DOI 10.37882/2223-2966.2023.12-2.03

# РАЗРАБОТКА МОДЕЛИ УДАЛЕННОГО КОНТРОЛЯ И ЗАЩИТЫ ДАННЫХ В ИНФОРМАЦИОННО-АНАЛИТИЧЕСКИХ СИСТЕМАХ

## DEVELOPMENT OF A MODEL FOR REMOTE CONTROL AND DATA PROTECTION IN INFORMATION AND ANALYTICAL SYSTEMS

***E. Amelyutin***
***A. Selin***
***A. Zotov***

*Summary*. The article presents the results of the development of a model for remote control and data protection in information and analytical systems based on the Take-Grant model. The review of modern models of remote control and data protection is carried out. Software has been developed based on the proposed model.

*Keywords*: remote control and data protection model, access control, information security.

***Амелютин Евгений Вячеславович***
*Доцент, МИРЭА — Российский технологический университет*
*amelyutin9@yandex.ru*
***Селин Андрей Александрович***
*Доцент, МИРЭА — Российский технологический университет*
*chuknor@yandex.ru*
***Зотов Артём Олегович***
*МИРЭА — Российский технологический университет*
*lartem890@gmail.com*

*Аннотация*. В статье приведены результаты разработки модели удаленного контроля и защиты данных в информационно-аналитических системах, основанной на модели Take-Grant. Проведен обзор современных моделей удаленного контроля и защиты данных. На основе предложенной модели разработано программное обеспечение.

*Ключевые слова*: модель удаленного контроля и защиты данных, управление доступом, информационная безопасность.

В современном мире удаленный контроль доступа играет ключевую роль в обеспечении безопасной и эффективной работы с информационными системами и ресурсами. Он широко используется в организациях, чтобы обеспечить сотрудникам и партнерам доступ к корпоративным ресурсам, а также в сфере информационной безопасности для обеспечения безопасного удаленного доступа к системам и данным.

Удаленный доступ (Remote Access) — это возможность получить доступ к системам и ресурсам из-за пределов их локальной сети или физических местоположений [1–3].

Удаленный контроль доступа (Remote Access Control) — это процесс управления и регулирования доступом к информационным ресурсам, системам или устройствам извне, часто через сети или интернет. Этот процесс позволяет авторизованным пользователям или устройствам получить доступ к нужным ресурсам, даже если они физически находятся в другом месте [4].

Управляющие системы могут вести журнал доступа и мониторинга, фиксируя действия пользователей или устройств, что помогает в обнаружении несанкционированного доступа или аномального поведения.

Контроль доступа часто включает в себя возможность управлять правами доступа и ресурсами удаленно. Администраторы могут добавлять, изменять или удалять доступ для пользователей и устройств на удаленных ресурсах.

Учитывая потенциальную уязвимость при передаче данных через сети, в удаленном контроле доступа важную роль обеспечивает шифрование (процесс преобразования данных в зашифрованный формат для обеспечения конфиденциальности при передаче через сети), что обеспечивает безопасность данных, передаваемых между удаленными устройствами и ресурсами.

В рамках удаленного контроля доступа важно вести журналирование, то есть запись и хранение информации о действиях пользователей или устройств при доступе к системам [6–7].

Модель Take-Grant — это математическая модель, используемая в компьютерной безопасности и управлении доступом для представления и анализа потока разрешений или привилегий в системе [8]. Модель часто используется для понимания и обоснования политик контроля доступа и безопасности в компьютерных системах.

Take-Grant моделирует систему защиты, состоящую из набора состояний и переходов. Составляется ориентированный граф, который показывает связи между

узлами системы. Узлы представляют субъекты или объекты модели. Направленные ребра между узлами представляют права, которые один узел имеет над связанным узлом.

На рисунке 1 показано графическое представление структуры каталогов.

На рисунке 2 показан, граф, на котором P1 и P2 представляют субъектов (возможных пользователей), а D и F представляют объекты, каталоги и файлы. Если субъект/объект имеет возможность чтения (взятия — take) объекта, он может использовать любую из возможностей, которыми обладает данный объект. Аналогично, если субъект/объект имеет возможности записи (предоставления — grant) объекту, он может предоставить этому объекту любую из своих возможностей.

В модели имеется обозначение [8]:

*O* — множество объектов (файлы, сегменты памяти и т. д.);

*S* — множество субъектов (пользователи, процессы системы);

$R = \{r_1, r_2, r_3, r_4, \ldots, r_n\}\ U\ \{t, g\}$ — множество прав доступа;

*t* [*take*] — право брать «права доступа»;

*g* [*grant*] — право давать «права доступа»;

$G = (S, O, E)$ — конечный, помеченный, ориентированный граф без петель;

*x*, *y* — объекты, элементы множества *O*;

*s* — субъекты, элементы множества *S*;

$E \in O \times O \times R$ — дуги графа (состояние системы описывается её графом);

(*x*, *y*) — набор прав доступа на ребре от узла *x* до узла *y*. Если *r* является элементом (*x*, *y*), то узел *x* имеет право *r* для узла y.

Рассмотрим правила перехода состояний модели Take-Grant:

1) Правило 1 (взятие — take) (рисунок 2).

