. БД защищается системой разграничения доступа.

Для вычисления контрольных сумм могут использоваться алгоритмы: MD5-Digest, SHA1 и ГОСТ Р 34.11-94.

2.2 Контроль целостности ППО посредством встроенных механизмов ПП «СКАДА А-СОФТ»

В ПП «СКАДА А-СОФТ» реализован механизм контроля целостности ППО. Контроль целостности ППО обеспечивается подсчетом контрольных сумм в соответствии с ГОСТ Р 34.11-94.

Подсчет контрольных сумм осуществляется в режиме исполнения для следующих конфигураций:

– модулей сбора данных – Modbus, OPC, Логический уровень;

– подключения баз данных;

– транспортов;

– редактор графических интерфейсов пользователя;

– общая контрольная сумма для всех контролируемых конфигураций.

Результат подсчета контрольных сумм выводится в конфигуратор ПП «СКАДА А-СОФТ».

3 Настройка и запуск механизмов контроля целостности базового ПО и ППО

3.1 Настройка регламентного контроля целостности базового ПО

Настройки хранятся в конфигурационном файле /etc/afick.conf. В файле содержатся указания о том, какие файлы и каталоги подвергаются контролю целостности и с какими правилами. В конфигурационном файле также можно задать местоположение файла отчета.

По умолчанию в конфигурационном файле присутствует ряд настроек. Кроме различных путей, например, к файлам БД:

```
database:=/var/lib/afick/afick
```

где содержится указание о том, какие файлы/каталоги подвергаются контролю целостности и с какими правилами.

Правило PARSEC выглядит следующим образом:

```
PARSEC = p+d+i+n+u+g+s+b+md5+m+e+t
```

где p+d+i+n+u+g+s+b+md5+m означает слежение за всеми стандартными атрибутами файла и использование хэш-функции MD5-Digest для слежения за целостностью содержимого файлов;

+e+t означает контроль расширенных атрибутов: мандатной метки и флагов аудита, соответственно. Контроль ACL осуществляется при установке флага +g.

Правило GOST выглядит следующим образом:

```
GOST = p+d+i+n+u+g+s+b+gost+m+e+t
```

где p+d+i+n+u+g+s+b+gost+m означает слежение за всеми стандартными атрибутами файла и использование хэш-функции ГОСТ Р 34.11-94 для слежения за целостностью содержимого файлов;

+e+t означает контроль расширенных атрибутов: мандатной метки и флагов аудита, соответственно. Контроль ACL осуществляется при установке флага +g.

Правило для каталогов:

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

Правило означает слежение за правами доступа, метаданными, количеством ссылок и другими стандартными атрибутами (подробнее см. /etc/afick.conf).

В файле конфигурации задаются пути к файлам и каталогам, контролируемым afick, например:

```
/boot GOST
/lib/modules PARSEC
/sbin PARSEC
/lib64/security PARSEC
```

Кроме того, на выбор администратора представлен ряд дополнительных путей с правилами. Соответствующие строки помечены знаком комментария # и могут быть активированы снятием этого знака.

Параметр `report_url:=stdout` задает местоположение файла-отчета.

При запуске afick с параметром -i:

afick -i

будет создан файл /var/lib/afick/afick. Это и есть БД формата ndbm.

При запуске afick автоматически установит ежедневное задание для cron. Файл с заданием находится в /etc/cron.daily/afick_cron.

При изменении конфигурационного файла можно обновить конфигурацию командой:

afick -C

В приложении 1 приведен пример конфигурационного файла /etc/afick.conf. Контролю подлежат папки, в которые устанавливается ПП «СКАДА А-СОФТ».

3.1.1 Запуск afick для проверки целостности

Чтобы не ждать запуска afick по cron, можно набрать команду:

afick --update

На рисунке 1 приведен результат запуска afick, если не было изменений.

