

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М.В.ЛОМОНОСОВА
(МГУ)

№ госрегистрации 114100140107

УТВЕРЖДАЮ

Проректор - начальник
Управления научной политики и
организации научных исследований
МГУ, д.ф.-м.н.


А.А.Федянин
«29» декабря 2015 г.



О ПРИКЛАДНЫХ НАУЧНЫХ ИССЛЕДОВАНИЯХ

Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации

по теме:

ТЕОРЕТИЧЕСКИЕ ИССЛЕДОВАНИЯ (3-ЕЙ ОЧЕРЕДИ)
ПОСТАВЛЕННЫХ ПЕРЕД ПНИ ЗАДАЧ

(промежуточный)

Этап 3

ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014-2020 годы»

Соглашение о предоставлении субсидии от 27.06.2014 № 14.604.21.0056


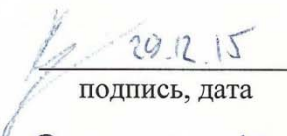
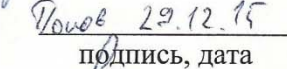
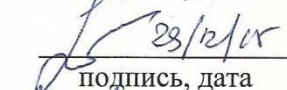
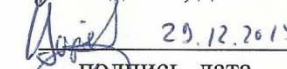
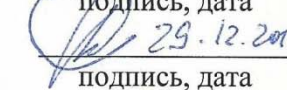
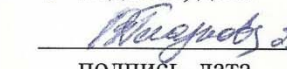
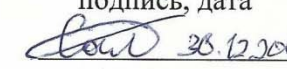
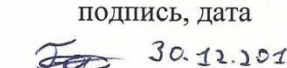
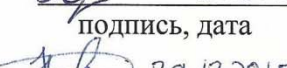
Руководитель ПНИ,
д.ф.-м.н., профессор


29.12.2015
подпись, дата

И.В.Машечкин

Москва 2015

СПИСОК ИСПОЛНИТЕЛЕЙ

Руководитель проекта д.ф.-м.н., профессор	 29.12.2015 подпись, дата	И.В. Машечкин (введение, заключение, реферат, приложение Б)
Исполнители темы		
к.ф.м.н., доцент	 29.12.15 подпись, дата	М.И. Петровский (подразделы 1.1, 1.2, 1.3, приложение А)
м.н.с.	Попов 29.12.15  подпись, дата	И.С. Попов (подраздел 1.2, приложения В, Г, Д)
к.ф.м.н., доцент	 29/12/15 подпись, дата	А.Н. Терехин (подраздел 1.5, приложения Ж, Л)
математик 1-й кат.	 29.12.2015 подпись, дата	Д.В. Царёв (подраздел 1.4, 1.5, приложение А)
программист	 29.12.2015 подпись, дата	А.Ю.Корчагин (приложения Ж, И)
нормоконтролер к.ф.м.н., ассистент	 29.12.2015 подпись, дата	В.В. Глазкова (приложение А, Ж)
математик 1-й кат	 30.12.2015 подпись, дата	П.М.Саликов (разделы 2, 3)
математик 1-й кат	 30.12.2015 подпись, дата	О.Е.Горохов (разделы 2, 3)
программист	 29.12.2015 подпись, дата	Д.А.Никифоров (разделы 2, 3)

РЕФЕРАТ

Отчет 356 с., 1 ч., 96 рис., 3 табл., 18 источников, 9 прил.

КОМПЬЮТЕРНАЯ БЕЗОПАСНОСТЬ, ПОВЕДЕНЧЕСКАЯ БИОМЕТРИЯ, АКТИВНАЯ АУТЕНТИФИКАЦИЯ, ИДЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЕЙ, ДИНАМИКА РАБОТЫ С КЛАВИАТУРОЙ И МЫШЬЮ, ОБНАРУЖЕНИЕ ВНУТРЕННИХ ВТОРЖЕНИЙ, ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ, МАШИННОЕ ОБУЧЕНИЕ, МОДЕЛИРОВАНИЕ ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЯ.

Объектом исследования являются методы анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач компьютерной безопасности, включая задачи активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации.

Цель работы — исследование и разработка комплекса научных решений, направленных на создание программных средств анализа индивидуальных особенностей поведения пользователей компьютерных систем (поведенческой биометрии) при работе в рамках стандартного человеко-машинного интерфейса, с целью создания инновационной технологии построения систем компьютерной безопасности.

В рамках настоящих ПНИ проводились работы, соответствующие третьему этапу «Теоретические исследования (3-ой очереди) поставленных перед ПНИ задач».

В отчете содержится информация о разработке ЭО ПК для обеспечения компьютерной безопасности на основе анализа поведенческой информации работы пользователей компьютерных систем. В приложениях к отчету содержатся: отчет о дополнительных патентных исследованиях, программная документация на ЭО ПК.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	8
1 Разработка ЭО ПК для обеспечения компьютерной безопасности на основе анализа поведенческой информации работы пользователей компьютерных систем.....	10
1.1 Разработка пользовательских сценариев работы с ЭО ПК.....	10
1.1.1 Модуль сбора, предобработки и классификации поведенческой биометрической информации.....	10
1.1.2 Модуль консолидации поведенческой информации.....	14
1.1.3 Модуль построения поведенческих моделей.....	15
1.1.4 Модуль идентификации.....	18
1.1.5 Консоль управления.....	19
1.2 Проектирование архитектуры ЭО ПК.....	20
1.2.1 «Подсистема 1» для сбора и анализа биометрической информации об особенностях динамики работы пользователя с устройствами ввода-вывода.....	21
1.2.2 «Подсистема 2» для сбора и анализа данных о работе пользователя с информационными и вычислительными ресурсами вычислительной системы (системные, прикладные, пользовательские и специальные журналы).....	23
1.2.3 «Подсистема 3» для сбора и анализа информации об особенностях работы пользователя с текстовой информацией, предназначена для решения задач непрерывной фоновой идентификации и обнаружения попыток хищения конфиденциальной информации.....	26
1.3 Реализация структур представления биометрических данных, процедуры их сбора, хранения, управления ими и предварительной обработки.....	29
1.3.1 Реализация структур представления биометрических данных, процедур их сбора, хранения, управления и предварительной обработки для поведенческой биометрической информации об особенностях работы пользователя со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор).....	29
1.3.2 Реализация структур представления биометрических данных, процедур их сбора, хранения, управления и предварительной обработки для поведенческой биометрической информации об особенностях работы пользователя с текстовой	

информацией, включая факты создания, чтения, редактирования, удаления, копирования для различных типов текстовой информации.....	42
1.3.3 Реализация структур представления биометрических данных, процедур их сбора, хранения, управления и предварительной обработки для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами компьютерной системы	50
1.4 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей.....	58
1.4.1 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей со стандартными устройствами ввода-вывода.....	58
1.4.2 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы.....	67
1.4.3 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с текстовой информацией различных типов	83
1.5 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей.....	89
1.5.1 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач активной аутентификации без использования секретной информации.....	89
1.5.2 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач идентификации пользователей.....	95
1.5.3 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач раннего обнаружения внутренних вторжений и попыток хищения конфиденциальной информации.....	111
2 Обеспечение работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по проекту, проведение	

регламентных работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту.....	127
3 Построение программно-аппаратного стенда для проведения экспериментальных исследований и оценки результатов	130
3.1 Программно-аппаратный стенд для проведения исследований и оценки результатов «Подсистемы 1»	131
3.2 Программно-аппаратный стенд для проведения исследований и оценки результатов «Подсистемы 2»	133
3.3 Программно-аппаратный стенд для проведения исследований и оценки результатов «Подсистемы 3»	135
ЗАКЛЮЧЕНИЕ.....	138
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	140
ПРИЛОЖЕНИЕ А Отчет о патентных исследованиях.....	142
ПРИЛОЖЕНИЕ Б Описание применения.....	166
ПРИЛОЖЕНИЕ В Программный компонент «Подсистема 1» ЭО ПК. Текст программы ...	182
ПРИЛОЖЕНИЕ Г Программный компонент «Подсистема 1» ЭО ПК. Описание программы.....	187
ПРИЛОЖЕНИЕ Д Программный компонент «Подсистема 2» ЭО ПК. Текст программы ...	229
ПРИЛОЖЕНИЕ Ж Программный компонент «Подсистема 2» ЭО ПК. Описание программы.....	234
ПРИЛОЖЕНИЕ И Программный компонент «Подсистема 3» ЭО ПК. Текст программы ...	310
ПРИЛОЖЕНИЕ К Программный компонент «Подсистема 3» ЭО ПК. Описание программы.....	315
ПРИЛОЖЕНИЕ Л Акт №2 исполнения обязательств по работам на этапе №3 Плана-графика, выполненных за счет внебюджетных средств	355

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем отчете о НИР применяют следующие термины с соответствующими определениями.

ТЗ	Техническое задание на выполнение прикладных научных исследований (Приложение 1 к Соглашению № 14.604.21.0056 о предоставлении субсидии от 27.06.2014).
Контент документа	Содержимое документа.
ИБ	Информационная безопасность.
ИАД	Интеллектуальный анализ данных.
ППК	Прикладной программный комплекс.
ЭО ПК	Экспериментальный образец программного комплекса.

ВВЕДЕНИЕ

Целью настоящего этапа является проведение теоретических исследования 3-й очереди поставленных перед ПНИ в части исследования и разработки комплекса научных решений, направленных на создание программных средств анализа индивидуальных особенностей поведения пользователей компьютерных систем (поведенческой биометрии) при работе в рамках стандартного человеко-машинного интерфейса, с целью создания инновационной технологии построения систем компьютерной безопасности.

В задачу настоящего этапа исследований входит разработка, на основе результатов, полученных на 1-м и 2-м этапах настоящего ПНИ, экспериментального образца программного комплекса (ЭО ПК) для обеспечения компьютерной безопасности на основе анализа поведенческой информации работы пользователей компьютерных систем.

Отчет ПНИ содержит описание результатов разработки ЭО ПК, в частности: разработку пользовательских сценариев функционирования ЭО ПК; разработку архитектуры ЭО ПК; реализацию структур представления биометрических данных, процедуры их сбора, хранения, управления ими и предварительной обработки; разработку программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей; разработку программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей.

В отчете представлено описание работ, выполненных за счет внебюджетных источников. Представлено описание регулярных работы выполняющихся по обеспечению работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по проекту, проведение регламентных работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту. Также в отчете представлены результаты построения за счет внебюджетных источников программно-аппаратного стенда для проведения экспериментальных исследований и оценки результатов.

Приложения отчета ПНИ содержат:

- разработанную на ЭО ПК программную документацию в составе следующих документов: Описания применения на ЭО ПК в целом; Описание программы на каждый из трех компонентов, составляющих ЭО ПК («Подсистема 1» – для сбора и анализа биометрической информации об особенностях динамики работы пользователя с

устройствами ввода-вывода в рамках стандартного человеко-машинного интерфейса (клавиатура, мышь), предназначена для решения задач активной аутентификации и непрерывной фоновой идентификации; «Подсистема 2» – для сбора и анализа данных о работе пользователя с информационными и вычислительными ресурсами компьютерной системы (системные, прикладные, пользовательские и специальные журналы), предназначена для решения задач непрерывной фоновой идентификации и раннего обнаружения внутренних вторжений; «Подсистема 3» – для сбора и анализа информации об особенностях работы пользователя с текстовой информацией, предназначена для решения задач непрерывной фоновой идентификации и обнаружения попыток хищения конфиденциальной информации); Текст программы на оптическом CD на каждый из трех компонентов, составляющих ЭО ПК;

- отчет о дополнительных патентных исследованиях (получены охранный результаты интеллектуальной деятельности (РИД) - Свидетельство о государственной регистрации программы для ЭВМ №2015661555 от 30.10.2015 «Система двухфакторной аутентификации на основе анализа поведенческой биометрической информации об особенностях работы пользователя с клавиатурой компьютера»);
- копию акта (от 29 декабря 2015г.) об исполнении обязательств по работам на этапе 3 Плана-графика по Соглашению о предоставлении субсидии № 14.604.21.0056 от 27.06.2014г., выполненных за счет внебюджетных средств, по теме: «Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации».

1 Разработка ЭО ПК для обеспечения компьютерной безопасности на основе анализа поведенческой информации работы пользователей компьютерных систем

1.1 Разработка пользовательских сценариев работы с ЭО ПК

Согласно требованиям подпункта 4.1.2.11 ТЗ в ходе выполнения ПНИ разрабатываемый ЭО ПК для обеспечения компьютерной безопасности на основе анализа поведенческой информации работы пользователей компьютерных систем должен иметь следующую логическую структуру, которая представляется в виде совокупности модулей:

1. модуль сбора, предобработки и классификации поведенческой биометрической информации;
2. модуль консолидации поведенческой информации;
3. модуль построения поведенческих моделей;
4. модуль идентификации;
5. консоль управления.

Настоящий подраздел посвящён разработке пользовательских сценариев работы с ЭО ПК на основе приведённой логической структуры в виде модулей и теоретических исследований 1-ой и 2-ой очередей, представленных в отчётах за предыдущие этапы настоящих ПНИ.

1.1.1 Модуль сбора, предобработки и классификации поведенческой биометрической информации

В задачи модуля сбора, предобработки и классификации поведенческой биометрической информации входят:

1. Сбор поведенческой информации в соответствии с заданным режимом сбора (см. рисунок 1), включая:
 - a. информацию, регистрируемую в выбранных журналах операционной системы и журналах приложений;

- b. специально собираемую дополнительную информацию, например, информацию о фактах работы с внешними устройствами, о фактах сетевых соединений, о фактах работы с электронными сообщениями и документами;
 - c. информацию о динамике работы пользователя с устройствами ввода-вывода (клавиатура, мышь+монитор);
 - d. «теневые копии» «перехваченных» электронных текстовых документов, сообщений и web-контента, с которым работает пользователь
2. Нормализация и предобработка собранных данных (см. рисунок 1):
- a. выявление значений и заполнение обязательного набора атрибутов, содержащих время и длительность события, имя пользователя, компьютера, приложения, с которым связана собранная информация и т.д.;
 - b. первичная предобработка и агрегация событий, не требующая связи с модулем идентификации, например, вычисление длительности события работы процесса по парам событий «запуск» – «остановка» процесса, агрегация суммарных объемов переданной или полученной информации по последовательности операций чтения-записи файла и т.д.;
3. Промежуточное локальное хранение собранной информации с целью оптимизации нагрузки на сеть передачи данных (в соответствии с заданным режимом передачи) или в случае отсутствия соединения с модулями консолидации и/или идентификации (см. рисунок 1);

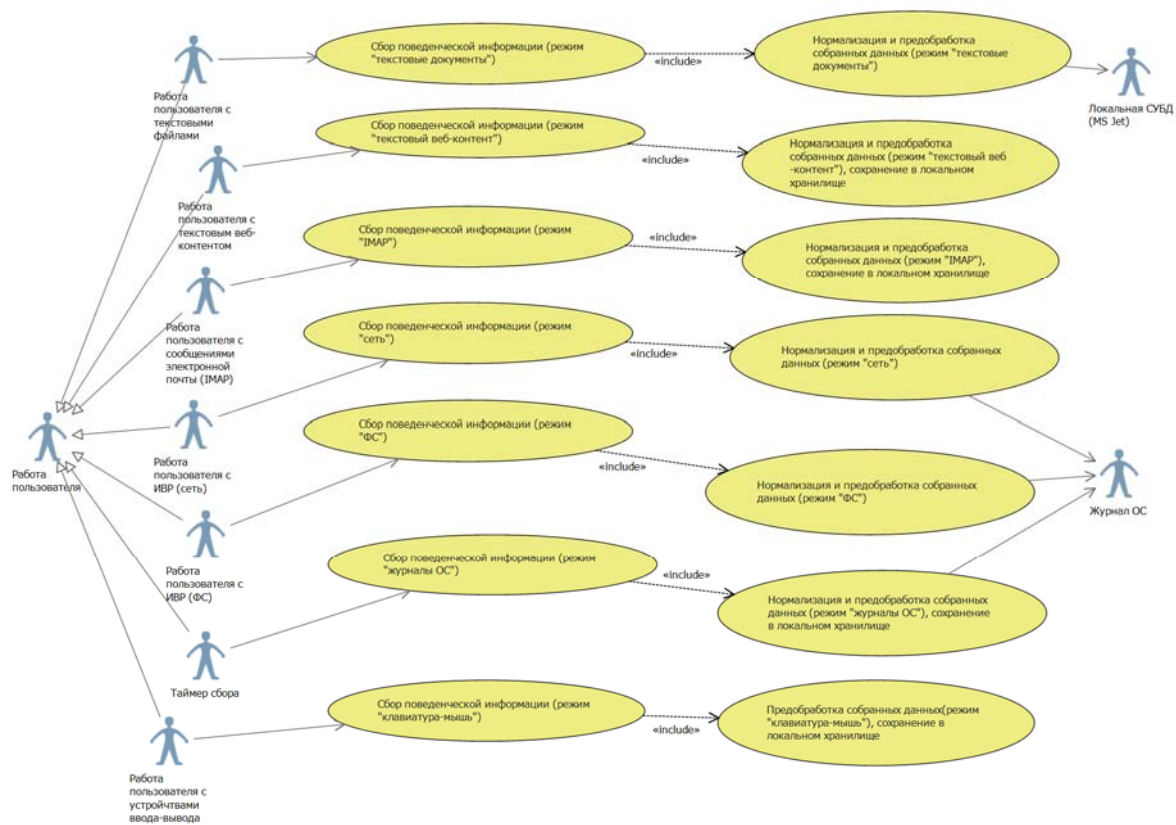


Рисунок 1 — Сбор поведенческой информации, нормализация и предобработка собранных данных, промежуточное локальное хранение собранной информации.

4. Передача собранной информации модулю консолидации в соответствии с заданным режимом передачи (см. рисунок 2);



Рисунок 2 — Передача собранной информации модулю консолидации.

5. Классификация собранной информации путем передачи ее модулю идентификации в соответствии с заданным режимом идентификации, получение ответа и реакция на него, например, запрос дополнительного пароля или отказ в доступе пользователю, если

модуль идентификации распознал его текущую активность как аномальную или опасную (см. рисунок 3);

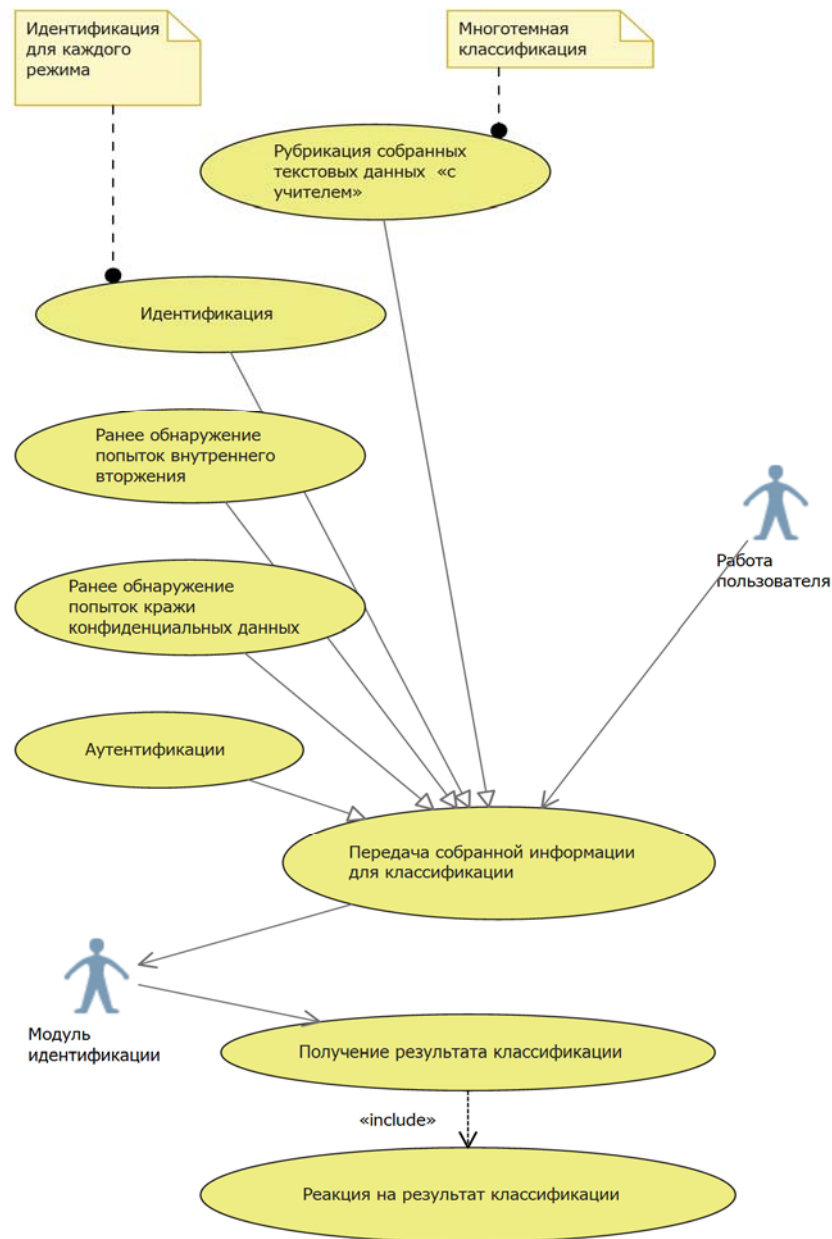


Рисунок 3 — Классификация собранной информации.

6. Задание режимов его работы (см. рисунок 4).

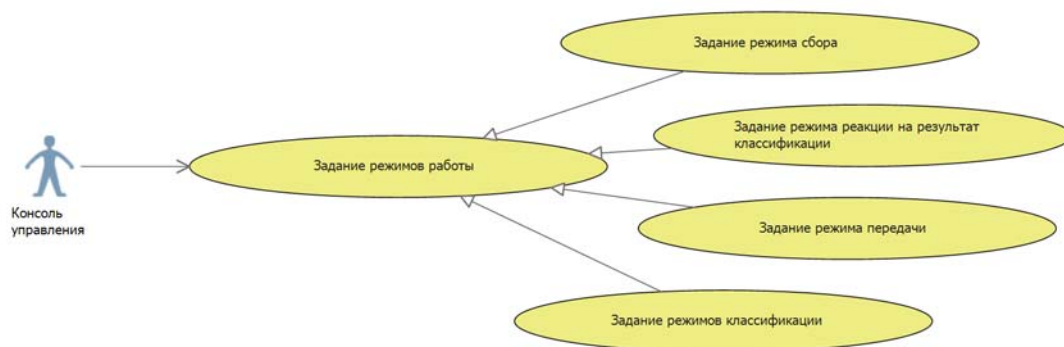


Рисунок 4 — Задание режимов работы модуля сбора.

1.1.2 Модуль консолидации поведенческой информации

Модуль консолидации поведенческой информации, предназначенный для хранения и управления поведенческой биометрической информацией, включая хранение теневого копий текстовых данных, журналов работы пользователей с вычислительными и информационными ресурсами защищаемой компьютерной системы, а также с устройствами ввода-вывода (см. рисунок 5).

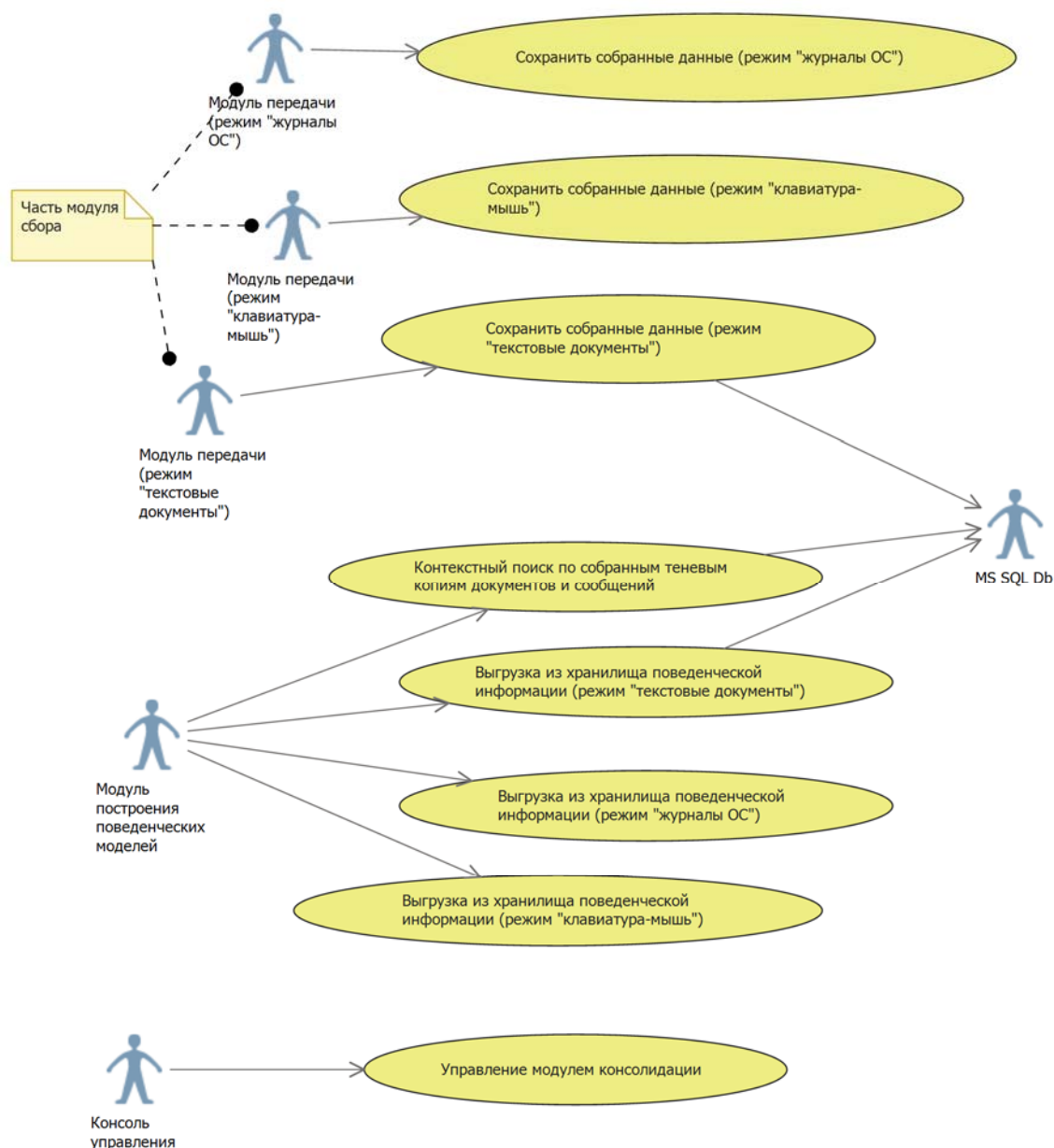


Рисунок 5 — Модуль консолидации.

1.1.3 Модуль построения поведенческих моделей

Модуль построения поведенческих моделей, предназначенный для решения следующих задач:

1. формирование и выгрузка из хранилища поведенческой информации в витрину данных срезов собранных событий, расчет агрегированных показателей, формирование динамических отчетов для оперативного многомерного анализа статистической информации о работе пользователей корпоративной сети (см. рисунок 6);

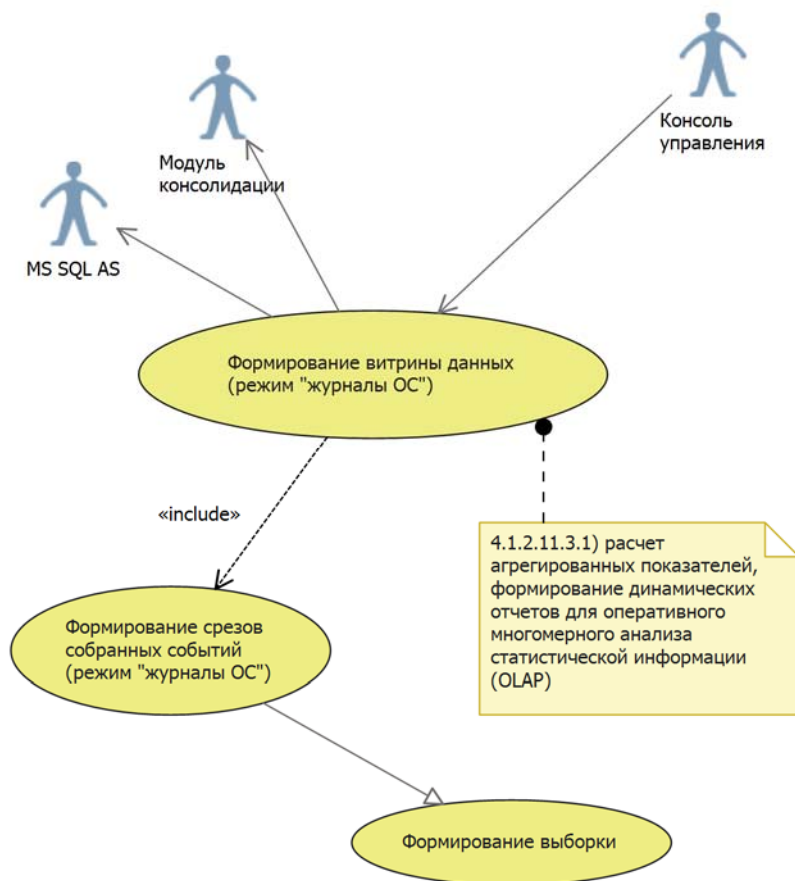


Рисунок 6 — Формирование витрины данных.

2. выявление знаний в собранных текстовых данных, формирование интерактивной базы знаний о потоках текстовой информации, циркулирующих в защищаемой компьютерной сети, включая решение следующих задач предобработки текстовых данных (см. рисунок 7):

- a. автоматическое аннотирование больших текстовых документов и выявление ключевых тематик в потоках коротких текстовых сообщений;
- b. рубрикация собранных текстовых данных на основе машинного обучения «с учителем», т.е. формирование моделей распознавания тематик документов, формируемых экспертом, на основе подобранных им примеров;
- c. группировка собранных текстовых данных на основе машинного обучения «без учителя» с выявлением ключевых тематик и ключевых слов, характерных как для отдельных пользователей, так и для их групп.

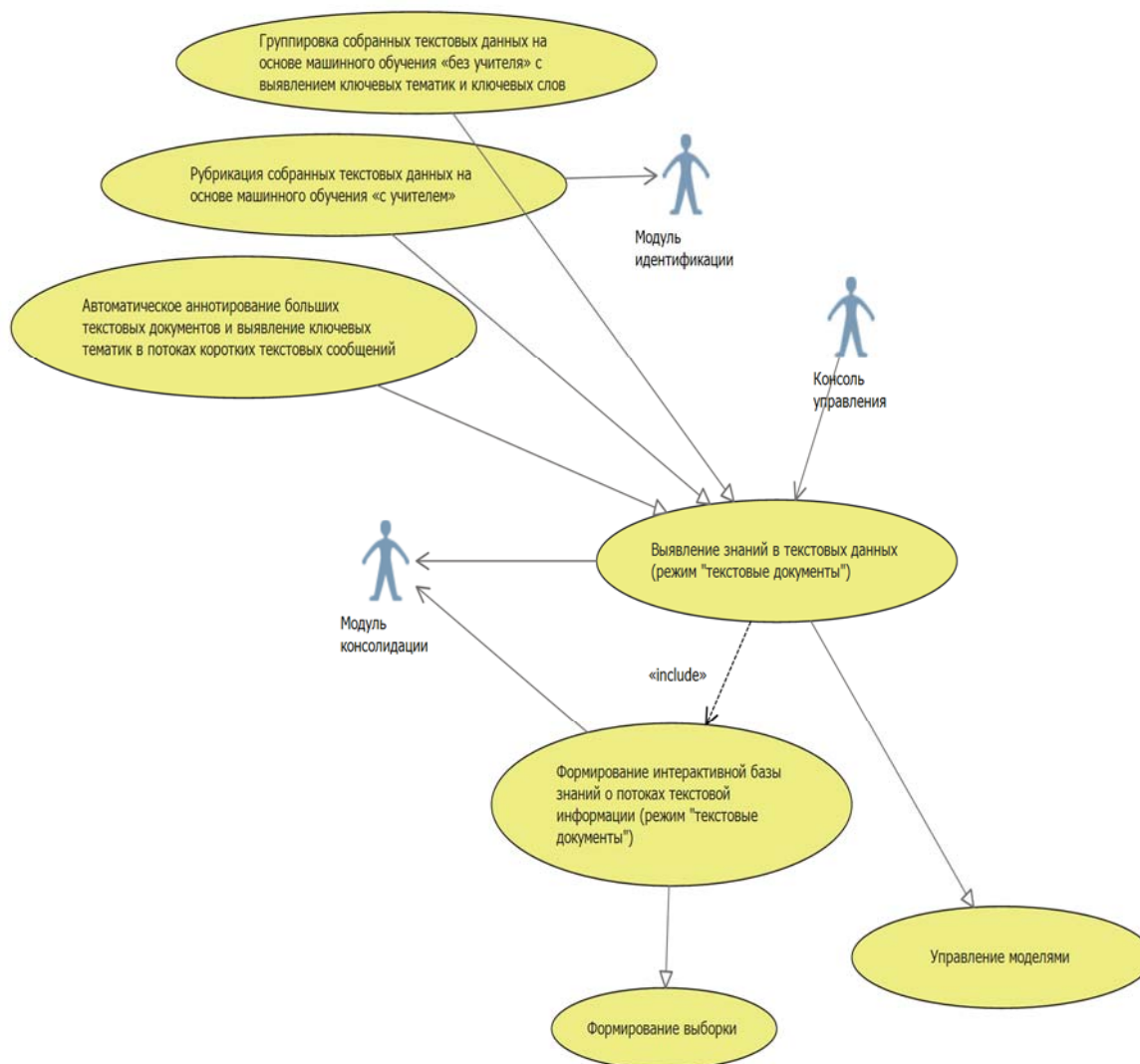


Рисунок 7 — Выявление знаний в собранных текстовых данных.

3. построение, визуализация, валидация и управление поведенческими моделями пользователей для решения задач (см. рисунок 8):

- a. аутентификации с использованием информации об особенностях работы со стандартными устройствами ввода-вывода;
- b. фоновой идентификации пользователей с использованием информации об особенностях работы со стандартными устройствами ввода-вывода, информации о работе с информационными и вычислительными ресурсами защищаемой компьютерной системы, информации об особенностях работы с текстовыми данными;
- c. раннего обнаружения попыток внутреннего вторжения или кражи конфиденциальных данных с использованием информации об особенностях

работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы, информации об особенностях работы с текстовыми данными.

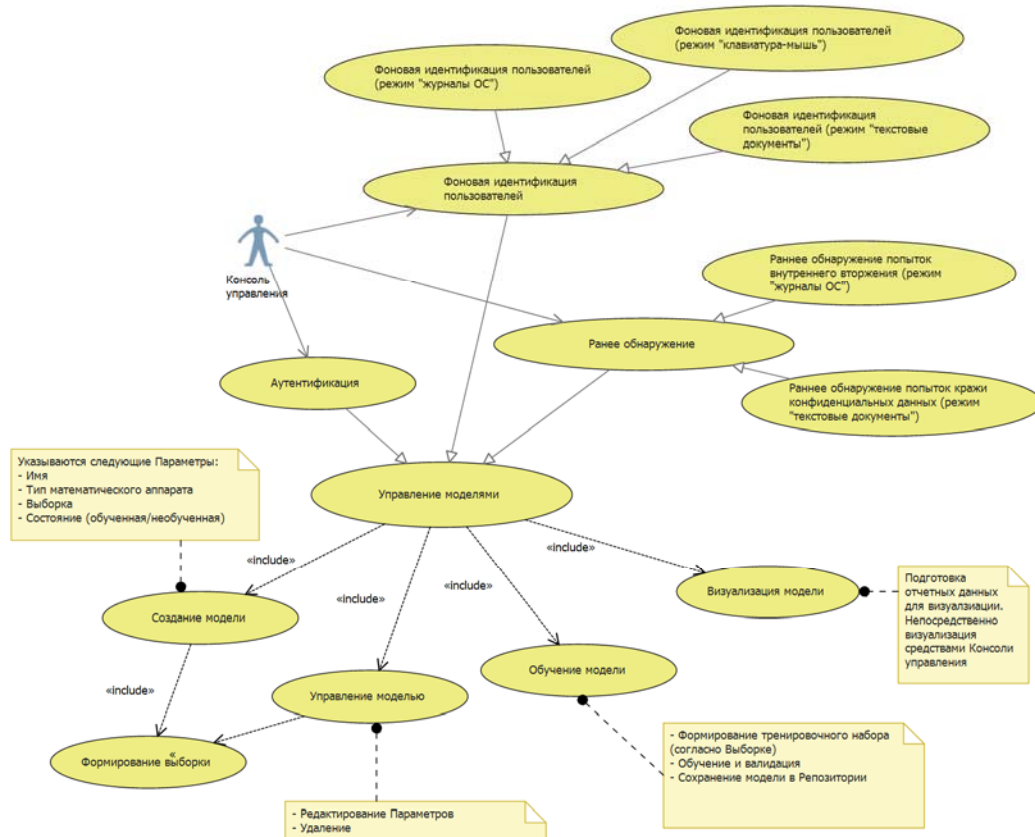


Рисунок 8 — Построение, визуализация, валидация и управление поведенческими моделями.

1.1.4 Модуль идентификации

Модуль идентификации, предназначенный для решения следующих задач (см. рисунок 9):

1. аутентификация и фоновая идентификация пользователя на основе его поведенческих моделей по запросу агента;
2. идентификация и классификация действий пользователя в соответствии с поведенческими моделями по запросу агента;
3. классификация фрагментов текстовой информации в соответствии с моделями языково-независимой предобработки текстовых данных, обнаружение конфиденциальной информации по запросу модуля сбора, предобработки и классификации поведенческой биометрической информации или модуля построения поведенческих моделей;
4. оповещение пользователя об обнаруженных аномалиях или нарушениях.

