МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ имени ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT-инжиниринг

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой
PhD, доцент
_____ А.А. Досжанова
«____» _____ 2020 г.

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

На тему: Модели и методы разработки системы контроля и управления доступом

Специальность 6М070300 – «Информационные системы»

Выполнил магистрант группы МИСн-18 _____ Б.Т. Ахметов

Научный руководитель PhD, доцент _____ А.А. Досжанова
«____» _____ 2020 г.

Консультанты по применению
вычислительной техники: ст. преп. _____ Ж.С. Айткулов
«____» _____ 2020 г.

Нормоконтролер: ст. преп. _____ Б.Р. Абсатарова
«____» _____ 2020 г.

Рецензент: к.п.н, доцент _____ К.М. Махаббат
«____» _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ имени ГУМАРБЕКА ДАУКЕЕВА»
Институт систем управления и информационных технологий
Кафедра IT-инжиниринг

Специальность 6М070300 – «Информационные системы»

ЗАДАНИЕ
на выполнение магистерской диссертации

Магистранту Ахметов Батыржан Тлеулесович

Тема проекта: Модели и методы разработки системы контроля и управления доступом

Утверждена приказом по университету № 43 от «18» марта 2020 г.

Срок сдачи законченного проекта «_4_» _____июня_____ 2020 г.

Исходные данные к проекту (требуемые параметры результатов исследования (проектирования) и исходные данные объекта): Техническая документация по C#, операционная система Windows 10, среда разработки Microsoft Visual Studio.

Перечень вопросов, подлежащих разработке в диссертационной работе, или краткое содержание диссертационной работы:

а) обзорно-аналитическая часть;
б) проектирование системы учета доступа;
в) разработка модели;

Основная рекомендуемая литература:

1. Gupta, Brij, Dharma P. Agrawal, and Shingo Yamaguchi, eds. Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, 2016.

2. Liu, X., Zhu, P., Zhang, Y., & Chen, K. (2015). A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. IEEE Transactions on Smart Grid, 6(5), pp. 2435–2443.

3. Jasiul, B., Szpyrka, M., & Śliwa, J. (2014). Detection and modeling of cyber attacks with Petri nets. Entropy, 16(12), pp. 6602–6623.

Консультации по проекту с указанием относящихся к ним разделов работы

Раздел	Консультант	Сроки	Подпись
Программная часть	Айткулов Ж.С.	05.05 – 02.06.20	
Нормоконтролер	Абсатарова Б.Р.	26.05 – 03.06.20	

График
подготовки диссертации

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Аналитическая часть работы	31.01.19	
Проектирование системы учета доступа	02.09.19	
Разработка модели	17.04.20	

Дата выдачи задания «25» __октября_____ 2019 г.

Заведующий кафедрой _____ А.А. Досжанова

Научный руководитель проекта _____ А.А. Досжанова

Задание принял к исполнению магистрант _____ Б.Т. Ахметов

АНДАТПА

Магистрлік диссертация қол жетімділікті басқару мен басқарудың заманауи жүйелерінің тиімділігін арттыру үшін қолданылатын әртүрлі әдістер мен технологияларды зерттеуге арналған. Магистранттың жұмысының нәтижесінде келесі нәтижелер алынды: киберқауіпсіздік ақпараттыққомпьютерлік желіні адаптивті басқарудың тұжырымдамалық моделі сипатталды, қол жеткізуді басқару және басқару жүйесін енгізуге арналған әртүрлі құрылымдық компоненттер ұсынылды; компьютерлік желідегі киберқауіптерді азайту немесе бейтараптандыру үшін пайдаланушы профилін түзету процедураларын автоматтандыру мүмкіндіктері көрсетілген; Қол жеткізуді және басқаруды басқарудың бәсекеге қабілетті жүйесінің прототипі әзірленді және сыналды, оның құрамына бухгалтерлік есеп, басқару және қол жеткізуді басқару элементі болып табылатын таймердің бағдарламалық жасақтамасы, сонымен қатар басқаруға арналған аппараттық-бағдарламалық құрал кіреді. Ұсынылған шешім браузері бар және сымсыз Wi-Fi желісіне қосылу мүмкіндігі бар кез келген құрылғыны қолдана отырып жүзеге асырылуы мүмкін. Әзірленген басқару және қол жеткізуді басқару жүйесінің Қазақстан нарығында баламасы жоқ; Жасалған тәсіл, қолданыстағыдан айырмашылығы, кәсіпорындағы бизнес-процестерді жалпы цифрландырудың қазіргі тенденцияларын ескеретіні көрсетілген. Минималды ресурстарды қолдана отырып, кәсіпорындар өздерінің ақпараттық ресурстарын қорғау аймақтарында қол жеткізуді басқарудың және басқарудың тиімді жүйесін енгізе алады.