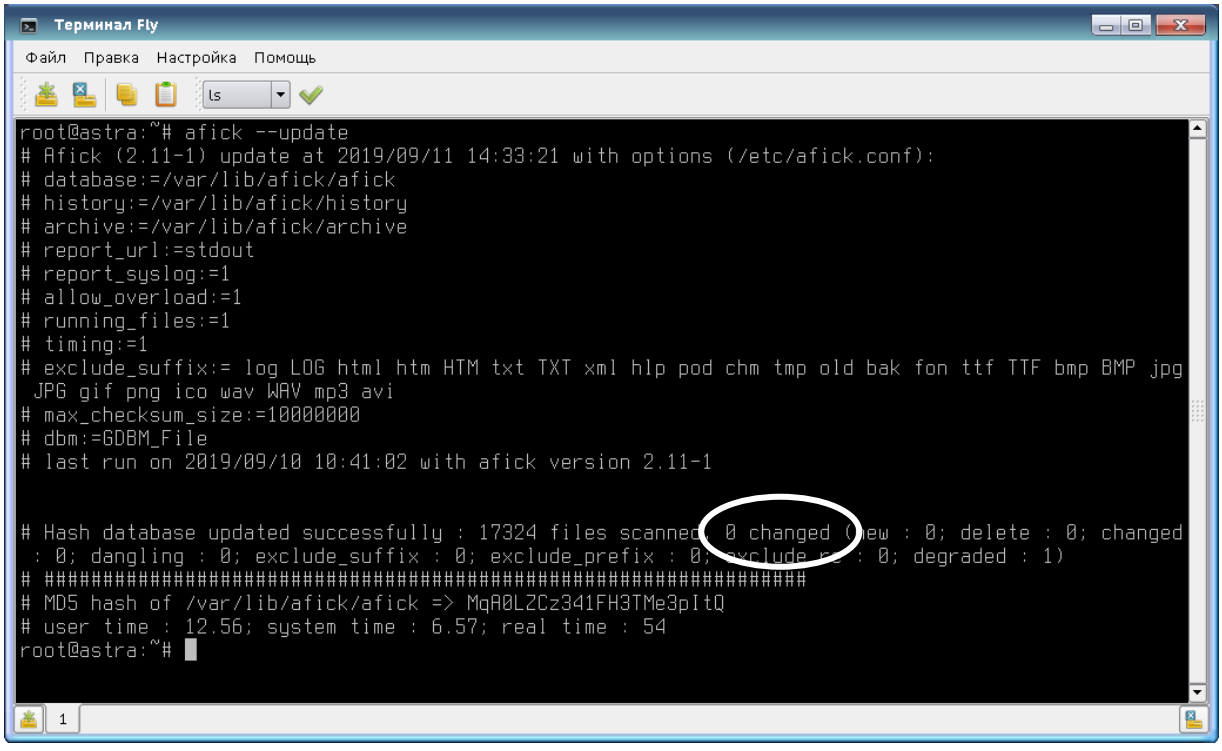

Рисунок 1 – Результат запуска afick при отсутствии изменений

На рисунке 2 приведен результат запуска afick после удаления файла из контролируемой папки. Результатом являются два изменения: удален файл и изменена папка.