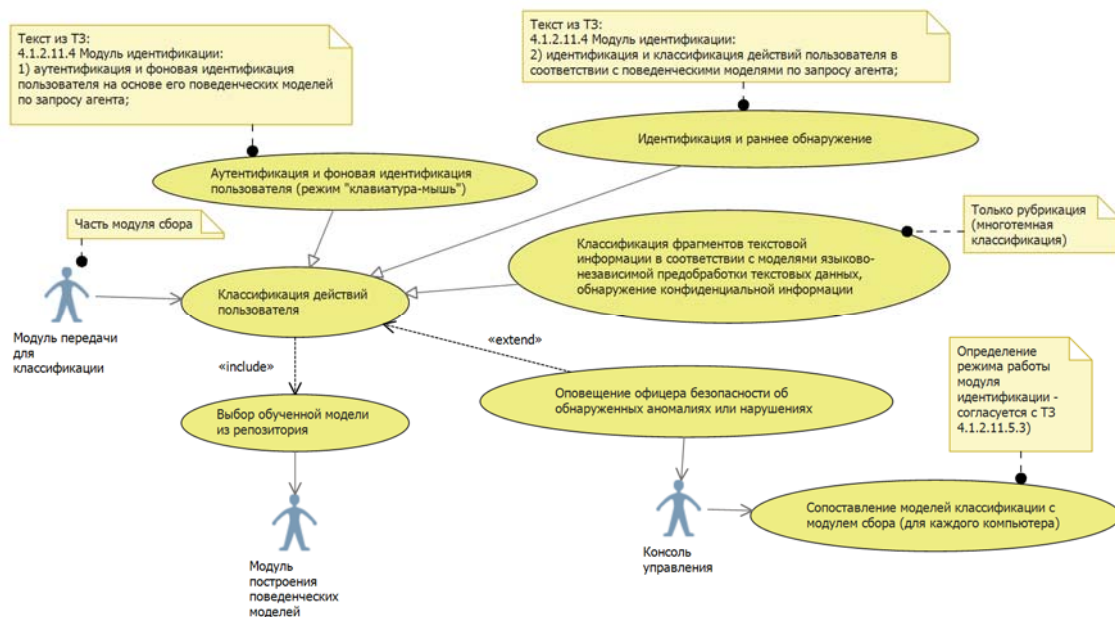


Рисунок 9 — Модуль идентификации.

1.1.5 Консоль управления

Консоль управления, предназначенная для решения задач (см. рисунок 10):

1. распространение, установка и конфигурация ЭО ПК;
2. определение режимов работы модуля сбора, предобработки и классификации поведенческой биометрической информации, включая режимы сбора, передачи, идентификации и классификации собираемой поведенческой информации;
3. управление модулями консолидации, построения поведенческих моделей и идентификации.

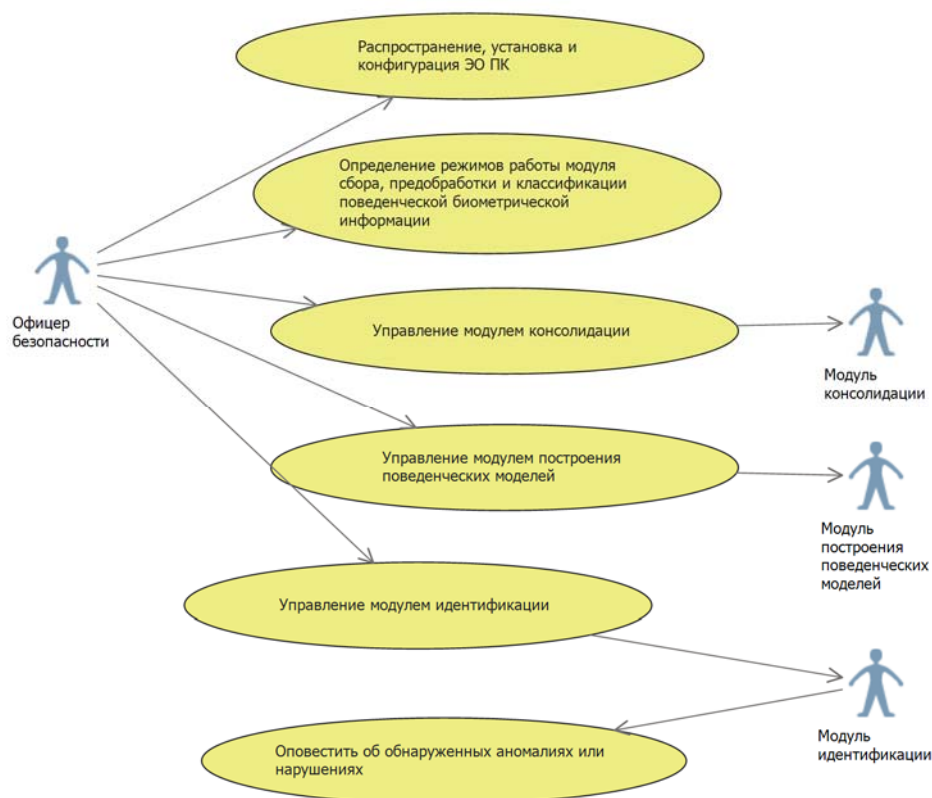


Рисунок 10 — Консоль управления.

1.2 Проектирование архитектуры ЭО ПК

Разработка архитектуры ЭО ПК проводилась в соответствии с требованиями ТЗ п. 3.11.2, а также с результатами ПНИ, полученными на 1-м и 2-м этапах.

ЭО ПК состоит из трех подсистем:

- «Подсистема 1» – для сбора и анализа биометрической информации об особенностях динамики работы пользователя с *устройствами ввода-вывода* в рамках стандартного человеко-машинного интерфейса (клавиатура, мышь), предназначена для решения задач *активной аутентификации* и *непрерывной фоновой идентификации*.
- «Подсистема 2» – для сбора и анализа данных о работе пользователя с *информационными и вычислительными ресурсами* компьютерной системы (системные, прикладные, пользовательские и специальные журналы), предназначена для решения задач *непрерывной фоновой идентификации* и *раннего обнаружения внутренних вторжений*.

- «Подсистема 3» – для сбора и анализа информации об особенностях работы пользователя с *текстовой информацией*, предназначена для решения задач непрерывной *фоновой идентификации* и обнаружения попыток *хищения конфиденциальной информации*.

1.2.1 «Подсистема 1» для сбора и анализа биометрической информации об особенностях динамики работы пользователя с устройствами ввода-вывода

Общая схема организации подсистемы (см. рисунок 11):

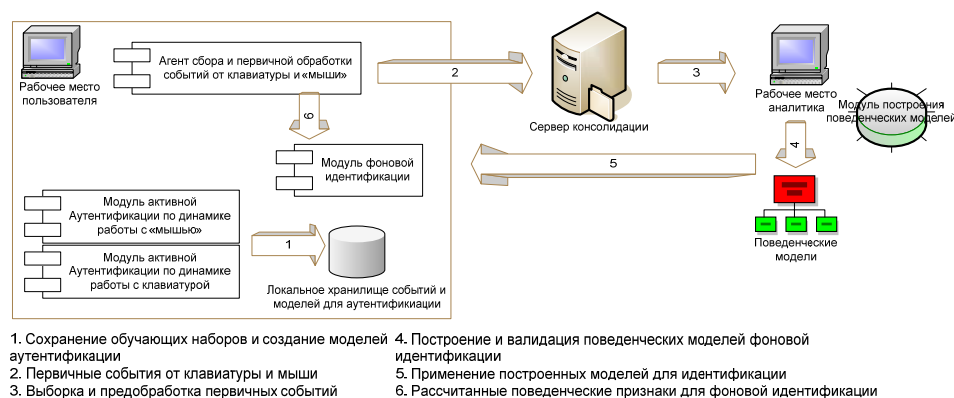


Рисунок 11 — Общая схема «Подсистемы 1».

Подсистема состоит из следующих компонент:

1. Модуль *активной аутентификации* на основе анализа динамики *клавиатурного ввода* ключевого (не секретного) слова, реализован на базе технологии GINA и решает задачи:
 - a. *Построение* поведенческой модели, включая ввод и локальное хранение обучающего набора - серии повторов ключевого слова, выявление отличительных признаков динамики ввода, настройка и построение одноклассовой модели распознавания по тренировочному набору.
 - b. *Применение* поведенческой модели на этапе входа пользователя в систему с использованием технологии GINA для подтверждения или отказа в доступе в систему.

- c. *Управление* настройками обучения и применения, а также созданными поведенческими моделями.
- 2. Модуль *активной аутентификации* на основе анализа динамики работы пользователя с манипулятором *мышь* при вводе (не секретного) графического символа на основе сгенерированного шаблона, реализован на базе технологии GINA и решает задачи:
 - a. *Построение* поведенческой модели, включая ввод и локальное хранение обучающего набора - серии повторов графического ввода символов по шаблону, выявление отличительных признаков динамики ввода, настройка и построение одноклассовой модели распознавания по тренировочному набору.
 - b. *Применение* поведенческой модели на этапе входа пользователя в систему с использованием технологии GINA для подтверждения или отказа в доступе в систему.
 - c. *Управление* настройками обучения и применения и созданными поведенческими моделями.
- 3. *Агент сбора и первичной обработки* событий о работе пользователя с устройствами ввода-вывода для непрерывной фоновой идентификации, реализованного по технологии Windows hook и решающего следующие задачи:
 - a. *Сбор* системных событий о работе пользователя с клавиатурой и манипулятором *мышь*.
 - b. *Сохранение* собранных событий на сервере консолидации.
 - c. Разбиение потоков пользовательских событий на временные окна и сессии, *расчет отличительных признаков* поведения по полученным временным окнам.
 - d. *Передача* полученных векторов признаков в модуль *онлайн идентификации*.
- 4. *Сервер консолидации* – централизованное хранилище первичных событий о работе пользователей с устройствами ввода вывода, реализован на файловой системе.
- 5. *Рабочее место аналитика* – комплекс программных компонент, реализованных на Python, решающих задачи:
 - a. *предобработки и подготовки* данных на основе собранных событий;
 - b. *построения и валидации* одноклассовых и многоклассовых *моделей фонового распознавания* пользователей;
 - c. поддержка *репозитария моделей* для применения на этапе онлайн идентификации.

6. Модуль *идентификации* – применяет построенные поведенческие пользовательские модели к векторам признаков полученных от агентов сбора с целью распознавания пользователя в режиме близком к онлайн.

1.2.2 «Подсистема 2» для сбора и анализа данных о работе пользователя с информационными и вычислительными ресурсами вычислительной системы (системные, прикладные, пользовательские и специальные журналы)

Общая схема организации подсистемы изображена на рисунке 12.

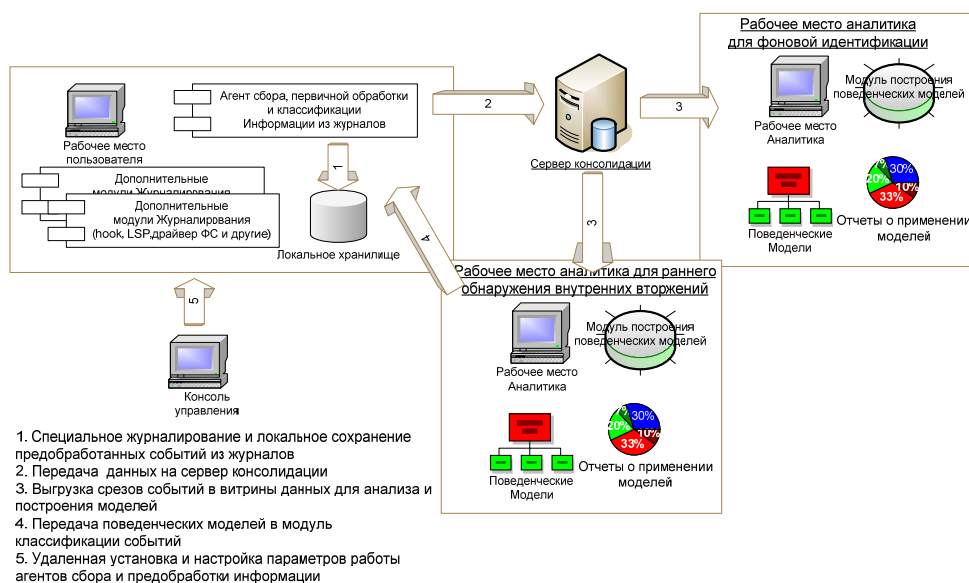


Рисунок 12 — Общая схема «Подсистемы 2».

Подсистема состоит из следующих компонент:

1. *Агенты сбора и предобработки* поведенческой информации о работе пользователей с *вычислительными и информационными ресурсами* компьютерной системы, реализованные как набор запускаемых модулей, сервисов и динамических библиотек, в их задачу входит:
 - a. *сбор стандартной журналируемой информации* в соответствии с заданной *политикой сбора* — выбранные журналы операционной системы, журналы приложений, информацию о работе с внешними устройствами;

- b. *журналирование* с помощью дополнительных модулей и *сбор специальной информации* в соответствии с заданной *политикой сбора*, включая информацию о работе с внешними и разделяемыми сетевыми устройствами, данные о сетевых соединениях (с помощью компоненты LSP, внедряемой в стек протокола TCP/IP), статистику о работе с устройствами ввода-вывода (с помощью компоненты на базе windows hook), информацию о работе с выбранными директориями файловой системы (с помощью собственного драйвера ФС);
 - c. *нормализация и предобработка* собранных данных: выявление значений и заполнение обязательного набора атрибутов, содержащих время и длительность события, имя пользователя, компьютера, приложения, с которым связана собранная информация; первичная предобработка и агрегация событий, не требующая связи с серверами идентификации, такая как, вычисление длительности события работы процесса по парам событий «запуск»-«остановка» процесса, агрегация суммарных объемов переданной или полученной информации по последовательности операций чтения-записи файла и т.д.
 - d. *промежуточное локальное хранение* собранной информации с целью оптимизации нагрузки на сеть передачи данных (в соответствии с заданной *политикой передачи*) или в случае отсутствия соединения с серверами *консолидации* и/или *идентификации*;
 - e. *передача* собранной информации на *сервер консолидации* в соответствии с заданной *политикой передачи*;
 - f. *классификация* собранной информации путем взаимодействия с модулем *раннего обнаружения вторжения*;
 - g. получение и установка *обновлений* ПО агентов и политик их работы.
2. *Сервер консолидации* - специализированное надежное высокопроизводительное хранилище, реализованное на C++ как отдельный исполняемый процесс и предназначенное для хранения и управления поведенческой информацией о работе пользователей с вычислительными и информационными ресурсами защищаемой компьютерной системы.

3. *АРМ аналитика безопасности* для решения задачи *раннего обнаружения попыток внутреннего вторжения* с использованием информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы. Реализован как ММС (Microsoft Management Console) компонент, и позволяет:
- a. формировать и *выгружать* из хранилища поведенческой информации в *витрину данных* срезы собранных событий
 - b. рассчитывать агрегированные показатели, формировать *динамические отчеты* для оперативного многомерного анализа статистической информации о работе пользователей корпоративной сети;
 - c. Строить (обучать), визуализировать, валидировать, применять и управлять *моделями поведения* для раннего обнаружения попыток *внутренних вторжений*, позволяющих оценить степень аномальности каждого отдельного события в журнале.
 - d. *Экспортировать* полученные модели для решения задачи раннего обнаружения внутренних вторжений в режиме близком к онлайн.
4. *АРМ аналитика безопасности* для решения задачи *фоновой идентификации пользователей* с использованием информации об особенностях работы с информационными и вычислительными ресурсами защищаемой компьютерной системы. Реализован как ММС компонент, и позволяет:
- a. формировать из выгруженных в *витрину данных* собранных событий тренировочные и валидационные наборы, задавать параметры формирования временных окон событий;
 - b. рассчитывать агрегированные показатели о поведении пользователей в виде *временных рядов* с использованием методов латентно-семантического анализа слабо структурированных данных в журналах;
 - c. строить (обучать), визуализировать, валидировать, применять и управлять *моделями поведения* для *фоновой идентификации* пользователей, позволяющих обнаружить изменение закономерностей в поведении пользователей при работе с информационными и вычислительными ресурсами защищаемой компьютерной системы на основе методов анализа временных рядов.

5. *Модуль раннего обнаружения вторжения* - работает в режиме близком к онлайн, использует поведенческие модели, подготовленные на *APM аналитика безопасности* для классификации событий в соответствии с поведенческими моделями.

6. *Консоль управления*. Рабочее место администратора, реализованное как ММС компонент, используется для централизованного управления компонентами всей подсистемы, включая:

- a. распространение, *установка* и конфигурация агентов подсистемы;
- b. определение *политик работы агентов*, включая политики сбора, передачи, идентификации и классификации собираемой поведенческой информации;

1.2.3 «Подсистема 3» для сбора и анализа информации об особенностях работы пользователя с текстовой информации, предназначена для решения задач непрерывной фоновой идентификации и обнаружения попыток хищения конфиденциальной информации.

Подсистема состоит из следующих модулей (см. рисунок 13):

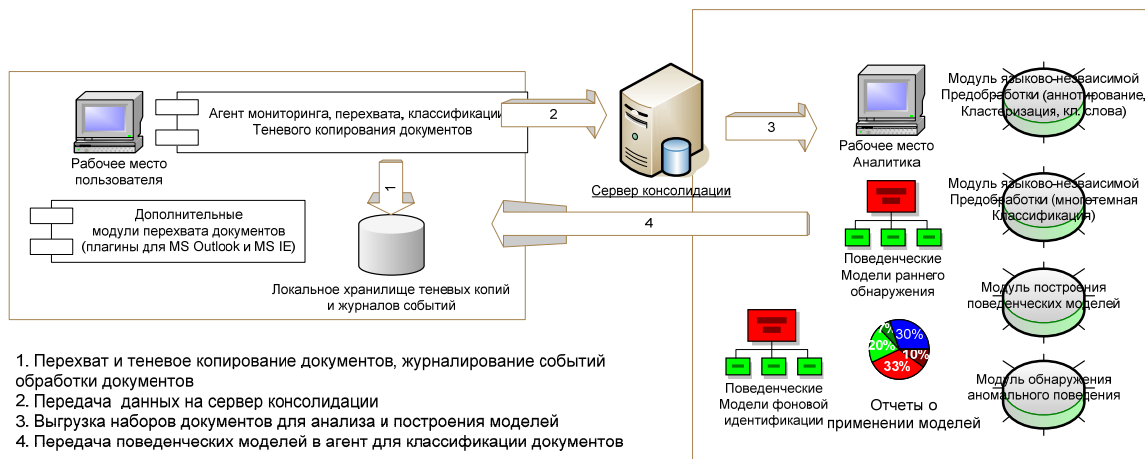


Рисунок 13 — Общая схема «Подсистемы 3».

1. *Агент мониторинга*. Программный агент, устанавливаемый на рабочее место пользователя, включающий драйвер файловой системы и совокупность параллельно

работающих модулей — *модуль сбора* и *модуль классификации*. Реализован в виде набора Windows служб и драйвера-минифilterа, обеспечивающего мониторинг файловой системы.

- a. *Модуль сбора* осуществляет сбор поведенческой информации о работе пользователей с текстовыми данными, включая перехват фактов работы и содержимого документов с помощью минидрайвера файловой системы, а также с помощью плагинов для обозревателя Internet Explorer (на технологии VHO) и плагина для Microsoft Office Outlook. Поддерживает «теневое копирование» документов и сохранение журналов событий действий с ними, а также обеспечивает промежуточное локальное хранение, передачу собранной информации модулю консолидации в соответствии с заданным режимом передачи;
 - a. *Модуль классификации* служит для применения поведенческих моделей раннего обнаружения попыток хищения конфиденциальной информации в режиме близкого к онлайн времени к собираемой в локальном хранилище поведенческой информации.
6. *Модуль консолидации поведенческой информации*. Реализован как исполняемый файл на языке C#. Является программным агентом, который служит для обеспечения консолидации в едином хранилище поведенческой информации, получаемой от *агентов мониторинга*. Кроме хранения поведенческой информации в задачи модуля также входит предоставление доступа к хранилищу поведенческой и текстовой информации для формирования аналитических выборок, которые используются при создании и применении поведенческих моделей. Доступ к хранилищу используется и для применения других средств выявления знаний в поведенческих данных путём агрегированного анализа как операций с электронными документами, так и их текстового содержимого. Например, доступ к хранилищу также использоваться для применения средств языково-независимой предобработки текстовой информации.
7. Модуль для *языково-независимой* предобработки собираемой текстовой информации, включая автоматическое аннотирование больших текстовых документов и группировки собранных текстовых данных на основе машинного обучения «без учителя» (кластеризация) с выявлением ключевых тематик и ключевых слов; Реализован в виде COM-компоненты и набора исполняемых файлов, написанных на языке C++ и реализующих математический аппарат. Вызывается из рабочего места аналитика.

8. Модуль для *языково-независимой* предобработки собираемой текстовой информации для рубрикации собранных текстовых данных на основе машинного обучения «с учителем» (многотемная классификация с не взаимоисключающими классами). Реализован в виде динамических библиотек, написанных на языке C++. Вызывается из рабочего места аналитика.
9. *Модуль построения поведенческих моделей*. Реализован в виде СОМ-компоненты и набора исполняемых файлов, написанных на языке C++ и реализующих математический аппарат. На основе выборки данных поведенческой текстовой информации модуль выполняет процедуру построения поведенческой модели, для решения задач *фоновой идентификации* и *раннего обнаружения попыток хищения конфиденциальной информации*. После формирования соответствующих структур модели производится её сохранение в *хранилище моделей*.
10. *Модуль идентификации* служит для применения поведенческих моделей в отложенном режиме, т.е. поведенческие модели применяются к выборке поведенческих данных, сформированной из поведенческих данных хранилища *модуля консолидации*. При этом формирование выборок может осуществлять либо администратор в ручном режиме, либо формирование выборок и последующее применение моделей будет происходить автоматически по заданному расписанию и параметрам формирования соответствующих выборок. Реализован в виде СОМ-компоненты и набора исполняемых файлов, написанных на языке C++ и реализующих математический аппарат.
11. *Рабочее место аналитика*. Реализована в виде ММС компоненты и представляет графический интерфейс, реализующий следующие варианты использования ЭО ПО:
 - a. *Создание* поведенческих моделей — формирование выборки поведенческой текстовой информации из сервера консолидации, выбор типа и настроек моделей, обучение моделей;
 - b. *Применение* поведенческих моделей — формирование выборки поведенческой информации, на которой будет применяться обученная модель, выбор модели, применение ее к выборке, формирование *отчетов* о результатах применения;
 - c. *Настройка отложенного режима* — задание расписания и других параметров для автоматического построения и применения моделей типа *фоновой идентификации* к данным из хранилища модуля консолидации;

1.3 Реализация структур представления биометрических данных, процедуры их сбора, хранения, управления ими и предварительной обработки

На первом и втором этапах ПНИ, согласно требованиям ТЗ (пункты 3.4, 3.6, 3.8), были разработаны:

- структуры данных, методы сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор);
- структуры данных, методы сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.
- структуры данных, методы сбора, предобработки, хранения и управления для поведенческой информации об особенностях работы пользователя с текстовой информацией, включая факты создания, чтения, редактирования, удаления, копирования для различных типов текстовой информации.

На основе данных результатов ПНИ, согласно требованиям ТЗ (пункт 3.11.3) были реализованы соответствующие структуры представления биометрических данных, процедуры их сбора, хранения, управления ими и предварительной обработки.

1.3.1 Реализация структур представления биометрических данных, процедур их сбора, хранения, управления и предварительной обработки для поведенческой биометрической информации об особенностях работы пользователя со стандартными устройствами ввода-вывода (клавиатура, мышь, монитор)

1.3.1.1 Модули авторизации пользователя

Сбор биометрической информации об использовании пользователем клавиатуры и мыши реализован в виде программных модулей, интегрируемых в систему авторизации операционной системы Windows, со следующими базовыми функциями:

- Формирование лога событий нажатия/отпускания клавиш клавиатуры и лога движений мыши в виде временных текстовых файлов
- Обработка данных файлов и преобразование во внутренние структуры данных для последующего использования непосредственно в процедуре авторизации.

Для функционирования схемы авторизации с учетом биометрических данных пользователя при использовании клавиатуры/мыши требуется предварительно составить его биометрическую модель поведения. Построение модели поведения начинается со сбора информации о нажатии клавиш (движении мыши) во время процедуры обучения во временный текстовый файл. Далее, на основании этих данных строится модель нечёткого множества таким образом, чтобы эталонные попытки авторизации (по мыши или клавиатуре) имели максимальную принадлежность к множеству.

1.3.1.1.1 Описание структур данных

Для данных, поступающих от клавиатуры, используются следующие характеристики:

- Время события (нажатия/отпускания)
- Интервал между двумя нажатиями клавиш
- Интервал между нажатием и отпусканием клавиши
- Интервал между нажатием текущей клавиши и отпусканием предыдущей

Для данных, поступающих от мыши в процессе очерчивания тестового шаблона, сохраняются следующие характеристики:

- Время
- X-координата положения курсора мыши на экране
- Y-координата положения курсора мыши на экране.

Построение модели поведения пользователя при авторизации с использованием клавиатуры/мыши реализовано на основе данных, полученных с соответствующих устройств ввода, в виде математической модели нечёткого множества со следующими характеристиками соответствующих структур данных:

- Матрица ядер
- Положение центра нечёткого множества
- Список расстояний от центра до элементов нечёткого множества

- Список принадлежностей элементов ко множеству

1.3.1.2 Структуры представления данных в модулях идентификации пользователя

Для процедуры идентификации результатом работы модуля сбора информации о поведении пользователя при работе с устройствами ввода является сессия – набор файлов, соответствующий промежутку времени, в течении которого подсистема сбора была активна.

Сессия представляет собой директорию, содержащую в своём названии имя пользователя, дату и время начала сессии:

<имя пользователя>_<гггг_мм_дд чч_мм>.

Директория включает в себя следующие файлы:

- KEYBOARD.csv
- MOUSE.csv
- systeminfo.log

В текстовые файлы *KEYBOARD.csv* и *MOUSE.csv* в формате CSV записываются данные об активности пользователя, собранные за время сессии. В файл *systeminfo.log* в формате JSON записывается информация об аппаратно-программной конфигурации клиентской машины, соответствующей времени начала сессии.

1.3.1.2.1 Описание структур для событий ввода

В данном пункте описывается содержимое файлов *KEYBOARD.csv* и *MOUSE.csv*.

Общие поля:

- time - время события в формате hh:mm:ss.mmmmm
- username - имя пользователя-автора события
- processname - короткое имя процесса, сгенерировавшего событие
- PID – идентификатор процесса, сгенерировавшего событие
- hwnd /hwnd_orig – идентификатор окна, в котором произошло событие
- hookmode – в настоящее время не используется, всегда 0

Поля клавиатуры:

- virtualcode – виртуальный код кнопки

scancode – аппаратно-зависимый код кнопки
keyup – 0/1 (кнопка нажата/отжата)
prev_keystate – предыдущее состояние keyup кнопки
repeat-count - число повторений при удержании кнопки
alt_down – 0/1 кнопка Alt нажата/ненажата
extended_kbd – 0/1 кнопка принадлежит основной части клавиатуры/дополнительному блоку
langID – код языка
keybdID – код раскладки клавиатуры
raw_data_debug – отладочные данные

Поля мыши:

message_id – код события, соответствует значениям констант Windows для событий мыши (WM_MOUSEMOVE и пр.)
x_pos – X-координата события
y_pos – Y-координата события
hitTest - значение параметра «hit-test»
extra_info – отладочные данные

1.3.1.2.2 Описание программно-аппаратной информации

В данном пункте описывается содержимое файла *systeminfo.log*. Информация из данного файла является необходимой для последующей обработки информации о событиях ввода, непосредственным образом влияя на дальнейший расчёт признаков.

Поля с информацией о программе-сборщике:

Version – версия сборщика
Опции запуска (секция Options):
provocationMode – значение маски опций режима провокации

Системные поля:

OS – информация о версии Windows
ComputerName – имя компьютера
IPAddress – стандартный IP-адрес компьютера (адрес локального хоста)

Поля с характеристиками процессора:

frequencyMhz – частота ядра в Mhz

architectureType – константа с типом архитектуры (см. таблица 1)

oemID – oemID процессора

count – число ядер процессора

type – тип процессора

activemask – активная маска

Таблица 1 — Числовые константы типов архитектур процессора.

9	x64 (AMD or Intel)
6	Intel Itanium Processor Family (IPF)
0	x86
0xffff	Unknown processor

Поля с характеристиками дисплея:

width - ширина основного дисплея в пикселях

height - высота основного дисплея в пикселях

hmmSize - ширина основного дисплея в миллиметрах

vmmSize - высота основного дисплея в пикселях

numMonitors – количество дисплеев в системе

Поля с характеристиками клавиатуры (см. таблица 2):

type/subtype – константа типом/подтипом клавиатуры

functionKeysType – число функциональных клавиш

repeatDelay – константа задержки удерживаемой клавиши перед повторным событием, значение в диапазоне от 0 (250 миллисекунд) до 3 (1 секунда)

repeatSpeed – константа скорости генерируемых повторных событий при удержании клавиши, значение в диапазоне от 0 (примерно 2.5 повтора в секунду) до 31 (30 повторов в секунду)

Таблица 2 — Числовые константы типов/подтипов клавиатур.

Клавиатура	Тип	Подтип
US (стандартная)	4	0
Japan1	7	1
Japan2	7	2
Korean	8	3

Поля с характеристиками мыши:

doubleClickMs – максимальный диапазон времени в миллисекундах, в течении которого 2 одиночных клика мыши считаются за двойной клик

doubleClickXYRange – размер прямоугольника в пикселях, 2 одиночных клика мыши внутри которого могут считаться за двойной клик

buttonsCount – число кнопок мыши

dragXYRange - размер прямоугольника в пикселях, который ограничивает возникновение события drag&drop при нажатии и последующем движении мышью

speed – константа значения скорости указателя мыши, диапазон от 1 (медленно) до 20 (быстро)

acceleration – значение ускорения (по умолчанию 1), применяемое к скорости.

wheelHorizontal – число символов для одного события горизонтального скроллинга

wheelVertical - число строк для одного события вертикального скроллинга

Поля с информацией об устройствах ввода (секция InputDevices):

Keyboard<№> - системная строка-описание клавиатуры с номером <№>. Нумерация осуществляется с 0.

Mouse<№> - системная строка-описание манипулятора мышь с номером <№>. Нумерация осуществляется с 0.

1.3.1.3 Процедуры сбора, хранения и управления биометрическими данными об особенностях работы пользователей со стандартными устройствами ввода-вывода

Активация подсистемы сбора модуля идентификации пользователей осуществляется путем запуска исполняемого файла приложения. Дальнейшая работа производится автоматически в фоновом режиме. Активность пользователя записывается в течении всего промежутка времени (сессии), пока приложение активно. Запуск может осуществляться традиционными способами запуска фоновых приложений (опции Startup, скрипты и пр.) при входе пользователя в систему. Режим работы определяется комбинацией опций запуска(флагов), переданных в командной строке.

Для сохранения данных об активности пользователя при работе с устройствами ввода на сервере (или любой другой директории), запуск производится с ключом:

-f <путь к директории>

например: `hookstub.exe -f "\\SERVER\Biodata"`

В данном примере директория "Biodata", расположенная на сервере "SERVER" должна быть доступна с правами достаточными для записи процессами, от имени которых будут запускаться экземпляры подсистемы сбора на персональных компьютерах пользователей..

Для указания периодичности записи накопленных событий в файлы используется опция:

-T <время_в_миллисекундах>

например, инициировать запись каждые 3 секунды: `hookstub.exe -T 3000`

По умолчанию, запись накопленных данных в целевые файлы производится один раз в секунду.

Целевой каталог для накопления данных определяется опциями запуска согласно приведенному описанию. При запуске приложения в целевом каталоге создаются (в случае отсутствия) логи программы, которые могут использоваться в дальнейшем в информационно-отладочных целях. Данные текущей сессии сохраняются в каталог с именем *<имя_компьютера>* , соответствующим локальному имени целевого компьютера. Внутри данного каталога сохраняются каталоги, соответствующие сессиям (промежуток между очередным запуском и завершением сборщика). Название сессии формируется из имени текущего пользователя и даты/времени начала сессии. Внутри каталога создаются три конечных файла, соответствующих текущей сессии: описание активной аппаратно-

программной конфигурации, активность пользователя при работе с клавиатурой, активность пользователя при работе с манипулятором мышь.

Окончанием активной сессии сбора и завершением сбора данных является завершение процесса приложения. Завершение может быть осуществлено естественным образом – при выключении компьютера/завершении сеанса Windows для данного пользователя и пр., либо форсированным способом: при одновременном нажатии ключевой комбинации клавиш (Ctrl+Shift+S) и подтверждении действия.

1.3.1.4 Предварительная обработка структур представления биометрических данных об особенностях работы пользователей со стандартными устройствами ввода-вывода

При анализе данных от клавиатуры и мыши, собираемых с помощью подсистемы сбора информации об активности пользователя с устройствами ввода в модулях идентификации был выявлен ряд проблем, возникающих при использовании различных версий ОС Windows, определенных характеристик, аппаратуры, системных настроек. Ввиду того, что часть данных проблем не может быть решена непосредственно на стороне программы-сборщика, было решено ввести стадию предобработки (фильтрации данных), которая будет проводиться перед этапом выделения признаков. Стадия предобработки позволяет избавиться от шумовых событий, негативно влияющих на рассчитываемые для пользователя признаки, тем самым это позволит увеличить их качество и повысить точность итоговой классификации.

1.3.1.4.1 Предобработка данных о динамике работы пользователя с клавиатурой

Предобработка (фильтрация) данных, полученных с клавиатуры, осуществляется в два этапа.

На первом этапе удаляются события, про которые заранее известно, что они являются нехарактерными для пользователя) (например события, собранные при использовании клавиатурного тренажеров/симуляторов) - было экспериментально установлено, что если не отсекал такие события, то точность классификации на открытом и закрытом наборах значительно ухудшается. Предположительно, такое поведение наблюдается из-за того, что клавиатурный тренажер провоцирует пользователя работать в нестандартном для себя режиме, влияя как на темп ввода, так и на его методику (данный клавиатурный тренажер

позиционируется как тренер, обучающий десятипальцевой методике ввода, и пользователь, который, например, в обычной жизни для набора пользуется только одной рукой/только определенными пальцами, вынужденно использует все пальцы)

На втором этапе происходит удаление шумовых событий (например, есть событие нажатия, но нет события отжатия, и наоборот). Делается это в два прохода по списку событий. На первом проходе все клавиши помечаются как не нажатые, и алгоритм идет сверху вниз и выполняет следующие шаги:

1. Определяется тип события (нажатие/отжатие)
2. Если это нажатие, то клавиша с данным кодом помечается как нажатая
3. Если это отжатие, то проверяется статус клавиши: если она была нажата, то помечаем ее как не нажатую и переходим к следующему событию, иначе (если она была отжата, но события нажатия нет) данное событие удаляется и алгоритм переходит к следующему событию

Как только достигается конец списка, все клавиши помечаются как не отжатые, и начинают выполняться следующие шаги (проход по списку событий осуществляется снизу вверх):

1. Определяется тип события (нажатие/отжатие)
2. Если это отжатие, то клавиша с данным кодом помечается как отжатая
3. Если это нажатие, то проверяется статус клавиши: если она была отжата, то помечаем ее как не отжатую и переходим к следующему событию, иначе (если она была нажата, но события отжатия нет) данное событие удаляется и алгоритм переходит к следующему событию.

1.3.1.4.2 Предобработка данных о динамике работы пользователя с мышью

В ходе анализа собранных данных были выявлены следующие проблемы и найдены возможные пути их решения:

Фильтрация записей, не содержащих типа событий или координат мыши

В ходе проведенных экспериментов было выявлено, что в ряде собираемых записей могут отсутствовать некоторые критически важные поля, например: поле `message_id` (тип события) или поля `x_pos` и `y_pos` (координаты мыши по осям `x` и `y` соответственно). Возникновение таких записей может являться следствием аварийного завершения

программы-сборщика событий, сбояв ОС, а также ошибками, возникающими при обработке потока событий операционной системой. Данные записи было решено удалять.

Фильтрация записей, содержащих типы событий, не характеризующие движение мыши, нажатия клавиши или работу с колесиком

К событиям, относящимся к мыши, также принадлежат события, описывающие положение мыши относительно запущенных окон, взаимодействие с ними. Данные события не характеризуют индивидуальные особенности работы пользователя с манипулятором, а используются операционной системой для своих нужд. Ввиду этого данные события также было решено удалять, если они появились в потоке записанных событий.

Удаление дубликатов событий

В ходе анализа собранных данных было выявлено, что в ряде случаев могут присутствовать события, совпадающие по всем полям, например, последовательные события движения мыши, в которых не менялась временная метка. Данные дубликаты событий также подвергались удалению.

Сортировка записей по времени

При анализе данных были выявлены ситуации, при которых события мыши могли быть обработаны операционной системой не в порядке их возникновения. Для решения данной проблемы производится сортировка записей по времени.

Стандартизация кодов событий

Было выявлено, что в ряде случаев, при работе пользователя с окном какого-либо приложения, события мыши, происходящие не в рабочей области окна, а в области меню или границы окна, имеют другие коды (коды, относящиеся к группе NONCLIENTAREA). Данные события используются операционной системой для взаимодействия пользователя с контекстным меню приложения и для изменения размеров окна. Они дублируют стандартные события, однако имеют относительно них предопределенное смещение. В связи с этим была проведена стандартизация кодов событий: событиям, происходящим в области NONCLIENTAREA, были присвоены обычные коды (коды событий, происходящих в основной области окна). В полученном после преобразования событии (в битовом поле record_info) устанавливался специальный флаг NONCLIENTAREA.