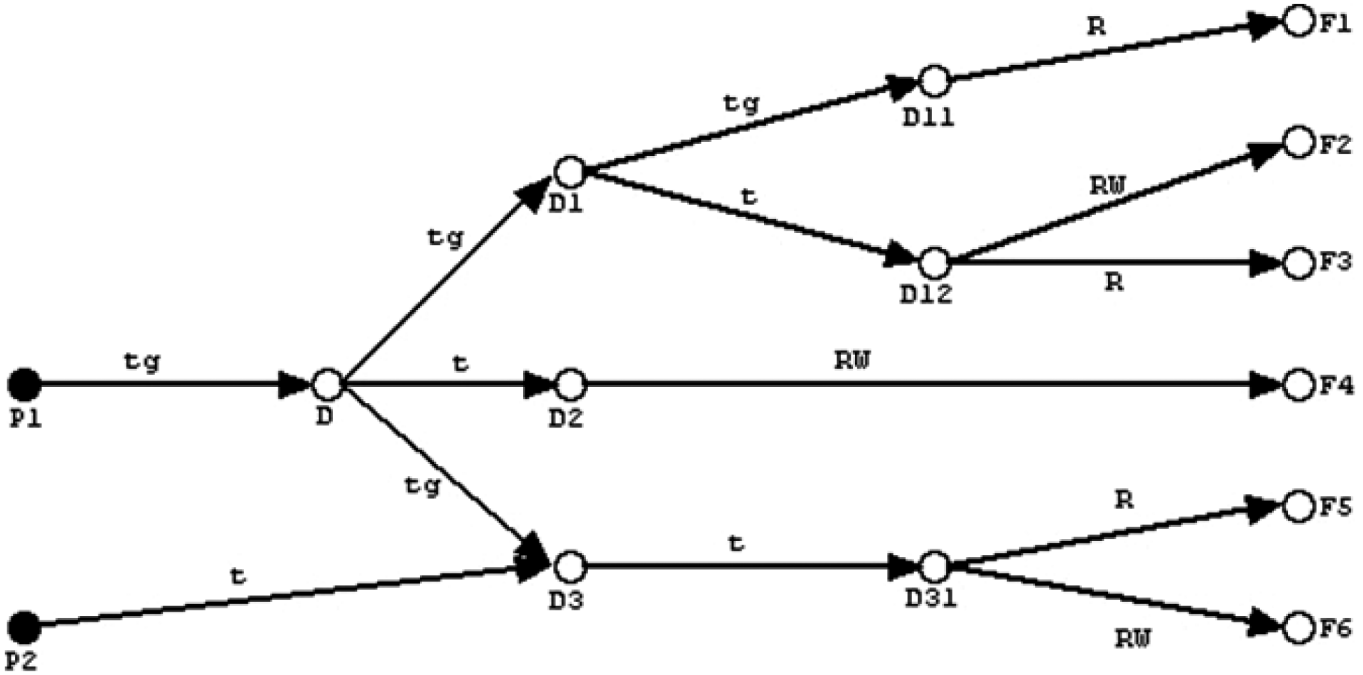


Рис. 1. Пример графического представления структуры каталогов по модели Take-Grant

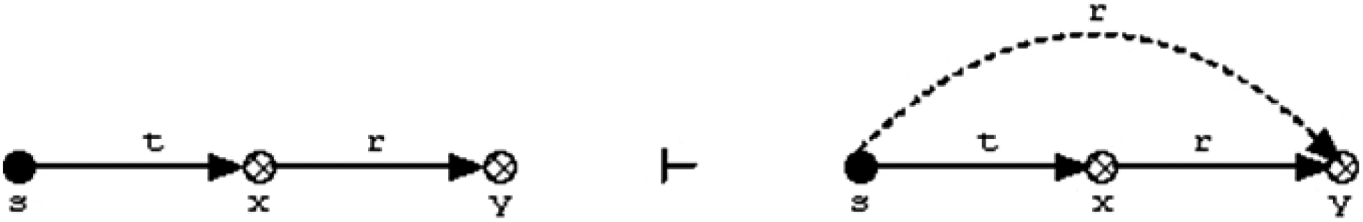


Рис. 2. Правило взятия прав доступа

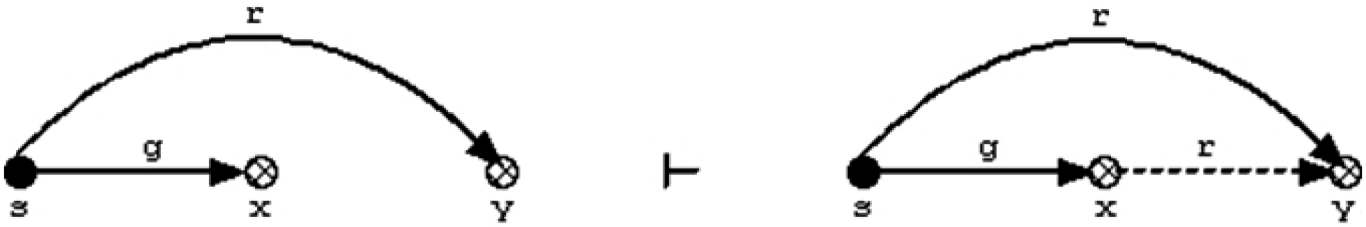


Рис. 3. Правило предоставления прав доступа

Пусть:
— *s* — субъект;
— *x* и *y* — объекты;
— *t* — элемент (*s*, *x*), то есть право взятия объекта *s* на *x*;
— *r* — элемент (*x*, *y*), то есть право объекта *x* на *y*.

Если субъект *s* хочет получить права *r* на объект *y*, то он берет их у объекта *x*:

$$\text{Брать} = take(r, x, y, s), r \in R \quad (1)$$

$$s\ Take\ r\ for\ y\ from\ x \quad (2)$$

2) Правило 2 (предоставление — grant) (рисунок 3).

Пусть:
— *s* — субъект;
— *g* — элемент (*s*, *x*), право предоставления субъекта *s* на объект *x*;
— *r* — элемент (*s*, *y*), право субъекта *s* на объект *y*.

Если объекту *x* нужны права *r* на объект *y*, то они могут быть ему предоставлены субъектом *s*, уже обладающими этими правами:

$$\text{Давать} = grant(r, x, y, s), r \in R \quad (3)$$

$$s\ Grant\ r\ for\ y\ to\ x \quad (4)$$

3) Правило 3 (создать — create) (рисунок 4).


Рис. 4. Правило создания прав доступа

Если *s* — субъект, а *p* — набор прав, то команда добавит новый узел *x* и установит (*s*, *x*) = *p*:

$$s\ Create\ p\ for\ new\ \{subject\ or\ object\}\ x \quad (5)$$

Узел *x* может быть либо субъектом, либо объектом.