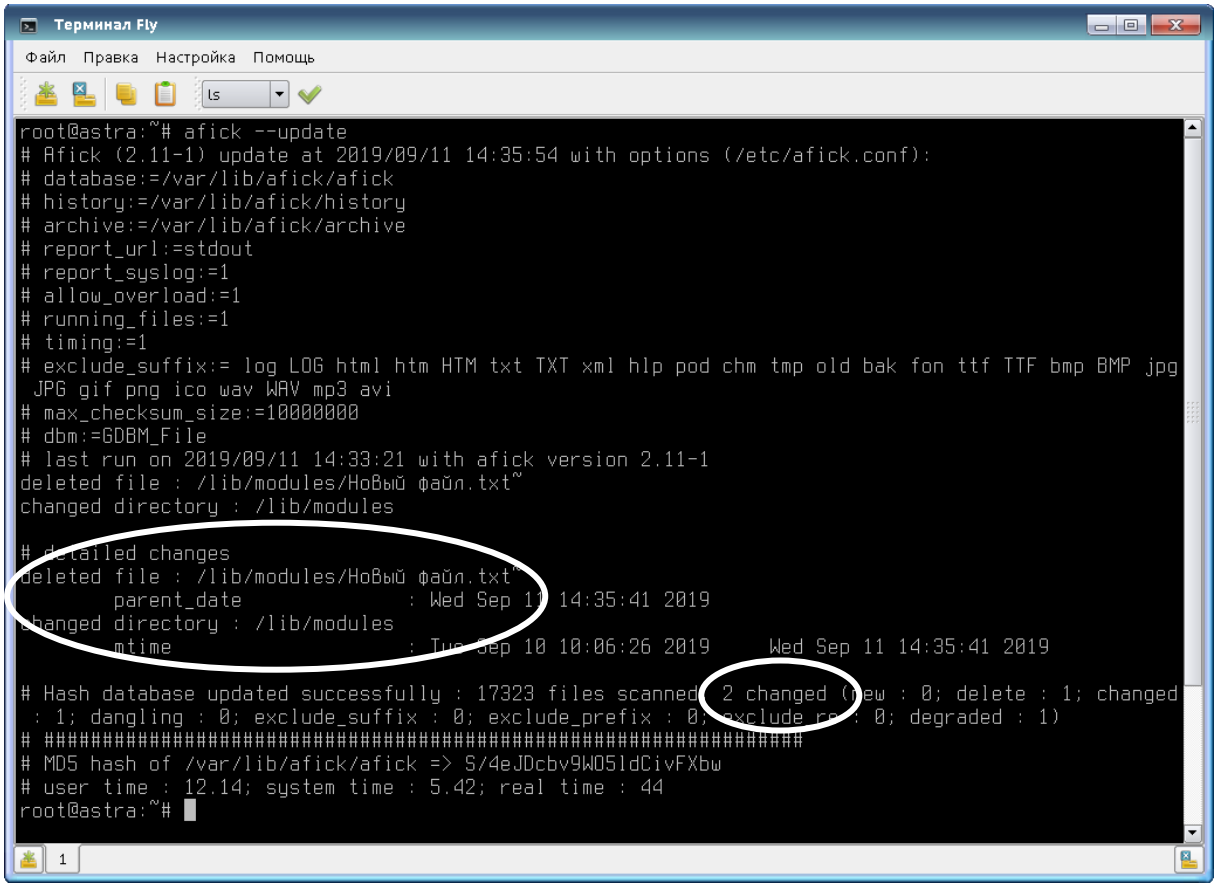

Рисунок 2 - Результат запуска afick после изменений

3.2 Настройка контроля целостности ПО посредством встроенных механизмов ПП «СКАДА А-СОФТ»

В ПП «СКАДА А-СОФТ» реализован механизм контроля целостности ППО. Контроль целостности ППО обеспечивается подсчетом контрольных сумм в соответствии с ГОСТ Р 34.11-94.

Подсчет контрольных сумм осуществляется в режиме исполнения для следующих конфигураций:

– конфигурация модулей сбора данных – Modbus, OPC, Логический уровень (для каждого контроллера отдельно);

– библиотека шаблонов подсистемы «Сбор данных»;

– конфигурация подключения баз данных (отдельно для конфигурации каждой базы);

– конфигурация транспортов;

– общая контрольная сумма конфигурации рабочей станции (для всех контролируемых конфигураций, кроме редактора графических интерфейсов пользователя);

– редактор графических интерфейсов пользователя.

У каждого контролируемого модуля в конфигураторе есть пункт *Hash*, при раскрытии которого на экран выводится контрольная сумма текущей конфигурации. На рисунке 3 показан подсчет контрольной суммы для контроллера «1» модуля «Логический уровень» подсистемы «Сбор данных».