АННОТАЦИЯ

Магистерская работа посвящена исследованию различных методов и технологий, используемых для повышения эффективности современных систем управления и контроля доступа. В результате выполнения магистерской работы были получены такие результаты: описана концептуальная модель адаптивного управления киберзащитой информационно-вычислительной сети, а также предложены различные структурные компоненты для реализации системы контроля и управления доступом; показаны возможности автоматизации процедур корректировки профиля пользователя для минимизации или нейтрализации киберугроз в информационно-вычислительной сети; разработан и испытан прототип конкурентоспособной системе управления контроля и доступом, которая включает программную реализацию таймера, являющегося элементом учета, управления и контроля доступа, а также аппаратно-программного устройства для контроля. Предложенное решение может быть реализовано с применением любого устройства, имеющего браузер и возможность подключения к беспроводной сети Wi-Fi. Разработанная система управления контроля и доступом не имеет аналогов на рынке Казахстана; показано, что подложенный подход, в отличие от существующих, учитывает современные тенденции тотальной цифровизации бизнес-процессов на предприятии. Используя минимальные ресурсы, предприятия могут имплементировать эффективную систему контроля и управления доступом в своих зонах защиты информационных ресурсов.

ANNOTATION

The master's thesis is devoted to the study of various methods and technologies used to increase the effectiveness of modern access control and control systems. As a result of the master's work, the following results were obtained: a conceptual model of adaptive control of a cyber-security information-computer network was described, and various structural components for implementing an access control and management system were proposed; the possibilities of automating the procedures for adjusting the user profile to minimize or neutralize cyber threats in the computer network are shown; A prototype of a competitive control system for access and control was developed and tested, which includes software implementation of a timer, which is an element of accounting, management and access control, as well as a hardware-software device for control. The proposed solution can be implemented using any device that has a browser and the ability to connect to a wireless Wi-Fi network. The developed control and access control system has no analogues in the market of Kazakhstan; It is shown that the forged approach, unlike the existing ones, takes into account current trends in the total digitalization of business processes in the enterprise. Using minimal resources, enterprises can implement an effective access control and management system in their information resource protection zones.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В данной магистерской работе описаны модели и алгоритмы создания и использования систем контроля и управления доступом в информационно-вычислительную среду, а также в помещение, в котором, например, находится сервер. Подробно описан процесс моделирования и практическое применение подобных систем контроля у управления доступом или система контроля и управления доступом. Тема магистерской работы представляется чрезвычайно актуальной, поскольку все процессы в нашей жизни стремятся к полной автоматизации, идентификация людей не является исключением.

Цель и задачи исследования. Целью магистерской работы является исследование и разработка системы контроля и управления доступом в информационно-вычислительную среду объекта информатизации, а также в помещение.

Целевая направленность исследования обусловила формулировки и решение в процессе написания работы следующих задач:

- раскрыть сущность и исследовать общие понятия о системе контроля и управления доступом;
- разработать систему контроля и управления доступом с применением любого устройства, имеющего браузер и возможность подключения к беспроводной сети Wi-Fi;
- оценить производительность представленного в работе решения.

Объектом исследования являются процессы обеспечения и применения система контроля и управления доступом в условиях роста количества киберугроз и попыток несанкционированного доступа к ресурсам предприятий и организаций.

Предметом исследования является теоретико-методические основы система контроля и управления доступом в пределах локальных сетей различных объектов информатизации.

Методы исследования. В работе применены общенаучные и специальные методы познания: анализ, синтез, индукция и дедукция, системный метод, метод группировки, детализации и теоретического обобщения - для комплексного исследования систем контроля и управления доступом; структурно-логический анализ - для построения структуры работы; аналогия и оптимизация моделирования - для разработки модели системы контроля и управления доступом.

Информационная база исследования. Теоретическую и методологическую основу исследования составили труды современных казахских и зарубежных ученых в области систем контроля и управления доступом и цифровой безопасности.

Научная новизна исследования заключается в решении важной для отечественных предприятий задачи обеспечения защиты своих помещений путем внедрения цифровых система контроля и управления доступом.

Практическое значение полученных результатов заключается в создании новой системы контроля и управления доступом, которая, в отличие от существующих подходов, позволяет использовать для идентификации пользователя любое устройство, имеющее браузер и возможность подключения к беспроводной сети Wi-Fi.