4) Правило 4 (удалить — delete) (рисунок 5).

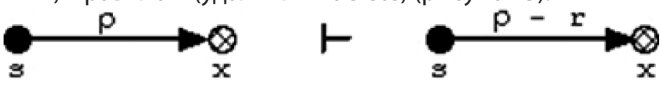


Рис. 5. Правило удаления прав доступа

Если *s* — субъект, а *x* — узел, то команда удалит право *r* из (*s*, *x*):

$$s\ Remove\ r\ for\ x \quad (6)$$

Узел *x* может быть либо субъектом, либо объектом.

Модель Take-Grant является полезным инструментом для упрощения и визуализации отношений контроля доступа, что делает ее ценным инструментом для предварительного анализа безопасности [5]. Однако его ограничения включают его простоту, отсутствие контекста реального мира и неспособность учитывать все аспекты безопасности, что делает его менее подходящим для моделирования очень сложных и специфических сценариев управления доступом.

## Разработка архитектуры модели удаленного контроля и защиты данных

Моделирование системы на языке UML (англ. Unified Modeling Language — унифицированный язык моделирования) позволяет описать проектируемую систему с разных точек зрения. Для этого служит набор различных поведенческих и структурных диаграмм. Поведенческие описывают систему в некоторой динамике действий, а структурные показывают, из каких частей состоит система «Диаграмма прецедентов», которая выделяет действия, выполняемые программой, и связывает их с группами пользователей (рисунок 6).

С системой работает только администратор.

Вначале он должен запустить клиентов на рабочих станциях сотрудников. Для этого он выполняет обязательные (связь include) действия:
— установка клиентов на рабочие станции;
— включение режима прослушивания команд.

Далее для непосредственного контроля и управления правами доступа над файлами, согласно разработанной ранее модели, администратор должен выбрать соответствующую рабочую станцию. Для этого он выполняет такие действия в программе:
— ввод начального и конечного IP адреса для сканирования сети;
— система выполняет автоматическое сканирование сети на наличие активных ПК;

— администратор осуществляет выбор ПК из списка активных.

Далее администратор подключается к нужному ему ПК (удаленно через программу), система автоматически выводит список дисков на устройстве.

После этого можно проводить контроль и управление правами доступа к файлам, который включает такие операции:

— выбор папки;
— загрузка содержимого;
— просмотр прав доступа;
— установка прав доступа.

При этом информационная система информирует администратора о выполняемых операциях в специальном терминале.

На рисунке 7 представлена диаграмма последовательностей работы с системой.

В представленной схеме четко задана последовательность операций, и выделены отдельно административная панель и клиентское приложения.

Диаграммы UML позволяют описать одну и ту же проектируемую систему с разных точек зрения, делая упор на роли и действия (диаграмма прецедентов), а также на модули и их взаимодействие посредством команд и ответов (диаграмма последовательностей).

В общем случае система клиент-серверная, потому что имеется одна административная панель и множество рабочих станций сотрудников. На всех ПК установ-