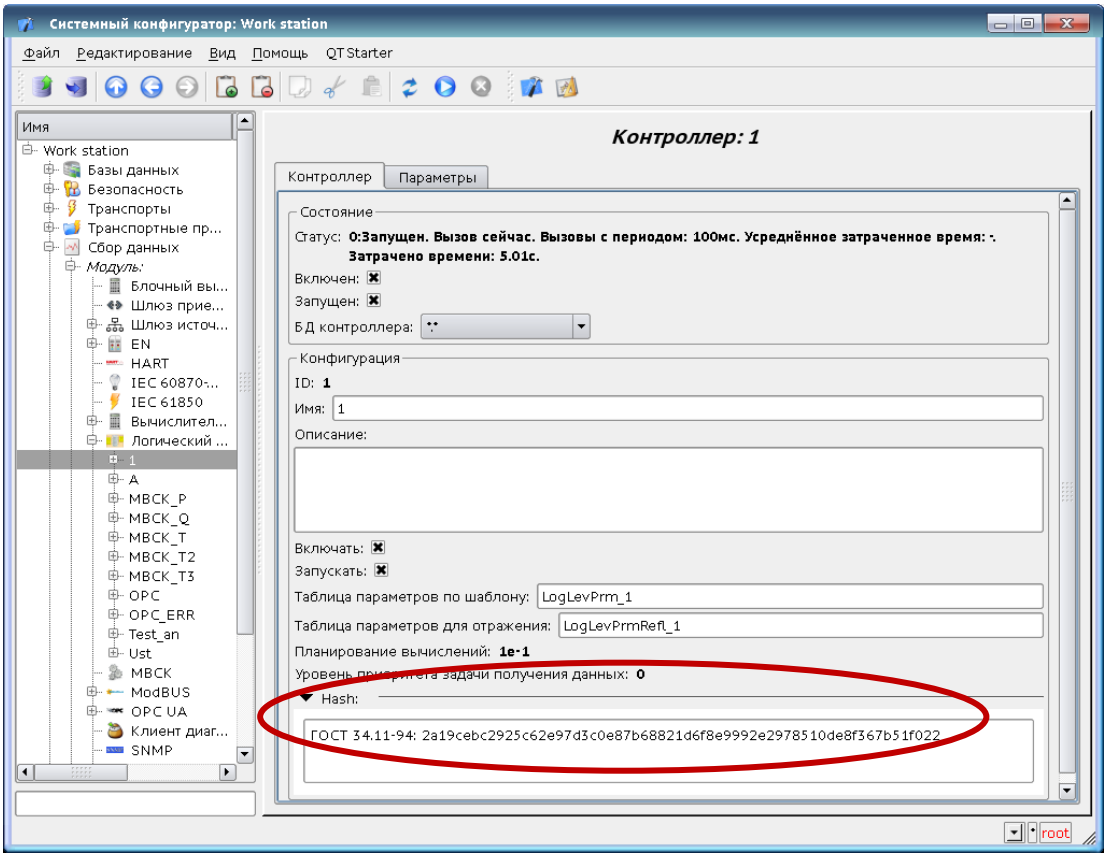

Рисунок 3 - Контрольная сумма логического контроллера

Аналогично производится подсчет контрольной суммы для остальных конфигураций. На рисунке 4 приведен пример подсчета общей контрольной суммы для конфигураций рабочей станции.