1 ОБЗОР ТЕХНОЛОГИЙ И СРЕДСТВ КОНТРОЛЯ ДОСТУПА В ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

Информационные технологии стали неотъемлемой частью нашей жизни и существенно влияют на успех бизнеса, его конкурентную способность как внутри страны, так и на мировой арене. По данным Департамента труда США долгосрочный рост экономики, начиная с 1948 года и в течение 50 лет, обеспечивался развитием технологий производства, однако за последние 10 лет эта роль перешла к информационным и телекоммуникационным технологиям, которые способны обеспечить более эффективное решение вопросов организации бизнес-процессов. По оценкам западных экспертов в данный момент судьба взноса информационных технологии в увеличение производительности труда составляет 52%.

В условиях роста конкурентной борьбы, а, следовательно, и значимости информационных и телекоммуникационных технологий, особое значение приобретает задача обеспечения необходимого уровня защищенности компьютерных сетей. Существенное влияние на защищенность сети делает наличие уязвимостей - слабых мест в системах и приложениях, использование которых злоумышленником может привести к реализации угрозы, а также неправильное конфигурирование программных и аппаратных средств, составляющих ее инфраструктуру.

Сегодня принято характеризовать такую конфигурацию, ее соответствие установленным корпоративным стандартам и политикам безопасности, термином «здоровье» (health).

Локальные пользователи и администраторы компьютерных сетей могут оценить и настроить безопасность своих систем с помощью шаблонов безопасности операционных систем Windows, которые впервые появились в пакете обновлений Service Pack 4 (SP4) для Windows. Шаблон безопасности является текстовым файлом, содержащим параметры безопасности, его можно применить единственной командой, позволяет практически моментально настроить безопасность отдельного компьютера или сети (с помощью групповых политик в домене). Недостаток использования шаблонов заключается в том, что они не являются средством, обеспечивающим постоянный контроль безопасности системы. Единственный способ гарантировать, что параметры остаются в силе, - регулярно применять шаблон вручную или создать групповую политику, использует шаблон. Но использовать групповые политики можно только при наличии домена. Другой недостаток шаблонов заключается в том, что они позволяют оценить и настроить только параметры безопасности системы и вовсе не позволяют оценить ее защищенность: не проверяется наличие обновлений и заплаток для операционных систем и установленного программного обеспечения, а также

не проверяется наличие антивируса и актуальность используемых сигнатурных баз.

1.1 Анализ способов несанкционированного доступа к локальной вычислительной сети

Предприятие - это сложная система, состоящая из одной или нескольких производственных, административных зданий и сооружений и обладает собственной инфраструктурой, включая информационные системы, гидро-, энерго- и транспортные коммуникаций. Все они требуют особого подхода в оснащении автоматизированными системами для обеспечения бесперебойной работы.

Одной из главных таких систем является система управления контроля доступа на предприятии.

Установка системы контроля и управления доступом является одной из наиболее важных и необходимых систем в структуре любого предприятия. Функционал системы автоматизируют производственные процессы, контролирует доступ на территорию предприятия и к его сооружениям, идентифицируют и управляют пользователями, предотвращают несанкционированный доступ, позволяют зонировать помещение с разными правилами организации доступа. Все это стало возможным благодаря использованию информационных технологий, которые стремительно развиваются, и тем самым предоставляют новые возможности для усиления контроля и безопасности не только к объектам, а и к информации предприятия. На сегодня существует множество вариантов использования система управления контроля доступа в зависимости от режима секретности предприятия и от его потребностей и возможностей, см. рис. 1.1–1.4.