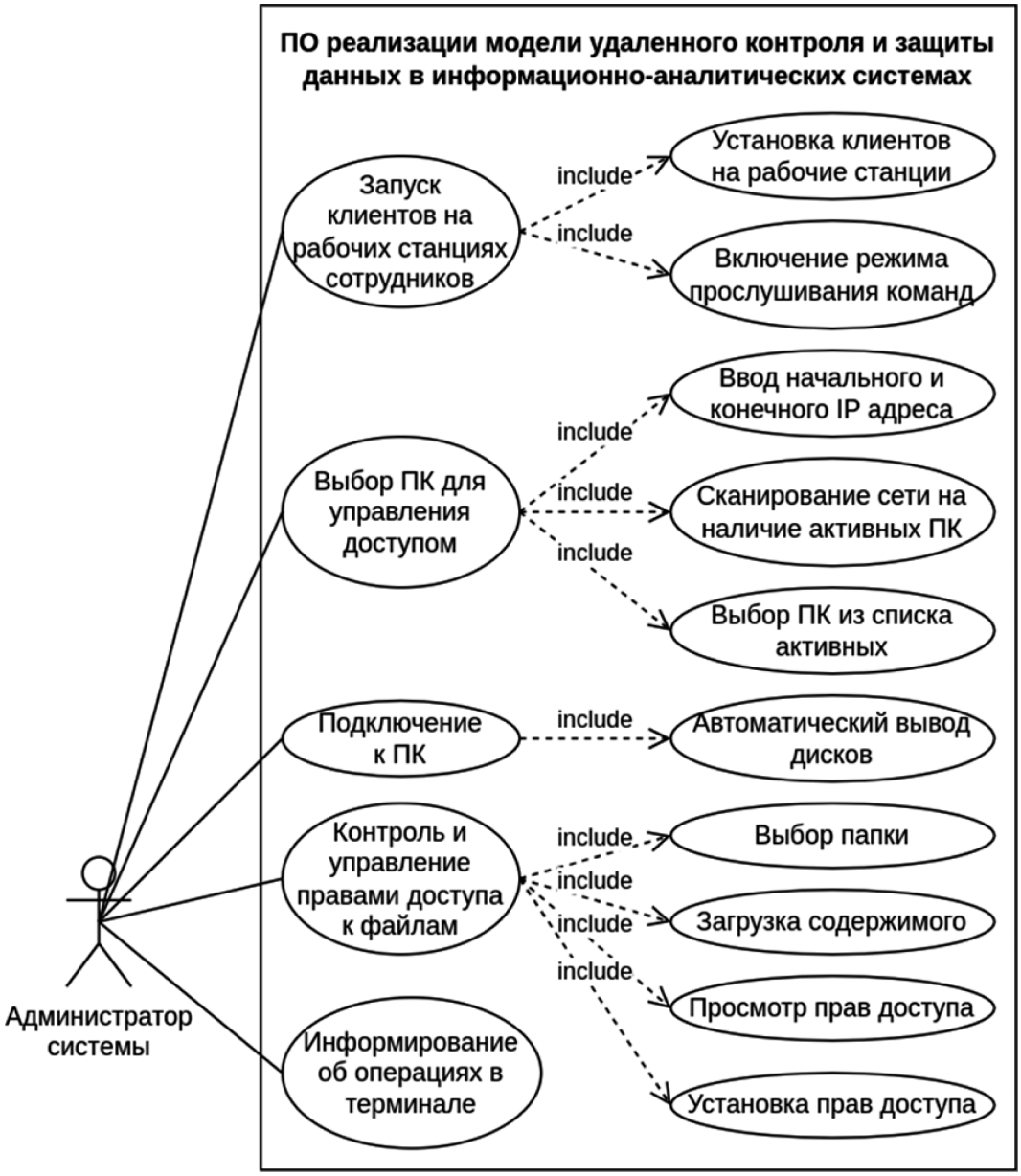


Рис. 6. Диаграмма прецедентов программы

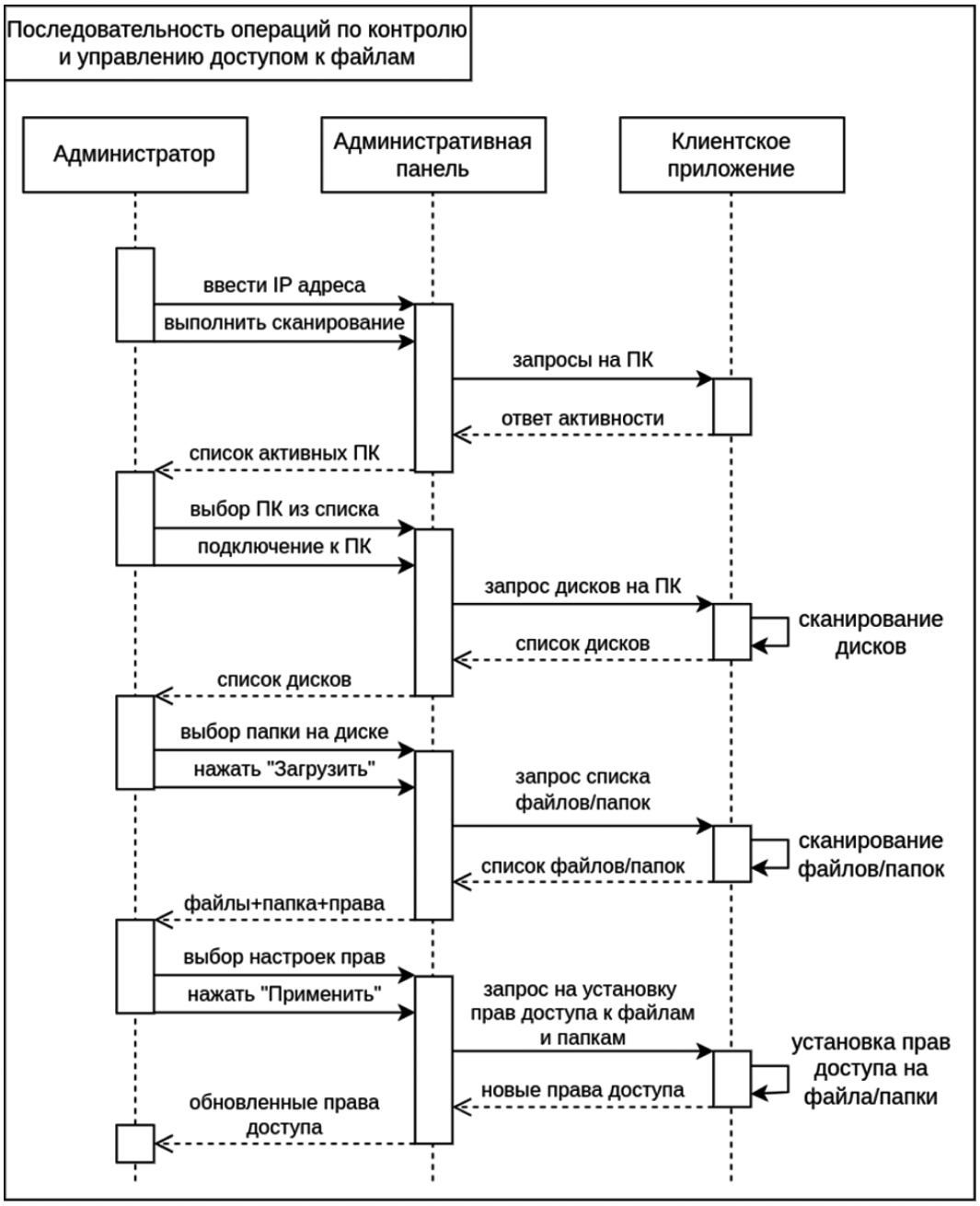


Рис. 7. Структура последовательностей работы с системой

лены ОС Windows, на которых в среде CLR выполняются программы на C#.

Для взаимодействия между административной панелью и клиентскими приложениями используется механизм сокетов.

## Разработка алгоритмической модели удаленного контроля и защиты данных

Рассмотрим разработанный алгоритм работы функции клиента, которая прослушивает порт на подключение сервера, получает команду и выполняет ее (рисунок 8).

Вначале происходит получение IP-адреса и порта сервера, которые передаются в функцию в качестве параметров. Далее идет создание и запуск объекта TcpListener.

После происходит прослушивание канала в ожидании подключения сервера с целью ожидания подключения ПК администратора.

Если сервер подключен, то выполняет создание нового потока и буфера для приема данных и запускается бесконечный цикл, в котором происходит чтение данных со стороны сервера в буфер.