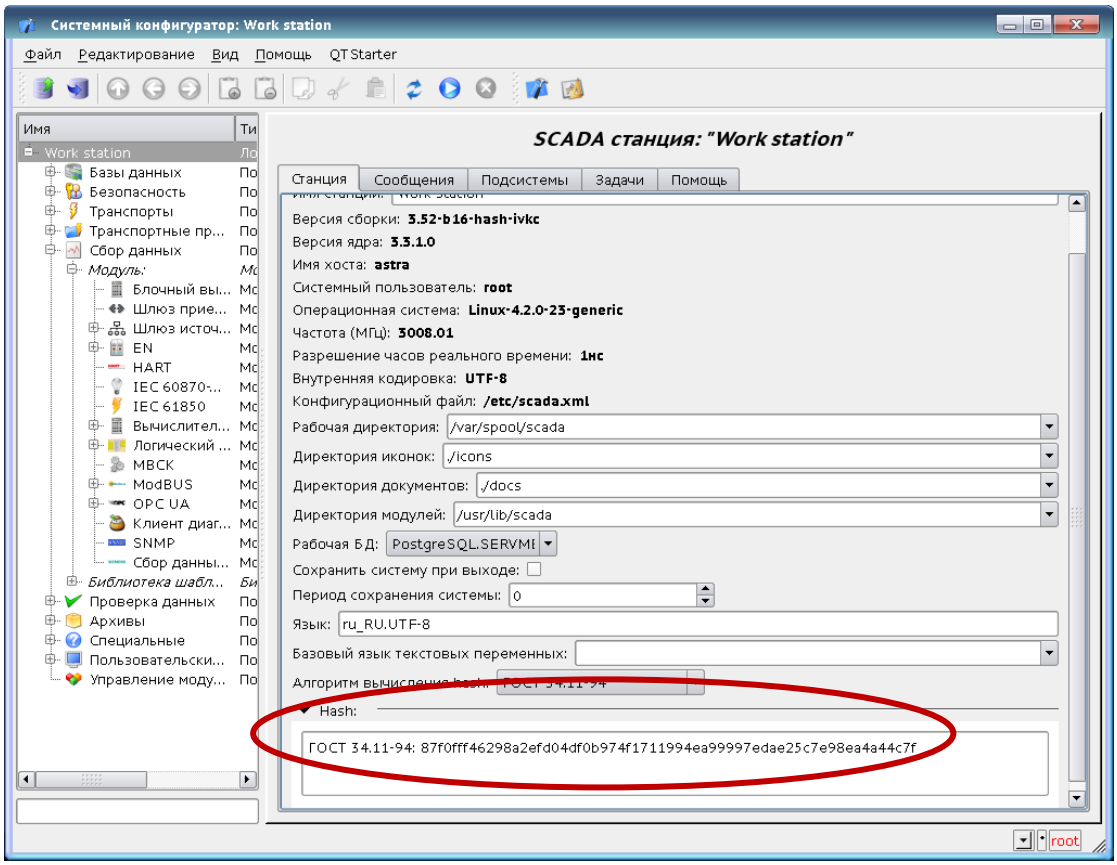

Рисунок 4 - Контрольная сумма рабочей станции

Для проверки контрольных сумм необходимо после установки ППО, созданного на базе ПП «СКАДА А-СОФТ», запустить подсчет контрольных сумм для всех контролируемых частей, сделать скриншоты полученных результатов и сохранить их.

Далее для контроля достаточно проверить контрольную сумму рабочей станции (см. рисунок 4). В случае, если контрольная сумма не совпадет с подсчитанной ранее, необходимо сравнить контрольные суммы отдельных модулей.

Отдельно производится подсчет контрольных сумм для графической части ППО (библиотека виджетов и проект). Подсчет контрольных сумм можно произвести, запустив графический редактор, либо в системном конфигураторе открыть вкладку «Пользовательские интерфейсы» → «Движок среды визуализации» → «Проекты» → «*Имя_проекта*» (рисунок 5). Развернуть пункт *Hash*, в окне будет выведен результат подсчета контрольной суммы выбранного проекта.