Стандартизация событий двойного клика

В ходе анализа документации и проведения серии экспериментов на семи различных компьютерах (включая ноутбуки) было выявлено, что последовательность событий, представляющих двойной клик, на разных компьютерах разная. Это связано с тем, что в ОС Windows не все окна приложений генерируют события двойного клика - данный параметр зависит от наличия у окна приложения стиля CS_DBLCLKS. Также, в зависимости от драйвера мыши, настроек самой операционной системы и свойств окна приложения может генерироваться различная последовательность событий движения мыши во время двойного клика. Данные отличия приведены в таблице 3.

Таблица 3 — Возможные наборы событий, составляющих двойной клик, в ОС Windows.

Название компьютера/ Название программы	Comp1	Comp2	Comp3	Comp4	Comp5	Comp6	Comp7
Word	513 514 512 515 514	513 514 512 515 512 514	513 512 514 515 514	513 514 512 515 512 514	Программа не установлена	513 514 512 515 512 514	513 514 512 515 512 514
Explorer	513 514 512 513 514	513 514 512 513 514	513 512 514 512 513 514	513 512 514 512 515 512 514	513 514 512 513 514	513 512 514 512 515 512 514	513 514 512 513 514
Explorer (desktop)	513 512 514 512 515 512 514	513 512 514 512 515 512 514	513 512 512 514 512 515 512 514	513 512 514 512 515 512 514	513 512 514 512 515 512 514	513 512 512 514 512 515 512 514	513 512 514 512 515 512 514
Excel	513 512 514 512 515 512 514	513 514 512 515 512 514	513 512 514 512 515 512 514	513 514 512 515 512 514	Программа не установлена	513 514 512 515 514	513 512 512 514 515 512 514
Chrome	513 514 512 515 514	513 514 512 515 514	513 512 514 512 515 514	Программа не установлена	513 514 512 515 514	513 512 514 512 515 514	513 512 514 512 515 514
Notepad	513 514 512 515 514	513 514 512 515 514	513 512 514 512 515 514	513 514 512 515 514	513 514 512 515 514	513 514 512 515 514	513 514 512 515 514
Skype	513 514 515 514	513 514 515 514	Программа не установлена	513 514 515 514	Программа не установлена	513 514 515 514	513 514 515 514
Calculator	513 514	513 514	Программа не установлена	513 514	Программа не установлена	513 514	513 514

	515 514	515 514		515 514		515 514	515 514
Paint	513 514 512 515 514	513 514 512 515 514	Программа не установлена	513 514 512 515 514	Программа не установлена	513 514 512 515 514	513 514 512 515 514

Возможны двойные клики левой, правой и средней клавишами мыши.

Для левой клавиши мыши на разных компьютерах могут генерироваться следующие последовательности событий во время двойного клика:

513-514-513-514 или 513-514-515-514, где

513 – событие нажатия левой клавиши,

514 – событие отпущения левой клавиши,

515 – событие двойного клика левой клавишей.

При этом между каждыми данными событиями могут стоять события движения мыши (512) в любом количестве (например, 513-512-512-514-513-512-514).

Для правой и средней кнопок мыши данные комбинации аналогичны (отличаются только коды событий работы с клавишами).

В связи с этим, необходимо в базе с записанными событиями искать двойные клики самостоятельно.

В ходе первоначальной предобработки все события 515 заменялись на 513 (для других клавиш мыши – аналогично).

Далее стояла задача выделять последовательности событий, составляющих двойной клик. Данная последовательность (на примере левой клавиши) должна содержать события 513-514-513-514, возможно разделенные событиями 512 в любом количестве.

Для двойного клика играют роль такие системные параметры, как максимальное время, в течение которого два одинарных клика засчитываются за двойной, а также размер площадки в пикселях, за которую нельзя заступать во время выполнения двойного клика. Также стоит учитывать, что во время совершения двойного клика не может меняться программа (процесс), в которой выполняется работа.

В результате было реализовано программное средство, которое находит такие последовательности, второе событие 513 в последовательности заменяет на 515 (для левой клавиши, для других – аналогично), оставляет все события 512 без изменения и изменяет для событий, входящих в двойной клик, значение соответствующего флага (поле record_info,

пятый бит данного поля выставляется в единицу, если данное событие входит в последовательность событий двойного клика).

Для двойных кликов правой и средней клавишами изменяется этот же бит (также выставляется в единицу).

В ходе данных операций представление двойных кликов во всех снятых данных стандартизуется.

Удаление записей о движении мыши, при которых не менялись координаты курсора

Было установлено, что в ряде случаев может генерироваться последовательность событий движения мыши, в то время как на самом деле она не двигалась (в частности, сама мышь может генерировать такую последовательность, проблема может быть в драйвере мыши). В связи с этим было принято решение удалять все события движения мыши (WM_MOUSEMOVE), у которых с течением времени не менялась координата.

Различие между операцией перетаскивания и одинарным кликом

В ряде случаев между нажатием и отпусканием клавиш мыши возникают события движения мыши. Для того чтобы различить событие одинарного клика и событие перетаскивания была использована временная метка (системный параметр), а также размер площадки, при попадании событий нажатия и отпускания в которую, в системе фиксируется одинарный клик.

Анализ ситуаций, при которых в данных присутствуют события движения мыши с разными координатами, но одинаковым временем события

В анализируемых данных были обнаружены случаи, когда подряд идут события движения мыши с одинаковой временной меткой, но с различными координатами. Данная проблема носит программно-аппаратную природу. Ввиду того, что данная проблема имела единичный характер (возникла на одном из пользователей), поиск ее решения был отложен.

Невозможность определения монитора, с которым работает пользователь (при работе с персональным компьютером с несколькими мониторами)

Ввиду того, что несколько мониторов, подключенных к одному компьютеру, представляются в ОС Windows в едином координатном пространстве, и пользователь может менять положение мониторов относительно друг друга, расширяя координатную ось по всем

направлениям, определение монитора, с которым работал пользователь, а также координат внутри конкретного монитора, затруднено. Однако, на текущем этапе исследований, данная информация не является критичной, поэтому поиск решения данной проблемы был отложен.

Анализ перемещения курсора мыши при использовании цифровых клавиш (NumPad)

Системные настройки ОС Windows позволяют использовать клавиши секции NumPad для манипуляций с мышью (движение мыши, нажатие клавиш) при ее возможном отсутствии. При этом данные события никоим образом не отличаются от событий, возникающих при работе с манипулятором типа мышь (коды событий одинаковы при работе с мышью и с навигационной секцией NumPad). Однако, по умолчанию данная настройка отключена, а также подавляющее большинство пользователей не знают о ее наличии, поэтому поиск решения данной проблемы был отложен.

1.3.2 Реализация структур представления биометрических данных, процедур их сбора, хранения, управления и предварительной обработки для поведенческой биометрической информации об особенностях работы пользователя с текстовой информацией, включая факты создания, чтения, редактирования, удаления, копирования для различных типов текстовой информации

Исходя из представленного в пункте 1.2.3 описания архитектуры «Подсистемы 3» (для сбора и анализа информации об особенностях работы пользователя с потоками текстовой информации) следует, что для реализации сбора поведенческой биометрической информации учувствуют агент мониторинга и сервер консолидации (модуль консолидации). Далее приводятся исследования данных программных модулей, которые включают реализацию структур представления биометрических данных, процедур их сбора, предобработки, хранения и управления. Приведённые работы основываются на уже завершённых работах 1 этапа настоящих ПНИ, а именно «Разработка структур данных, методов сбора, предобработки, хранения и управления для поведенческой информации об особенностях работы пользователя с текстовой информацией» [1].

1.3.2.1 Реализация агента мониторинга поведенческой информации

Как уже отмечалось (в пункте 1.2.3) агент мониторинга устанавливается на наблюдаемое рабочее место пользователя и состоит из совокупности параллельно работающих модулей — *модуль сбора* и *модуль классификации*. По существу, данные агенты выполняют сбор, обработку и передачу поведенческих данных, такие агенты в [2] отнесены к типу *информационных* — управление информацией из множества различных источников, в том числе и физически разных.

Согласно п. 4.1.2.5.1 ТЗ требуется реализовать сбор и обработку фактов работы пользователя с документами следующих типов и форматов:

1. с любыми электронными документами в виде локальных файлов, файлов на внешних носителях и разделяемых сетевых ресурсах в защищаемой компьютерной системе, в текстовых форматах;
2. с незакодированными сообщениями электронной почты, получаемыми и передаваемыми по протоколам IMAP и HTTP (на почтовые Web-системы) с использованием одного из веб-обозревателей (Microsoft Internet Explorer версии 9 и выше);

Реализация 1-го пункта приведённого требования будет достигнута за счёт мониторинга изменений, происходящих в файловой системе (ФС) наблюдаемого компьютера.

Реализация 2-го пункта требования сводится к 1-ому пункту путём выполнения:

- Сохранения электронных почтовых сообщений, получаемых (по протоколу IMAP) и отправляемых пользователем с помощью почтового клиента, установленного на наблюдаемом компьютере, в виде файлов в специальные директории агента мониторинга;
- Сохранения отправляемых форм и посещаемых web-страниц по протоколу HTTP пользователем с помощью браузера, установленного на наблюдаемом компьютере, в виде файлов в специальные директории агента мониторинга.

Таким образом, факты работы пользователей с документами всех типов из 2-го пункта требования 4.1.2.5.1 ТЗ и их содержимое (контент) будут собраны и обработаны путём мониторинга ФС за счёт их сохранения в виде файлов. Поэтому далее речь пойдёт только о реализации мониторинга файлов, а разработка программных компонент, сохраняющих документы из 2-го пункта требования ТЗ в файлы, приведена в подпунктах 1.4.3.5 и 1.4.3.6.

Поток текстовой информации — последовательность изменений состояния электронного документа и описание операций, вызвавших данные изменения [1]. Для

текстовых файлов на компьютере и подключаемых внешних носителях это операции: создание, изменение, чтение, перемещение, удаление. В случае создания файла или его первой регистрации в системе агенту мониторинга достаточно сохранять его содержимое и путь, в случае перемещения сохранять новый путь, в случае изменения содержимого – новое содержимое (задача теневого копирования). В случае удаления – просто пометить его как удаленный. После чего всю собранную информацию необходимо передавать в центральное хранилище модуля консолидации. Таким образом, на каждом компьютере, для которого производится мониторинг, необходимо решение агентом следующих задач:

1. Модуль сбора:

- a. *Мониторинг файловой системы.* Получение данных об изменениях в файловой системе локального компьютера для мониторинга операций с файлами и подключения внешних носителей;
- b. *Фильтрация файлов и операций.* Для разных организаций разные файлы и операции над ними представляют интерес, поэтому необходимо иметь средство задания правил, по которым будут определяться требуемые файлы, которые мы будем называть «документами», и требуемые операции над ними;
- c. *Сохранение операций с документами.* Сохранение в локальное хранилище.
- d. *Сохранение содержимого документов.* Сохранение теневых копий.
- e. *Передача данных об информационных потоках в центральное хранилище.*

2. Модуль классификации:

- a. *Актуализация поведенческих моделей* пользователей для наблюдаемого компьютера;
- b. *Применение поведенческих моделей.* Выборка данных из локального хранилища, применение модели, сохранение результата.

Из приведённого списка задач следует, что реализация структур представления биометрических данных, процедур их сбора, предобработки, хранения и управления входит в задачи *модуля сбора*. Поэтому далее приводятся описания реализации решений данных задач. Модуль классификации агента мониторинга служит только для применения поведенческих моделей к собранным данным, поэтому он далее в настоящем пункте рассматриваться не будет.

1.3.2.2 Мониторинг файловой системы

В ОС Windows используются специальные структуры данных ядра, называемые IRP-пакетами (англ. *I/O Request Packet* — пакет запроса ввода/вывода), для обеспечения обмена

данными между приложениями и драйвером, а также между драйвером и драйвером. Таким образом, обращение к файлам — это фактически формирование соответствующих IRP и посылка их драйверам файловой системы [3]. Операции быстрого ввода/вывода (англ. *Fast I/O*), специально предназначенные для быстрого синхронного ввода/вывода в кэшируемых файлах, мы не учитываем, т.к. они служат для передачи данных непосредственно между пользовательскими буферами и системным кэшем в обход файловой системы и стеков драйверов устройств [4].

Фильтрация IRP — это общий и универсальный механизм, его используют при разработке антивирусов, файловых архиваторов, файлового шифрования и т.д. Для реализации фильтрации IRP есть документированные возможности — написание драйвера и присоединение его к стеку драйверов файловой системы. Начиная с Windows XP SP2, возможно написание драйверов – *минифильтров* ФС [5], предназначенных специально для мониторинга (и фильтрации) IRP-пакетов ФС. Важной особенностью минифильтров является поддержка двунаправленного небуферизированного канала обмена сообщениями между драйвером и приложениями пользовательского режима, в качестве которых обычно используют службы Windows [6]. Общий механизм мониторинга IRP изображён на рисунке 14.

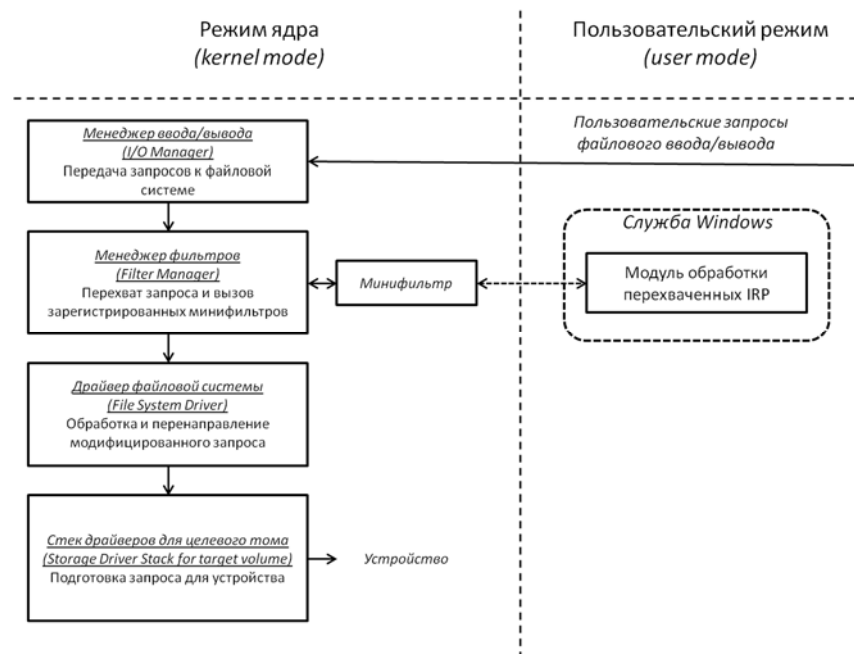


Рисунок 14 — Мониторинг IRP.

Для мониторинга файловой системы требуется разработать драйвер-минифильтр ФС. Для ведения жизненного пути файла, достаточно перехватывать операции открытия и закрытия: IRP-пакеты IRP_MJ_CREATE и IRP_MJ_CLEANUP.

- *Создание нового файла или открытие существующего файла* — факт наличия IRP на открытие файла (IRP_MJ_CREATE).
- *Изменение файла* — наличие флага с доступом на запись в IRP на открытие.
- *Перемещение файла* — несоответствие путей при IRP_MJ_CREATE и IRP_MJ_CLEANUP.
- *Удаление файла* — наличие соответствующего флага в IRP на закрытие.

Таким образом можно отслеживать любые обращения к файлам, однако не все файлы и не все операции с файлами представляют интерес (например, системные файлы и операции системных процессов), особенно учитывая направленность функционала целевой системы на мониторинг файлов, содержащих текстовую информацию. Поэтому необходима реализация механизма фильтрации файлов и операций над ними.

1.3.2.3 Фильтрация файлов и операций

С помощью мониторинга файловой системы можно получать структурированное описание операций с файлами. Соответственно, для определения того, требуется ли вести наблюдение за файлом, над которым выполняется операция, будет использоваться информация о выполняемой операции (пример атрибутов операций: имя файла, путь к файлу, имя процесса, имя пользователя, тип операции и т.п.).

Далее под наблюдаемой операцией с файлом будем понимать операцию, удовлетворяющую заранее заданным свойствам её атрибутов, а файл, над которым выполняется наблюдаемая операция, будем называть наблюдаемым документом. Таким образом, под поведенческими данными работы пользователей с текстовой информацией далее мы будем понимать данные о наблюдаемых операциях и содержимое ассоциированных с ними наблюдаемых файлов.

Задача определения, является ли операция с файлом наблюдаемой, затруднена тем, что в различных организациях могут использоваться различные правила (политики) для задания наблюдаемых операций. Соответственно, необходимо предоставить экспертам возможность задания правил, по которым будет производиться данная классификация. Для задания сложных правил удобны так называемые скриптовые языки (англ. *scripting languages*), разработанные для записи «сценариев», последовательностей операций, которые

пользователь может выполнять на компьютере. Сценарии обычно интерпретируются, а не компилируются. Многие языки являются встраиваемыми: реализации предоставляют интерфейсы (в том числе для C/C++) для интерпретации сценариев в период выполнения. Сейчас наиболее распространённым является язык Python [7]. Для него есть JIT-компилятор Pyso, позволяющий транслировать исходный код в машинный, во время первого запуска. Это позволяет существенно увеличить производительность. Кроме того, для Python имеется множество дополнительных свободных библиотек, в частности, для работы с различными форматами текстовых файлов, работы с кодировками и т.п. Однако, если правила, определяющие наблюдаемые операции, не очень сложны, то их можно задавать путём обычного конфигурационного файла (например, формата XML).

1.3.2.4 Сохранение наблюдаемых операций и содержимого документов

Поступающую от драйвера информацию об операциях с ФС, после прохождения через соответствующие фильтры, необходимо где-то сохранять. Кроме описаний самих операций нужно хранить связанные с ними данные о процессах и пользователях. Также необходимо осуществляя эффективную выборку данных для их последующей передачи в центральное хранилище. Поэтому хранение информации об операциях реализовано в «легковесной» реляционной СУБД, например, MS Access, тем более, библиотеки для работы с ней предустановлены практически на любой рабочей станции с ОС Windows. Схема реляционной базы данных приводится ниже в подпункте 1.4.3.3.

Помимо сохранения информации о наблюдаемых операциях с документами необходимо сохранять и содержимое документов. Содержимое документов сохраняется в случае операции создания документа или любого обращения к существующему документу, но который ранее не был зарегистрирован в системе, а также в случае последующих операций изменения содержимого документов. Теневые копии содержимого документов помещаются в специальную директорию, а информация о скопированных агентом документах также сохраняется в РСУБД наряду с другими операциями. Копии документов решено было сохранять в виде файлов, т.к. в этом случае БД агента не будет перегружаться большим объёмом информации типа BLOB (англ. Binary Large Object) и копии документов всегда будут напрямую доступны для различных дополнительных операций предобработки.

Таким образом, структура хранения поведенческой информации реализована следующем образом — наблюдаемые операции с документами сохраняются в реляционной базе данных, а содержимое соответствующих документов сохраняется в виде файлов в специальной директории агента мониторинга.

В отличие от описания операций с документами содержимое самих документов, как правило, представляет гораздо больший объем данных, поэтому необходима эффективная организация его хранения. Также эффективное хранения накопленных контентных данных важно и с точки зрения уменьшения объема передаваемых данных (см. подпункт 1.3.2.5). Однако, теневые копии документов необходимы модулю классификации (который также входит в состав агента мониторинга) непосредственно для вычисления значения аномальности содержимого документа и ассоциированной с ним наблюдаемой операции пользователя. Поэтому было решено организовать сжатие накопленных теневых копий документов по расписанию, а удаление теневых копий будет выполняться после завершения их обработки модулем классификации (см. подпункт 1.3.2.5).

1.3.2.5 Передача данных об информационных потоках в центральное хранилище

В функции агента мониторинга входит передача собранных поведенческих данных модулю консолидации для последующего их анализа в *отложенном режиме*. Консолидация данных подразумевает передачу больших объемов собранной информации по сети. Так как существует ряд сетей, где скорость передачи невелика, а объем трафика имеет значение, актуальным является наличие методов сжатия передаваемых данных и планирования передачи данных. Также требуются механизмы защиты данных, передаваемых по сети, с целью гарантирования достоверности собранных и обрабатываемых данных, а также минимизации рисков утечки информации.

От агента мониторинга требуется передача следующих собранных данных, формирующих поведенческую информацию:

1. Описание операций над документами. Информация об операциях хранится на агенте в виде таблиц локальной БД, поэтому экспорт данных из БД можно реализовать средствами самой СУБД.
2. Содержимое документов. Содержимое документов хранится на агенте в виде файлов. Список файлов, требующийся для передачи, определяются на основе данных об экспортируемых операциях.

Пакет файлов, содержащий данные об операциях с документами и файлы содержимого документов, перед отправкой сжимается архиватором *gzip* [8]. Архиватор *gzip* был выбран, т.к. предоставляет хорошую степень сжатия при высокой скорости работы, кроме того он имеет свободные реализации на всех популярных платформах.

Для обеспечения распределения нагрузки на сеть и балансировки нагрузки на наблюдаемые компьютеры предложено организовывать передачу собранных данных по одной или нескольким из следующих стратегий:

1. *Фиксированными объемами данных.* Агент накапливает определенный объем информации или фиксированное количество записей в базе данных и затем передает их на сервер консолидации.
2. *Через равные промежутки времени.* Агент через равные промежутки времени передает все имеющиеся у него в локальном хранилище данные независимо от их объема.
3. *Немедленная передача.* Агент сразу же передает полученные данные при появлении каждой новой записи в журнале. Данная стратегия наиболее требовательная к ресурсам компьютера.

Указанный механизм реализован заданием параметров, описывающих максимально возможный объем переданных данных и максимально возможный интервал времени, в течение которого агент может не передавать данные. Как только одно из максимально возможных значений достигнуто, все собранные данные помещаются в очередь на сжатие и последующую отправку.

Если после выполнения процедуры сжатия теневые копии помечены как обработанные модулем классификации, то они и соответствующие им данные о наблюдаемых операциях удаляются из локального хранилища агента. Таким образом, в случае неуспешной отправки данных модулю консолидации в локальном хранилище останутся только сжатые данные.

Для обеспечения безопасности все передаваемые по сети данные шифруются с помощью криптографического протокола SSL. Применяется двусторонняя авторизация для невозможности подмены принимающей или передающей стороны. Протокол обеспечивает конфиденциальность обмена данными между клиентом и сервером, использующими TCP/IP, причём для шифрования используется асимметричный алгоритм с открытым ключом [9].

1.3.2.6 Реализация модуля консолидации поведенческой информации

Агенты мониторинга реализуют передачу собираемых поведенческих данных о работе пользователей с текстовой информацией *модулю консолидации*. Таким образом, *модуль консолидации* должен обеспечивать принятие данных об операциях с электронными документами и контенте документов от множества агентов мониторинга и помещать полученные данные в единое хранилище. Основной особенностью работы модуля

консолидации является необходимость параллельного получения данных от большого количества агентов (тысячи и даже десятки тысяч), что объясняется масштабами современных сетей. Поэтому возникает необходимость реализации:

1. Эффективного представления передаваемых по сети данных — это реализуется на агентах мониторинга путём сжатия пакета собранных поведенческих данных;
2. Механизмов распределения нагрузки на сеть — это реализуется на агентах мониторинга путём реализации стратегий передачи данных.

Таким образом, *модуль консолидации* реализует приём, распаковку и сохранение в единое хранилище поведенческих данных, поступающих от множества агентов мониторинга. Также в задачи модуля консолидации входит предоставление доступа к единому хранилищу для других программных модулей (см. рисунок 13), например, модуля построения и модуля применения поведенческих моделей.

Структура хранения поведенческой информации реализована аналогично структуре, используемой агентом мониторинга — информация о наблюдаемых операциях хранится в таблицах БД (см. подпункт 1.4.3.3), а копии документов хранятся в директориях ФС, соответствующих каждому зарегистрированному агенту. Для разграничения прав доступа к поведенческой информации задаются соответствующие права как на БД, так и на директории агентов.

1.3.3 Реализация структур представления биометрических данных, процедур их сбора, хранения, управления и предварительной обработки для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами компьютерной системы

Структуры представления биометрических данных реализованы, согласно форматам, разработанным на 2 этапе данного ПНИ (Отчет ПНИ, Этап 2, пункт 2.1). Основными реализациями структур представления данных о поведенческой информации об особенностях работы с информационными и вычислительными ресурсами компьютерной системы являются:

- *Записи в журнале windows event logs* как создаваемые стандартными компонентами аудита ОС Windows, так создаваемые с помощью специально разработанных компонент.
- Представление собранных фактов в виде *локального хранилища событий*, полученных в результате чтения событий из windows event log.

- *Перманентное хранилище данных на сервере консолидации*, формируемое из информации, полученной от всех агентов сбора.
- *Витрины данных для анализа*, используемые рабочим местом аналитика безопасности для выгрузки интересующего среза событий (по времени, компьютерам и типа событий) и построения аналитических моделей для решения задач фоновой идентификации пользователей и раннего обнаружения внутренних вторжений.

1.3.3.1 Реализация структур представления данных о поведенческой информации об особенностях работы с информационными и вычислительными ресурсами компьютерной системы являются в виде записей в Windows Event Log.

Используются как стандартные записи аудита так и создаваемые с помощью специально разработанных компонент сбора. К стандартным записям аудита относятся:

- записи о фактах начала и окончания пользовательских сессий (события аудита с идентификаторами 528 и 538),
- записи о фактах запуска и остановки пользовательских и системных процессов (события аудита с идентификаторами 592 и 593),
- записи о фактах запуска и завершения работы ОС (события аудита с идентификаторами 512 и 513),
- записи о фактах установки и удаления пользователем программного обеспечения (события аудита с идентификаторами 1033 и 1034).

К дополнительным событиям, записываемым в windows event log специализированными дополнительно разработанными компонентами мониторинга относятся:

- записи о параметрах (используемых портах, количестве переданных и полученных байт, адресах и протоколах) TCP соединений, создаваемые специально разработанной компонентой на основе технологии LSP (Layered Service Provider, англ. многоуровневый поставщик услуг), позволяющей встроить собственную библиотеку в стек протокола TCP/IP в Windows Socket,
- записи об изменении аппаратной конфигурации контролируемого компьютера (список устройств и тип действия – удаление или добавление), создаваемые агентом сбора информации с помощью стандартных API SetupDiEnumDeviceInfo,
- записи по статистике действий (число событий в единицу времени, по умолчанию – за минуту) пользователя с использованием клавиатуры и манипулятора мышь (в рамках

выбранных приложений), собираемые с помощью отдельной реализованной компоненты на основе технологии Windows Hook,

- записи о фактах и параметрах (имя файлов, директорий, тип операции, количество прочитанных или записанных байт) доступа и работы с файлами в выбранных директориях с использованием технологии Windows mini filter FS.

Все записи в журналах аудита, как стандартные, так и собранные с помощью дополнительных компонент мониторинга содержат общий обязательный набор атрибутов: время события, имя пользователя, имя и путь процесса, породившего событие, имя компьютера и домена. Также реализован расчет и сохранение в свойствах событий в windows log дополнительные параметры в зависимости от типа событий.

1.3.3.2 Реализация структур представления данных о поведенческой информации об особенностях работы с информационными и вычислительными ресурсами компьютерной системы являются в виде специализированного локального хранилища событий на агенте сбора данных.

Представление собранных фактов в виде *локального хранилища событий*, полученных в результате чтения событий из windows event log. Процесс формирования локального хранилища осуществляется специальной программной компонентой - агентом сбора информации, реализованного в виде windows service. Реализация представления данных фактов основывается на использовании локального хранилища, размещенного на агенте сбора и реализованного на основе файловой системы, представляющего из себя совокупность справочников, хранящих значения символьных атрибутов событий (имена, пути процессов и другие), а также файлов с данными, содержащих описание фактов каждого типа с числовыми признаками и числовыми ключами для поиска значения символьного признака в соответствующем словаре. В последствии данные из локального хранилища доставляются на сервер консолидации в соответствии с заданной стратегией передачи данных:

- *Фиксированными объемами данных.* Агент накапливает определенный объем информации или фиксированное количество записей журналов, а затем передает их на сервер консолидации.
- *Через равные промежутки времени.* Агент через равные промежутки времени передает все имеющиеся у него в локальном хранилище данные независимо от их объема.

- *Немедленная передача.* Агент немедленно передает данные о каждой вновь прочитанной записи в журнале регистрации.

Реализация процедур хранения и консолидации поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами компьютерной системы основывается на разработке методов хранения данных представленных в отчете ПНИ за второй этап (Отчет ПНИ Этап 2 пункты 2.3, 2.4). Реализованы процедуры хранения данных агента сбора и сервера консолидации, основанные на использовании файловой системы. Для представления записей журналов регистрации на агенте сбора используется хранилище на основе файловой системы. Файлы такого хранилища делятся на три части:

- *Основные параметры*, присутствующие во всех записях всех журналов: тип события, время генерации, имя пользователя, имя процесса, компьютера и домена.
- *Вспомогательные параметры*, описывающие дополнительную информацию о событии.
- *Справочники* для хранения реальных строковых значений как основных, так и вспомогательных параметров.

При чтении событий из windows event log и сохранении их в локальном хранилище производится фильтрация событий по заданным шаблонам с использованием правил на основе регулярных выражений (например, можно задать ограничения на идентификаторы собираемых стандартных событий, имена и пути процессов, имена пользователей и так далее). Помимо фильтрации производится расчет дополнительных характеристик событий, которые невозможно рассчитать на этапе сохранения событий, в частности, суммарное время сеанса работы пользователя (рассчитывается по временным меткам пары событий: факта начала и факта окончания сеанса), суммарное время работы процесса (рассчитывается по временным меткам пары событий: факта начала и факта окончания работы процесса), именование портов и «расшифровка» сетевых адресов по IP для фактов о сетевых соединениях. Запись о событии из журнала разделяется между двумя базовыми файлами и файлами справочников: файл для хранения идентификаторов основных параметров, файл для хранения идентификаторов вспомогательных параметров и файлы справочников. Файлы справочников содержат только одну копию каждого значения и позволяют по идентификаторам (значениям хэш-функции) параметров получать текстовые значения атрибутов событий. Три справочника необходимы для хранения имен пользователя, имен параметров и всех остальных значений, соответственно. Разделение на три справочника

вместо одного общего оправдывается выигрышем производительности: время поиска значения в справочнике возрастает с размером справочника. Имя пользователя и имена параметров требуются для анализа каждой записи журнала регистрации, поэтому количество обращений к справочникам именно за этими значениями велико.

1.3.3.3 Реализация структур представления данных о поведенческой информации об особенностях работы с информационными и вычислительными ресурсами компьютерной системы являются в виде перманентного хранилище данных на сервере консолидации.

Для хранения данных на сервере консолидации используется тот же подход с расширением организации хранения файлов событий и справочников в файловой системе. Хранилище организовано в виде дерева (см. рисунок 15), в корне которого находятся каталоги с именем домена, внутри каждого каталога с именем домена находятся каталоги с именем компьютера этого домена. Внутри каталогов с именем компьютера находятся каталоги для журналов регистрации, внутри каталога журнала регистрации находятся файлы с собранными записями, разделенные по датам, например, по дням. Файлы-справочники могут храниться в произвольном месте на диске.

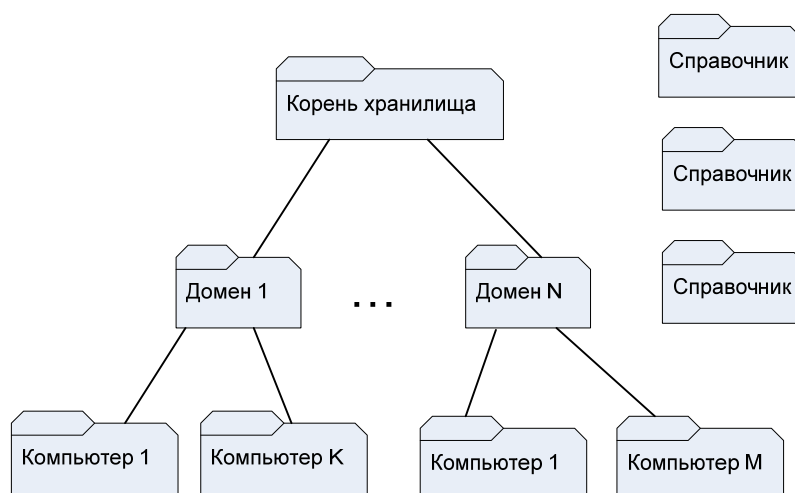


Рисунок 15 — Структура организации хранилища.

Хранилище данных на агенте сбора и на сервере консолидации реализовать на основе файловой системы NTFS — это позволит обеспечить защиту доступа к данным с помощью расстановки прав доступа к соответствующим файлам и каталогам. Также дополнительно реализованы механизмы обеспечения отказоустойчивости при работе со справочниками и файлами данных. Для решения защиты собираемой информации от противодействий и

аварийных ситуаций на различных уровнях используются средства разграничения прав доступа, основанный на SSL механизм шифрования, предложенные бинарные форматы представления данных, разработанные механизмы справочников, механизмы контрольных точек и резервного копирования данных.

1.3.3.4 Реализация структур структур представления данных о поведенческой информации об особенностях работы с информационными и вычислительными ресурсами компьютерной системы являются в виде *Витрины данных для анализа*.

Для проведения анализа необходимый срез данных выгружается их хранилища сервера консолидации в «витрину данных». Основными требованиями к витрине данных являются: скорость загрузки данных в витрину и возможность сравнительно простого чтения данных из витрины различными компонентами анализа. Витрина данных реализована на основе реляционной СУБД MSSQL Server 2005. Безопасность выгруженных данных обеспечивается встроенными в MSSQL 2005 средствами разграничения прав доступа, а необходимая скорость заполнения с помощью разработанных методов заполнения витрины. Важным требованием к витрине данных является занимаемый базой размер, а так же возможность применения выбранных типов анализа. Размер в первую очередь определяется схемой базы данных. Для минимизации размера, как и на сервере консолидации, можно использовать справочники, роль которых в витрине данных выполняют SQL таблицы. Похожая структура данных может использоваться и для применения технологии OLAP. Задача формирования мер, измерений и иерархий решается на уровне структуры витрины данных. В таком случае реализация предложенной схемы построения измерений на основе выделения подстрок, формирования иерархий и циклических иерархий проводится в момент загрузки данных в витрину. В качестве схемы базы данных для применения технологии OLAP может быть использована схема типа «звезда» или «снежинка», основными составляющими данных схем являются таблицы фактов и множество таблиц измерений (справочников). Используя БД на основе схемы типа «звезда» или «снежинка» (на основе SQL запросов или с помощью представлений) так же можно получить факты активности в виде набора значений атрибутов, что требуется для построения ассоциативных правил и поиска аномалий. Таким образом, витрина данных предложено представлять в виде набора SQL таблиц – справочников, и набора таблиц фактов. Например, часть базы, для описания фактов работы пользователей в сети имеет следующую структуру (см. рисунок 16).

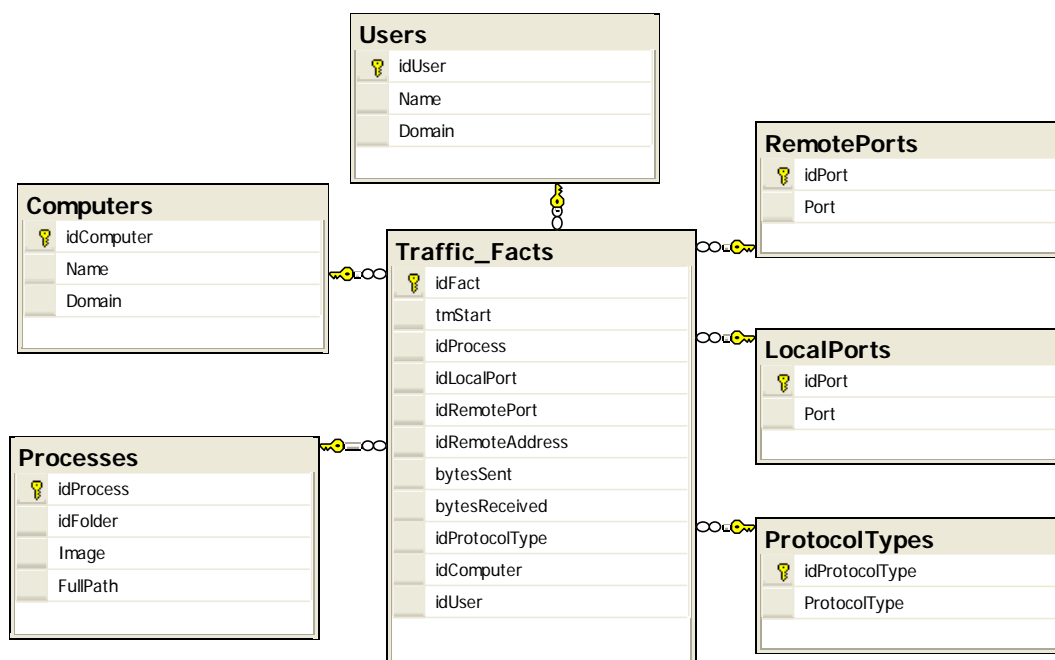


Рисунок 16 — Схема витрины данных для фактов работы в сети.

Общей таблицей, содержащей записи о фактах сетевого взаимодействия, является таблица Traffic_Facts, она содержит информацию о времени факта (tmStart), объеме переданных и полученных данных (bytesSent, bytesReceived), а так же идентификаторы значений справочников, где хранятся значения остальных текстовых атрибутов, таких как имя пользователя (idUser), имя компьютера (idComputer) и других. Следует заметить, что справочники для разных типов фактов активности могут быть общими. Для всех фактов общими являются справочники для хранения обязательных атрибутов (имя пользователя и компьютера). Атрибут «имя процесса» так же присутствует практически во всех фактах активности пользователей, так как все действия пользователи осуществляют в какой-то программе (процессе). Примером циклического справочника может служить справочник для записи полного пути к файлу (см. рисунок 17).