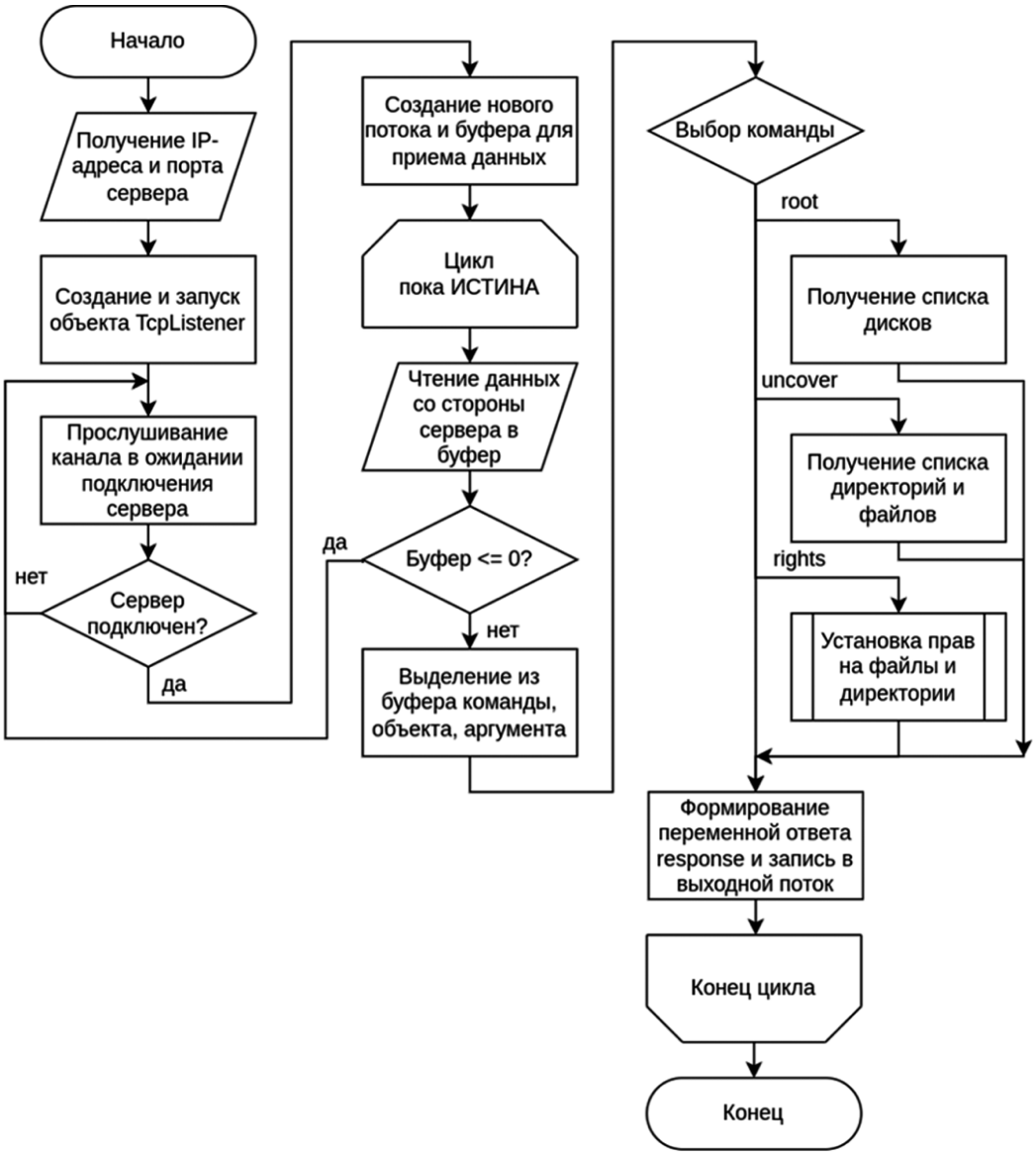


Рис. 8. Блок-схема алгоритма работы клиента

Если буфер пустой, это сигнал отключения сервера, и система переходит к ожиданию нового подключения, в противном случае идет выделение из буфера команды, объекта, аргумента.

Далее в зависимости от команды выполняет одно из трех действий:

— получение списка дисков;
— получение списка директорий и файлов;
— установка прав на файлы и директории.

Результаты должны быть отправлены на сервер, поэтому происходит формирование переменной ответа «response» и запись в выходной поток.

На рисунке 9 представлена блок-схема алгоритма проверки прав доступа на директорию или файл.

Вначале алгоритм получает в качестве параметров путь и права на объект, наличие которых надо проверить.

Далее идет получение атрибутов объекта и установка первоначальных параметров наличия прав доступа в FALSE.

Затем считывается значение списка контроля доступа и заданных объекту правил доступа. Если эти списки пустые, то права доступа не обнаружены у объекта — выход из функции.

Потом запускается цикл по присутствующим у объекта правилам доступа. Если искомое правило найдено, то проверяется, оно типа «Разрешить» или «Запретить».

Итоговые права назначаются как «permission_allow && !permission_deny».