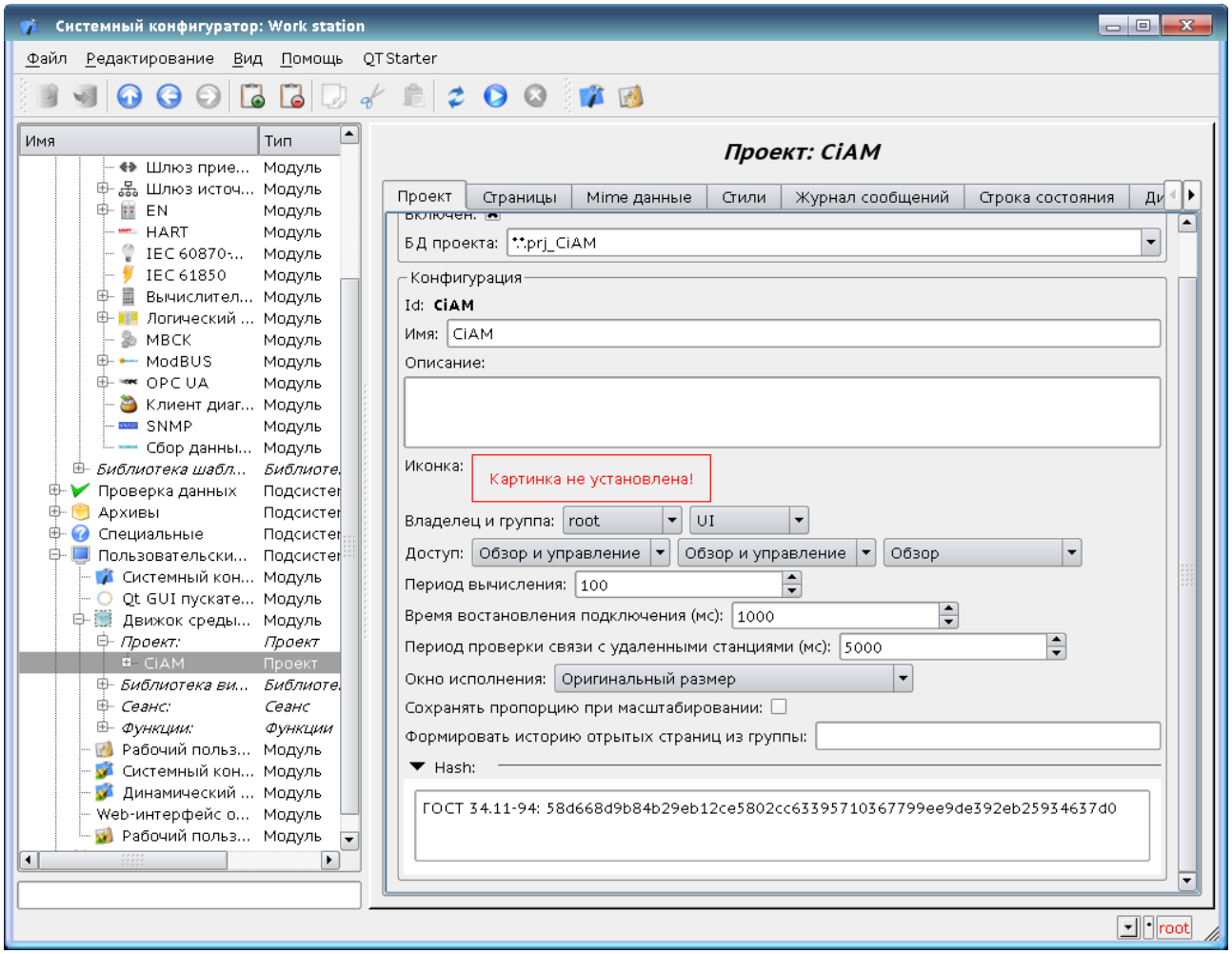

Рисунок 5 - Контрольная сумма проекта

Для подсчета контрольной суммы библиотек виджетов выбрать в системном конфигураторе вкладку «Пользовательские интерфейсы» → «Движок среды визуализации» → «Библиотека виджетов» → «*Имя_библиотеки*». Развернуть пункт *Hash*, в окне будет выведен результат подсчета контрольной суммы выбранной библиотеки (рисунок 6).