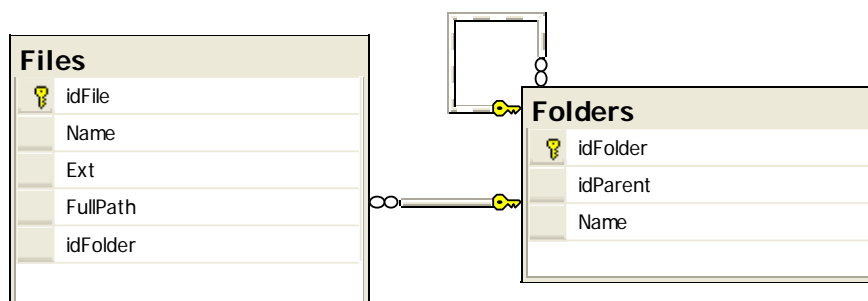


Рисунок 17 — Пример структуры циклического справочника.

Таблица Files является справочником, запись таблицы Files описывает один файл. На таблицу Files может ссылаться любой факт, описывающий активность пользователя, например, факт работы с файловой системой. В таблице Files сохраняется как полный путь к файлу, его имя и расширение, так и ссылка на запись таблицы Folders, содержащей имя каталога и ссылку на родительский каталог и так далее. Таким образом, таблица Folders так же является справочником, но который используется уже не из таблицы фактов, а из другого справочника. Предложенная структура позволяет построить иерархию каталогов на пути к файлу. Построение таких справочников проводится вручную для некоторых базовых сущностей ОС, таких как файл и процесс. Создание и заполнение витрины данных с предложенной структурой, очевидно, требует информации о семантике некоторых базовых или общих атрибутов (имя пользователя, имя компьютера, имя процесса, каталог файловой системы и некоторых других). Для помещения остальных фактов активности в витрину создаются простые справочники для текстовых атрибутов, состоящие из ключа и значения. Числовые же атрибуты хранятся в основной таблице факта. Основная таблица факта формируется на основе набора атрибутов факта. Учитывая то, что для построения факта на агенте указываются имена его атрибутов, то все атрибуты предложено разделять на обязательные, общие и остальные на основе имени атрибутов, что позволяет не использовать при заполнении витрины данных дополнительной информации о собранных фактах активности, а так же дополнительно не хранить исходные названия атрибутов. Заполнение витрины возможно по двум стратегиям: заполнение пустой витрины или дозаполнение уже заполненной ранее витрины. Основным требованием к методам заполнения витрины данных является их производительность. При этом для заполнения витрины с предложенной структурой требуется по значению каждого текстового атрибута проверять, присутствует ли оно уже в требуемом справочнике, и если нет, то добавлять. В случае реализации таких проверок и вставок с помощью SQL запросов (а именно SELECT и INSERT запросов), время заполнения не удовлетворяет требованиям производительности. Для решения указанной проблемы было предложено воспользоваться механизмом «объемной» (bulk) вставки. Идея механизма заключается в подготовке специальных файлов данных с разделителями. Такие файлы содержат помещаемые в SQL таблицу значения в виде, допускающем непосредственное отображение записей файлов в записи SQL таблиц. Подготовленный таким образом файл с помощью одного запроса загружается в базу, что позволяет избежать накладных расходов на разбор и выполнение большого количества INSERT запросов. Поиск уже добавленных и вычисление новых идентификаторов значений справочников производится не средствами реляционных СУБД, а в программе-загрузчике, что позволило

избежать накладных расходов выполнения SELECT запросов. Однако, учитывая тот факт, что изначально витрина уже может быть заполнена, перед добавлением данных требуется считывание из витрины уже добавленных значений и идентификаторов справочников. Таким образом, основные шаги предложенного алгоритма дозаполнения витрины данных следующие:

1. Чтение из витрины данных уже существующих в ней идентификаторов и значений записей справочников.
2. Чтение среза данных из хранилища, преобразование данных среза в новые записи витрины данных с учетом прочитанных из витрины записей. Сохранение новых записей в специальных файлах.
3. Выполнение операции объемной вставки (BULK INSERT) для подготовленных файлов.

Предложенный алгоритм реализован в модуле рабочее место аналитика.

1.4 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей

1.4.1 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей со стандартными устройствами ввода-вывода

В данном пункте отчета приложены результаты разработки программных компонент, входящих в состав «Подсистемы 1» ЭО ПК, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей со стандартными устройствами ввода-вывода, выполненных согласно пункту 3.11.4.1 ТЗ.

Программные компоненты разрабатывались на основании результатов исследований, полученных на 2 этапе данных ПНИ, а также разработанной в соответствии с пунктом 3.11.2 ТЗ архитектуре ЭО ПК.

Согласно архитектуре ЭО ПК, решения для указанных задач реализовывались в следующих логических модулях (компонентах) «Подсистемы 1» ЭО ПК:

- Модули активной аутентификации на основе анализа динамики клавиатурного ввода ключевого (не секретного) слова и динамики работы пользователя с

манипулятором *мышь* при вводе (не секретного) графического символа на основе сгенерированного шаблона;

- Агент сбора и первичной обработки событий о работе пользователя с устройствами ввода-вывода для фоновой идентификации;
- Хранилище первичных событий (сервер консолидации) о работе пользователей с устройствами ввода вывода при фоновой идентификации.

1.4.1.1 Модули активной аутентификации

Механизм расширения функциональности стандартных средств аутентификации в операционных системах семейства Microsoft Windows реализован на базе различных технологий: в младших версиях (ОС с ядром XP включительно) – используется технология GINA (Graphical Identification and Authentication dll), в более поздних версиях (начиная с ОС с ядром Windows Vista) используется технология Credential Provider.

На данном этапе ПНИ было принято решение о разработке модуля авторизации на базе технологии GINA, так как он предоставляет более удобный доступ к событиям ввода (непосредственно из интегрируемых элементов пользовательского интерфейса), а также более удобен для процесса разработки и отладки. Реализация модулей активной аутентификации на базе технологии Credential Provider предполагается на следующем этапе ПНИ.

Назначение и общее описание

Модуль активной аутентификации является частью реализации механизма двухфакторной авторизации пользователя в операционной системе Windows XP - на основании предварительно собранной информации о биометрических характеристиках пользователя при нажатии клавиш, задержках между нажатиями, характерных конкретному пользователю, а также о движениях мыши: скорости движения на прямых участках, при очерчивании сложных форм, скорость смены направления движения и других.

Сбор биометрической информации об использовании пользователем клавиатуры и мыши реализован в виде программы на языке C++, со следующими функциями:

- формирование лога событий нажатия(отпускания) клавиши, а также лога движений мыши.

- Обработка данного файла для дальнейшего использования собранных данных в процедурах обучения и авторизации.

Сборка и установка

Модуль реализован на языке программирования C++ с использованием стандартной библиотеки шаблонов (STL). Для сборки программных компонент из исходных кодов и установки на целевую систему требуется интегрированная среда разработки программного обеспечения Microsoft Visual Studio 2013 и выполнение следующих действий:

- Открыть файл `ginafull.sln` в Visual Studio и запустить сборку решения (Сборка->Собрать решение).

В результате работы команды в папке `bin/` должен сформироваться файл `ginafull.dll`

- Данный файл необходимо скопировать в каталог `C:\WINDOWS\System32` целевой системы
- Добавить в ветку реестра `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon` строковый параметр `GinaDLL` со значением `ginafull.dll`.
- Перезагрузить компьютер

После этого запуск модуля осуществляется автоматически во время загрузки операционной системы.

Основные функции и настройка

Вход в систему

Окно входа в систему (см. рисунок 18) по умолчанию доступно при загрузке операционной системы (при условии успешной установки модуля). Для того чтобы начать процедуру авторизации пользователя необходимо в данном окне нажать контрольную комбинацию клавиш `CTRL + ALT + DELETE`.

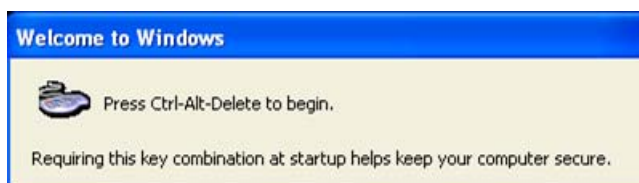


Рисунок 18 — Окно входа в систему.

Интерфейс авторизации пользователя

Окно авторизации пользователя (см. рисунок 19) предоставляет возможность авторизации пользователя с помощью клавиатуры или с помощью мыши, назначение полей:

- «User Name». Определяет имя пользователя. По умолчанию подставляется имя последнего успешно авторизовавшегося в операционной системе пользователя. Имя может состоять из букв русского и английского алфавита, цифр, других символов (кроме \, /, ?, :, *, ", >, <, |, _).
- «Password». Определяет пароль, вводимый пользователем. Может состоять из любых символов или быть пустым.
- «Domain». Определяет компьютер, на котором пользователь желает провести авторизацию. Если требуется авторизация на текущем компьютере, требуется оставить поле пустым.
- «LOG W/O». Позволяет авторизоваться без проверки биометрических признаков, стандартными средствами авторизации ОС Windows (при этом всё равно требуется правильное введение пароля).
- «OK». Позволяет авторизоваться пользователю, используя биометрические данные о динамике ввода пароля пользователем.
- «MouseLog». Позволяет провести авторизацию с использованием данных о движениях мыши пользователем.
- «Cancel». Отменяет процесс авторизации.
- «Shutdown». Выключает компьютер.

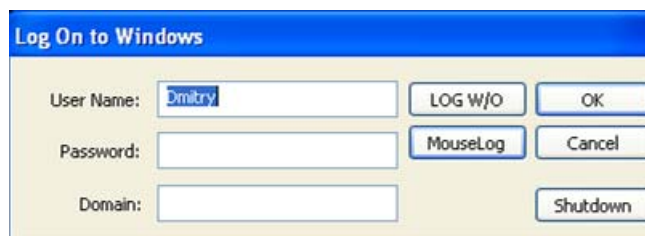


Рисунок 19 — Окно интерфейса авторизации.

Режим авторизации с использованием биометрических данных о динамике работы с клавиатурой

Для авторизации с использованием биометрических данных о динамике ввода пароля пользователем на клавиатуре требуется:

- Существование предварительно созданной модели поведения пользователя (см. далее 1.5.1.1).
- Ввод имени пользователя

- Набор пароля в привычном пользователю ритме.
- После нажатия ОК в случае успешной авторизации окно авторизации исчезнет, и пользователь сможет начать работать с операционной системой. В ином случае - в процессе авторизации произошла ошибка, и пользователь должен ввести пароль повторно или использовать другой способ авторизации.

Режим авторизации с использованием биометрических данных о динамике работы с мышью

Для авторизации с использованием мыши (см. рисунок 20) требуется:

- Предварительно созданная модель использования мыши пользователем. (см. далее 1.5.1.1)
- Нажать в окне авторизации кнопку MouseLog.
- Очертить появившийся шаблон с текстом путём нажатия левой кнопки мыши и последующего обвода шаблона.
- Подтвердить текущую попытку входа нажатием на кнопку Ассепт. В случае если пользователь считает, что попытка получается неудачной, пользователь может нажать на кнопку Clear и обвести шаблон заново.
- В случае успешной проверки биометрических данных окно авторизации исчезнет, и пользователь сможет приступить к работе с операционной системой. В ином случае в процессе авторизации произошла ошибка, и пользователь должен повторно повторить процесс авторизации с использованием мыши или выбрать другой способ авторизации.



Рисунок 20 — Окно интерфейса авторизации с использованием манипулятора мышь.

Дополнительные настройки

Окно Security (см. рисунок 21) вызывается путём нажатия контрольной комбинации CTRL + ALT + DELETE во время работы в операционной системе (т.е. после успешной авторизации в системе), предлагая следующие настройки:

- “Lock computer“ позволяет заблокировать систему.
- “Change Password...” позволяет проводить процедуры построения биометрических моделей пользователя.
- “Log Off...” позволяет текущему пользователю выйти из системы.
- “Task Manager” вызывает диспетчер задач операционной системы.
- “Shut Down” выключает компьютер.
- “Cancel” закрывает окно.



Рисунок 21 — Окно дополнительных настроек.

1.4.1.2 Агент сбора и первичной обработки событий о работе пользователя с устройствами ввода-вывода

Данный модуль предназначен для фонового сбора данных об использовании устройств ввода на клиентской машине, их первичной обработке и подготовке для передачи в модуль идентификации пользователей. Работа модуля осуществляется на клиентских машинах под управлением ОС Microsoft Windows с целью сбора пользовательской активности при работе клавиатурой и мышью, а также сохранения информации о программно-аппаратном обеспечении данной клиентской машины.

Назначение и общее описание

Модуль предназначен для работы на ОС семейства Windows i386/AMD64, начиная с ядра версии XP SP2, физически состоит из 2 файлов:

- динамическая библиотека DLL, предназначенная для встраивания в процессы Windows и перехвата целевых событий;
- исполняемый файл, предназначенный для управления библиотекой.

Интеграция перехватчика локальных событий от клавиатуры и мыши осуществляется средствами Windows API с помощью функций *SetWindowsHookEx* и *UnhookWindowsHookEx*.

По результатам предварительных исследований оказалось, что ОС Windows реализует разный принцип работы перехватчиков событий в ОС с ядром XP и в более новых версиях. Кроме того, ОС накладывает жесткие временные ограничения на продолжительность обработки каждого перехваченного события. В связи с этим библиотека-перехватчик и основной исполняемый файл используют параллельную обработку и взаимодействие с помощью нитей (threads) и стандартных средств синхронизации Windows (критические секции, разделяемая память, мьютексы).

При перехвате новых событий основное приложение производит запись этих событий в определённые файлы с заданным промежутком времени. Описание соответствующих структур данных и файлов приведено в 1.3.1.2

Сборка и установка

Сборка модуля производится на системе с предустановленным программным обеспечением Microsoft Visual Studio 2010 или новее с поддержкой языка C++. Для компиляции компонента необходимо выполнить следующие действия:

- скопировать файлы из дистрибутива в каталог в целевой системе. В результате копирования в целевом каталоге должна быть сформирована следующая структура каталогов и файлов:
 - hooks/
 - hookstub/
 - hookinfo.h
 - hookstub.sln
- открыть файл решения hookstub.sln в Microsoft Visual Studio;

- произвести сборку программных компонент на языке C++.

Для установки компонента необходимо скопировать скомпилированные файлы в каталог в целевой системе, при необходимости добавить hookstub.exe в список автозагрузки ОС. В результате копирования в целевом каталоге должна быть сформирована следующая структура каталогов и файлов:

```
hookstub.exe  
hooks.dll.
```

Основные функции и настройка

Основные функции агента сбора и первичной обработки следующие:

- Перехват всей пользовательской активности при работе с устройствами ввода и запись информации о событиях в файлы
- Сбор информации о программно-аппаратных характеристиках целевой системы и запись в файл
- Фильтрация перехваченных событий динамики работы пользователей с клавиатурой и мышью.
- Выделение составных событий динамики работы пользователей с клавиатурой и мышью.
- Разбиение последовательности событий динамики работы пользователей с клавиатурой на временные окна заданной длины.
- Расчет и сохранение в файл векторов признаков динамики работы с клавиатурой.
- Разбиение последовательности событий динамики работы пользователей с мышью на временные окна заданной длины.
- Расчет и сохранение в файл векторов признаков динамики работы с мышью.

Константы, влияющие на расчёт последовательностей в настоящее время размещены в подключаемых файлах исходных кодов. Доступны следующие блоки характеристик:

- настройки параметров разбиения последовательности событий действий пользователя с клавиатурой и мышью на временные окна;
- настройки параметров выделения признаков динамики работы пользователя с мышью;

- настройки параметров выделения признаков динамики работы пользователя с клавиатурой.

Запуск и завершение

Активация подсистемы сбора модуля идентификации осуществляется путем запуска исполняемого файла приложения. Настройки режимов работы определяется комбинацией опций запуска(флагов), переданных исполняемому файлу в командной строке.

Окончанием активной сессии сбора и завершением сбора данных является завершение приложения. Завершение может быть осуществлено естественным образом – при выключении компьютера/завершении сеанса Windows для данного пользователя и пр., либо форсированным способом: при одновременном нажатии комбинации клавиш Ctrl+Shift+S. При появлении информационного диалога(см. рисунок 22) перехват данных временно останавливается (данный режим может быть использован для ввода паролей, и другой конфиденциальной информации).

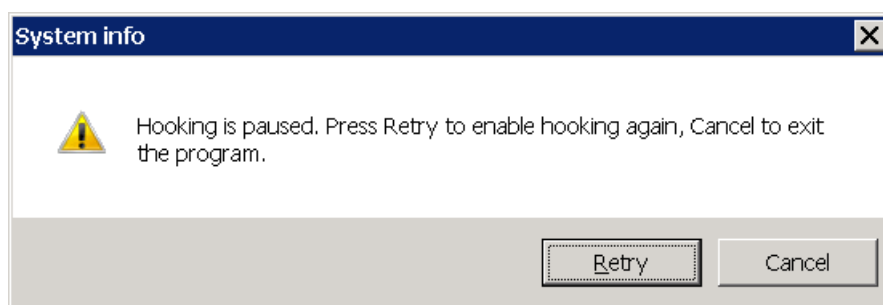


Рисунок 22 — Управление модулем сбора.

Возможные действия после появления информационного диалога:

Retry (Повторить) – продолжить сбор данных (в той же сессии),

Cancel (Завершить) - завершить сбор (форсировать закрытие сессии).

1.4.1.3 Сервер консолидации данных

Сервер консолидации данных предназначен для хранения и управления набором первичных событий о работе пользователей с устройствами ввода вывода. На данном этапе разработок хранилище является логической (виртуальной) компонентой "Подсистемы_1", и

не содержит в себе отдельных программных единиц - необходимая функциональность обеспечивается стандартными средствами файловой системы операционных систем семейства Windows. В частности,

- Хранение данных для модулей авторизации – реализовано в виде временных файлов (формируются в процессе осуществления попыток режима обучения, либо попытки авторизации), и файлов-моделей (результат режима обучения), сохраняемых локально.
- Хранение данных для модуля идентификации – реализовано в виде структурированного дерева каталогов ("имя_компьютера"->"имя_пользователя _ дата_начала_сессии") согласно разделу 1.3.1.2, сохраняемых на разделяемом ресурсе локальной сети (или в локальной файловой системе).

Организация прав доступа к соответствующим разделяемым каталогам формируется администратором ЭО ПК. Обеспечение синхронизации доступа к необходимым данным со стороны модулей "Подсистемы_1" реализовано на базе стандартных механизмов синхронизации Windows (таймеры, разделяемая память, критические секции) в соответствующих модулях.

1.4.2 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы

В данном пункте отчета приложены результаты разработки программных компонент, входящих в состав «Подсистемы 2» ЭО ПК, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы, выполненных согласно пунктам 3.11.4 и 3.11.4.2 ТЗ.

Программные компоненты, сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с информационными и вычислительными ресурсами защищаемой компьютерной системы разрабатывались на основании результатов исследований, полученных на 2 этапе данных ПНИ (разработка

структур данных, методов сбора, предобработки, хранения и управления для поведенческой биометрической информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы), а также разработанной в соответствии с пунктом 3.11.2 ТЗ на данном этапе ПНИ архитектуре ЭО ПК.

1.4.2.1 Процедуры сбора данных

Основной задачей агентов сбора данных является подготовка на основе журналируемых событий из различных источников необходимых для анализа фактов активности и их передача на сервер консолидации. Основной особенностью сбора журналируемых данных в современных корпоративных сетях является количество и разнородность журналов и событий внутри журналов. Часть данных, относящихся к разным фактам активности, распределена среди журналов операционных систем и прикладных программ. При этом возможна ситуация, что журналируются не все требуемые параметры и требуется их дополнение из других источников. При этом некоторые необходимые события могут не содержаться в доступных журналах вообще, в данном случае единственным способом получения требуемой информации является разработка специализированных модулей слежения за параметрами работы пользователей с указанными ресурсами. Степень полноты журналирования необходимых данных определяется операционной системой и набором установленного программного обеспечения. В рамках каждой операционной системы требуется решать задачу выбора источников необходимой журналируемой информации (журналов), выбора типов событий в рамках журналов и выбора модулей слежения, которые требуется реализовать.

Для различных типов источников событий и журналов применяются различные методы сбора. Однако так как ряд модулей слежения за работой пользователей может быть реализован только отдельно от агента сбора, например, на уровне драйверов ОС, то передача данных от модулей слежения на агент сбора может прерываться, например, в случае остановки агента. Для решения проблемы предлагается журналировать данные, полученные модулями слежения, с целью последующего их чтения из журналов агентом сбора. Для сбора всех необходимых событий из различных журналов требуются методы унифицированного представления событий и унифицированного чтения несистемных журналов, так как задача чтения текстовых журналов может быть решена без учета ОС, и не может решаться отдельно для каждого журнала. В то же время в рамках каждой ОС требуется разработка модуля чтения системных журналов.

Для выделения требуемых данных из журналов необходимы методы фильтрации журналируемых событий, при этом методы фильтрации должны учитывать как типы событий, так и значения их атрибутов, с целью сбора только необходимых данных и минимизации потока событий. Полученные из различных журналов события полностью или частично описывают необходимые параметры работы пользователей. Требуется дополнить, если необходимо, собранные события недостающими атрибутами, затем сформировать на их основе факты активности, описывающие необходимые характеристики работы пользователей с требуемыми ресурсами. С целью реализации указанных преобразований предложена модульная схема построения агента сбора. Для каждого преобразования предложено использовать свой модуль (см. рисунок 23).

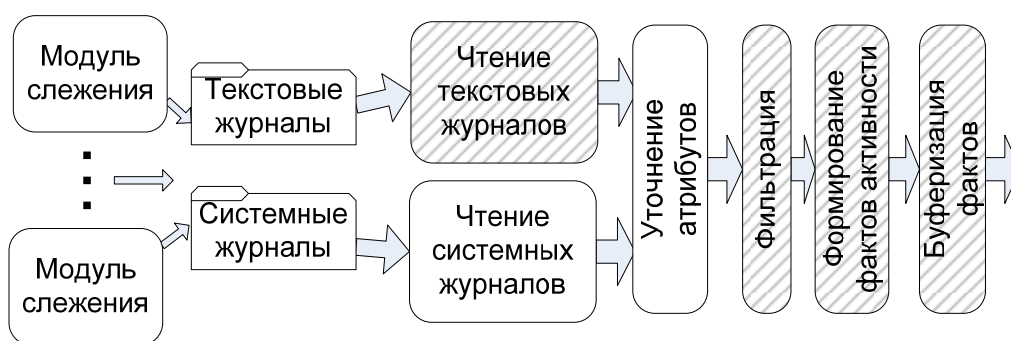


Рисунок 23 — Схема обработки данных в подсистеме сбора.

Разделение на модули позволяет реализовывать отдельно от агента сбора функциональность, зависящую от операционной системы (отмечено светлым фоном на рисунке). Штрихом на схеме выделены общие модули, реализация которых не зависит от конкретной операционной системы.

Таким образом, можно выделить следующие этапы обработки данных:

1. Журналирование необходимых параметров работы пользователей с помощью модулей слежения за требуемой активностью.
2. Сбор событий из различных журналов.
3. Уточнение набора и значений атрибутов событий.
4. Фильтрация считываемых событий с целью выделения из журналов только требуемых событий. Фильтрация выполняется после уточнения атрибутов, так как может быть основана на измененных значениях атрибутов или на вновь добавленных атрибутах.
5. Построение на основе собранных событий фактов активности, описывающих параметры работы пользователей с ресурсами.

6. Буферизация построенных фактов активности с целью последующей отправки на сервер консолидации, реализация стратегий передачи данных на сервер консолидации.

Согласно архитектуре ЭО ПК, решения задач сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с информационными и вычислительными ресурсами реализовывались в следующих логических модулях (компонентах) «Подсистемы 2» ЭО ПК:

- Агенты сбора и предобработки поведенческой информации;
- Сервер консолидации;
- Консоль управления;
- АРМ аналитика безопасности.

Агенты сбора и предобработки поведенческой информации реализованы в виде сервисов ОС Windows. Агенты сбора и предобработки поведенческой информации обеспечивают сбор журналируемой информации, содержащей сведения, а фактах активности пользователей, их предобработку, передачу на сервер консолидации. Агенты сбора и предобработки поведенческой информации устанавливаются и работают на АРМ пользователей. Для каждого агента сбора создается профиль агента, структура, содержащая информацию о параметрах настройки сбора, локального хранения, передачи данных на сервер консолидации. Для управления работой агентов и сбором данных разработана Консоль управления сбором данных - программные модули, обеспечивающие настройку сбора данных.

1.4.2.2 Процедуры передачи и сохранения данных на сервер консолидации

С целью обеспечения защищенного и надежного буфера данных в рабочем каталоге агента для группы типов фактов активности создается четыре папки. Пример папок для фактов, построенных на основе событий стандартного журнала безопасности (Security) ОС Windows, приведен на рисунке 24:

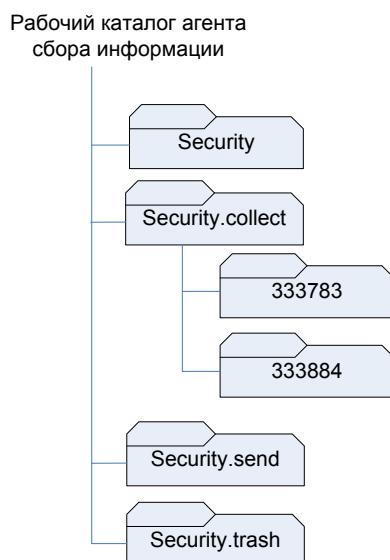


Рисунок 24 — Структура файлов каталога агента.

В каталоге Security хранятся временные параметры работы с журналом, такие как номер последнего прочитанного из журнала события. В каталоге Security.collect располагаются собранные данные, разбитые по часам. Разбивка буфера агента на части дает возможность более гибкого планирования передачи за счет уменьшения максимального размера пакета, а так же уменьшает влияние сбоев в работе компьютера агента с файлами, например, в случае аварийного выключения питания, так как работа ведется только с одним пакетом из буфера.

Несмотря на одновременную работу лишь с частью буфера аварийные выключения компьютера, на котором установлен агент, могут приводить к повреждению части собранной, но еще не отправленной информации. Возможность такой ситуации следует исключить, так как злоумышленник может сознательно воспользоваться данной уязвимостью при удалении следов своей деятельности. Для контроля целостности данных агент при старте выполняет проверки и корректировки тех файлов, в которые производился сбор до остановки или аварийного завершения работы. В результате корректировок у файлов «отрезаются» испорченные окончания и дописываются заново данными из журнала.

Файлы каталога Security.send представляют собой очередь данных для отправки. В каталог Security.send перемещаются все папки из Security.collect, как только срабатывает одно из условий стратегий передачи данных. Каталог Security.trash служит для буферизации данных, если в них возникают неустранимые ошибки при сборе или при передаче.

Агент периодически просматривает все папки <Имя журнала>.send и вызывает процедуру отправки данных на сервер консолидации для каждого из найденных пакетов. Так как процедуры сбора и передачи работают параллельно, очень важно, чтобы данные

помещались на отправку мгновенно с точки зрения процедуры передачи, иначе могут возникнуть ошибки, связанные, например, с передачей не полного набора данных. Для помещения данных на отправку используется атомарный системный вызов перемещения/переименования каталога в папку <Имя журнала>.send.

Так как хранилище данных реализовано на файловой системе, целостность данных может быть нарушена любыми сбоями работы сервера, например, в момент аварийного выключения питания. Сбои могут приводить к тому, что не все содержимое, дописанное сервером в файл с момента его открытия, будет физически записано на жесткий диск в момент аварийного выключения. Гарантией того, что данные уже находятся на жестком диске, в ОС Windows является закрытие файла, но только в случае использования системных вызовов ОС с режимом работы без буферизации. Использование таких вызовов и свойства атомарности операций переименования и удаления файлов лежит в основе предлагаемых в системе алгоритмов защиты от аварийных выключений.

Характер работы с файлами и справочниками, заключающийся в том, что запись производится только в конец файла, позволяет рассматривать в качестве потенциального места ошибок, только конец файла, куда производилась запись.

Как было показано выше, данные хранятся в виде двух основных файлов, описывающих факт активности, и трех справочников, описывающих значения атрибутов факта, поэтому требуется механизм защиты от сбоев и основных файлов и файлов – справочников. Для обеспечения надежности хранилища реализуется механизм транзакций для добавления данных, основанный на реализации собственных механизмов контрольных точек файлов и избыточного хранения для справочников. Рассмотрим предложенные механизмы подробнее.

Под транзакцией понимается неделимая операция импорта одного из пришедших от агента пакетов (набора фактов). Если в момент импорта пакета происходит сбой в работе сервера консолидации, связанный с аварийным выключением питания или другими неполадками, импорт данного пакета повторяется заново после повторного включения сервера. Предложенная схема реализуется следующим алгоритмом:

1. При открытии файлов данных проверяется наличие незакрытой контрольной точки. Признак незакрытой контрольной точки – присутствие файла checkpoint.tmp, располагающегося в том же каталоге что и файлы данных.
2. Если файл checkpoint.tmp уже существует, происходит откат на предыдущую контрольную точку.

3. При открытии файлов с атрибутами фактов создается файл `checkpoint.tmp`, служащий флагом того, что работа с файлами не была завершена и следующая контрольная точка еще не установлена.
4. Осуществляется импорт очередного пакета данных.
5. Файлы данных закрываются, что гарантирует их полную запись на диск. Определяются их размеры и записываются в файл `checkpoint.tmp`.
6. Файл `checkpoint.tmp` закрывается, что гарантирует его полную запись на диск.
7. Файл `checkpoint.tmp` переименовывается в файл `checkpoint`, с помощью атомарного системного вызова `MoveFileE`. Успешное переименование файла означает установку контрольной точки и завершение транзакции.

Откат на предыдущую контрольную точку описывается следующим алгоритмом:

1. Из файла `checkpoint` считываются размеры файлов данных на момент создания контрольной точки.
2. Длины файлов данных устанавливаются равными размерам, считанным из файла контрольной точки, что равносильно отмене всех изменений данных файлов с момента создания контрольной точки, так как модификация файлов производится только путем дописывания в конец.
3. Файл `checkpoint.tmp` удаляется.

Справочники используются в системе для получения по идентификаторам значений атрибутов их реальных значений, поэтому ситуация, когда справочник содержит больше данных, чем «требуется» основным файлам вполне допустима. Приведенное допущение позволяет организовать механизм защиты справочников, не используя контрольные точки, а используя метод избыточного копирования данных, базирующийся на следующих принципах:

1. Во время работы со справочником всегда хранится две его копии, одна из которых является резервной и всегда целостной, вторая является рабочей, и в случае возникновения сбоя может быть утрачена.
2. Периодически или в определенные контрольные моменты, а так же при закрытии справочника запускается механизм резервного копирования, результатом работы которого является обновленная версия резервной копии справочника.

Механизм резервного копирования реализован по следующей схеме (см. рисунок 25):

1. Рабочий файл справочника закрывается и перемещается во временный файл, это гарантирует, что он уже не подвержен сбоям.

2. Файл резервной копии переименовывается в файл рабочей версии.
3. Производится копирование новых данных из конца временного файла в конец файла резервной копии.
4. Файл рабочей версии замещает файл резервной копии.
5. Временный файл вновь становится новой рабочей версией, путем переименования.

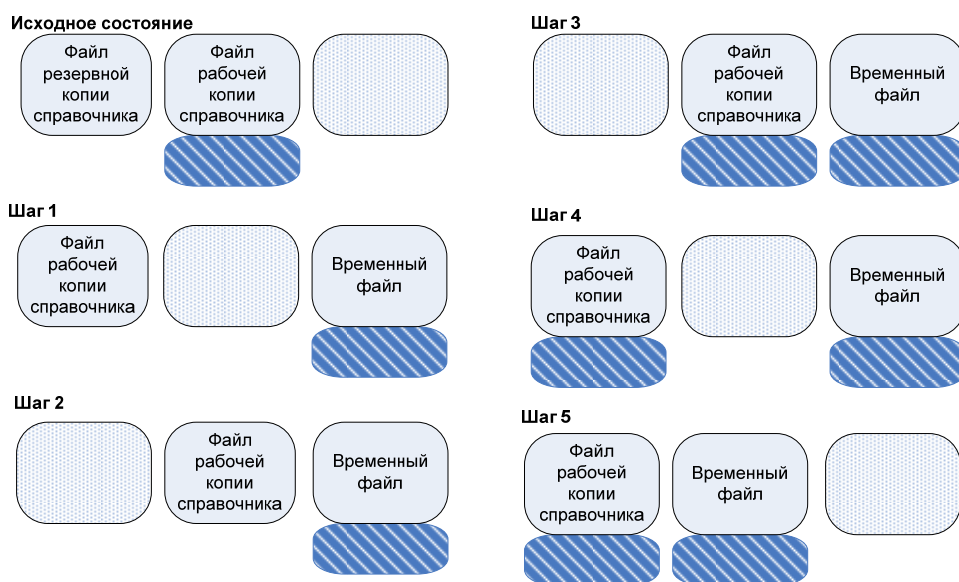


Рисунок 25 — Схема создания резервной копии справочника.

Однако сбой может произойти и в момент выполнения любого из этапов описанного механизма дампирования. Эта проблема решается во время открытия справочника. Как видно из алгоритма, как минимум один из справочников (резервная копия или временный файл) всегда гарантированно находится в целостном состоянии. Загрузка осуществляется следующим образом.

1. Производится попытка загрузить резервную копию справочника.
2. Если резервная копия доступна, то файл резервной копии копируется в файл рабочей копии и справочник считается открытым.
3. Если файл резервной копии не найден или поврежден, то осуществляется попытка загрузки временного файла, в случае удачи он так же копируется в рабочий и справочник считается открытым.
4. Если временный файл так же не удалось открыть, то фиксируется ошибка.

Основным требованием к серверу консолидации является его производительность. Для этого промежуточная буферизация и хранилище на сервере консолидации реализовано

на основе файловой системы. В рабочей папке сервера консолидации расположены три каталога:

1. data – каталог для хранения консолидированных данных.
2. receive – каталог для промежуточного хранения принимаемых данных.
3. import – каталог, содержащий принятые, но еще не обработанные и не помещенные в хранилище пакеты данных от агентов.

Файлы – справочники хранилища так же располагаются внутри рабочего каталога сервера консолидации (см. рисунок 26).

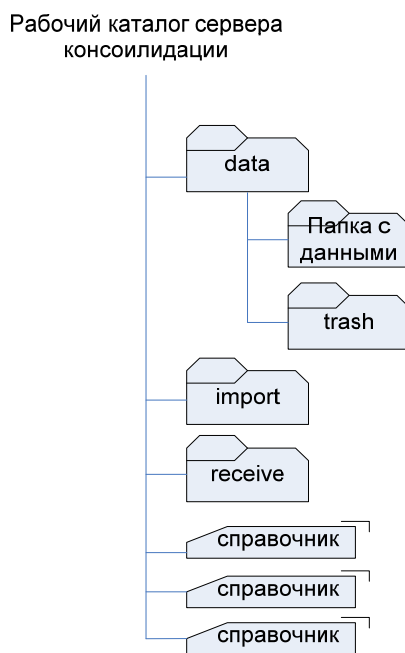


Рисунок 26 — Структура каталогов сервера консолидации.

В момент приема данных от агента внутри каталога receive создается временная папка с уникальным именем. В названии временной папки зашифрован идентификатор агента, от которого были получены передаваемые данные, и текущее время. Как только все пять файлов успешно приняты от агента во временную папку, она перемещается в каталог import, данная операция является атомарной, и означает добавление пакета данных в очередь импорта. Если в момент передачи данных были обнаружены неустраняемые ошибки, то временная папка удаляется, а агент повторяет попытку передачи через некоторое время. Таким образом, очередь пакетов данных для импорта полностью описывается каталогами внутри папки import.

Агент периодически просматривает папку import и вызывает процедуру добавления в хранилище каждого найденного пакета, при этом, принадлежность пакета к конкретному

компьютеру определяется по имени обрабатываемого каталога. Обработка пакета начинается с изменения имени папки пакета – к имени дописывается специальный флаг, показывающий, что обработка пакета начата. В случае аварийного завершения работы, после перезапуска сервер консолидации продолжит обработку очереди импорта именно с этого пакета. В случае любых ошибок во время обработки пакета данных, пакет перемещается в специальную папку data/trash, что позволяет впоследствии проводить анализ ошибок. В случае успешного завершения импорта данных, пакет удаляется.

Важно заметить, что операции изменения имени или удаления каталогов и файлов, несмотря на то, что доступ к ним пользователями ОС не осуществляется, не всегда могут быть выполнены в любой момент времени. Как показывает практика, файлы могут блокироваться на небольшие промежутки времени, например, антивирусным ПО. В таком случае, сервер консолидации не может продолжать нормальное функционирование, пока файлы не будут разблокированы. Такие ситуации обрабатываются везде, где это критично с помощью итерационных попыток обращения к файлам.

Предложенная организация файлов сервера и использование файловой системы для синхронизации работы потоков приема и импорта данных, а так же для описания параметров пакетов позволяет избежать большинства сложностей, связанных с потерей данных из оперативной памяти, которые могли бы возникнуть в случае аварийного завершения работы сервера. При использовании предложенной схемы, при повторном запуске сервера требуется проводить лишь восстановление не до конца дописанных файлов хранилища и выбор консистентной версии справочников.

1.4.2.3 Автоматизированная установка и управление агентами и настройками параметров сбора данных через управляющую консоль

Разработана Консоль управления позволяющая подготовить установочный пакет агентов, настроить работу агентов, наблюдать за параметрами работы агентов и сервера консолидации.

Разработан пользовательский интерфейс консоли управления сбором, состоящий из двух пунктов:

1. «Агенты сбора» – позволяет создавать/изменять/удалять профили настроек агентов сбора, добавлять/удалять/настраивать агенты сбора. Опрашивать и отображать статус работы агентов.
2. «Сервер консолидации» – позволяет просматривать статистику работы сервера консолидации, производить выгрузку и архивирование собранных данных.