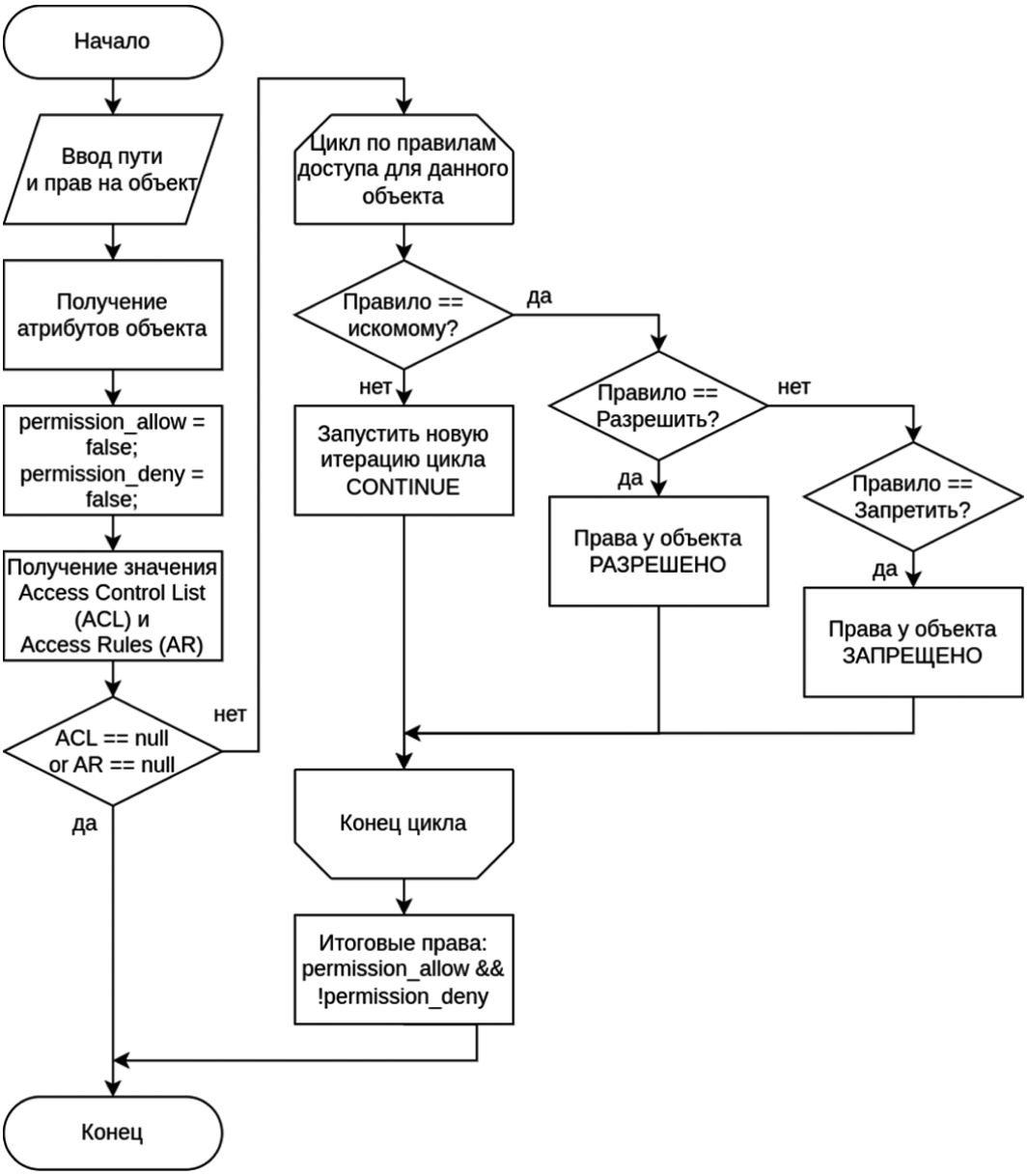


Рис. 9. Блок-схема алгоритма проверки прав доступа на директорию или файл

На рисунке 10 представлена разработанная блок-схема алгоритма функции исполнения команд на клиенте со стороны сервера.

Функция получает IP-адрес клиента, команду, объект и аргумент. Если команда не пустая, то формируется строка запроса для клиента, после чего она кодируется в UTF-8, и ее байтовая последовательность записывается в выходной поток.

После отправки функция ожидает ответа со стороны клиента, инициализирует входной буфер, выполняет чтение данных в него, раскодирование байтовой последовательности в кодировку UTF-8.

Если в данном процессе произошла ошибка, она обрабатывается. В противном случае возвращается прочитанная строка.

## Пример функционирования приложения реализации модели удаленного контроля и защиты данных

Для запуска программы реализации модели удаленного контроля и защиты данных в информационно-аналитических системах требуется на каждой рабочей станции установить клиентские приложения, а затем их запустить. Далее ввести адрес административной панели и порт, а также нажать на кнопку «Начать прослушивание».

Система на стороне рабочей станции перейдет в режим прослушивания подключения административной панели, чтобы была возможность получать от нее и выполнять команды контроля и управления доступом к файлам, находящимся на ПК работника (рисунок 11).