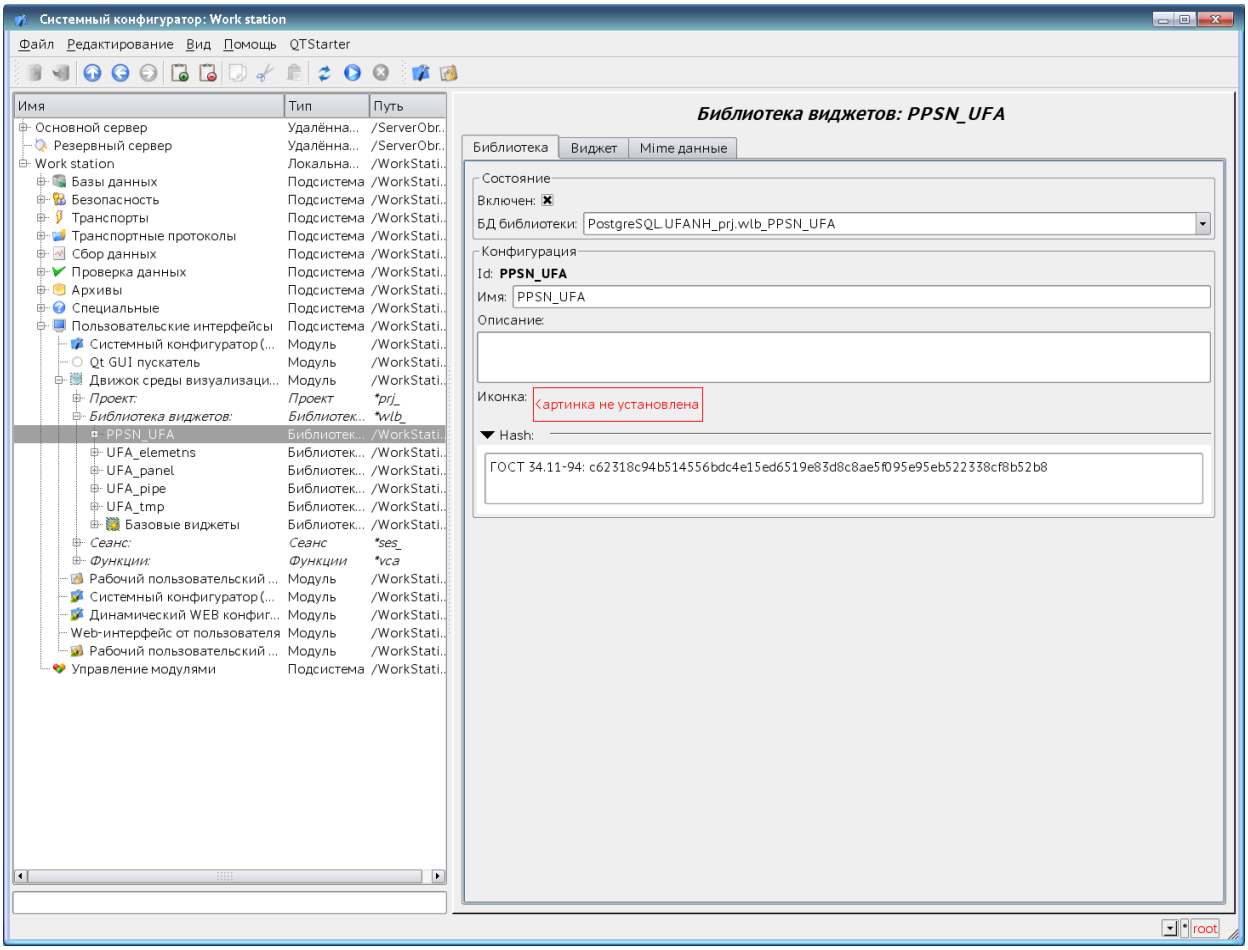

Рисунок 6 - Контрольная сумма библиотеки

4 Входные и выходные данные

Входными данными для программных средств ОС, обеспечивающих контроль целостности, является конфигурационный файл afick.conf, содержащий список папок с установленной ПП «СКАДА А-СОФТ».

Выходными данными являются сообщения *afick* о состоянии целостности файлов ПП «СКАДА А-СОФТ».

Входными данными для программ подсчета контрольных сумм ППО являются конфигурации контролируемых частей ППО.

Выходными данными для программ подсчета контрольных сумм ППО являются контрольные суммы, выводимые на экран.

5 Сообщения

5.1 Сообщения программных средств ОС регламентного контроля целостности базового ПО

В результате запуска afick на экран выводится информация, содержащая:

– перечень измененных папок, добавленных, измененных и удаленных файлов;

– статистику изменений:

– общее количество просканированных файлов;

– общее количество ошибок;

– количество новых файлов;

– количество удаленных файлов;

– количество измененных файлов.

5.2 Сообщения встроенных механизмов контроля целостности ППО ПП «СКАДА А-СОФТ»

Сообщениями встроенного механизма контроля целостности ППО в ПП «СКАДА А-СОФТ» являются подсчитанные контрольные суммы, которые выводятся в соответствующие окна контролируемых элементов.

6 Перечень принятых сокращений

БД	- база данных
ОС	- операционная система
ПО	- программное обеспечение
ПП	- программная платформа
ППО	- прикладное программное обеспечение
СВТ	- средство вычислительной техники
СКАДА(SCADA)	- диспетчерское управление и сбор данных (Supervisory Control And Data Acquisition)