Профили агентов позволяют объединять одно или более АРМ пользователей, на которые еще не установлены агенты в конфигурационные группы. Для каждой группы могут быть заданы единые настройки работы и правила установки. Впоследствии параметры работы агентов могут быть изменены как для всей группы (профиля) в целом, так и для отдельного агента внутри группы.

Для создания профиля настройки агентов разработан и реализован пользовательский интерфейс (см. рисунок 27).

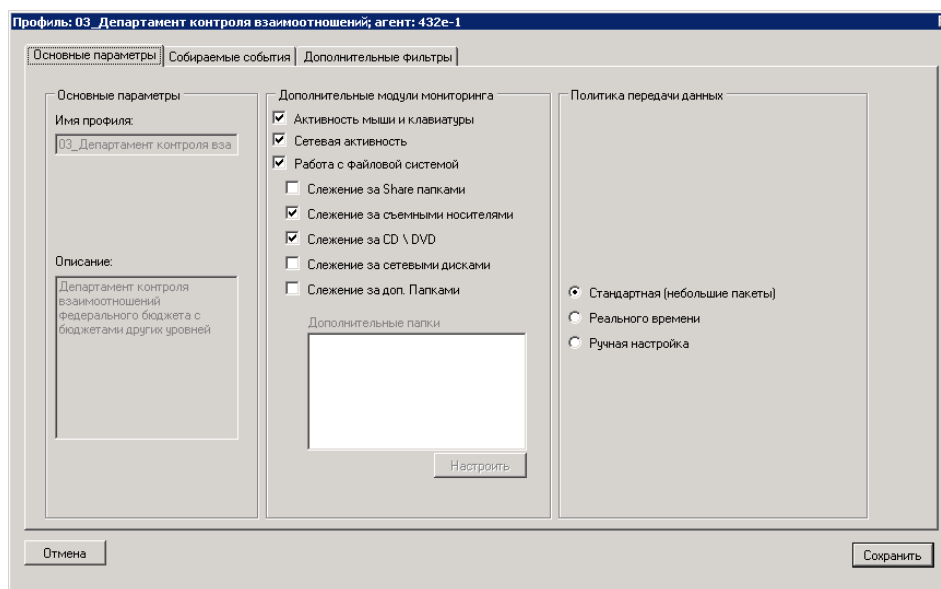


Рисунок 27 — Редактирование основных параметров профиля.

Вкладка «Основные параметры» состоит из трёх разделов:

1. Основные параметры.
2. Дополнительные модули мониторинга.
3. Политика передачи данных.

В разделе «Основные параметры» необходимо указать имя профиля и его описание.

Раздел «Дополнительные модули мониторинга» Позволяет указать, какие дополнительные типы активности пользователя агент будет отслеживать на АРМ. Возможные дополнительные типы собираемой активности: активность работы с мышью и клавиатурой в приложениях, активность работы в сети, работа с файловой системой.

Если требуется мониторинг доступа к папкам на жестком диске АРМ пользователя, то необходимо указать полный путь к папкам, используя кнопку «Настроить». Указывать следует локальные имена папок на АРМ пользователя. Сравнение осуществляется по принципу содержания указанной папки в полном пути. Например, для журналирования

операций с папкой [Program Files]\Advanced Algorithms\AdvAlg IDS Agent, можно задать «AdvAlg IDS Agent».

Раздел «Политика передачи данных» позволяет задать схему передачи данных от агентов на сервер консолидации. Следует выбрать один из способов передачи:

1. «Стандартная» – передача осуществляется либо после сбора 100 событий, либо не реже 1 раза в 100 минут.
2. «Реального времени» – данные передаются сразу же после наступления события (но не чаще, чем раз в 5 секунд). Данный режим требователен к ресурсам АРМ пользователей, ЛВС и сервера консолидации. Не рекомендуется выбирать данный режим работы, не имея на то причин.
3. «Ручная настройка» – передача данных начинается, когда достигнут один из максимумов (объем собранных данных, количество собранных фактов, максимальное время сбора), Указанные параметры необходимо задать в окне, которое появится после выбора данного режима.

В закладке «Собираемая активность» следует указать, какие события из журналов регистрации будут считываться агентами. «Стандартный профиль» содержит полный список событий (см. рисунок 28).

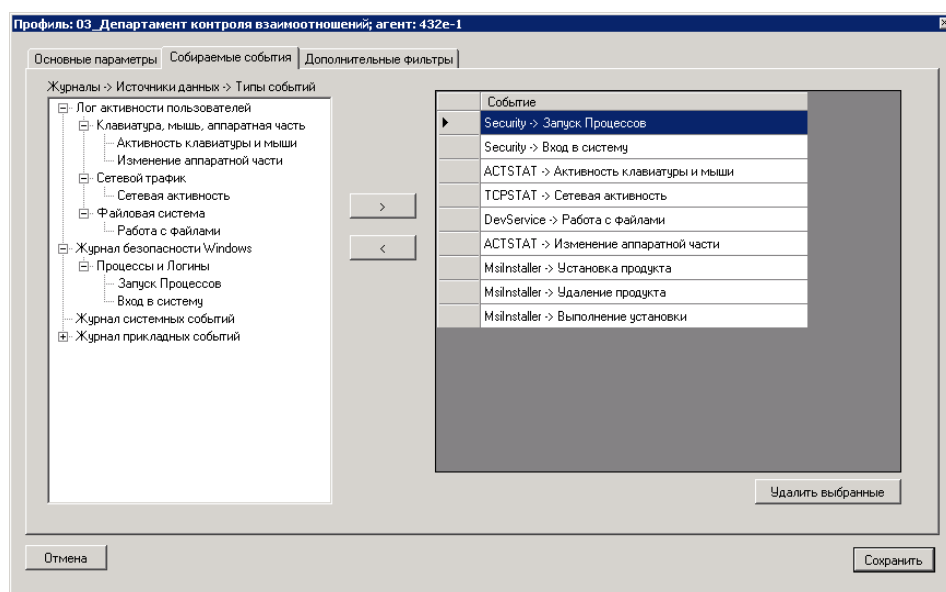


Рисунок 28 — Выбор набора собираемых событий профиля.

В левом окне отображается дерево с типами событий. В правом только те события, которые будут считываться агентом. Для добавления нового события выделите его на дереве и нажмите «<>» или дважды щелкните по нему левой кнопкой мыши. Если такого события еще нет в списке, оно добавится. Для удаления событий выделите их в правом окне

нажатием на левый столбец строки (для выделения нескольких событий используйте клавиши «ctrl» или «shift» или движение мышью), затем нажмите «<>» или кнопку «Удалить выбранные».

В закладке «Дополнительные фильтры» можно указать фильтры на собираемые агентами из журналов регистрации данные (см. рисунок 29). Агент будет передавать на сервер консолидации только данные, прошедшие фильтрацию.

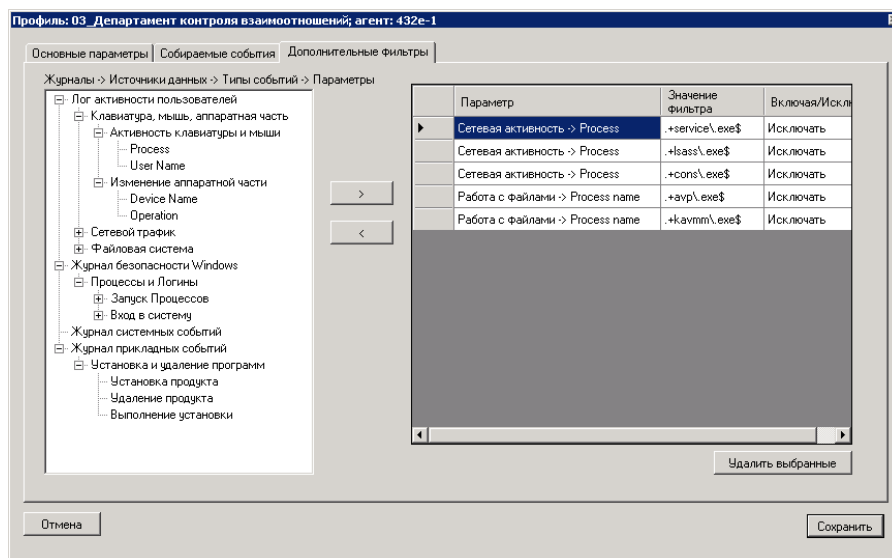


Рисунок 29 — Задание фильтров для профиля.

В левом окне отображается дерево с именами параметров каждого события. Справа - правила включения/исключения события при сборе информации.

Выберите событие в дереве. Нажмите «>» или дважды щелкните по нему. Появится окно «Условие для фильтрации параметра».

Введите условие (см. рисунок 30) и нажмите «Готово». В фильтр будут включены все значения, содержащие данную подстроку.

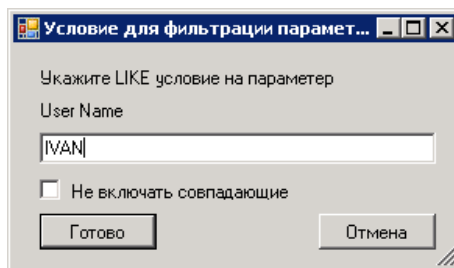


Рисунок 30 — Ввод условия фильтрации для параметра.

Для исключения условия – выделите «Не включать совпадающие» и нажмите «Готово».

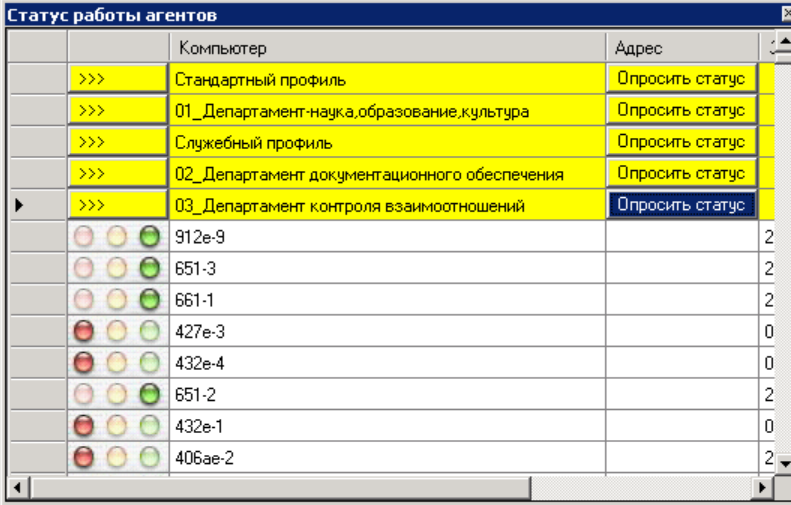
При необходимости, значение фильтра можно редактировать прямо в таблице, для этого выделите нужное значение и щелкните по нему левой кнопкой мыши.

Для удаления фильтров на параметры выделите их в списке и нажмите «<>» или «Удалить выбранные».

Для удаления профиля нажмите правой кнопкой мыши по профилю. В появившемся меню выберите «Удалить профиль». Подтвердите удаление. Нельзя удалить профиль, в рамках которого установлены агенты. Если в рамках профиля установлены агенты, система предложит так же их удалить. Нельзя удалить профиль «Стандартный профиль».

В консоле управления разработан модуль для оперативного получения сведений о функционировании агентов сбора работающих на АРМ сети, а также для получения сведений о сервере консолидации. Для этих целей разработаны программные компоненты, позволяющие получать следующую информацию (см. рисунок 31):

- красный цвет – выключено АРМ пользователя;
- жёлтый цвет – агент работает, но некоторые компоненты сбора отключены;
- зелёный цвет – агент работает, и включены все компоненты сбора.



Статус работы агентов	
Компьютер	Адрес
>>> Стандартный профиль	Опросить статус
>>> 01_Департамент-наука,образование,культура	Опросить статус
>>> Служебный профиль	Опросить статус
>>> 02_Департамент документационного обеспечения	Опросить статус
>>> 03_Департамент контроля взаимоотношений	Опросить статус
912e-9	2
651-3	2
661-1	2
427e-3	0
432e-4	0
651-2	2
432e-1	0
406ae-2	2

Опросить Закрыть Нормальные Проблемные Выключенные

Рисунок 31 — Отображение статуса работы агентов.

В случае если не удалось установить связь с агентом, а АРМ пользователя работает, вся строка с именем агента будет выделена красным цветом. При нажатии левой кнопкой мыши на строке, описывающей статус работы, будет выведено окно с расширенной информацией о статусе работы. Пример детализации статуса работы агента приведен на рисунке 32.

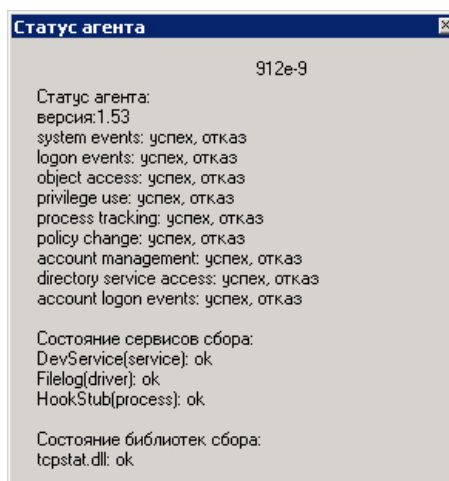


Рисунок 32 — Статус агента сбора.

Состояние сервисов и библиотек сбора информирует о включенности дополнительных модулей сбора на агенте: DevService и Filelog – сбор фактов работы с файловой системой, HookStub – сбор фактов работы с клавиатурой и манипулятором типа «мышь», tcpstat.dll – сбор фактов работы в сети.

В случаях, когда АРМ пользователя включено, но сеанс работы еще не начат, статус работы агента может отображаться как «желтый», а в детализации можно увидеть отключенную компоненту HookStub. Это связано с тем, что компонента сбора автоматически запускается для каждого сеанса работы пользователя с АРМ.

Для получения информации об актуальном состоянии сервера консолидации разработан специальный программный компонент, позволяющий получать следующие сведения (см. рисунок 33).

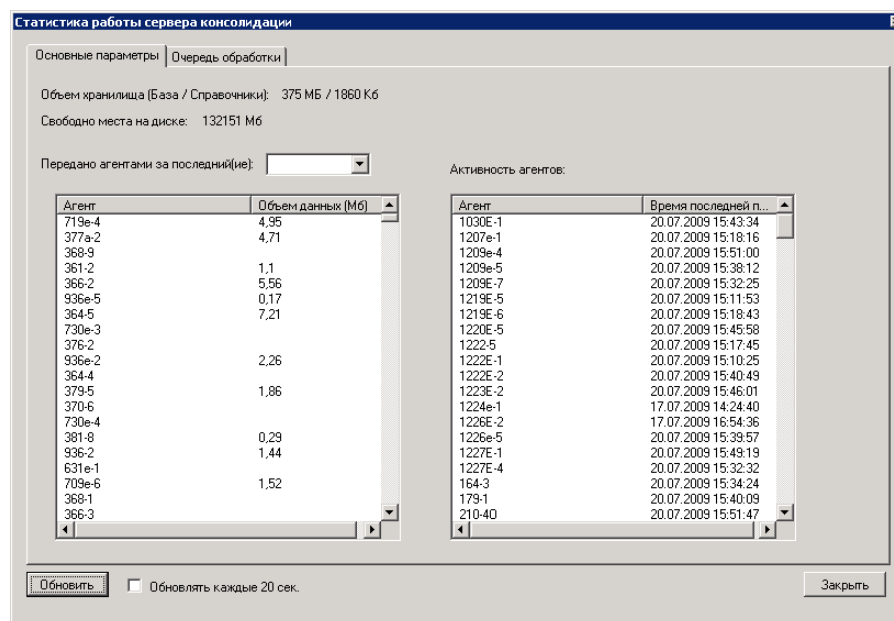


Рисунок 33 — Статистика работы сервера консолидации.

Вкладка «Основные параметры» отражает количество данных принятое от каждого агента за указанный временной промежуток, время последней передачи данных каждым из агентов, а так же объемы консолидированных данных и свободное место на диске сервера консолидации.

Вкладка «Очередь обработки» отражает, какие из пакетов, принятых от агентов еще не были помещены в хранилище («Ожидание»), а какие обрабатываются в данный момент («В обработке»).

Обновление статистики работы сервера консолидации может занять некоторое время (до 5 минут).

Сервер консолидации, представляет собой специальным образом разработанное хранилище, организованное над файловой системой, в функции которого входит получение, предобработка и консолидация данных, поступающих от агентов сбора, работающих на АРМ пользователей. Разработаны программные модули управления сервером консолидации, обеспечивающие реализацию функций создания удаления, сервера консолидации, остановку и запуск его работы.

Установка сервера консолидации сопровождается созданием службы сервера, о чем установщик сообщит информационным сообщением «Служба успешно установлена». Имя службы в системе «LogConsolidator». Имя процесса службы «cons.exe».

Реализована возможность удаления сервера консолидации (для удаления необходимо запустить тот же пакет установки, но выбрать опцию «Удалить») и временной приостановки. Для временной остановки сервера консолидации требуется остановить службу

LogConsolidator на сервере консолидации. Для продолжения работы сервера требуется заново запустить службу LogConsolidator.

1.4.3 Разработка программных компонент, предназначенных для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с текстовой информацией различных типов

Исходя из представленного в пункте 1.2.3 описания архитектуры «Подсистемы 3» (для сбора и анализа информации об особенностях работы пользователя с потоками текстовой информации) следует, что программные компоненты, предназначенные для сбора, предобработки, хранения и управления информацией входят в состав агента мониторинга и сервер консолидации (модуль консолидации). Далее приводится описание процесса разработки перечисленных программных компонент в рамках их функционирования на агенте мониторинга и модуле консолидации. Приведённые работы основываются на уже завершённых работах 1 этапа настоящих ПНИ, а именно «Разработка структур данных, методов сбора, предобработки, хранения и управления для поведенческой информации об особенностях работы пользователя с текстовой информацией» [1].

1.4.3.1 Разработка программных компонент агента мониторинга

В настоящем подпункте приводится описание архитектуры разработанного программного агента мониторинга, а также детали его реализации.

Агент мониторинга состоит из следующих основных программных компонент:

1. *Драйвер–минифильтр*. Получение данных об операциях с файлами на наблюдаемом компьютере и подключаемых к нему внешних носителях.
2. *Служба Windows мониторинга поведенческой информации*. Обработывая сообщения от драйвера–минифильтра, производит сохранение атрибутов наблюдаемых операций и теневое копирование соответствующих документов.
3. *Служба Windows классификации поведенческой информации*. Выполняет применение поведенческих моделей к собранной поведенческой информации.
4. *Служба Windows передачи данных*. Передача собранной поведенческой информации и результатов применения поведенческих моделей модулю консолидации.

Связь между перечисленными программными компонентами изображена на рисунке 34.

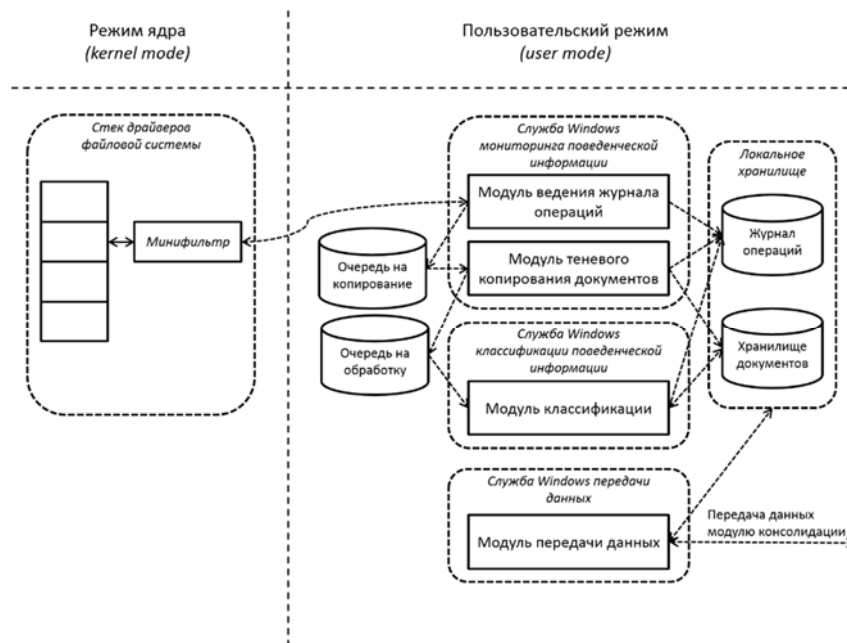


Рисунок 34 — Архитектура агента мониторинга.

Кроме того, для сбора поведенческой информации с незакодированными сообщениями электронной почты, получаемыми и передаваемыми по протоколам IMAP и HTTP (на почтовые Web-системы) с использованием одного из веб-обозревателей (Microsoft Internet Explorer версии 9 и выше) были разработаны дополнительные программные компоненты, описание которых приведено в подпунктах 1.4.3.5 и 1.4.3.6.

Из приведённого списка программных компонент следует, что за сбор, предобработку, хранение и управления информацией об особенностях работы пользователей с текстовой информацией различных типов отвечают следующие модули агента: *драйвер–минифильтр*, *служба Windows мониторинга поведенческой информации*, *служба Windows передачи данных*. Поэтому далее приводятся описания данных программных компонент. реализации решений данных задач. *Служба Windows классификации поведенческой информации* служит только для применения поведенческих моделей к собранным данным, поэтому она далее в настоящем пункте рассматриваться не будет.

1.4.3.2 Драйвер–минифильтр

Разработан драйвер–минифильтр, который обрабатывает IRP-пакеты типа IRP_MJ_CREATE и IRP_MJ_CLEANUP, составляя на каждый обработанный IRP (IRP-пакеты, отвергнутые нижележащими драйверами, игнорируются) сообщение для службы режима пользователя (*Служба мониторинга поведенческой информации*) с полями:

- Путь наблюдаемого файла (в двухбайтовой кодировке);

- Путь к исполняемому файлу процесса, вызвавшему данный запрос;
- SID пользователя, которому принадлежит данный процесс;
- Указатель на структуру FILE_OBJECT (уникальный идентификатор открытого файла [10]);
- Размер файла;
- Флаг директория/файл;
- Флаг, открыт файл с правами на изменение или нет (только для IRP_MJ_CREATE);
- Флаг, отмечен файл на удаление или нет (только для IRP_MJ_CLEANUP).

Драйвер–минифильтр написан на языке C, так как это единственный поддерживаемый Microsoft язык для программирования драйверов.

1.4.3.3 Служба Windows мониторинга поведенческой информации

Разработана системная служба Windows мониторинга поведенческой информации, которая выполняет буферизацию сообщений, поступающих от *драйвера–минифильтра*, и их последующую обработку с помощью двух параллельно выполняющихся нитей (см. рисунок 34): *ведение журнала операций, теневого копирования документов*.

Модуль ведения журнала операций для каждого сообщения от *драйвера–минифильтра*, находящегося в буфере службы, выполняет следующие действия:

1. Производит классификацию операций (т.е. определяет, является ли операция наблюдаемой) по следующим атрибутам: полное имя файла, полное имя исполняемого файла процесса, SID пользователя, имя пользователя, размер файла.
2. Делает запись о наблюдаемой операции с документом в *журнале операций*. Сохраняется следующие данные об операции (см. рисунок 35): путь к файлу, время операции, тип операции (создание, изменение, чтение, перемещение, удаление), информация о процессе (инициализирующем данную операцию), информация о пользователе и вспомогательный путь (новый путь к файлу в случае операции типа «перемещение»).
3. В случае если документ, над которым выполняется наблюдаемая операция, был изменён или впервые зарегистрирован агентом, то добавляет запись с указанием пути наблюдаемого документа в *очередь на копирование*.

Модуль теневого копирования документов для каждой записи в *очереди на копирование* выполняет следующие действия:

1. Копирование соответствующего наблюдаемого документа в *Хранилище документов*, т.е. создание теневой копии;
2. Добавление записи о теневой копии в *Журнал операций*;
3. Добавление записи о наблюдаемой операции и пути соответствующей теневой копии в очередь на обработку.

Журнал операций представляет собой базу данных MS Access. Используемая в *Журнале операций* реляционная модель данных, описывающих поведенческую информацию, представлена на рисунке 35. Копии документов помещаются в специальную директорию, а информация о скопированных агентом документах также сохраняется в *Журнале операций*. Таким образом, *Локальное хранилище* агента представляет собой файл базы данных MS Access (*Журнал операций*) и директорию ФС (*Хранилище документов*), права доступа к которым обычные пользователи не имеют.

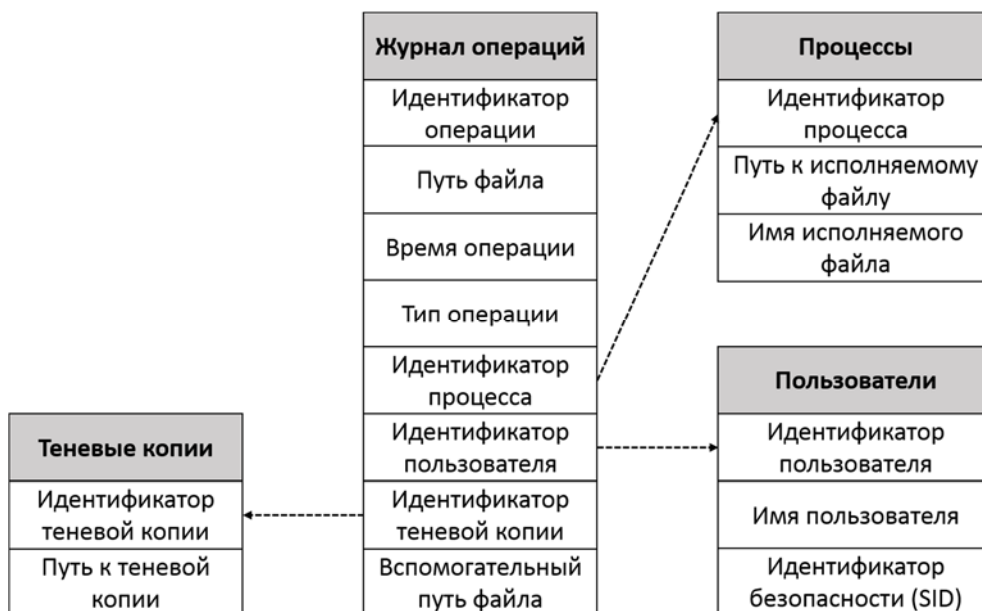


Рисунок 35 — Реляционная модель описания поведенческой информации.

Служба Windows мониторинга поведенческой информации реализована на языке C++, основным критерием выбора являлось быстродействие, которое обеспечивает язык.

1.4.3.4 Служба Windows передачи данных

Разработана системная служба Windows передачи данных, которая выполняет планирование передачи данных путём задания параметров, описывающих максимально

возможный объем переданных данных и максимально возможный интервал времени, в течение которого агент может не передавать данные. Как только одно из максимально возможных значений параметров достигнуто, все собранные данные помещаются в очередь на отправку.

Схема работы при наступлении события начала передачи данных следующая:

1. Формирование пакета файлов для отправки, содержащего данные об операциях с документами (данные из таблиц *Журнала операций*) и файлы теневых копий соответствующих документов. Если отправляемые поведенческие данные обработаны модулем классификации, то:
 - a. в пакет для отправки включается файл с данными таблицы, содержащей результаты применение поведенческих моделей;
 - b. выполняется очистка *Локального хранилища* агента, т.е. удаляются данные из соответствующих таблиц *Журнала операций* и файлы теневых копий.
2. Сформированный пакет файлов для отправки сжимается архиватором *gzip*.
3. В специальную таблицу в *Журнале операций* заносятся данные о том, какая информация была выбрана для отправки, для этого генерируется уникальный идентификатор данного сеанса передачи данных.
4. Выполняется передача данных по сети *модулю консолидации*;
5. Ожидается получение подтверждения приема данных, если таковое получено, то данный факт фиксируется в *Журнале операций*. Если подтверждение не пришло, то попытка передачи повторится при повторном наступлении события начала передачи данных.

Служба Windows передачи данных реализована на языке C++. Для обеспечения безопасной передачи данных по сети используется библиотека OpenSSL [11].

1.4.3.5 Мониторинг электронных сообщений MS Outlook

Задачей мониторинга электронных сообщений является перехват отправляемых и получаемых электронных писем через почтовый клиент Microsoft Outlook. Для решения данной задачи был разработан VSTO-плагин [12], который добавляет следующую функциональность к почтовому клиенту Microsoft Outlook:

- Сохранение текста тела входящего электронного письма (в том числе полученного по протоколу IMAP) в виде файла, а также сохранение прикрепленных к нему файлов. Сохранение перечисленных файлов производится в специальную директорию для

входящих электронных писем, которую можно рассматривать как часть локального хранилища агента мониторинга;

- Сохранение текста тела отправленного электронного письма в виде файла, а также сохранение прикрепленных к нему файлов. Сохранение перечисленных файлов производится в специальную директорию для отправленных электронных писем, которую можно рассматривать как часть локального хранилища агента мониторинга.

Разработанный VSTO-плагин не содержит ресурсоёмких операции, поэтому для удобства и повышения скорости разработки был выбран язык C#.

1.4.3.6 Мониторинг веб-страниц MS Explorer

Задачей мониторинга web-страниц является перехват отправляемых форм и посещаемых web-страниц через браузер Internet Explorer. Для решения данной задачи был разработан ВНО-плагин (англ. Browser Helper Object) [13], который добавляет следующую функциональность к браузеру Internet Explorer:

- Сохранение посещаемой web-страницы в виде файла в специальную директорию для посещаемых страниц, которую можно рассматривать как часть локального хранилища агента мониторинга;
- Сохранение отправляемой форм методом POST протокола HTTP в виде файла (файлов в случае отправки формы типа «multipart/form-data») в специальную директорию для отправленных форм, которую можно рассматривать как часть локального хранилища агента мониторинга. При этом в случае отправки формы типа «multipart/form-data» происходит сохранение всех файлов, содержащихся в форме, по отдельности.

Разработанный ВНО-плагин не содержит ресурсоёмких операции, поэтому для удобства и повышения скорости реализации был выбран язык C#.

1.4.3.7 Разработка программных компонент агента мониторинга

Модуль консолидации должен обеспечивать приём, распаковку и сохранением в единое хранилище поведенческих данных, поступающих от множества агентов мониторинга. Также в задачи модуля консолидации входит предоставление доступа к единому хранилищу для других программных модулей (см. рисунок 13), например, модуля построения и модуля применения поведенческих моделей.

При разработке модуля консолидации поведенческой информации использовались перечисленные далее подходы. Для обеспечения безопасного обмена данными по сети

используется библиотека OpenSSL [11]. Распаковка данных, получаемых от агентов мониторинга, выполняется с помощью архиватора *gzip* [8]. Единое хранилище представляет собой базу данных в СУБД Microsoft SQL Server 2012. Для хранения поведенческой информации используется аналогичная организация данных, которая реализована в агентах мониторинга. Т.е. информация о наблюдаемых операциях хранится в таблицах БД, а копии документов хранятся в директориях ФС, соответствующих каждому зарегистрированному агенту. Для разграничения прав доступа к поведенческой информации задаются соответствующие права как на БД, так и на директории агентов.

Модуль консолидации поведенческой информации реализован в виде отдельного исполняемого файла, написанного на языке C#.

1.5 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей

1.5.1 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач активной аутентификации без использования секретной информации

В данном пункте отчета приставлены результаты разработки программных компонент, входящих в состав «Подсистемы 1» ЭО ПК, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач активной аутентификации без использования секретной информации, выполненных согласно пункту 3.11.5.1 ТЗ.

Программные компоненты разрабатывались на основании результатов исследований, полученных на 2 этапе данных ПНИ, а также разработанной в соответствии с пунктом 3.11.2 ТЗ архитектуре ЭО ПК.

Согласно архитектуре ЭО ПК, программные компоненты для решения указанных задач разрабатывались в следующих логических модулях (компонентах) «Подсистемы 1»:

- Модули активной аутентификации на основе анализа динамики клавиатурного ввода ключевого (не секретного) слова и динамики работы пользователя с

манипулятором мышью при вводе (не секретного) графического символа на основе сгенерированного шаблона;

1.5.1.1 Модули активной аутентификации

Базовая функциональность программных компонент модулей активной аутентификации пользователей по динамике работы с клавиатурой и мышью описана в разделах 1.3.1.1. и 1.4.1.1. В данном разделе описывается функциональность подсистем модулей аутентификации, предназначенных именно для работы с поведенческими моделями пользователей, для чего реализуется следующая функциональность:

- построение модели использования клавиатуры пользователем (ввод текстового пароля);
- построение модели использования мыши пользователем;
- авторизация пользователя в системе на основании построенной модели ввода пароля;
- авторизация пользователя в системе на основании построенной модели использования мыши;

Построение модели поведения пользователя реализовано в виде программы на языке C++, вычисляющей принадлежность очередной попытки авторизации к соответствующей модели и, при условии, что принадлежность превышает установленный порог, авторизация считается успешной, и информация об этом передаётся в операционную систему. По запросу пользователя обе модели авторизации (и клавиатуры, и мыши) можно модифицировать.

Конфигурация и настройка

Доступ к настройкам осуществляется в окне *Password change* (см. рисунок 36), которое вызывается путём нажатия контрольной комбинации CTRL + ALT + DELETE, и последующего нажатия кнопки *Password Change...* Опции пользовательского интерфейса, управляющие параметрами алгоритма построения модели:

- OutKPart - Доля попыток, которые будут считаться неудачными и не будут использованы в создании модели.
- FuzzDeg – Степень размытия нечёткого множества.
- Sigma – Корень из дисперсии распределения
- AuthLimit – Порог принадлежности, при превышении которого авторизация считается успешной.

Значение всех параметров лежит в диапазоне от 0.0 до 1.0.

После изменения нужных параметров алгоритма новые параметры необходимо сохранить путём нажатия кнопки *Save*.

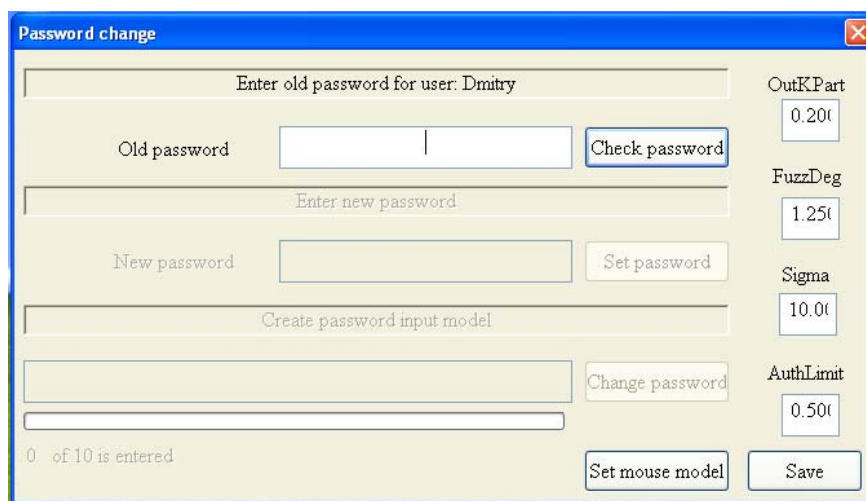


Рисунок 36 — Окно управления алгоритмом построения моделей.

Создание модели авторизации пользователя по динамике работы с клавиатурой

Для того чтобы создать соответствующую модель авторизации пользователя (см. рисунок 37) требуется осуществить следующую последовательность действий:

- В окне *Security* (см. пункт 3.2.7) нажать кнопку *Change password....*
- Ввести старый пароль в поле *Old password*.
- Нажать кнопку *Change password*. В случае успешной проверки старого пароля кнопка *Set password* и поле *New password* станут активными (см. рисунок 38). Иначе будет выведено сообщение об ошибке и потребуется ввести старый пароль заново.
- Ввести в поле *New password* новый (непустой) пароль.
- Подтвердить новый пароль нажатием кнопки *Set password*. В случае успешной установки нового пароля активируется поле для контроля динамики ввода пароля пользователем в пункте *Create password input model*. (см. рисунок 39)
- Ввести новый пароль в разблокированное поле, пока не заполнится счётчик повторений в левом нижнем углу окна. При введении последнего символа пароля поле автоматически очищается, а счётчик попыток увеличивается на 1. Если во время введения очередной попытки произошла ошибка, потребуется повторить ввод данной попытки.
- После заполнения счётчика попыток, активируется кнопка *Change password*.
- Нажатие на эту кнопку завершит процедуру построения модели. Результат будет показан в виде информационного сообщения. (см. рисунок 40)

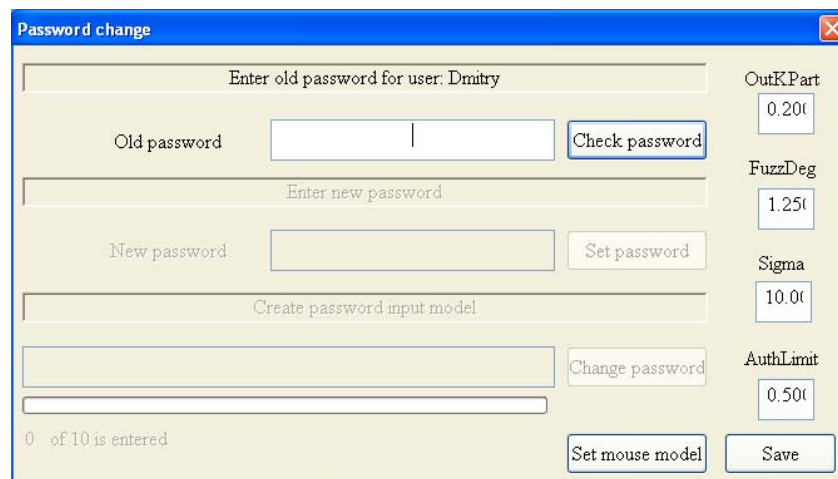


Рисунок 37 — Окно интерфейса для создания модели поведения пользователя при работе с клавиатурой.