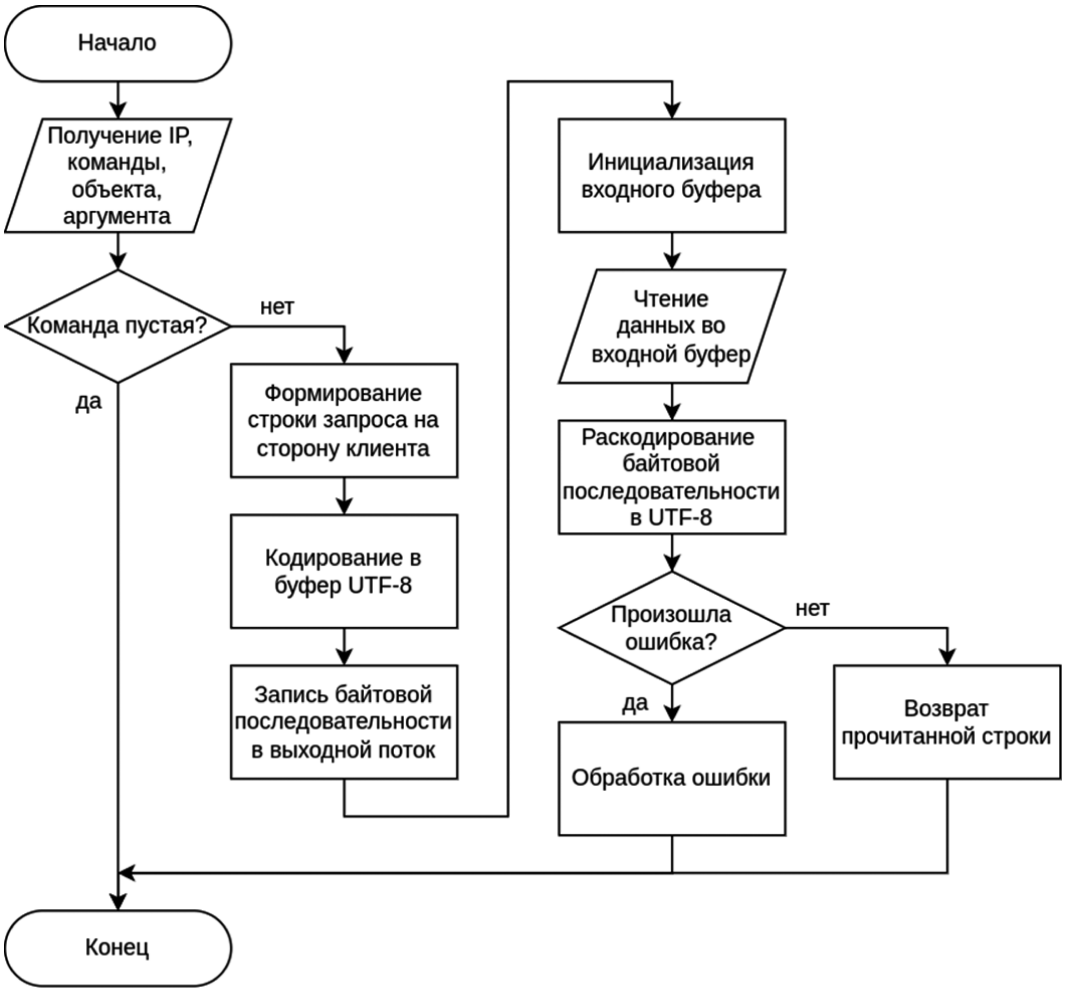


Рис. 10. Блок-схема алгоритма функции исполнения команд на клиенте со стороны сервера

После того, как администратор установил клиентские приложения, в серверной программе надо ввести диапазон адресов для сканирования сети, а затем нажать на кнопку «Выполнить сканирование активных ПК в сети».

Приложение выдаст список активных устройств (рисунок 12). В данном случае это один ПК (тестирование модели велось на одном ноутбуке).

После нажатия на кнопку «Подключиться к ПК» на выделенное в списке устройство будет послана команда вывода дисков, которые отобразятся на административной панели (рисунок 12).

Информирование в терминале. Получение списка дисков удаленного ПК.

Администратор может устанавливать права как на директории, так и на отдельные файлы. Надо выбрать объект в дереве, выставить права на доступ, и нажать на кнопку «Применить». Можно применить право на все вложенные папки и файлы (рисунок 12).

Доступ к файлу запрещается.

На стороне клиента команда будет получена и исполнена (рисунок 13).

После исполнения команды будет перегружен вышестоящий каталог, то есть в данном случае диск Z. Тогда можно будет просмотреть измененные права на файл.

Таким образом, реализованное приложение позволяет администратору удаленно контролировать и управлять доступом к файлам на рабочем станции сотрудника.

## Заключение

Были рассмотрены и проанализированы математические методы для удаленного контроля и защиты данных в информационно-аналитических системах. Разработана алгоритмическая модель удаленного контроля и защиты данных, основанная на модели Take-Grant. Разработано и протестировано приложение удаленного контроля и защиты данных.