Приложение 1

Конфигурационный файл afick.conf

```
# afick config sample file
# $Id: afick.conf 987 2006-12-21 15:31:16Z gerbier $
# see afick.conf documentation for more informations

####################
# directives section
####################
# binary values can be : yes/1/true or no/0/false
# database : name with full path to database file
database:=/var/lib/afick/afick
# history : name with full path to history file
history := /var/lib/afick/history
# archive : name with full path to directory for archived results
archive := /var/lib/afick/archive
# report_url : where to send the result : stdout/stderr/null
report_url := stdout
# report_syslog : send output to syslog ?
report_syslog := yes
# verbose : (obsolete) boolean value
# use debug parameter below
verbose := no
# debug : set a level of debugging messages, from 0 (none) to 4 (full)
debug := 0
# warn_dead_symlinks : boolean : if set, warn about dead symlinks
warn_dead_symlinks := no
# follow_symlinks : boolean : if set, do checksum on target file (else on
target file name)
follow_symlinks := no
# allow_overload : boolean : if set, allow to overload rules (the last rule
wins), else put a warning
allow_overload := yes
# report_full_newdel : boolean : if set, report all changes, if not set,
report only a summary on top directories
report_full_newdel := no
# warn_missing_file  : boolean : is set, warn about selected files (in this
config), which do not exists
warn_missing_file := no
# running_files : boolean : if set, warn about files changed during a program
run
running_files := yes
# timing : boolean : if set, print timing statistics about the job
timing := yes
# ignore_case : boolean : if set, ignore case on file name
ignore_case := no
#   max_checksum_size   :   numeric   :   only   compute   checksum   on   first
max_checksum_size bytes ( 0 means unlimited)
max_checksum_size := 10000000

# exclude_suffix : list of suffixes to ignore
# text files
exclude_suffix := log LOG html htm HTM txt TXT xml
# help files
exclude_suffix := hlp pod chm
# old files
exclude_suffix := tmp old bak
# fonts
exclude_suffix := fon ttf TTF
# images
exclude_suffix := bmp BMP jpg JPG gif png ico
# audio
exclude_suffix := wav WAV mp3 avi

# exclude_prefix : list of prefixes to ignore
#exclude_prefix :=
```

```
# exclude_re : list of patterns (regular expressions) to ignore (apply on
full path)
#exclude_re :=

#################
# macros section
#################
# used by cron job (afick_cron)
# define the mail adress to send cron job result
@@define MAILTO root@localhost
# truncate the result sended by mail to the number of lines (avoid too long
mails)
@@define LINES 1000
# REPORT = 1 to enable mail reports, =0 to disable report
@@define REPORT 1
# VERBOSE = 1 to have one mail by run, =0 to have a mail only if changes are
detected
@@define VERBOSE 0
# define the nice value : from 0 to 19 (priority of the job)
@@define NICE 18
# = 1 to allow cron job, = 0 to suppress cron job
@@define BATCH 1

###############
# 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
GOST = p+d+i+n+u+g+s+b+gost+m+e+t

##############
# file section
##############
# 3 syntaxe are available :
# file action
#     to scan a file/directory with "action" parameters
# ! file
#     to remove file from scan
# = directory action
#   to scan the directory but not sub-directories
```

```
    # file with blank character have to be quoted

    #=/  DIR
    #/bin     MyRule

    #/boot    MyRule
    #!/boot/map
    #!/boot/System.map

    #/dev p+n
    # to avoid problems with pending usb
    # =/dev/scsi p+n

    #/etc     ETC
    #/etc/mtab ETC - i
    #/etc/adjtime ETC - md5
    # /etc/aliases.db ETC - md5
    # /etc/mail/statistics ETC - md5
    #/etc/motd ETC - i
    # /etc/ntp/drift ETC - i - md5
    # /etc/urpmi/urpmi.cfg Logs
    # /etc/urpmi/proxy.cfg Logs
    # !/etc/cups/certs/0
    # !/etc/map
    # !/etc/postfix/prng_exch
    #!/etc/samba/secrets.tdb
    # !/etc/webmin/sysstats/modules/

    #/lib     MyRule
    #/lib/modules MyRule -m
    # /lib/dev-state MyRule -u

    #/root MSEC
    #!/root/.viminfo
    #!/root/.bash_history
    #!/root/.mc
    # !/root/tmp
    #/var/ftp MyRule
    #/var/log Logs
    # !  /var/log/ksymoops
    # /var/www MyRule
    # ! /var/www/html/snortsnarf
    /boot GOST
    /lib/modules PARSEC
    /etc/security PARSEC
    /etc/pam.d PARSEC
    /lib/x86_64-linux-gnu/security PARSEC
    /lib/security PARSEC
    /sbin PARSEC
    /etc/fstab PARSEC
    /usr/sbin PARSEC
    #file or directory of scada
    /lib/systemd/system/scada.service GOST
    /etc/scada.xml GOST
    /etc/init.d/scada GOST
    /usr/lib/libscada.so.3.0.0 GOST
    /usr/lib/scada GOST
    /usr/share/locale/ru/LC_MESSAGES GOST
    /usr/share/applications/scada.desktop GOST
    /usr/share/doc/scada GOST
    /usr/share/menu/scada GOST
    /usr/bin/scada_start GOST
    /usr/bin/scada GOST
    /usr/lib/libscada.so.3 GOST
    ##########################################
    # to allow easier upgrade, my advice is too separate
    # the default configuration file (above) from your
    # local configuration (below).
    # default configuration will be upgraded
    # local configuration will be kept
##########  put your local config below ####################
```

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					