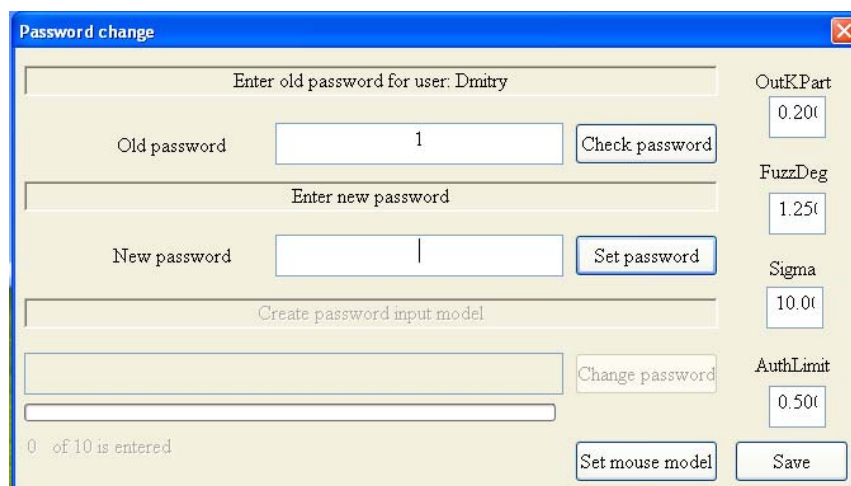


Рисунок 38 — Подготовка к установке нового клавиатурного пароля.

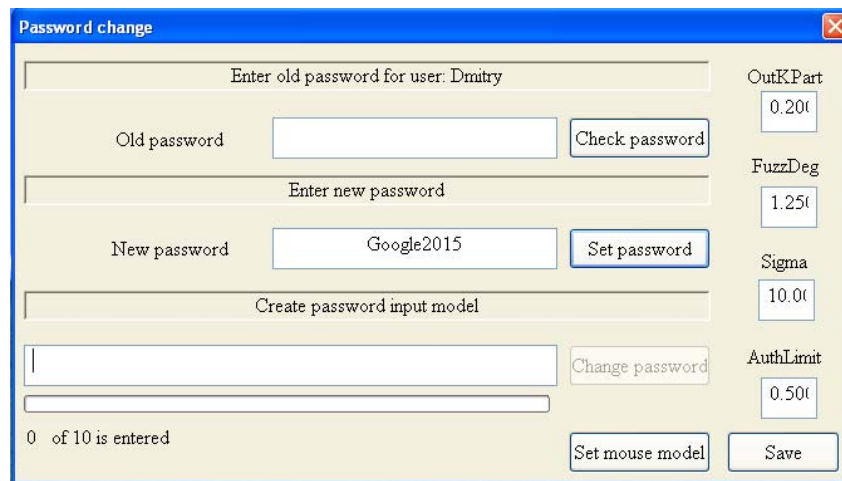


Рисунок 39 — Установка нового клавиатурного пароля.

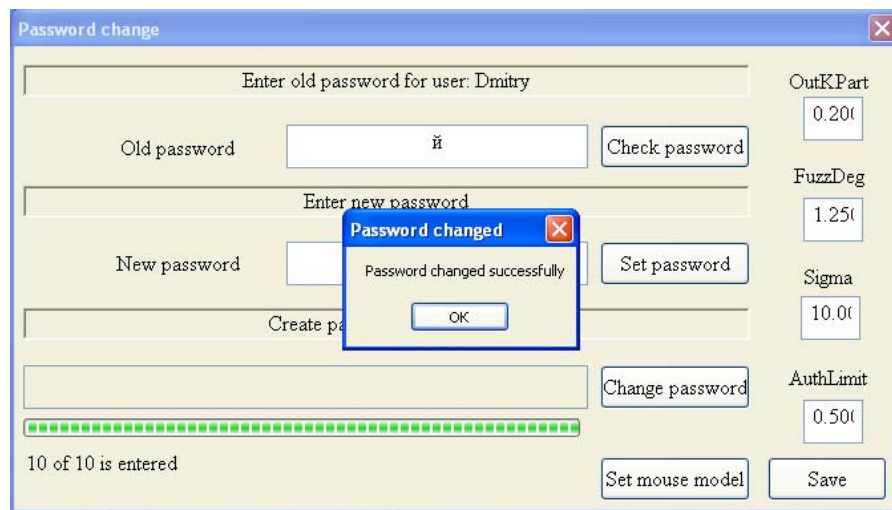


Рисунок 40 — Формирование биометрической модели на основе нового пароля.

Создание модели авторизации пользователя по использованию мыши

Для создания модели авторизации пользователя по использованию мыши (см. рисунок 41) требуется:

- В окне *Security* (см. рисунок 19) нажать кнопку *Change password....*
- В появившемся окне *Password change* нажать кнопку *Set mouse model*.
- Очертить появившийся шаблон с текстом путём нажатия левой кнопки мыши и последующего обвода шаблона (см. рисунок 42).

- Подтвердить текущую попытку входа нажатием на кнопку *Accept*. В случае если пользователь считает, что обвод шаблона получился неудачным, пользователь может нажать на кнопку *Clear* и очертить шаблон заново.
- После десяти попыток окно с шаблоном для обвода закроется, далее требуется закрыть окно смены пароля.



Рисунок 41 — Окно интерфейса для создания модели поведения пользователя при использовании мыши (предустановленный шаблон).

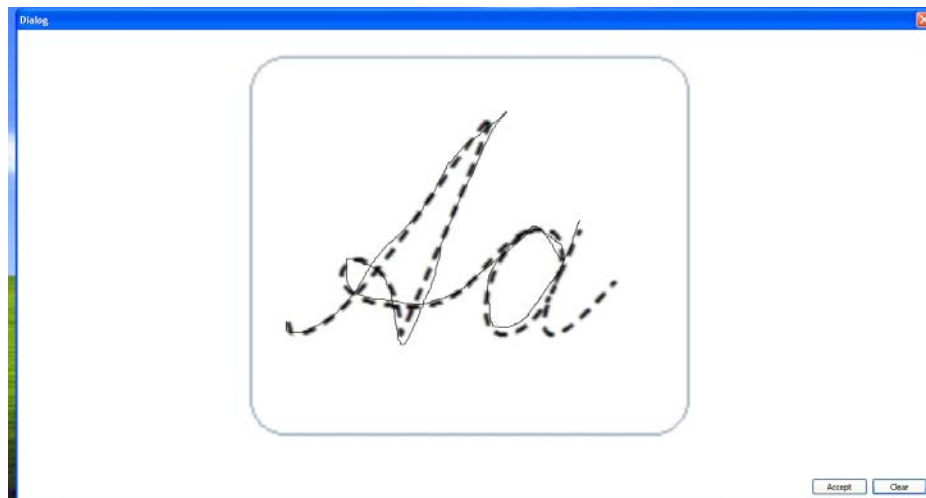


Рисунок 42 — Процедура ввода шаблона с использованием мыши.

1.5.2 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач идентификации пользователей

В данном пункте отчета приставлены результаты разработки программных компонент, входящих в состав всех подсистем ЭО ПК, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач идентификации, выполненных согласно пункту 3.11.2 ТЗ. Программные компоненты разрабатывались на основании результатов исследований, полученных на 2 этапе данных ПНИ, а также разработанной в соответствие с пунктом 3.11.2 ТЗ архитектуре ЭО ПК.

1.5.2.1 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задачи идентификации пользователей на основе поведенческой биометрии работы пользователя с текстовыми данными

Разработка методов машинного обучения и математической статистики для построения и применения поведенческих моделей на основе поведенческой биометрии работы пользователя с текстовыми данными была выполнена на 2-ом этапе настоящих ПНИ [14]. В основе разработанных поведенческих моделей, которые применяются как для идентификации пользователей, так и для раннего обнаружения попыток хищения конфиденциальной информации, лежит тематическое моделирование текстовых данных. В качестве текстовых данных используются текстовые документы различных форматов, полученные из различных источников информации. Поэтому для обработки текстовых документов различных форматов и для построения и применения предложенных поведенческих моделей был разработан *единый* СОМ-объект.

Данный СОМ-объект должен реализовывать широкий функционал по обработки текстовых документов, поэтому для удобства и простоты разработки был выбран язык Python, который имеет множество дополнительных свободных библиотек, в частности, для работы с различными форматами текстовых файлов, работы с кодировками, обработки текстовой информации (разбиение текста на слова, фильтрация стоп-слов, приведение слов к нормализованной форме [15]) и т.п. Однако, весь ресурсоёмкий математический аппарат, служащий для тематического моделирования и прогнозирования многомерных временных рядов, реализован в виде отдельных исполняемых файла, написанных на языке C++.

Приведём список методов разработанного СОМ-объекта:

1. *Извлечение текста документа.* Извлечения текста из документов различных текстовых форматов основано на фильтрах *ifilter* от Microsoft [16]. Для получения доступа к соответствующим *ifilter*, входящим в Microsoft Index Server, используется Python-модуль `win32com.ifilter` [17].
2. *Тематическое моделирование.* Построение тематической модели по коллекции текстовых документов. Предварительная обработка текстов документов реализована на Python с использованием библиотеки NLTK [15]. Построение матриц тематической модели реализовано в виде отдельного исполняемого файла, написанного на языке C++ с использованием высокопроизводительной библиотеки линейной алгебры Eigen [18].
3. *Отображение документов в модельное тематическое пространство.* Представление коллекции документов в тематическом пространстве существующей тематической модели. Предварительная обработка текстов документов реализованы на Python с использованием библиотеки NLTK [15]. Построение матрицы тематического представления документов реализовано в виде отдельного исполняемого файла, написанного на языке C++ с использованием высокопроизводительной библиотеки линейной алгебры Eigen [18].
4. *Прогнозирование многомерного временного ряда.* Построение прогноза многомерного временного ряда на заданное число шагов с помощью разработанного метода прогнозирования на основе ортонормированной неотрицательной матричной факторизации [14]. Метод прогнозирования реализован в виде отдельного исполняемого файла, написанного на языке C++ с использованием высокопроизводительной библиотеки линейной алгебры Eigen [18].

Исходя из представленного в пункте 1.2.3 описания архитектуры «Подсистемы 3» (для сбора и анализа информации об особенностях работы пользователя с потоками текстовой информации) следует, что программные компоненты, предназначенные для построения, управления и применения пользовательских поведенческих моделей для задачи идентификации пользователей входят в состав только *рабочего места аналитика*.

Далее приводится описание процесса разработки перечисленных программных компонент в рамках их функционирования на *рабочем месте аналитика*.

Программный компонент построения пользовательских поведенческих моделей для задачи идентификации пользователей

Поведенческие модели данного типа основаны на прогнозировании тематической направленности пользователя за длительные интервалы времени на основе сложившихся в прошлом тенденций работы пользователя с текстовым контентом различных категорий. Поэтому для построения поведенческой модели первоочередной задачей является задание *модельного времени* [14] для анализируемого пользователя. Затем для текстовых документов анализируемого пользователя, входящих в заданное модельное время, выполняется следующий *аналитический конвейер*:

1. *Извлечение текста документа.* Поведенческие модели строятся и применяются к текстовой информации, поэтому необходимо извлекать текст в единой кодировке из наблюдаемых документов различных форматов (с помощью соответствующей функции разработанного СОМ-объекта), которые могут представлять собой текстовые файлы различных форматов (например: doc, rtf, pdf, и т.п.);
2. *Разбиение модельного времени на интервалы.* Для формирования поведенческой модели для идентификации пользователя заданное модельное время разбивается на последовательно измеренные через некоторые промежутки времени интервалы. Например, в качестве промежутка времени (шага) может быть выбран час, день, а также время, за которое происходит заданное число событий [14]. Далее для каждого полученного интервала все его тексты документов объединяются в один документ интервала.
3. *Тематическое моделирование.* К полученным документам интервалов применяется тематическое моделирование с помощью соответствующей функции разработанного СОМ-объекта. Таким образом, получаем матрицу изменения тематической направленности пользователя за модельное время [14].
4. *Прогнозирование тематической направленности.* Применение метода прогнозирования СОМ-объекта к полученной матрице изменения тематической направленности пользователя.

Таким образом, полученная поведенческая модель содержит файлы тематической модели и файл со значениями прогнозов весов тематик для заданного числа временных интервалов. Данные файлы модели сохраняются в сетевой папке — *хранилище моделей*.

Программный компонент применения пользовательских поведенческих моделей для задачи идентификации пользователей

Применение поведенческой модели данного типа, сохранённой в *хранилище моделей*, возможно только к n временным интервалам (при этом временной интервал соответствует интервалам модели), следующим за модельным временем, на основе которого формировалась модель (при этом n не должно превышать число точек прогноза модели). Тогда для документов каждого из n интервалов выполняется следующий *аналитический конвейер*:

1. *Извлечение текста документа.* Извлечение текста из документов, входящих в анализируемый интервал времени, с помощью соответствующей функции разработанного СОМ-объекта;
2. *Объединение текстов документов.* Объединение всех полученных текстов для анализируемого интервала в один документ интервала;
3. *Отображение документа интервала в модельное тематическое пространство.* Использую тематическую модель, входящую в состав поведенческой модели, получение весов тематик для документа интервала с помощью соответствующей функции разработанного СОМ-объекта;
4. *Вычисление уровня аномальности интервала времени.* Расчёт отклонения тематического представления документа интервала от спрогнозированных значений [14].

Полученное значение отклонения характеризует аномальность анализируемого временного интервала работы пользователя с текстовой информацией. Соответственно, чем больше значение отклонения, тем ниже достоверность того, что пользователь, работающий с защищаемой компьютерной системой, является действительно тем, от имени кого он авторизовался.

1.5.2.2 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задачи идентификации пользователей на основе динамики работы с устройствами ввода

Программное обеспечение Рабочего места аналитика и Модуля фоновой идентификации физически является единым набором программных компонент, разработанных на языке Python3, предоставляющим интерфейсы для решения следующих задач:

1. *Предобработка данных.*

2. *Вычисление признаков динамики работы с клавиатурой.*
3. *Вычисление признаков динамики работы с мышью.*
4. Классификации пользователей.
5. Обеспечение идентификации пользователей на основе предварительно построенных моделей в режиме близком онлайн

Функциональность разработанных программных компонентов следующая:

1. Программный компонент предобработки данных
 - Фильтрация событий динамики работы пользователей с клавиатурой и мышью.
 - Выделение составных событий динамики работы пользователей с клавиатурой и мышью.
2. Программный компонент вычисления признаков динамики работы с клавиатурой
 - Разбиение последовательности событий динамики работы пользователей с клавиатурой на временные окна заданной длины.
 - Расчет и сохранение в файл векторов признаков динамики работы с клавиатурой.
3. Программный компонент вычисления признаков динамики работы с мышью.
 - Разбиение последовательности событий динамики работы пользователей с мышью на временные окна заданной длины.
 - Расчет и сохранение в файл векторов признаков динамики работы с мышью.
4. Программный компонент классификации пользователей.
 - Дискретизация по квантилям рассчитанных признаков динамики работы пользователей с клавиатурой и мышью.
 - Определение принадлежности вектора признаков динамики работы пользователя с клавиатурой тому пользователю, на котором производилось обучение классификатора.
 - Определение принадлежности вектора признаков динамики работы пользователя с мышью тому пользователю, на котором производилось обучение классификатора.
 - Оценка качества классификации при наличии информации о том, какому из пользователей на самом деле принадлежат вектора признаков
5. Программный компонент идентификации пользователей в режиме близком к онлайн

- Установка режима взаимодействия(канала связи) с Модулями сбора и первичной обработки информации о динамике работы пользователя с устройствами ввода-вывода (Агенты).
- Получение очередных блоков информации (векторов признаков для соответствующих временных окон) от Агента соответствующего пользователя.
- Классификация пользователя согласно п.4 и формирование реакции по результатам проведенной классификации

Функциональность п.1 – п.3 частично дублируется с функциональностью модуля сбора информации о динамике работы пользователя с устройствами ввода-вывода.

Схема взаимодействия программных компонентов для построения, управления и применения пользовательских поведенческих моделей для задачи идентификации пользователей

Решение задачи идентификации обеспечивается взаимодействием следующих программных модулей "Подсистемы 1" ЭО ПК:

- Модуль сбора информации о динамике работы пользователя с устройствами ввода-вывода (*Агент*)
- Рабочее место аналитика (*РМ аналитика*)
- Модуль фоновой идентификации (*Модуль Идентификации*)

Схема функционирования:

1. Стадия сбора первичных событий о динамике работы пользователей с устройствами ввода-вывода

На данном этапе на локальных рабочих местах пользователей работают Агенты сбора, сконфигурированные для записи всех перехватываемой информации о работе с устройствами ввода в централизованное хранилище данных. В роли данного хранилища в настоящее время используются каталоги файловой системы Рабочего места аналитика, сконфигурированные для возможности удалённого доступа на чтение/запись всех Агентов. По окончании этапа Агенты сбора выключаются.

2. Стадия обработки и классификации

На данном этапе оператор РМ аналитика запускает модуль классификации, осуществляющий рекурсивный обход каталогов с информацией об активности пользователей собранных Агентами на Стадии 1. Во время данного этапа

осуществляется предобработка, расчёт признаков и классификация данных всех доступных пользователей, с сохранением модели каждого пользователя в локальном хранилище РМ Аналитика. Также, формируется набор наиболее значимых признаков, которые будут учитываться в дальнейшем при обработке пользовательских данных, полученных от клавиатуры (данный набор нужен для уменьшения вычислительного времени, необходимого для обработки данных, ввиду того что исходный набор клавиатурных признаков измеряется тысячами, что требует неоправданно высоких вычислительных затрат). Указанный набор целевых признаков для клавиатуры формируется в виде CSV файлов и записывается в исходные каталоги каждого пользователя.

3. *Запуск Агентов в режиме идентификации*

На данной стадии Агенты сбора запускаются в режиме идентификации: помимо сбора первичной информации об активности пользователя, аналогичной Стадии1, Агенты осуществляют предобработку, разбиение на временные окна, и формирование векторов признаков, соответствующих данным окнам. Построение векторов признаков для клавиатуры осуществляется согласно содержимому CSV-файлов со значимыми признаками для клавиатуры, полученных от РМ Аналитика на Стадии 2. Результаты работы агентов записываются в те же каталоги хранилища файловой системы.

4. *Запуск модуля фоновой идентификации*

На стенде, физически совмещенным с РМ Аналитика, запускается программный компонент модуля фоновой идентификации. Модуль производит рекурсивный обход каталогов файловой системы, с целью отслеживания изменений в каждом из подкаталогов, соответствующих Агенту сбора каждого из пользователей. При обнаружении изменений для любого из пользователей (появление нового рассчитанного вектора признаков) Модуль Идентификации производит вызов соответствующих программных компонент РМ Аналитика с целью классификации данного пользователя. Результат классификации в настоящее время записывается только в виде информационных сообщений РМ Аналитика и Модуля Идентификации. В дальнейшем планируется, что результат классификации будет возвращаться Агентам сбора, с целью обеспечения соответствующей реакции Агентами сбора непосредственно на локальных рабочих места пользователей.

1.5.2.3 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задачи идентификации пользователей на основе информации об особенностях работы пользователя с информационными и вычислительными ресурсами компьютерной системы

В состав экспериментального образца программного обеспечения входят следующие программные компоненты для построения и применения моделей идентификации пользователей на основе информации об особенностях их работы с информационными и вычислительными ресурсами компьютерной системы:

1. Программный компонент взаимодействия с пользователем, базой данных, первичной обработки данных и формирования на их основе временных окон журналов событий, характеризующих взаимодействие пользователя с системой.
2. Программный компонент построения модели пользователя на основе временных окон журналов событий, характеризующих взаимодействие пользователя с системой.
3. Программный компонент применения модели пользователя, характеризующих взаимодействие пользователя с системой.
4. Программный компонент прогнозирования временных рядов построенных на основе временных окон журналов, характеризующих взаимодействие пользователя с системой.

Программные компоненты ЭО реализуют алгоритмы и подходы, описанные на 2 этапе данного ПНИ (Отчет ПНИ, Этап 2, пункт 3.2.1) и решают следующие информационные задачи:

1. Программный компонент взаимодействия с пользователем, базой данных, первичной обработки данных и формирования на их основе документов, характеризующих взаимодействие пользователя с системой.
 - Взаимодействие с базой данных, содержащей системные и прикладные журналы;
 - Обработка данных журналов для формирования документов, характеризующих взаимодействие пользователя с системой;
 - Предоставление интерфейса для создания выборки, модели или отчета;
 - Создание таблиц выборок и графиков моделей и отчетов;
2. Программный компонент построения модели пользователя на основе временных окон журналов событий, характеризующих взаимодействие пользователя с системой.

- Выделение латентных тематик из документов, характеризующих взаимодействие пользователя с системой.
 - Построение временных рядов тематик на основе документов, характеризующих взаимодействие пользователя с системой.
3. Программный компонент применения модели пользователя, характеризующих взаимодействие пользователя с системой.
- Применение модели пользователя на основе документов, характеризующих взаимодействие пользователя с системой и модели пользователя.
4. Программный компонент прогнозирования временных рядов построенных на основе временных окон журналов, характеризующих взаимодействие пользователя с системой.
- Прогнозирование временных рядов тематик для выявления аномалий в работе пользователя с ситемой

ЭО ПО предназначен для решения следующих основных задач:

- создание выборок из журналов;
- построение моделей по выборкам;
- построение отчетов по применению модели;
- просмотр результатов.

Для реализации экспериментального образца программного обеспечения используется следующее общесистемное и стороннее ПО:

- интерпретатор Python 2.7.3 и выше в рамках 2.7;
- система управления базами данных (СУБД) SQL Server 2005 и выше;
- консоль Microsoft Management Studio версии 3.0
- модуль Python pywin 219 и выше;
- интегрированная среда разработки программного обеспечения Visual Studio 2013 и выше;
- программная платформа .Net версии 4.0 и выше
- комплект средств разработки MMC 3.0 SDK

Главное окно консоли состоит из 3 частей (см. рисунок 43). Слева отображается дерево узлов, посередине вид выбранного узла, справа меню для выбранного узла. Корневым узлом является «Идентификация пользователей». Меню настройки для этого узла задает базовые настройки ЭО. Меню обновить обновляет все дочерние вершины. У корневого узла есть 3 дочерних узла. Узел «Выборки событий» позволяет создавать выборки из журналов. Узел

«Модели идентификации» позволяет построить модели по выборкам. Узел «Результаты идентификации» позволяет построить отчеты по применению модели.

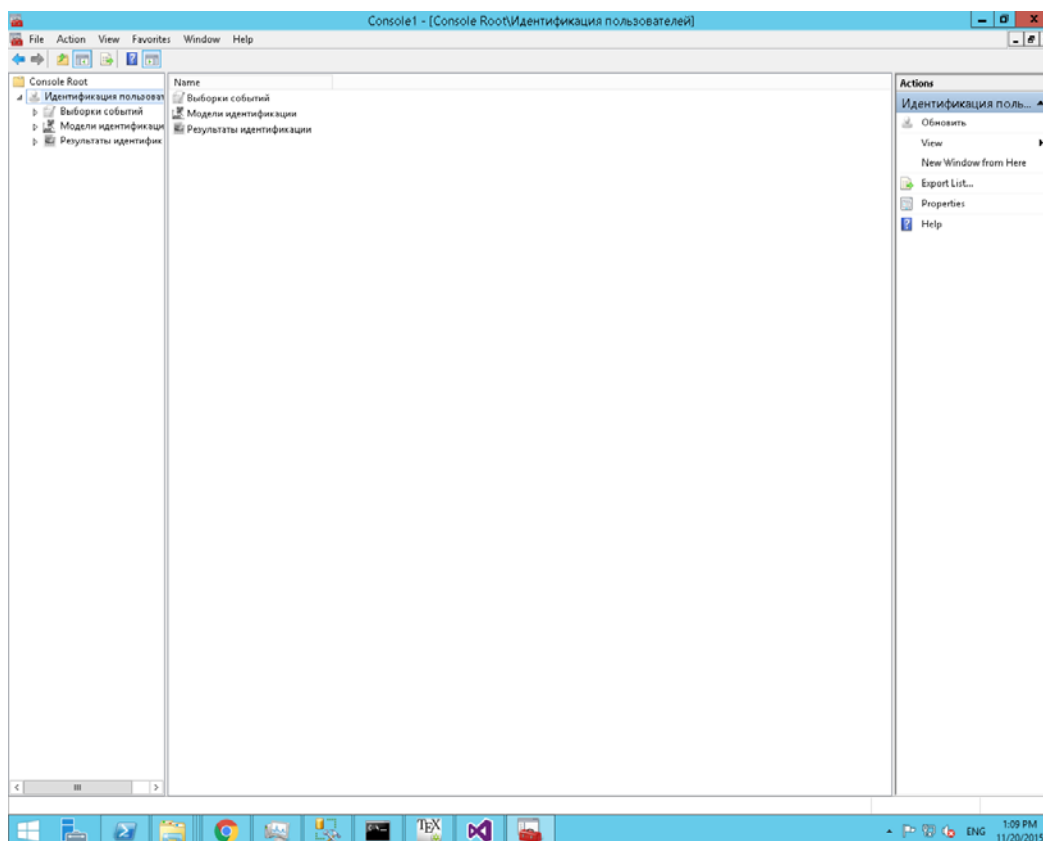


Рисунок 43 — Главное окно рабочего места аналитика.

Интерфейс создания выборок предоставляется меню «Создать выбоку» узла «Выборки событий» (см. рисунок 44).

- «Название выборки». Фиксирует уникальное имя. Имя может состоять из букв русского и английского алфавита, цифр, других символов (кроме \, /, ?, :, *, ", >, <, |, _).
- «Типы событий». Перечень типов событий, объединяемых в выборку. Задается с помощью семи флагов (Process_events, Login_events, Traffic events, Active_events, DeviceStat_events, Hardware_events, Software_events).
- «Атрибуты событий». Перечень атрибутов событий, включаемых в выборку. Задается с помощью флагов. Флаги становятся видимыми, когда выбран хотя бы один тип событий.
- «Временной интервал». Фиксирует временной интервал для всей выборки. Дата, с которой начинается интервал, должна быть раньше, чем дата, которой он кончается.
- «Дополнительные условия». Дополнительные условия на свойства атрибутов событий для каждого из типов. Состоят из трех частей. Правая часть задает столбец к которому применяется условие, средняя часть задает условие сравнения, левая – значение для сравнения.

- «Описание». Задаёт описание выборки в текстовом формате. Является необязательным для создания выборки.
- «ОК». Нажатие этой кнопки создаёт выборку.
- «Отмена». Нажатие этой кнопки отменяет создание выборки.

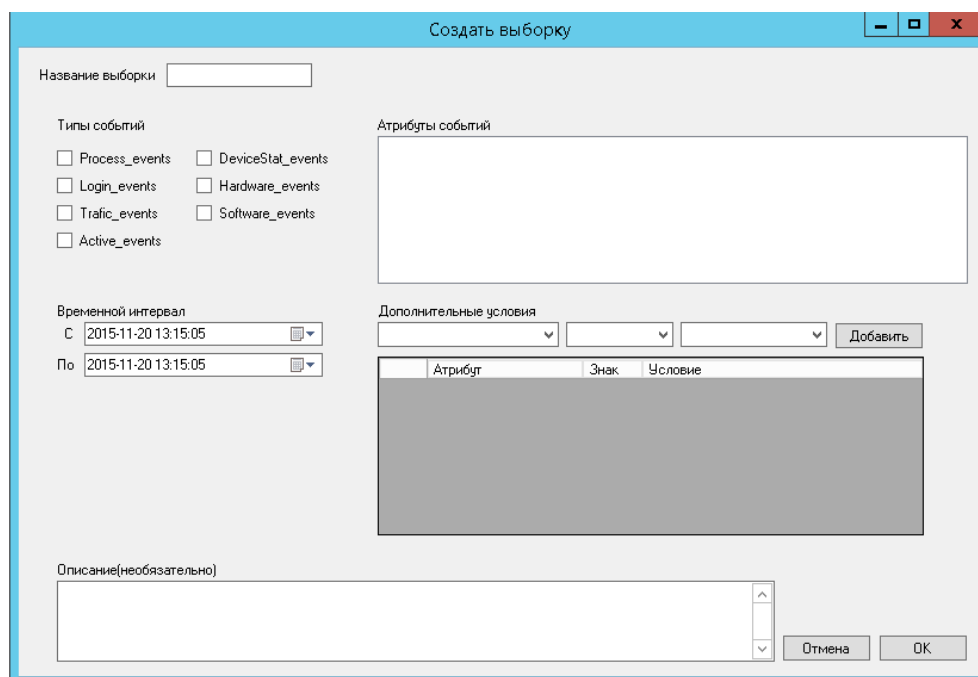


Рисунок 44 — Создание выборки.

Интерфейс построения моделей по выборкам предоставляется меню «Создать модель» узла «Модели идентификации» (см. рисунок 45).

- «Название модели». Фиксирует уникальное имя. Имя может состоять из букв русского и английского алфавита, цифр, других символов (кроме \, /, ?, :, *, ", >, <, |, _).
- «Выборка для обучения». Фиксирует имя существующего фильтра для обучения. Поле со списком отображает допустимые варианты – уже созданные выборки.
- «Описание фильтра». Отображает описание фильтра, если оно было задано при создании выборки.
- «Параметры модели»:
 - «Количество квантилей». Задаёт на сколько интервалов дискретизируются числовые столбцы журнала при составлении документов модели. Допустимое значение – целое число от 1 до 10. Значение по умолчанию 10.
 - «Шаг дискретизации». Задаёт размер временного окна для модели. Поле со списком справа позволяет задать тип временного интервала - либо абсолютное время в минутах, часах или днях, либо абсолютное значение числа событий. Допустимое значение – целое число от 1 до 10000. Значение по умолчанию 1.
 - «Число тематик». Фиксирует число тематик для модели. Допустимое значение – целое число от 1 до 10. Значение по умолчанию 1.
 - Доп. настройки алгоритма разложения и прогнозирования временных рядов

1. «Параметр ортогональности». Используется для построения матрицы обратной к матрице отображения событий в темы. Допустимое значение – целое число от 10 до 10000. Значение по умолчанию 100.
 2. «Минимальное значение DF». Минимальное значение частоты событий в документах. Допустимое значение – целое число от 1 до 100. Значение по умолчанию 2.
 3. «Использовать others». Сформировать отдельную тему для событий, не попавших не в одну тему.
 4. «Начальное число при генерации случайных чисел». Допустимое значение – целое число от 1 до 100. Значение по умолчанию 5.
 5. «Число итераций». Задаёт число итераций алгоритма разложения и прогнозирования временных рядов. Допустимое значение – целое число от 10 до 10000. Значение по умолчанию 100.
- «Создавать модель динамически с промежутком». Позволяет автоматически пересоздавать модель через равные промежутки времени, удаляя старые события из выборки и добавляя новые. Временной интервал между началом и концом событий в выборке сохраняется. Допустимое значение – целое число от 1 до 60. Значение по умолчанию 1. Поле со списком справа позволяет задать единицу измерения времени промежутка - минута, час или день.
 - «Описание». Задаёт описание модели. Является необязательным для создания модели.
- «ОК». Нажатие этой кнопки создаёт модель.
 - «Отмена». Нажатие этой кнопки отменяет создание модели.

Рисунок 45 — Создание модели.

Интерфейс построения отчетов по применению модели предоставляется меню «Создать отчет» узла «Результаты идентификации» (см. рисунок 46).

- «Название отчета». Фиксирует уникальное имя отчета. Имя может состоять из букв русского и английского алфавита, цифр, других символов (кроме \, /, ?, :, *, ", >, <, |, _).
- «Модель для применения». Фиксирует модель, по которой прогнозируются темы событий. Поле со списком отображает допустимые варианты – уже созданные модели.
- «Выборка для применения». Имя выборки для построения отчета. Поле со списком отображает допустимые варианты – уже созданные выборки.
- «Начало прогноза». Фиксирует начальное время, с которого выбираются события для сопоставления с моделью. Ограничено временем последнего события в выбранной выборке событий.
- «Задать начало с окна». Позволяет задать начальное время с определенного временного окна выборки. Допустимое значение – целое число от 0 до максимального окна в выборке. Разбиение на временные окна производится аналогично разбиению в выбранной модели.
- «Горизонт прогноза». Задает количество окон, на которое строится прогноз. Допустимое значение – целое число от 1 до 100. Значение по умолчанию 1.
- «Шаг сезонности». Задает периодичность функций весов тематик в окнах. Допустимое значение – целое число от 1 до 100. Значение по умолчанию 1.
- «Темы для отчета». Перечень тематик, включаемых в отчет. Задается с помощью флагов, соответствующих латентным тематикам, выделенным в модели.
- «Описание фильтра». Отображает описание фильтра, если оно было задано при создании выборки.
- «Описание модели». Отображает описание модели, если оно было задано при создании модели.
- «Описание отчета». Задает описание отчета. Является необязательным для создания отчета.
- «ОК». Нажатие этой кнопки создает отчет.
- «Отмена». Нажатие этой кнопки отменяет создание отчета.

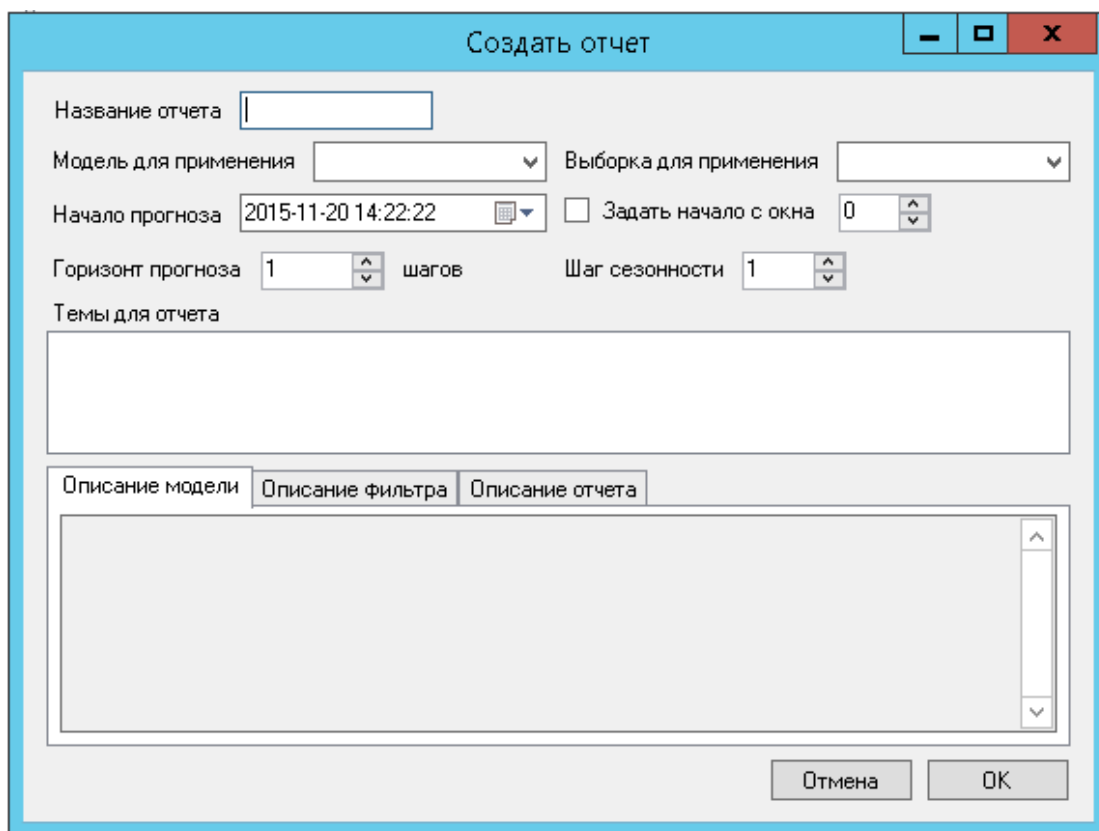


Рисунок 46 — Создание отчета.