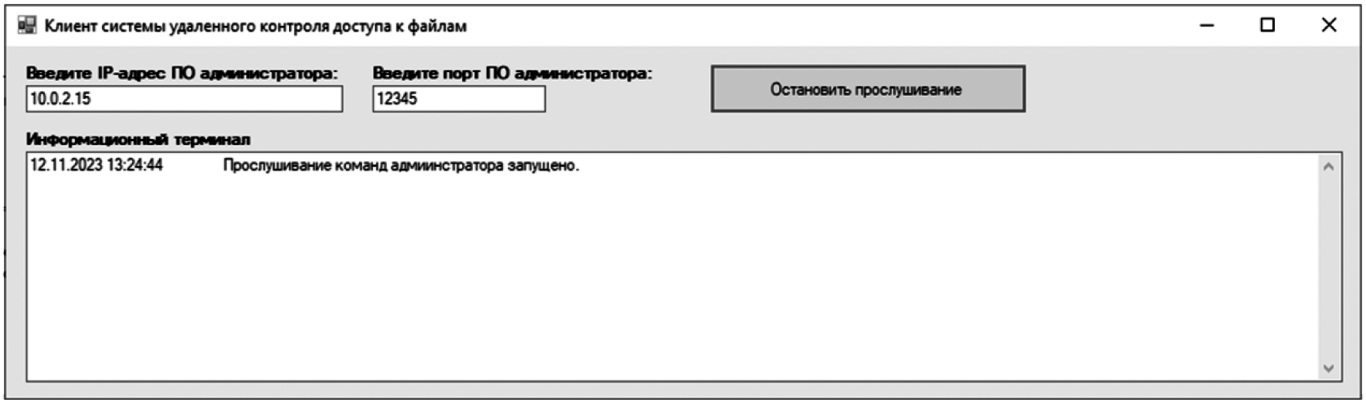


Рис. 11. Приложение на рабочей станции работника. Запуск режима прослушивания

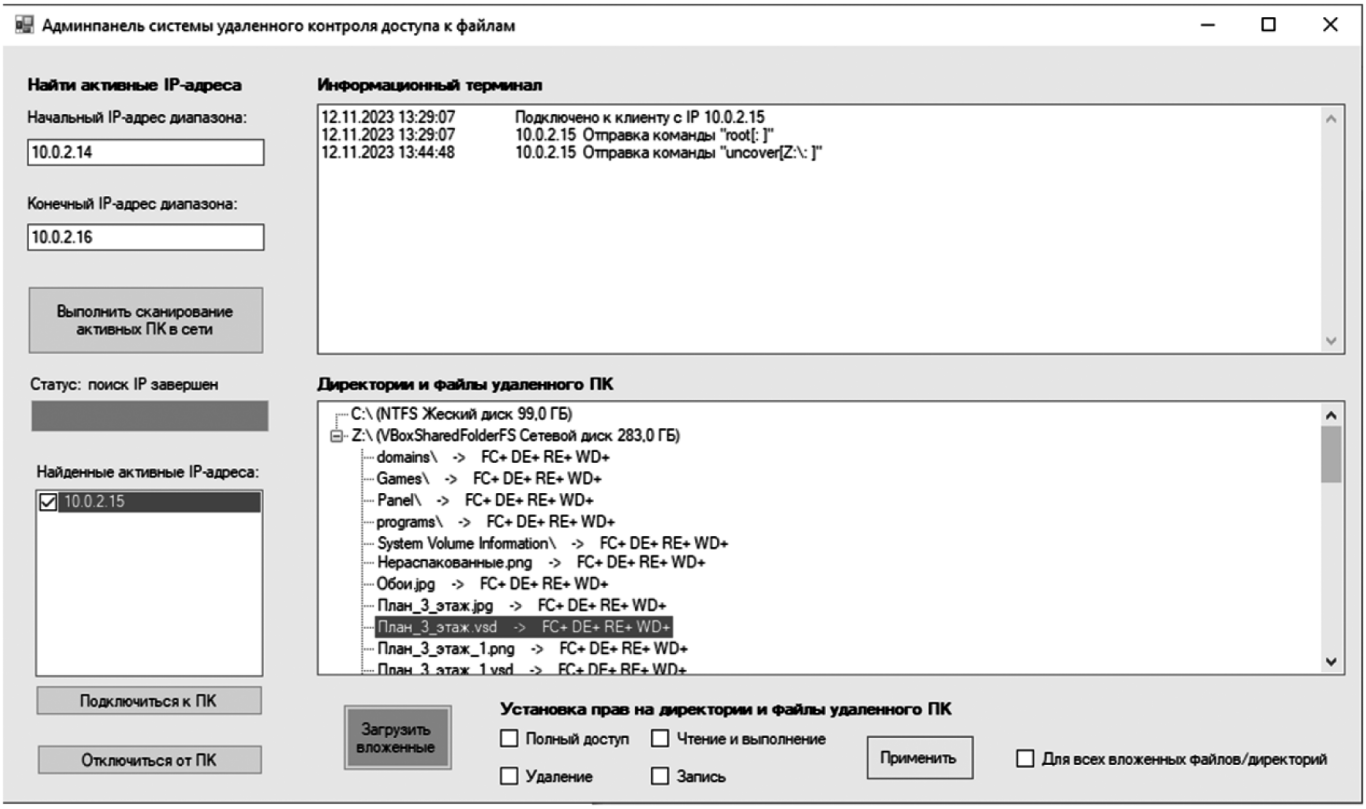


Рис. 12. Применение прав доступа к файлу

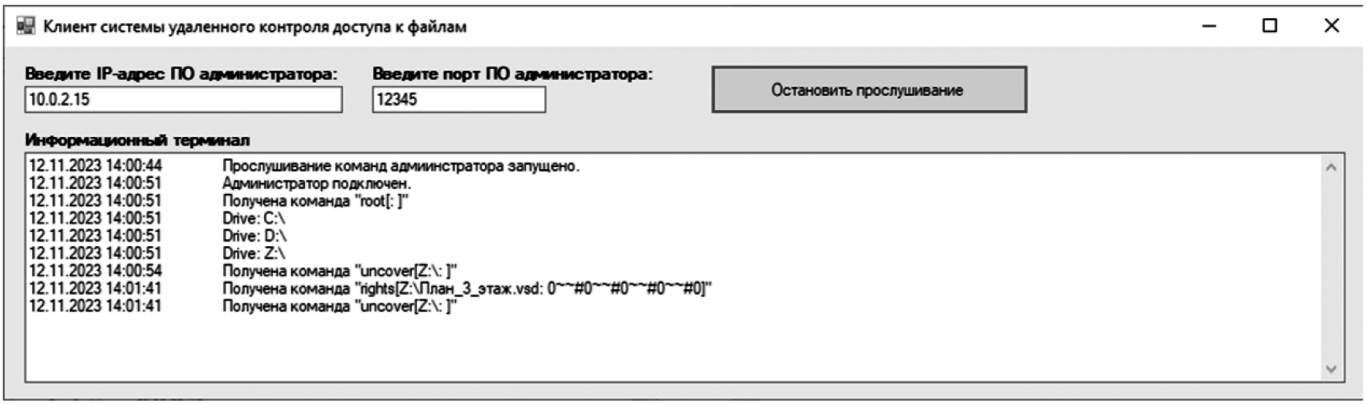


Рис. 13. Исполнение команды на стороне клиента

ЛИТЕРАТУРА

1. Балабанов Д.В. Управление доступом в корпоративных информационных системах. — Эксмо, 2007. — 208 страниц.
2. Деннинг, Дороти Э., «Криптография и безопасность данных», Addison-Wesley, Inc., Ридинг, Массачусетс, стр. 248–257, 1982.
3. Зухов А.С., Шорин А.В. Управление доступом и информационная безопасность в корпоративных информационных системах. — БХВ-Петербург, 2010. — 304 стр.
4. Левин М.А., Павленко Т.В. Управление доступом к информационным ресурсам предприятия. — КНОРУС, 2011. — 176 стр.
5. Леонтьев А.Ю. Администрирование и управление доступом в информационных системах. — Лань, 2016. — 312 стр.
6. Хусейнов Э.Н. Управление доступом к данным в корпоративных информационных системах. — Издательство НИУ ВШЭ, 2007. — 162 стр.
7. Яковлев В.С. Управление доступом к базам данных. — КНОРУС, 2015. — 240 стр.
8. Lipton R.J., Snayder L. A linear time algorithm for deciding subject security // Journal of ACM (Addison-Wesley). N.3. 1977. P.455–464