Каждая вершина выборки, модели или отчета содержит меню «Просмотреть», которое выводит выборку событий в таблице (см. рисунок 47). Таблица представляет собой последовательность событий, отсортированных по времени. Каждая строка таблицы соответствует одному событию, а столбцы соответствуют атрибутам события. В таблице отображаются атрибуты, выбранные при создании выборки, соответствующей данному узлу, а также время события. Для модели добавляется столбец показывающий временное окно, в которое попало событие и столбец, отображающий веса выделенных тем. Для отчетов добавляется столбец, показывающий аномальность события, а также события выделяются оттенком красного цвета более ярким для более аномальных событий (см. рисунок 48). При двойном нажатии на событие отображаются графики с красной вертикальной линией, показывающий временное окно, в которое произошло событие. Этот же функционал работает, если просто переключиться на узел. Также для вершин моделей и отчетов доступен просмотр графиков с помощью меню «Просмотреть графики», которое выводит графики в отдельном окне (см. рисунок 49). График «Тема» отображает функцию весов тематик для соответствующей темы. Для вершины отчета также начиная с точки прогноза пунктирной линией отображается спрогнозированный временной ряд. График «anomaly» отображает аномальность событий временных окон для отчета. График «approx» показывает

аппроксимацию событий, график «others» веса событий, не соответствующих выделенным тематикам. Поле флагов сверху позволяет выбрать какие графики должны быть видны в текущий момент.

	tmStart	source	type	duration	status	process	computer	userDomainName	userName
▶	3/28/2015 5:25:42 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/28/2015 9:17:12 AM		Service logon	0	Success logon	ADVAPI	XPX86	NT AUTHORITY	NETWORK S
	3/28/2015 9:18:33 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/28/2015 9:18:33 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/28/2015 9:18:33 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/28/2015 11:42:22 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/28/2015 12:06:16 PM					SRVTRACKER.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/28/2015 12:11:52 PM					SRVTRACKER.EXE	WIN7X86	NT AUTHORITY	система
	3/28/2015 12:16:52 PM					SVCHOST.EXE	WIN7X86	NT AUTHORITY	система
	3/28/2015 12:16:52 PM					SVCHOST.EXE	WIN7X86	NT AUTHORITY	система
	3/28/2015 12:16:52 PM					SVCHOST.EXE	WIN7X86	NT AUTHORITY	система
	3/28/2015 4:28:01 PM		Service logon	0	Success logon	ADVAPI	XPX86	NT AUTHORITY	NETWORK S
	3/29/2015 12:06:18 AM					SRVTRACKER.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/29/2015 12:12:02 AM					SRVTRACKER.EXE	WIN7X86	NT AUTHORITY	система
	3/29/2015 7:04:38 AM		Service logon	0	Success logon	ADVAPI	XPX86	NT AUTHORITY	NETWORK S
	3/29/2015 7:05:42 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/29/2015 7:06:12 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/29/2015 7:06:12 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/29/2015 9:20:30 AM					SVCHOST.EXE	WIN7X86	NT AUTHORITY	система
	3/29/2015 9:20:52 AM					SVCHOST.EXE	WIN7X86	NT AUTHORITY	система
	3/29/2015 9:30:01 AM					SVCHOST.EXE	XPX86	NT AUTHORITY	SYSTEM
	3/29/2015 12:12:11 PM					SRVTRACKER.EXE	WIN7X86	NT AUTHORITY	система
	3/29/2015 12:29:21 PM					SRVTRACKER.EXE	XPX86	NT AUTHORITY	SYSTEM

Рисунок 47 — Выборка событий.

tmStart	parent	duration	status	process	computer	userDomainName
3/27/2015 9:17:00 AM	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 9:32:00 AM	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 9:47:00 AM	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 10:02:00 ...	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 10:17:00 ...	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 10:32:00 ...	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 10:47:00 ...	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 11:02:00 ...	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 11:17:00 ...	SVCHOST.EXE	0	Process Run	MSFEEDSSYNC.EXE	XPX86	NT AUTHORITY
3/27/2015 12:18:02 ...	SMSS.EXE	0	Process Run	CSRSS.EXE	XPX86	NT AUTHORITY
3/27/2015 12:18:02 ...	SMSS.EXE	0	Process Run	WINLOGON.EXE	XPX86	NT AUTHORITY
3/27/2015 1:07:18 PM	SVCHOST.EXE	0	Process Run	WUAUCLT.EXE	XPX86	NT AUTHORITY
3/27/2015 1:50:46 PM	SMSS.EXE	0	Process Run	CSRSS.EXE	XPX86	NT AUTHORITY
3/27/2015 1:50:46 PM	SMSS.EXE	0	Process Run	WINLOGON.EXE	XPX86	NT AUTHORITY
3/27/2015 1:54:10 PM	SVCHOST.EXE	0	Process Run	WUAUCLT.EXE	XPX86	NT AUTHORITY
3/27/2015 3:16:38 PM	SMSS.EXE	0	Process Run	CSRSS.EXE	XPX86	NT AUTHORITY
3/27/2015 3:16:38 PM	SMSS.EXE	0	Process Run	WINLOGON.EXE	XPX86	NT AUTHORITY
3/27/2015 3:23:11 PM	SMSS.EXE	0	Process Run	CSRSS.EXE	XPX86	NT AUTHORITY
3/27/2015 3:23:11 PM	SMSS.EXE	0	Process Run	WINLOGON.EXE	XPX86	NT AUTHORITY
3/27/2015 4:24:12 PM	SVCHOST.EXE	0	Process Run	HELPSVC.EXE	XPX86	NT AUTHORITY
3/27/2015 4:24:14 PM	SVCHOST.EXE	0	Process Run	WMIPRVSE.EXE	XPX86	NT AUTHORITY
3/27/2015 4:46:48 PM	SMSS.EXE	0	Process Run	CSRSS.EXE	XPX86	NT AUTHORITY
3/27/2015 4:46:48 PM	SMSS.EXE	0	Process Run	WINLOGON.EXE	XPX86	NT AUTHORITY

Рисунок 48 — Выборка событий с оценкой аномальности.

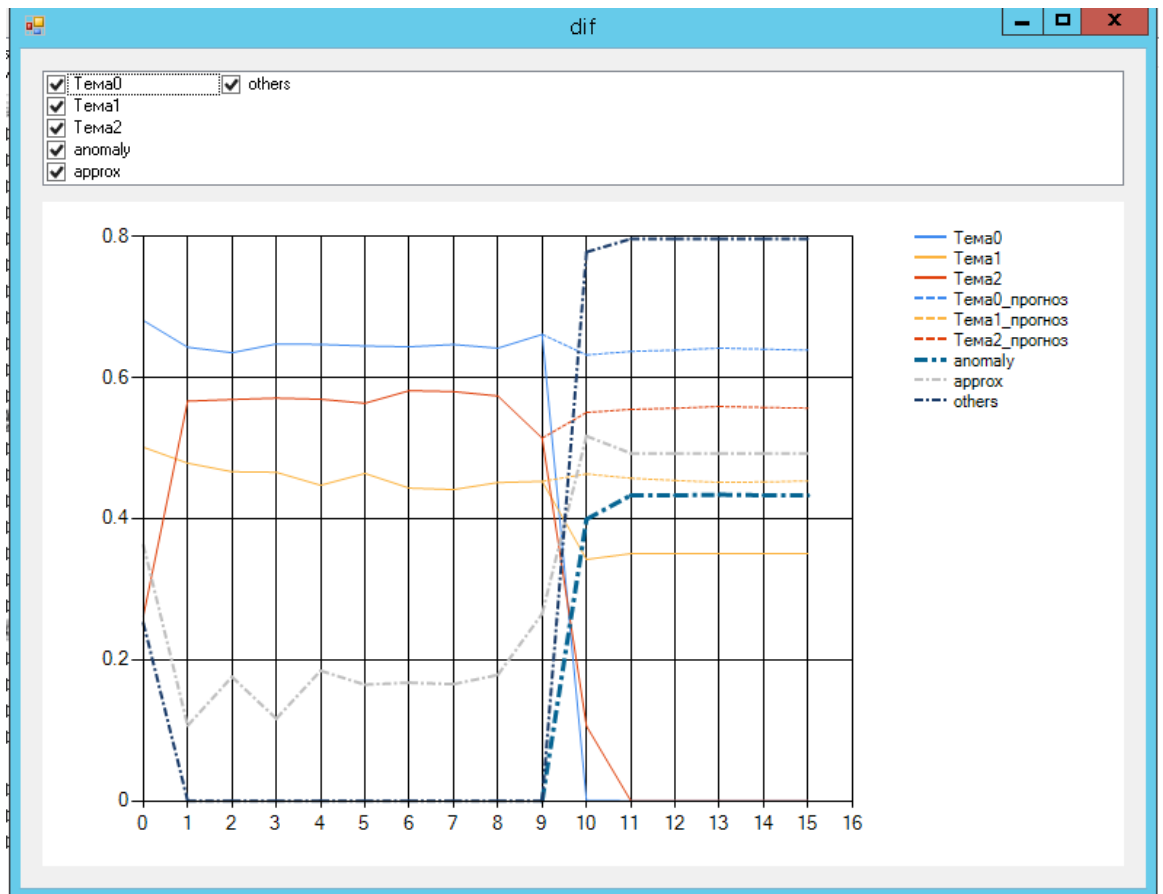


Рисунок 49 — Графическое представление результатов применения модели.

1.5.3 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задач раннего обнаружения внутренних вторжений и попыток хищения конфиденциальной информации

1.5.3.1 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задачи раннего обнаружения попыток хищения конфиденциальной информации на основе поведенческой биометрии работы пользователя с текстовыми данными

Исходя из представленного в пункте 1.2.3 описания архитектуры «Подсистемы 3» (для сбора и анализа информации об особенностях работы пользователя с потоками текстовой информации) следует, что программные компоненты, предназначенные для построения, управления и применения пользовательских поведенческих моделей для задачи

раннего обнаружения попыток хищения конфиденциальной информации входят в состав *рабочего места аналитика* и *модуля классификации*, который в свою очередь является частью *агента мониторинга* (см. подпункт 1.4.3.1). При этом построение и управление моделей выполняется только на *рабочем месте аналитика*, а применение — как на *рабочем месте аналитика*, так и на *модуле классификации*.

Далее приводится описание процесса разработки перечисленных программных компонент в рамках их функционирования на *рабочем месте аналитика* и *модуле классификации*.

Программный компонент построения пользовательских поведенческих моделей для задачи раннего обнаружения попыток хищения конфиденциальной информации

Поведенческие модели данного типа основаны на оценки принадлежности документа, с которым работает пользователь, к характерным тематикам для данного пользователя. Поэтому для построения поведенческой модели первоочередной задачей является задание *модельного времени* [14] для анализируемого пользователя. Затем для текстовых документов анализируемого пользователя, входящих в заданное модельное время, выполняется следующий *аналитический конвейер*:

1. *Извлечение текста документа*. Поведенческие модели строятся и применяются к текстовой информации, поэтому необходимо извлекать текст в единой кодировке из наблюдаемых документов различных форматов (с помощью соответствующей функции разработанного СОМ-объекта, см. подпункт 1.5.2.1), которые могут представлять собой текстовые файлы различных форматом (например: doc, rtf, pdf, и т.п.);
2. *Тематическое моделирование*. К полученным текстам документов применяется тематическое моделирование с помощью соответствующей функции разработанного СОМ-объекта (см. подпункт 1.5.2.1).

Таким образом, получаем матрицу «портрета» пользователя, которая и будет служить для представления документов в пространстве характерных тематик для данного пользователя [14]. Полученная поведенческая модель содержит только файлы тематической модели. Данные файлы модели сохраняются в сетевой папке — *хранилище моделей*.

Программный компонент применения пользовательских поведенческих моделей для задачи раннего обнаружения попыток хищения конфиденциальной информации

Для вычисления уровня аномальности наблюдаемой операции пользователя с документом необходимо применить поведенческую модель (соответствующую

пользователю, от имени которого была выполнена операция) к тексту документа, ассоциированного с данной операцией. Поэтому для каждой пары [*наблюдаемая операция / наблюдаемый документ*], являющейся частью сохранённой поведенческой информации пользователя, выполняется следующий *аналитический конвейер*:

1. *Извлечение текста документа*. Аналогично пункту 1 построения поведенческой модели;
2. *Выбор поведенческой модели*. Определение поведенческой модели, находящейся в *хранилище моделей*, по атрибутам наблюдаемой операции;
3. *Отображение текста документа в модельное тематическое пространство*. Использую тематическую модель, входящую в состав поведенческой модели, получение весов тематик для анализируемого текста документа с помощью соответствующей функции разработанного СОМ-объекта (см. подпункт 1.5.2.1);
4. *Вычисление уровня аномальности документа*. На основе вычисленных весов тематик для анализируемого текста документа выполнить расчёт оценки принадлежности текста к характерным тематикам анализируемого пользователя [14].

Полученное значение оценки принадлежности к характерным тематикам анализируемого пользователя показывает уровень аномальности текста наблюдаемого документа и соответствующей операции пользователя. Чем ниже значение принадлежности, тем более аномален факт поведения пользователя с анализируемым документом.

1.5.3.2 Разработка программных компонент, предназначенных для построения, управления и применения пользовательских поведенческих моделей для задачи раннего обнаружения попыток внутренних вторжений на основе информации об особенностях работы пользователя с информационными и вычислительными ресурсами компьютерной системы.

В АРМ аналитика безопасности разработаны компоненты, предназначенные для решения задач сбора, предобработки, хранения и управления информацией об особенностях работы пользователей с информационными и вычислительными ресурсами компьютерной системы. Данная разработка основывается на использовании технологий OLAP и работе с витринами данных. Для этих целей АРМ аналитика безопасности разработан пользовательский интерфейс. Меню АРМ аналитика безопасности отображается в виде дерева, вершиной которого является «Витрины данных». Внутри раздела «Витрины данных» располагается элемент «Все задачи» и витрины данных, создаваемые аналитиком. Элемент «Все задачи» позволяет отображать статус выполнения и управлять всеми задачами, запланированными и выполняющимися в рамках всех витрин данных. Создаваемые аналитиком витрины данных

предназначены для проведения анализа различных срезов данных. По одному срезу на витрину данных. Рекомендуется создавать отдельную витрину данных на подразделение или подмножество АРМ размером от 1 до 100 АРМ.

Каждая витрина данных состоит из следующих элементов:

- Список задач – отображение статуса и управление задачами, запланированными или выполняющимися в рамках витрины данных;
- Фильтры фактов – позволяет задавать фильтры над данными, находящимися в Data Mart;
- Статистические отчеты – позволяет создавать и просматривать статистические отчеты над собранными данными;
- Data Mining отчеты – позволяет создавать и просматривать Data Mining отчеты «Анализ изменения поведения» или «Отклонение от группы» как для пользователей так и для АРМ.
- Data Mining модели – позволяет создавать, обучать, просматривать и применять Data Mining модели на созданных фильтрах.

В рамках каждой витрины данных поддерживается набор задач. К задачам витрины данных относятся: заполнение витрины данных, обучение модели, применение модели, построение Data Mining отчетов. Любые задачи в рамках различных витрин данных могут выполняться параллельно. В рамках одной витрины данных одновременно не могут выполняться задачи заполнения витрины и построения отчетов, так же не могут выполняться задачи применения модели, для еще не построенной модели. АРМ аналитика безопасности позволяет указывать желаемое время запуска для каждой задачи, если задача может быть запущена в это время, она будет запущена, если нет, она будет запущена, как только будет возможно после заданного времени.

Общий вид меню витрины данных АРМ аналитика безопасности представлен на рисунке 50.

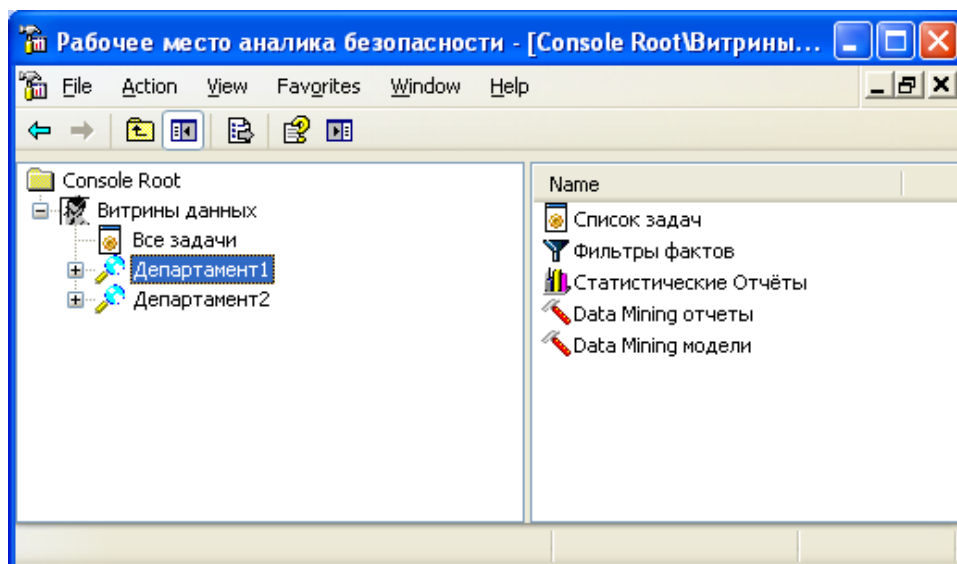


Рисунок 50 — Витрина данных АРМ аналитика безопасности.

В разделе «Все задачи» отображаются журнал выполнения задач, а также все запланированные и активные задачи в рамках всех витрин данных. Задачи отображаются по группам, каждая группа для соответствующей витрины данных. Для каждой задачи указывается название, статус, запланированное время запуска, процент выполнения. Существует возможность приостановить или удалить любую из задач, а так же все задачи. Для того чтобы приостановить выполнение задачи, требуется выделить задачу, щелкнув левой кнопкой мыши в колонке слева от ее имени, и нажать кнопку «Приостановить». Приостановка задач может быть полезна для освобождения вычислительных ресурсов, например, чтобы быстрее отобразить отчет в рамках другой витрины данных. Пример отображения списка задач приведен на рисунке 51.

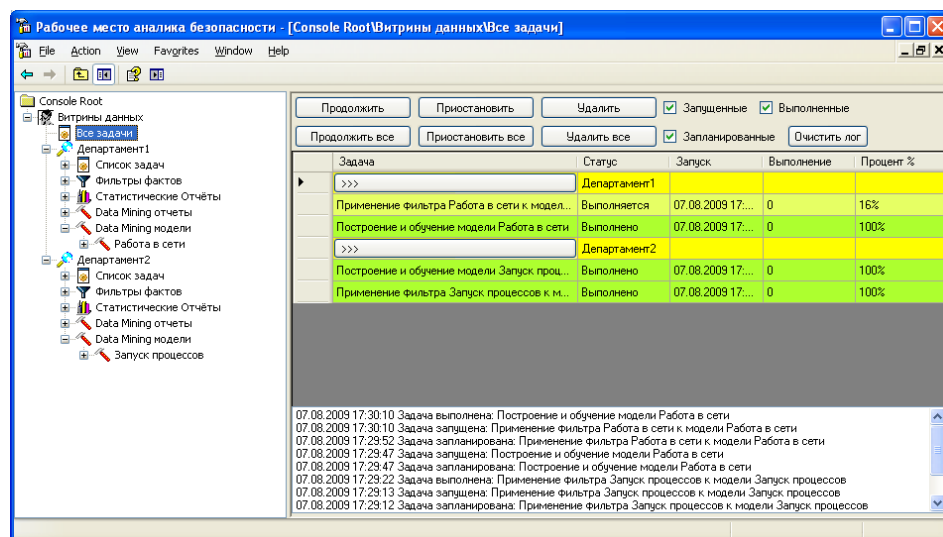


Рисунок 51 — Отображение списка задач.

Для создания витрин данных реализован следующая функциональность в пользовательском интерфейсе. Для создания новой витрины данных нажмите правой кнопкой мыши на «Витрины данных», в появившемся меню выберите «Создать Data Mart». Откроется диалоговое окно «Создание витрины данных», как показано на рисунке 52.

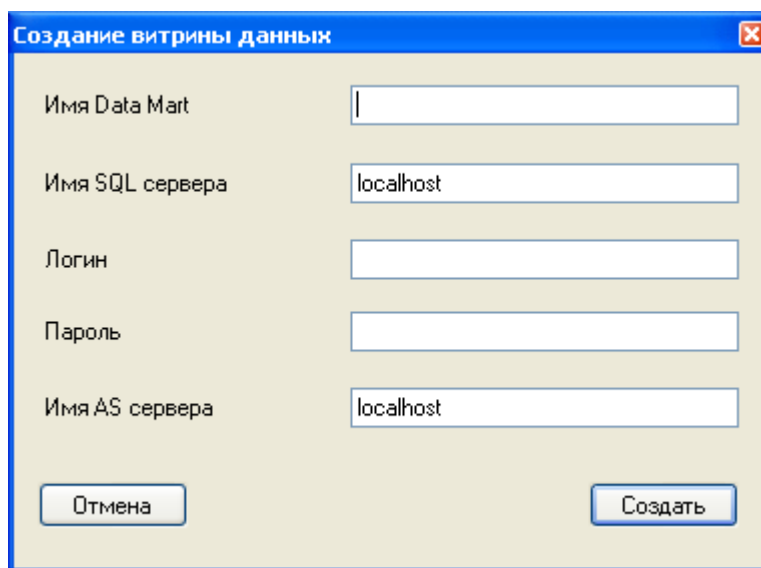


Рисунок 52 — Создание витрины данных.

Требуется указать уникальное имя витрины данных, откорректировать имя SQL сервера и AS (Analysis Services) сервера, логин и пароль (требуется, если SQL сервер устанавливался не на APM аналитика безопасности или не в стандартный «instance»). Нажмите кнопку «Создать».

Каждая витрина данных создается для анализа подмножества АРМ пользователей. Для запроса данных с сервера консолидации требуется единожды указывать список АРМ пользователей для витрины данных. Разработана форма «Настройка Data Mart» (см. рисунок 53).

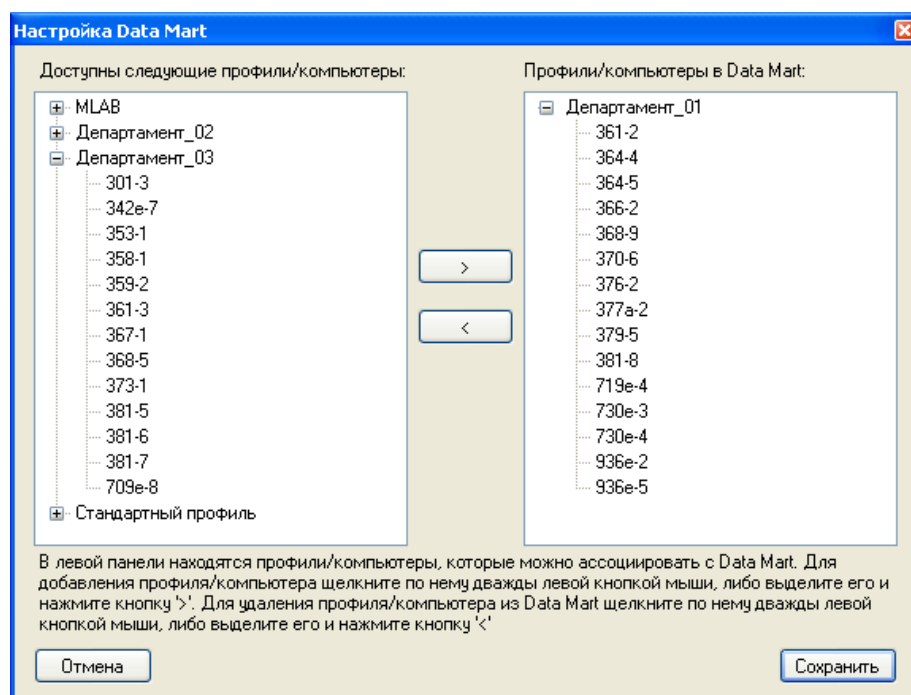


Рисунок 53 — Настройка витрины данных.

В левом окне отображается дерево с именами профилей агентов и именами АРМ, установленных внутри каждого профиля. В правом окне отображаются те АРМ, данные с которых будут запрашиваться с сервера консолидации для заполнения витрины данных. Для добавления нового АРМ выделите его в дереве и нажмите «>» или дважды щелкните по нему левой кнопкой мыши. Для добавления всех АРМ из профиля, аналогично добавьте весь профиль, выделив имя профиля. Если такого АРМ еще нет в правом списке, оно добавится. Для удаления АРМ из списка выделите его в правом окне, затем нажмите «<». То же справедливо и для удаления всего профиля. Возможно использовать витрину данных для анализа АРМ, установленных в рамках одного профиля, в таком случае, для связывания списка АРМ с витриной данных достаточно просто добавить в список требуемый профиль.

Для запроса и/или загрузки данных, собранных сервером консолидации, в витрину данных используется кнопка «Заполнить Data Mart». Разработана форма «Параметры заполнения» (см. рисунок 54).

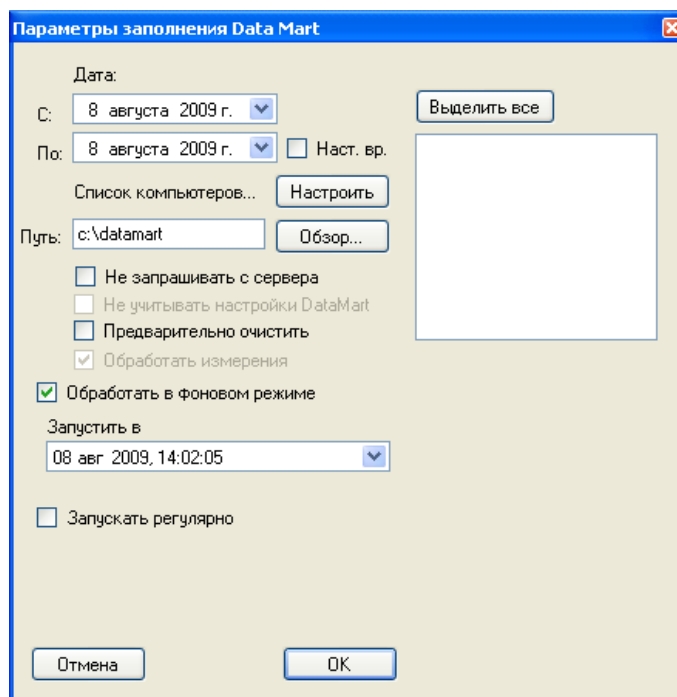


Рисунок 54 — Параметры заполнения витрины данных.

В окне возможно задать:

- «Дата». Временной промежуток, данные за который будут обработаны и помещены в Data Mart.
- «Предварительно очистить». Необходимо ли перед заполнением витрины данных очистить ее от предыдущих данных.
- «Не запрашивать с сервера». Позволяет выбрать запрашивать обновленные данные с сервера или работать только с данными, уже расположенными по адресу «Путь», например, ранее выгруженными с сервера консолидации для архивирования. В случае работы с данными по указанному пути, должна быть разрешена операция записи по указанному пути, т.е., например, нельзя загружать данные напрямую с компакт диска, требуется их перезапись на жесткий диск.
- «Не учитывать настройки DataMart». Доступно только в случае, если используются уже доступные данные (не запрашиваются с сервера). Позволяет игнорировать список АРМ, заданных для витрины данных и выбрать требуемые АРМ из списка доступных АРМ в данных. Кнопка «Выделить все» позволяет сразу отметить все доступные АРМ.

Вид окна «Параметры заполнения Data Mart» для загрузки данных, уже расположенных в папке «c:\datamart», не запрашивая данные с сервера, показан на рисунке 55.

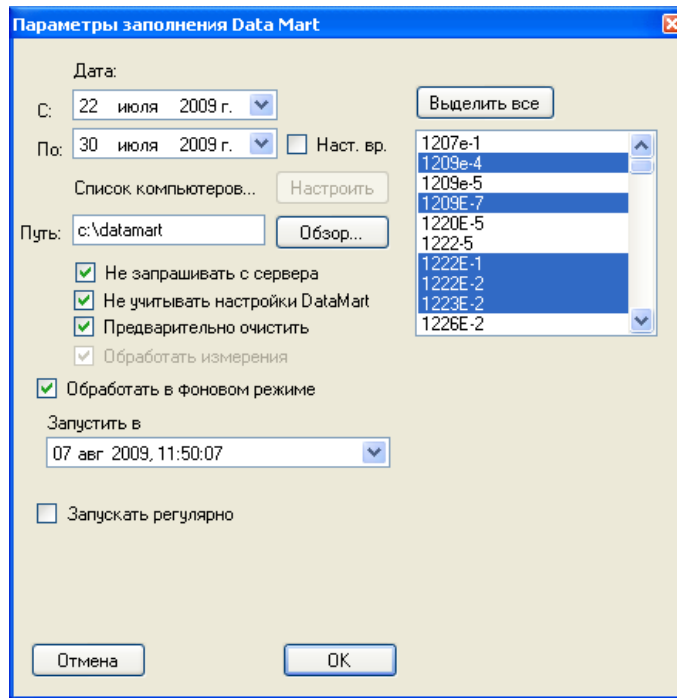


Рисунок 55 — Указанные параметры заполнения витрины данных.

Процедура заполнения Data Mart может потребовать значительного периода времени. Время заполнения определяется количеством выбранных АРМ и выбранным временным диапазоном. В случае предварительного запроса данных (не выбрано «заполнять не запрашивая») сервер консолидации должен быть включен, связь с АРМ, где он установлен, должна присутствовать.

Разработаны программные средства обеспечивающие фильтрацию фактов. Фильтры позволяют организовать выборку требуемых данных для анализа из Data Mart. Фильтрация может производиться как для фактов, так и для атрибутов фактов. Результатом фильтрации является подмножество записей, каждая из которых состоит из подмножества атрибутов. Для создания фильтра реализована форма «Фильтры фактов». Для создания фильтров разработана форма «Создать фильтр» на рисунке 56.

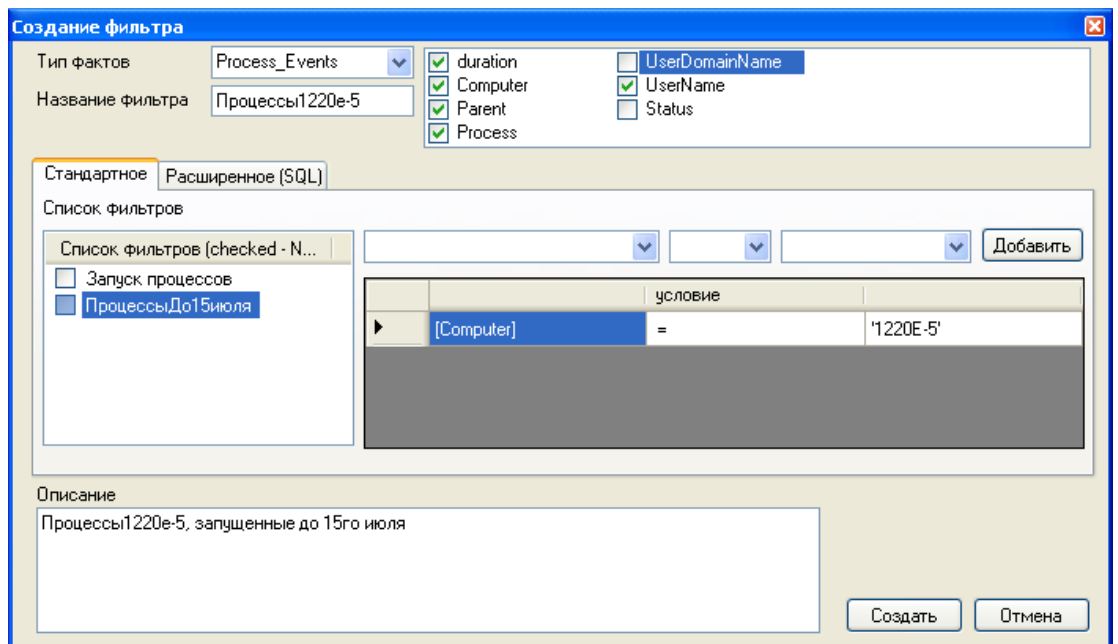


Рисунок 56 — Форма создания фильтра данных.

В графе «Тип фактов» возможно задание следующих типов событий, к которым будет применяться фильтр:

- Process_Events – факты запуска процессов.
- Login_Events – факты входа пользователей в свою учётную запись (логин).
- Traffic_Events – факты входящего/исходящего сетевого трафика.
- Active_Events – факты активность пользователей с клавиатурой и мышью.
- DeviceStat_Events – факты работы с файловой системой.
- Hardware_Events – факты изменения аппаратной части АРМ.
- Software_Events – факты изменения программной части АРМ.

В окне «Список фильтров» отображаются уже существующие фильтры для данного типа фактов, которые могут быть добавлены в качестве дополнительного условия к текущему фильтру. Если фильтр только отмечен (или не отмечен) - он к вашему фильтру не применяется. Если фильтр выделен – то он будет добавляться к критериям нового фильтра по принципу «И». Если выделен и отмечен – то к критериям нового фильтра будет добавляться его отрицание (NOT) по принципу «И». Если выделены (и отмечены) несколько фильтров – по принципу «И» будут добавлены эти фильтры (их отрицание), связанное операцией «ИЛИ». Создаваемый фильтр сохраняется в виде SQL запроса, просмотреть, а так же отредактировать SQL запрос можно, перейдя по вкладку «Расширенное (SQL)». Изменения, сделанные в SQL режиме, не будут отображаться в режиме «Стандартное» и будут сброшены при переходе назад вкладку «Стандартное», о чем система предупредит

оператора информационным сообщением. Изменение фильтров, входящих в состав создаваемого фильтра не будет влиять на созданный фильтр.

Реализована возможность просмотра фильтра, для этих целей разработана форма на рисунке 57.

IDFACT	DURATION	COMPUTER	PARENT	PROCESS	USERNAME
2241	0	1220E-5	SVCHOST.EXE	DFRGNTFS.EXE	SYSTEM
2242	0	1220E-5	SVCHOST.EXE	HELPSVC.EXE	SYSTEM
2243	0	1220E-5	SVCHOST.EXE	WMIPRVSE.EXE	SYSTEM
2244	0	1220E-5	EXPLORER.EXE	BASE32.EXE	KOTOV_DI
2245	0	1220E-5	SVCHOST.EXE	E0SENV*1.EXE	SYSTEM
2246	0	1220E-5	BASE32.EXE	E0SCRYPTOSV...	KOTOV_DI
2247	0	1220E-5	SVCHOST.EXE	WINWORD.EXE	SYSTEM
2248	0	1220E-5	SEARCHINDEX...	SEARCHPRTO...	SYSTEM
2249	0	1220E-5	SEARCHINDEX...	SEARCHFILTER...	SYSTEM
2250	0	1220E-5	WINWORD.EXE	DW20.EXE	KOTOV_DI
2251	0	1220E-5	WINWORD.EXE	OFFDIAG.EXE	KOTOV_DI
2252	0	1220E-5	DW20.EXE	WINWORD.EXE	KOTOV_DI
2253	0	1220E-5	SVCHOST.EXE	WINWORD.EXE	SYSTEM
2254	0	1220E-5	SEARCHINDEX...	SEARCHPRTO...	SYSTEM
2255	0	1220E-5	SEARCHINDEX...	SEARCHFILTER...	SYSTEM
2256	0	1220E-5	SVCHOST.EXE	WINWORD.EXE	SYSTEM

Рисунок 57 — Отображение данных после фильтрации.

Вместе с витриной данных создаются 7 стандартных фильтров – по одному на каждый тип активности пользователей. Они позволяют выбрать все атрибуты и факты данного типа. Стандартные фильтры нельзя удалить.

Модель поведения пользователей состоит из шаблонов поведения и профилей работы пользователей. *Шаблоны поведения* представляют собой ассоциативные правила, построенные над атрибутами фактов активности. Основным требованием к алгоритму построения ассоциативных правил является скорость работы. В работе для построения ассоциативных правил выбран алгоритм Apriori а именно его реализация Microsoft Association Algorithm, так как она позволяет автоматически производить дискретизацию числовых атрибутов с помощью алгоритма expectation maximization(EM), и обладает достаточной производительностью.

Основными параметрами алгоритма являются:

1. MINIMUM_SUPPORT, MAXIMUM_SUPPORT – минимальное и максимальное количество случаев появления группы атрибутов среди всех фактов активности обучающего набора, при котором будет построено правило.
2. MAXIMUM_ITEMSET_SIZE – максимальный размер группы атрибутов и их значений. Группы и, соответственно, правила с количеством атрибутов больше значения указанного параметра не будут строиться.

3. MINIMUM_PROBABILITY – минимальная достоверность построенных правил.

Предложенные методы оценки степени аномальности фактов активности реализованы на языке Си++, что позволило добиться необходимой производительности. Компонента применения построенной модели для оценки аномальности (ARulez) реализована с использованием технологии COM и работает независимо от служб анализа данных Microsoft Analysis Services. Для настройки компоненты ей необходимо передать список возможных атрибутов фактов с указанием типа (числовой или символьный) каждого атрибута. Для числовых атрибутов указать диапазоны дискретизации, которые использовались для построения ассоциативных правил, и передать в текстовом виде набор ассоциативных правил в форматах, стандартного интерфейса Microsoft OLEDB for Data Mining с помощью запроса "Select NODE_DESCRIPTION, NODE_PROBABILITY, NODE_SUPPORT Model.CONTENT where NODE_TYPE = 8" с указанием значений поддержки и достоверности каждого из правил. В данной компоненте реализованы вычислительно эффективные алгоритмы расчета аномальности факта в целом и отдельных атрибутов факта на основе метода, предложенного на предыдущем этапе проекта. Эта же компонента используется как в АРМ аналитика безопасности для построения отчетов по применению построенной модели поиска аномалий и раннего обнаружения внутренних вторжений в отложенном режиме, так и вызывается в агенте сбора событий в модуле онлайн обнаружения внутренних вторжений для обнаружения аномальных событий на этапе их сбора в режиме близком к реальному времени. Но сама модель в случае применения в режиме близком к онлайн должна быть предварительно построена в АРМ аналитика безопасности.

На рисунке приведен пример работы системы с целью поиска аномальных фактов активности, описывающих работу пользователей в сети. На данном примере для описания факта работы в сети используются следующие атрибуты: время передачи, имя процесса, локальный порт, удаленный адрес, удаленный порт, объем отправленных данных, объем полученных данных, тип протокола, имя компьютера, имя пользователя. По этим же атрибутам, кроме времени операции, ищутся аномальные факты активности (рисунок 58).

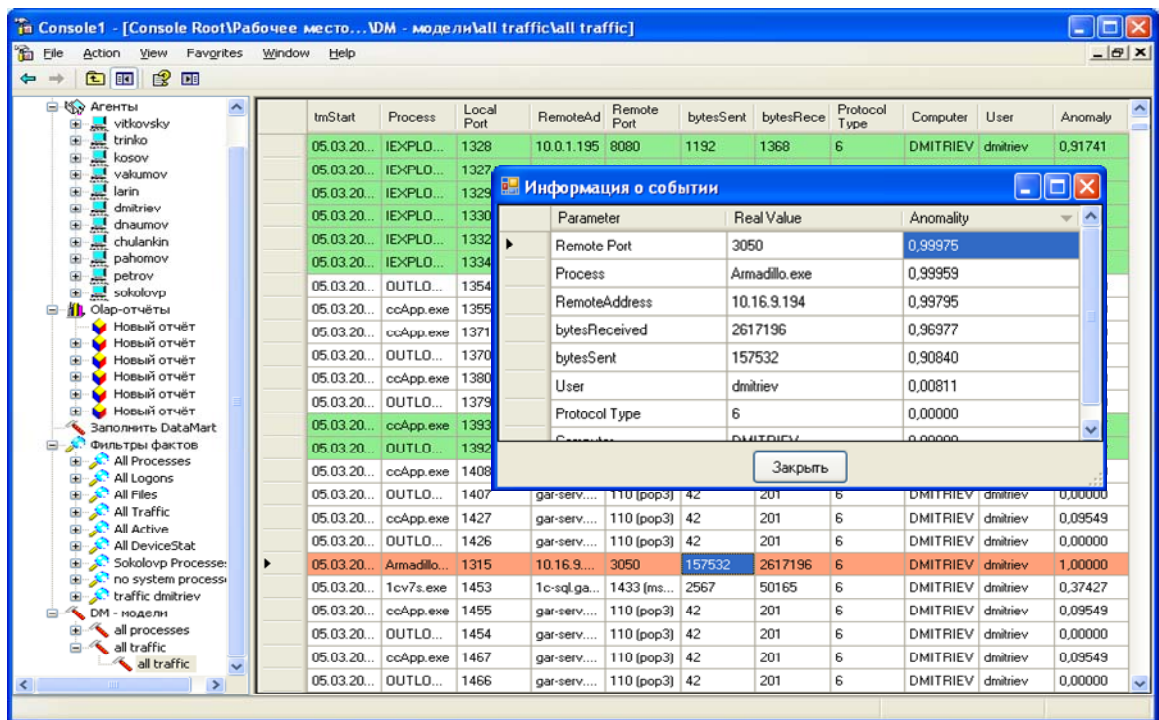


Рисунок 58 — Поиск аномалий в сетевом трафике.

Отчет представляет собой набор фактов активности, для каждого из которых оценена степень аномальности относительно модели. Аномальность приводится в последней колонке, значение «0» соответствует абсолютно нормальным фактам, значение «1» – аномальным. Для выделенного аномального факта дополнительно показаны причины его аномальности. Аномальность значения каждого конкретного атрибута считается при условии фиксированных значений остальных атрибутов. Как видно из рисунка, практически все значения атрибутов факта являются практически полностью аномальными, что может трактоваться экспертом как передача данных редко используемым для передачи процессом на редко используемый порт и адрес, т.е. как абсолютно аномальный факт передачи, похожий на факты обучающей выборки модели лишь типом используемого протокола. Другой пример аномального факта передачи данных приведен на рисунке 59.

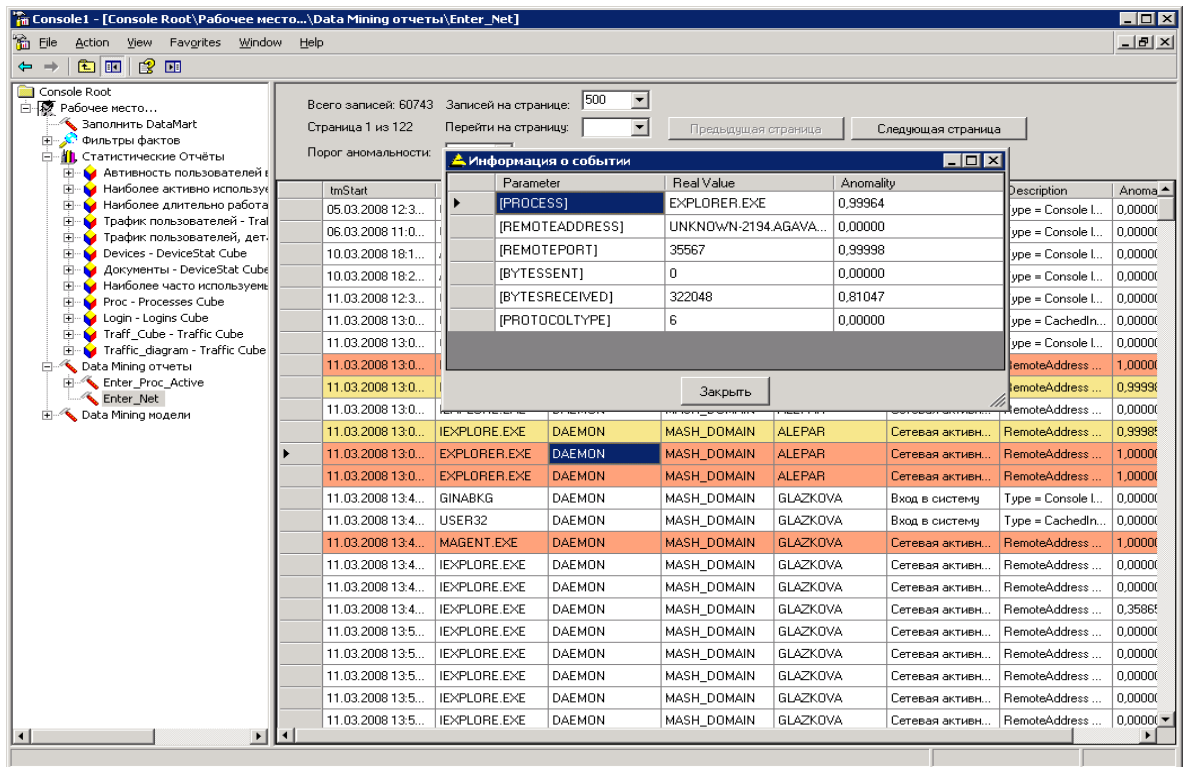


Рисунок 59 — Аномальный факт приема данных.

Аномальность данного факта может интерпретироваться следующим образом: Прием данных с указанного в атрибуте REMOTEADDRESS адреса осуществлялся с аномального порта и аномальным процессом.

Профили работы представимы в виде OLAP-куба. Для применения технологии OLAP в опытной программной реализации системы использовалась реализация из Microsoft Analysis Services 2005. Отображение соответствующих таблиц фактов и справочников в соответствующие кубы осуществляется средствами конкретной реализации технологии OLAP.

Срезы сформированного OLAP-куба могут быть визуализированы аналитику в виде таблиц или диаграмм. В рамках реализованного прототипа для визуализации применяются стандартные компоненты PivotTable и PivotChart из комплекта поставки Microsoft Office. Помимо использования данных компонентов расширены некоторые возможности визуализации OLAP кубов, в частности добавлена возможность накладывания фильтров на любые из измерений по «LIKE» условию. Предложенное расширение функциональности позволяет оперативно находить интересные значения атрибутов фактов, например, строить статистические отчет по работе с конкретным файлом.

Пример визуализации профиля работы в виде сводной таблицы – статистического отчета, описывающего объемы переданных и полученных данных по пользователям, процессам, и адресам приведен на рисунке 60.

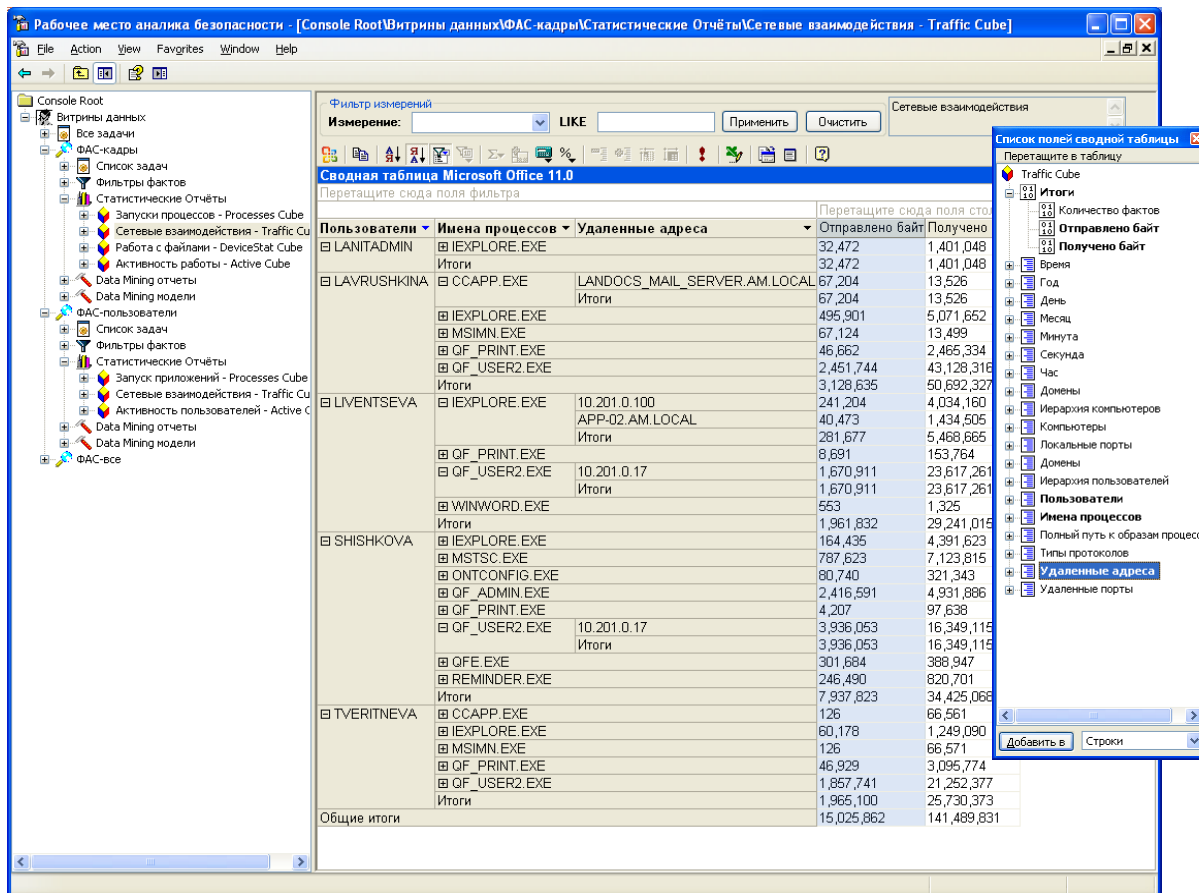


Рисунок 60 — Визуализация профиля работы в виде диаграммы.

Дополнительно приводится список полей сводной таблицы, эксперт может компоновать необходимые отчеты, используя доступные в профиле измерения и меры. При этом перестроение отчетов происходит в оперативном режиме.

Пример визуализации профиля работы в виде сводной диаграммы приведен на рисунке 61.

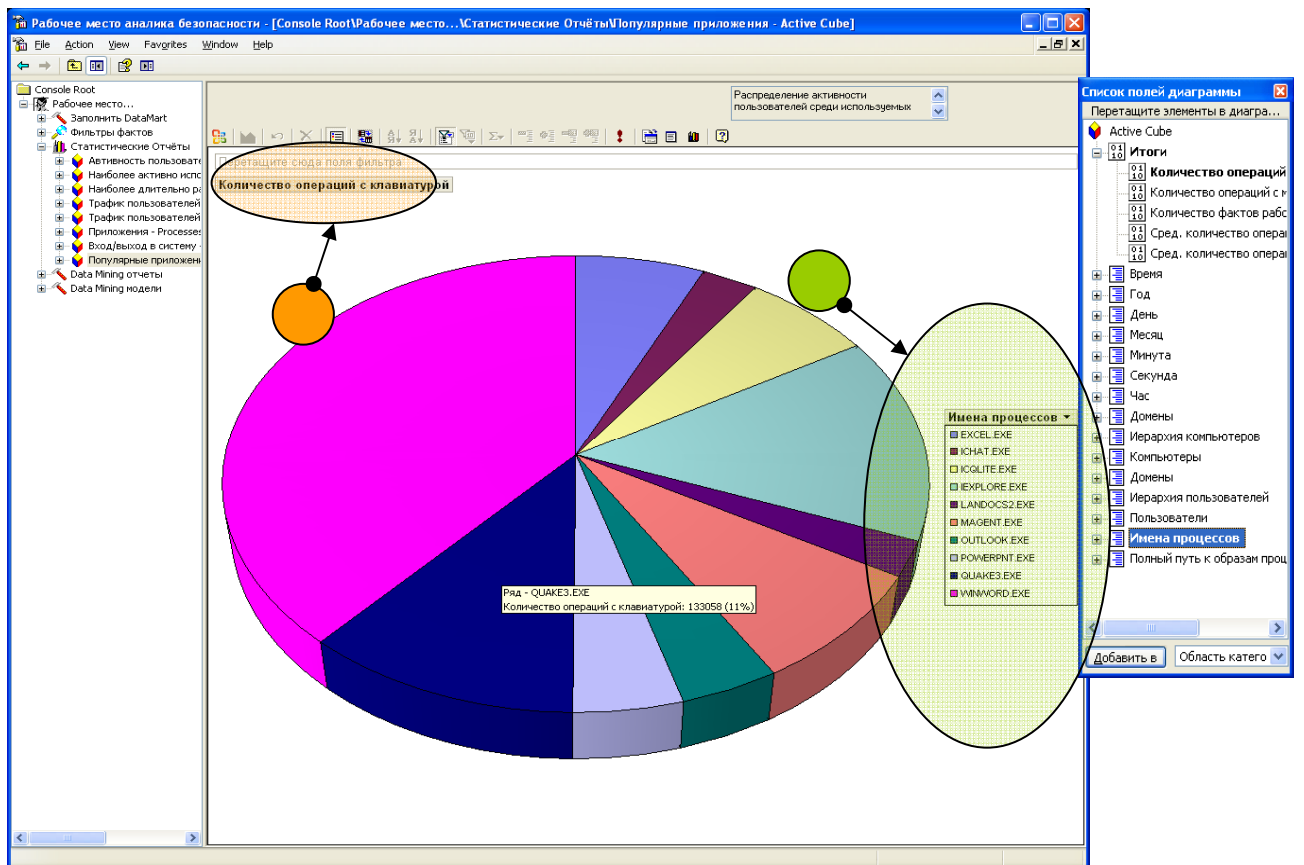


Рисунок 61 — Визуализация профиля работы в виде сводной таблицы.

Аналитик может выбирать вид отчета, а так же доступные в рамках профиля меры и измерения. Изменение диаграмм так же осуществляется в оперативном режиме, т.е. без длительного перестроения отчета.

2 Обеспечение работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по проекту, проведение регламентных работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту

Регламентное обслуживание обеспечивает необходимый набор услуг для поддержания оборудования и программного обеспечения в работоспособном и актуальном состоянии в процессе развития и эксплуатации инфраструктуры, включая обеспечение сохранности программ и данных, относящихся к проводимым работам по проекту.

В рамках работ по обеспечению работоспособности серверов и рабочих станций проводятся следующие процедуры:

- Обновление программного обеспечения (установка пакетов обновления поставляемых производителями программного обеспечения такими как `service-pack` или другие обновления). Периодичность работ — автоматические обновления там, где это возможно, ручная проверка обновлений и систем работоспособности систем автоматических обновлений — 1 раз в месяц.
- Поддержка антивирусной защиты. Периодичность работ по проверке корректного функционирования — 1 раз в месяц. Обновление лицензий антивирусного ПО, периодичность – 1 раз в год.
- Восстановление работоспособности сервера/рабочей станции после сбоя. Включает в себя повторную инсталляцию и настройку программного обеспечения, восстановление данных.
- Профилактические работы (чистка системных блоков, блоков питания, кулеров и их замена при необходимости. Периодичность работ — каждые 3 месяца для рабочих станций, 1 раз в месяц — для серверов.
- Установка дополнительного программного обеспечения. Периодичность — по мере необходимости.

- Проверка состояния жестких дисков (согласно SMART информации) в рабочих станциях и RAID-массивах серверов. При необходимости - замена жестких с дисков с восстановлением данных/консистентности RAID-массивов. Периодичность — 1 раз в месяц.
- Диагностика прочих физических неисправностей (включая периферийные устройства и соответствующий ремонт). Периодичность — по мере необходимости.

В рамках работ по обеспечению сохранности программ и данных, относящихся к проводимым работам по проекту, используются стратегии копирования и архивирования оперативных данных проекта в составе разрабатываемых программных средств/документов и отчетных материалов/экспериментальных данных на базе следующих возможностей:

- Физический уровень — настройка критических данных, хранимых на серверах, в режиме RAID1 ("зеркалирование") — массив из двух дисков, являющихся полными копиями друг друга. Периодичность работ по проверке консистентности RAID1-массива — 1 раз в месяц.
- Архивирование и восстановление данных на уровне файловой системы серверов — реализуется инкрементным копированием по расписанию средствами Windows Server Backup, встроенными в ОС семейства Windows Server. Дополнительно, используется резервное копирование и восстановление среды системы управления проектом на базе SharePoint: используются встроенные средства Microsoft SharePoint для защиты критических объектов среды (веб-приложение, сайт, база данных контента, библиотека документов, настройки, параметры конфигурации). Также, на сервере настроено автоматическое создание резервных копий баз данных системы контроля версий с целью обеспечения сохранности данных в случае программных либо аппаратных сбоев. Периодичность работ по инкрементальному архивированию критических данных проекта — ежедневно в автоматическом режиме (на базе возможностей соответствующего ПО). Периодичность работ по сохранению состояния среды — еженедельно в автоматическом режиме.
- Восстановление работоспособности рабочих станций на уровне файловой системы — реализуется на базе интегрированного в семейство ОС Microsoft Workstation средства System Restore (точки восстановления системы). Периодичность работ по созданию точек восстановления: еженедельно в автоматическом режиме, каждый раз после установки/обновления нового ПО.

Обеспечение бесперебойного электропитания. Для уменьшения вероятности потери/повреждения критических данных все сервера и рабочие станции подключены к источникам бесперебойного питания (ИБП) с возможностью автономной работы не менее 15 минут, необходимых для автоматического корректного выключения системы. Периодичность проверки состояния ИБП (и замены батарей в случае необходимости) — каждые 6 месяцев.

3 Построение программно-аппаратного стенда для проведения экспериментальных исследований и оценки результатов

Построение программно-аппаратного стенда (ПАС) для проведения экспериментальных исследований и оценки результатов проводилось в соответствии с требованиями пункта 4.1.2 ТЗ. Состав и характеристики стенда удовлетворяют спецификациям архитектуры ЭО ПК, разработанной в соответствии с пунктом 3.11.2 ТЗ.

В соответствии с разработанной архитектурой ПАС обеспечивает поддержку следующих подсистем ЭО ПК:

- «Подсистема 1» – для сбора и анализа биометрической информации об особенностях динамики работы пользователя с устройствами ввода-вывода в рамках стандартного человеко-машинного интерфейса (клавиатура, мышь), предназначена для решения задач активной аутентификации и непрерывной фоновой идентификации.
- «Подсистема 2» – для сбора и анализа данных о работе пользователя с информационными и вычислительными ресурсами компьютерной системы (системные, прикладные, пользовательские и специальные журналы), предназначена для решения задач непрерывной фоновой идентификации и раннего обнаружения внутренних вторжений.
- «Подсистема 3» – для сбора и анализа информации об особенностях работы пользователя с потоками текстовой информации, предназначена для решения задач непрерывной фоновой идентификации и обнаружения попыток хищения конфиденциальной информации.

Все компоненты ПАС объединены в локальную сеть с пропускной способностью 1 Гбит/сек и функционируют в домене Windows под управлением доменного контроллера Windows Server 2008R2 x64.

3.1 Программно-аппаратный стенд для проведения исследований и оценки результатов «Подсистемы 1»

Компоненты ПАС данной подсистемы обеспечивают поддержку функционирования следующих логических модулей:

- Модуль активной аутентификации на основе анализа динамики клавиатурного ввода ключевого (не секретного) слова.
- Модуль активной аутентификации на основе анализа динамики работы пользователя с манипулятором мышь при вводе (не секретного) графического символа на основе сгенерированного шаблона.
- Агенты сбора и первичной обработки событий о работе пользователя с устройствами ввода-вывода для непрерывной фоновой идентификации.
- Сервер консолидации – централизованное хранилище первичных событий о работе пользователей с устройствами ввода вывода.
- Рабочее место аналитика – комплекс программных компонент, решающих задачи предобработки данных, построения и валидации одноклассовых и многоклассовых моделей распознавания пользователей, а также интерфейсов для управления данными моделями.
- Модуль идентификации – применяет построенные поведенческие пользовательские модели к векторам признаков полученных от агентов сбора с целью распознавания пользователя в режиме близком к онлайн.

Для обеспечения указанной функциональности был построен программно-аппаратный стенд в составе объединённых в выделенную сеть персональных рабочих станций и вычислительных серверов, включающий в себя:

1. Одна рабочая станция под управлением операционной системы Windows XP SP3 для проведения экспериментов модулей активной аутентификации, реализованных по технологии Microsoft GINA. Базовые аппаратные характеристики:
 - процессор i386 с тактовой частотой 1 ГГц;
 - оперативная память объемом 512 Мбайт;
 - дисковый накопитель HDD, объемом 40 Гбайт;

Дополнительного СПО и ППО не требуется.

2. Одна рабочая станция под управлением операционной системы Windows XP SP3 для работы модуля сбора и первичной обработки, реализованного по технологии Windows hooks для операционных систем Windows с ядром XP i386. Базовые аппаратные характеристики:

- процессор i386 с тактовой частотой 1 ГГц;
- оперативная память объемом 1 Гбайт;
- дисковый накопитель HDD, объемом 80 Гбайт;
- сетевой адаптер с пропускной способностью 100Мбит

Дополнительного СПО и ППО не требуется.

3. Одна рабочая станция под управлением операционной системы Windows 7 (32bit) для работы модуля сбора и первичной обработки, реализованного по технологии Windows hooks для операционных систем с ядром Win7. Базовые аппаратные характеристики:

- процессор с тактовой частотой 2.4 ГГц, 2 ядра;
- оперативная память объемом 4 Гбайт;
- дисковый накопитель HDD, объемом 250 Гбайт;
- сетевой адаптер с пропускной способностью 100Мбит

Дополнительного СПО и ППО не требуется.

4. Одна рабочая станция под управлением операционной системы Windows 8.1 x64 для работы модуля сбора и первичной обработки, реализованного по технологии Windows hooks для операционных систем с ядром Win7 x64. Базовые аппаратные характеристики:

- процессор x64 с тактовой частотой 2.4 ГГц, 2 ядра;
- оперативная память объемом 4 Гбайт;
- дисковый накопитель HDD, объемом 500 Гбайт;
- сетевой адаптер с пропускной способностью 100Мбит

Дополнительного СПО и ППО не требуется.

5. Один вычислительный сервер под управлением операционной системы Microsoft Windows 2012R2 64bit для обеспечения функций централизованного хранилища, рабочего места аналитика и модуля идентификации. Базовые аппаратные характеристики:

- два процессора x64 с тактовой частотой 2.4 ГГц, 4 ядра;
- оперативная память объемом 64 Гбайт;
- два дисковый накопителя HDD, объемом 2Тбайт в режиме RAID1 (система);

- два дисковый накопителя HDD, объемом Тбайт;
- два сетевых адаптера с пропускной способностью 1 Гбит

Характеристики дополнительного ППО:

- интерпретатор Python 3;
- модуль Python pandas 0.16.2;
- модуль Python numpy 1.9.3;
- модуль user-agent 1.0.1.

3.2 Программно-аппаратный стенд для проведения исследований и оценки результатов «Подсистемы 2»

Компоненты ПАС данной подсистемы обеспечивают поддержку функционирования следующих логических модулей:

- Агенты сбора и предобработки поведенческой информации о работе пользователей с вычислительными и информационными ресурсами компьютерной системы.
- Сервер консолидации - специализированное надежное высокопроизводительное хранилище, предназначенное для хранения и управления поведенческой информацией о работе пользователей с вычислительными и информационными ресурсами защищаемой компьютерной системы.
- АРМ аналитика безопасности для решения задачи раннего обнаружения попыток внутреннего вторжения с использованием информации об особенностях работы пользователя с информационными и вычислительными ресурсами защищаемой компьютерной системы.
- АРМ аналитика безопасности для решения задачи фоновой идентификации пользователей с использованием информации об особенностях работы с информационными и вычислительными ресурсами защищаемой компьютерной системы.
- Модуль раннего обнаружения вторжения - работает в режиме близком к онлайн, использует поведенческие модели, подготовленные на АРМ аналитика безопасности для классификации событий в соответствии с поведенческими моделями.

- Консоль управления/рабочее место администратора - используется для централизованного управления компонентами всей подсистемы, включая распространение, установку и конфигурацию агентов подсистемы, а также определение политик работы агентов (включая политики сбора, передачи, идентификации и классификации собираемой поведенческой информации).

Для обеспечения указанной функциональности был построен программно-аппаратный стенд в составе объединённых в выделенную сеть персональных рабочих станций и вычислительных серверов, включающий в себя:

1. Одна рабочая станция под управлением операционной системы Windows XP SP3 для работы агентов сбора и предобработки поведенческой информации о работе пользователей с вычислительными и информационными ресурсами. Базовые аппаратные характеристики:
 - процессор i386 с тактовой частотой 1 ГГц;
 - оперативная память объемом 1 Гбайт;
 - дисковый накопитель HDD, объемом 80 Гбайт;
 - сетевой адаптер с пропускной способностью 100Мбит

Дополнительного СПО и ППО не требуется.

2. Три рабочие станции под управлением операционной системы Windows 7 для работы агентов сбора и предобработки поведенческой информации о работе пользователей с вычислительными и информационными ресурсами. Базовые аппаратные характеристики:
 - процессор Core2 с тактовой частотой 2.4 ГГц, 2 ядра;
 - оперативная память объемом 2 Гбайт;
 - дисковый накопитель HDD, объемом 250 Гбайт;
 - сетевой адаптер с пропускной способностью 100Мбит

Дополнительного СПО и ППО не требуется.

3. Один вычислительный сервер под управлением операционной системы Microsoft Windows 2008R2 64bit для реализации функций централизованного хранилища, рабочего места аналитика, консоли управления и модуля раннего обнаружения вторжений. Базовые аппаратные характеристики:
 - два процессора x64 с тактовой частотой 2 ГГц, 4 ядра;
 - оперативная память объемом 48 Гбайт;
 - два дисковый накопителя HDD, объемом 2Тбайт;

- два сетевых адаптера с пропускной способностью 1 Гбит

Характеристики дополнительного ППО:

- Microsoft Office 2007;
- Microsoft SQL Server 2005;
- Microsoft SQL Server 2005 Analysis Services;
- Microsoft SQL Server 2005 Management Studio;
- интерпретатор Python 2.7.3;
- модуль Python pywin 219;

3.3 Программно-аппаратный стенд для проведения исследований и оценки результатов «Подсистемы 3»

Компоненты ПАС данной подсистемы обеспечивают поддержку функционирования следующих логических модулей:

- Агент мониторинга - программный агент, устанавливаемый на рабочее место пользователя, включающий драйвер файловой системы, модуль сбора и модуль классификации.
- Модуль консолидации поведенческой информации - программный агент, который служит для обеспечения консолидации в едином хранилище поведенческой информации, получаемой от агентов мониторинга, а также предоставляет интерфейсы доступа к данному хранилищу.
- Модуль для языково-независимой предобработки собираемой текстовой информации, включая автоматическое аннотирование больших текстовых документов и группировки собранных текстовых данных на основе машинного обучения «без учителя» с выявлением ключевых тематик и ключевых слов.
- Модуль для языково-независимой предобработки собираемой текстовой информации для рубрикации собранных текстовых данных на основе машинного обучения «с учителем».
- Модуль построения поведенческих моделей - выполняет процедуру построения поведенческой модели, для решения задач фоновой идентификации и раннего обнаружения внутренних вторжений.
- Модуль идентификации служит - для применения поведенческих моделей в отложенном режиме, т.е. поведенческие модели применяются к выборке поведенческих данных, сформированной из поведенческих данных хранилища модуля консолидации.

- Рабочее место аналитика - представляет графический интерфейс, реализующий функциональность для создания и применения поведенческих моделей, настройки отложенного режима (задание расписания и других параметров для автоматического построения и применения моделей типа фоновой идентификации к данным из хранилища модуля консолидации) и настройка режима «близкого к онлайн»

Для обеспечения указанной функциональности был построен программно-аппаратный стенд в составе объединённых в выделенную сеть персональных рабочих станций и вычислительных серверов, включающий в себя:

1. Одна рабочая станция под управлением операционной системы Windows 8 для работы агентов сбора и предобработки поведенческой информации об особенностях работы пользователей с текстовой информацией. Базовые аппаратные характеристики:
 - процессор i386 с тактовой частотой 1 ГГц;
 - оперативная память объемом 1 Гбайт;
 - дисковый накопитель HDD, объемом 80 Гбайт;
 - сетевой адаптер с пропускной способностью 100Мбит
2. Три рабочие станции под управлением операционной системы Windows 7 для работы агентов сбора и предобработки поведенческой информации об особенностях работы пользователей с текстовой информацией. Базовые аппаратные характеристики:
 - процессор Core2 с тактовой частотой 2.4 ГГц, 2 ядра;
 - оперативная память объемом 2 Гбайт;
 - дисковый накопитель HDD, объемом 250 Гбайт;
 - сетевой адаптер с пропускной способностью 100Мбит
3. Один вычислительный сервер под управлением операционной системы Microsoft Windows 2008R2 64bit для реализации функций модуля консолидации, модуля языково-независимой предобработки, модуля построения поведенческих моделей, модуля идентификации и рабочего места аналитика. Базовые аппаратные характеристики:
 - два процессора x64 с тактовой частотой 2 ГГц, 4 ядра;
 - оперативная память объемом 48 Гбайт;
 - два дисковый накопителя HDD, объемом 2Тбайт;
 - два сетевых адаптера с пропускной способностью 1 Гбит.

Характеристики дополнительного прикладного программного обеспечения:

- Агент мониторинга дополнительно требует:
 - Python 2.6 (устанавливается инсталлятором);
 - Библиотека OpenSSL (устанавливается инсталлятором);
- Модуль консолидации и Автоматизированное рабочее место дополнительно требуют:
 - Microsoft SQL Server 2012;
- Модуль для языково-независимой предобработки, модуль построения поведенческих моделей и модуль идентификации дополнительно требуют:
 - Python 2.7;
 - Библиотека Python Win32 Extensions;
 - Библиотека Natural Language Toolkit (NLTK).

ЗАКЛЮЧЕНИЕ

В рамках работ на текущем этапе прикладных научных исследований проведены теоретические исследования 3-ей очереди и получены следующие основные результаты:

1. Разработан ЭО ПК для обеспечения компьютерной безопасности на основе анализа поведенческой информации работы пользователей компьютерных систем, в том числе:
 - 1.1. разработаны пользовательские сценарии работы (use cases) с ЭО ПК;
 - 1.2. спроектирована архитектура ЭО ПК;
 - 1.3. реализованы структуры представления биометрических данных, процедуры их сбора, хранения, управления ими и предварительной обработки;
 - 1.4. разработаны программные компоненты, предназначенные для сбора, предобработки, хранения и управления информацией об особенностях работы пользователей, в том числе со стандартными устройствами ввода-вывода, с информационными и вычислительными ресурсами защищаемой компьютерной системы, с текстовой информацией различных типов;
 - 1.5. разработаны программные компоненты, предназначенные для построения, управления и применения пользовательскими поведенческими моделями для задач активной аутентификации без использования секретной информации (пароля, ключа, секретных вопросов), идентификации пользователей, раннего обнаружения внутренних вторжений и попыток хищения конфиденциальной информации.
 - 1.6. Разработана программная документация на ЭО ПК.
2. Разработаны Программы и методики экспериментальных исследований ЭО ПК.

Проведенные теоретические исследования 3-ой очереди выполнялись на основе осуществлённого выбора направления исследований, предложенных методов и подходов, представленных в отчётах за первый и второй этапы настоящих ПНИ.

Кроме того, за счет финансирования из средств внебюджетных источников:

1. выполнены работы по обеспечению работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по

- проекту, проведены регулярные регламентные работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту;
2. выполнены работы по построению программно-аппаратного стенда для проведения экспериментальных исследований и оценки результатов.

В рамках выполнения этапа прикладного научного исследования получены охраноспособные результаты интеллектуальной деятельности (РИД) - Свидетельство о государственной регистрации программы для ЭВМ №2015661555 от 30.10.2015 "Система двухфакторной аутентификации на основе анализа поведенческой биометрической информации об особенностях работы пользователя с клавиатурой компьютера".

Уровень полученных результатов соответствует мировому. Поставленные на заданный отчетный период задачи выполнены полностью.

Сведения о ходе выполнения настоящих ПНИ размещены в открытом доступе на официальном сайте МГУ имени М.В.Ломоносова (система ИСТИНА — Интеллектуальная система тематического исследования научно-технической информации) по адресу: <http://istina.msu.ru/projects/7964619/>.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

- 1 Машечкин И.В. и др. Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации // Отчет о прикладных научных исследованиях (промежуточный) по теме «Выбор направления исследований. Теоретические исследования (1-ой очереди) поставленных перед ПНИ задач». — М., 2014.
- 2 М. ван Стеен, Таненбаум Э. Распределенные системы. Принципы и парадигмы. - Питер, 2003.
- 3 IRPs Are Different From Fast I/O [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2015. — Режим доступа: [http://msdn.microsoft.com/en-us/library/windows/hardware/ff548576\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff548576(v=vs.85).aspx). — 28.11.2015.
- 4 IRPs Are Different From Fast I/O [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2015. — Режим доступа: [http://msdn.microsoft.com/en-us/library/windows/hardware/ff548576\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff548576(v=vs.85).aspx). — 28.11.2015.
- 5 Filter Manager Concepts [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2015. — Режим доступа: [http://msdn.microsoft.com/en-us/library/windows/hardware/ff541610\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff541610(v=vs.85).aspx). — 28.11.2015.
- 6 Communication Between User Mode and Kernel Mode [Электронный ресурс]. — Электрон. дан. — [Б. м.] : Microsoft, 2015. — Режим доступа: [http://msdn.microsoft.com/en-us/library/windows/hardware/ff539277\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/windows/hardware/ff539277(v=vs.85).aspx). — 28.11.2015.
- 7 Python Programming Language [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: <http://www.python.org>. — 28.11.2015.
- 8 GNU zip compression utility [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: <http://www.gzip.org/>. — 28.11.2015.
- 9 OpenSSL Project [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: <http://www.openssl.org/>. — 28.11.2015.

- 10 FILE_OBJECT structure [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: [https://msdn.microsoft.com/en-us/library/windows/hardware/ff545834\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/hardware/ff545834(v=vs.85).aspx). — 19.12.2015.
- 11 OpenSSL Project [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: <https://www.openssl.org>. — 19.12.2015.
- 12 Outlook Solutions [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: <https://msdn.microsoft.com/en-us/library/bb386094.aspx>. — 19.12.2015.
- 13 Browser Helper Objects: The Browser the Way You Want It [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: [https://msdn.microsoft.com/en-us/library/bb250436\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/bb250436(v=vs.85).aspx). — 19.12.2015.
- 14 Машечкин И.В. и др. Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации // Отчет о прикладных научных исследованиях (промежуточный) по теме «Теоретические исследования (2-ой очереди) поставленных перед ПНИ задач». — М., 2015.
- 15 Natural Language Toolkit (NLTK) [Электронный ресурс]. — Электрон. дан. — [Б. м.] : 2015. — Режим доступа: <http://www.nltk.org>. — 28.05.2015.
- 16 Построение IFilter для поиска SharePoint 2010 и Windows Search с помощью C++, ATL и MFC [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: [https://msdn.microsoft.com/ru-ru/library/office/hh694268\(v=office.14\).aspx#odc_sp14_ta_HowToBuildAnIFilter_Introduction](https://msdn.microsoft.com/ru-ru/library/office/hh694268(v=office.14).aspx#odc_sp14_ta_HowToBuildAnIFilter_Introduction). — 19.12.2015.
- 17 Python for Windows Extensions [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2015. — Режим доступа: <http://starship.python.net/~skippy/win32/>. — 19.12.2015.
- 18 Eigen is a C++ template library for linear algebra [Электронный ресурс]. — Электрон. дан. и прогр. — [Б. м.] : 2014. — Режим доступа: http://eigen.tuxfamily.org/index.php?title=Main_Page. — 19.12.2015.

ПРИЛОЖЕНИЕ А
Отчет о патентных исследованиях

ПРИЛОЖЕНИЕ Б
Описание применения

ПРИЛОЖЕНИЕ В

Программный компонент «Подсистема 1» ЭО ПК. Текст
программы

ПРИЛОЖЕНИЕ Г

Программный компонент «Подсистема 1» ЭО ПК.

Описание программы

ПРИЛОЖЕНИЕ Д

Программный компонент «Подсистема 2» ЭО ПК. Текст
программы

ПРИЛОЖЕНИЕ Ж

Программный компонент «Подсистема 2» ЭО ПК.

Описание программы

ПРИЛОЖЕНИЕ И
Программный компонент «Подсистема 3» ЭО ПК. Текст
программы

ПРИЛОЖЕНИЕ К
Программный компонент «Подсистема 3» ЭО ПК.
Описание программы

ПРИЛОЖЕНИЕ Л

Акт №2 исполнения обязательств по работам на этапе №3

Плана-графика, выполненных за счет внебюджетных
средств

ПРИЛОЖЕНИЕ Л

УТВЕРЖДАЮ



«29» декабря 2015 г.

Декан факультета ВМК МГУ

Академик РАН Е.И.Моисеев

АКТ №2

от 29 декабря 2015 г.

исполнения обязательств по работам на этапе № 3 Плана-графика по Соглашению о предоставлении субсидии № 14.604.21.0056 от 27.06.2014г., выполненных за счет внебюджетных средств, по теме: «Исследование и разработка инновационной технологии построения программных средств обеспечения компьютерной безопасности, основанных на использовании методов машинного обучения и математической статистики для анализа данных поведенческой биометрии пользователей при работе в рамках стандартного человеко-машинного интерфейса, для решения задач активной аутентификации и идентификации пользователей, обнаружения внутренних вторжений и предотвращения попыток хищения конфиденциальной информации»

Настоящий акт составлен в том, что работы по соглашению о предоставлении субсидии № 14.604.21.0056 от 27.06.2014г. предусмотренные планом-графиком исполнения обязательств, а именно:

1. Обеспечение работоспособности рабочих станций, общего системного и специального программного обеспечения для проведения работ по проекту, проведение регламентных работ по обеспечению сохранности данных и программ, относящихся к проводимым работам по проекту;
2. Построение программно-аппаратного стенда для проведения экспериментальных исследований и оценки результатов.

проведены в полном объеме и надлежащем качестве за счет внебюджетных источников на сумму 1 250 000 (Один миллион двести пятьдесят тысяч) рублей.

Научный руководитель проекта

Профессор

Главный бухгалтер факультета ВМК МГУ



И.В.Машечкин



М.В.Сидорова