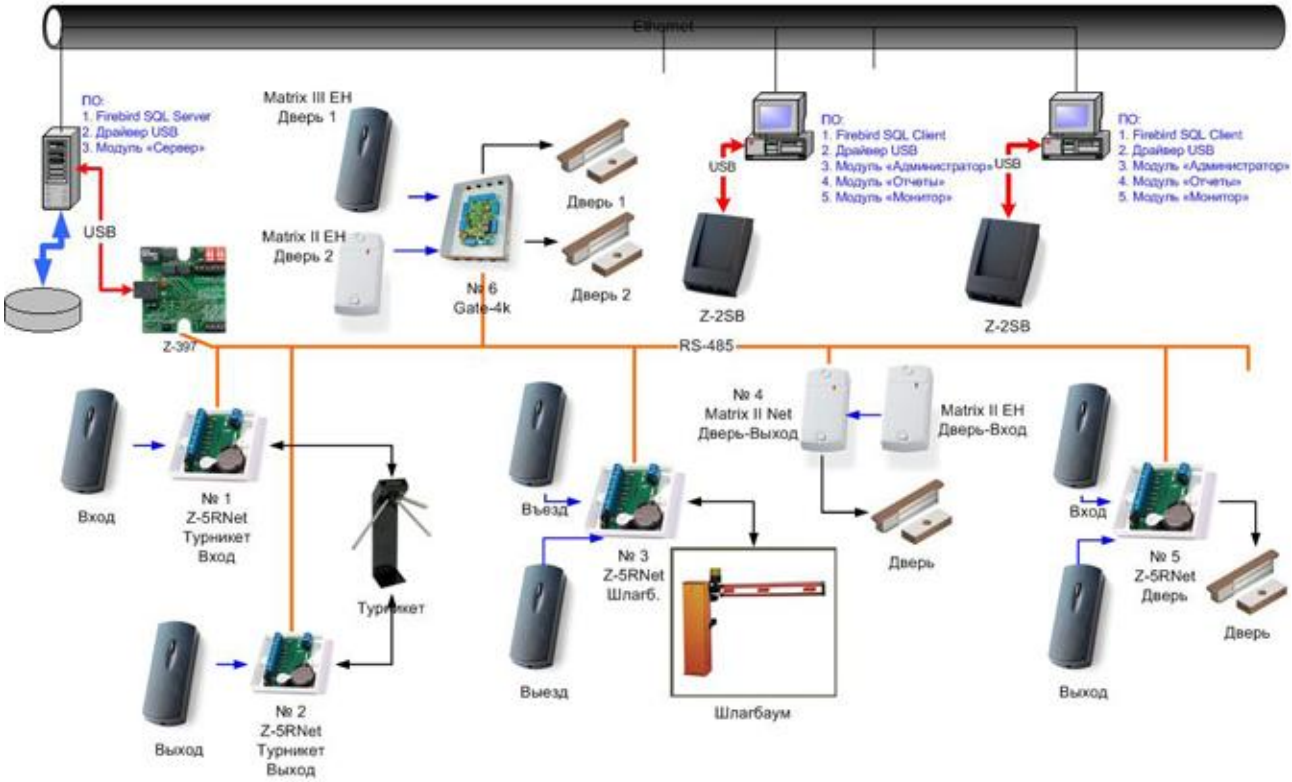


Рисунок 1.1 – Пример организации контроля доступа на предприятии

Рисунок 1.2 – Пример организации контроля доступа на предприятии с применением смартфона и кодового замка

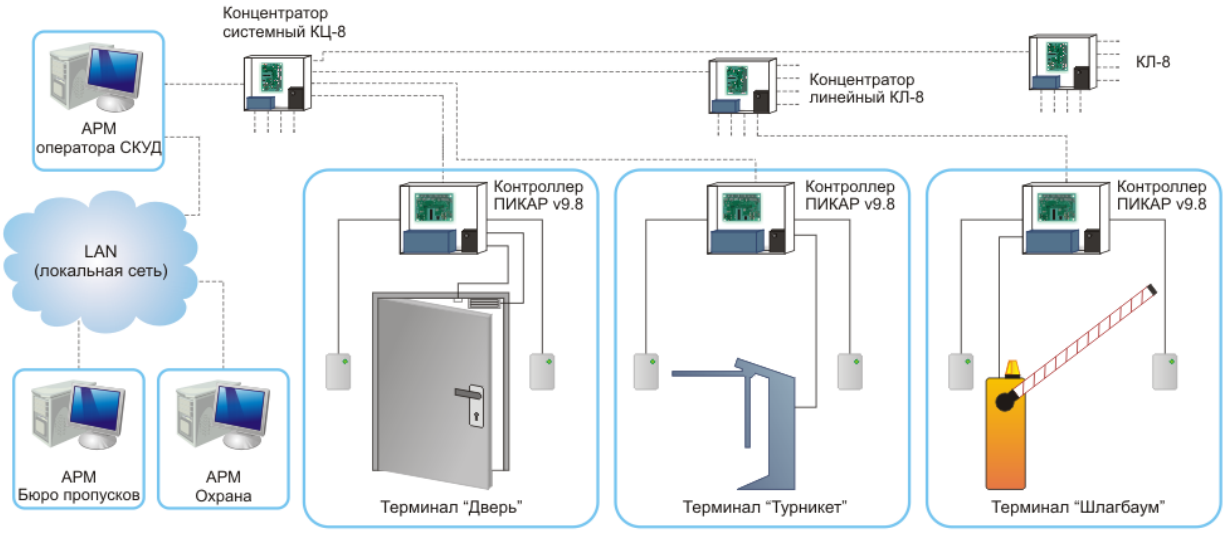


Рисунок 1.3 – Пример организации контроля доступа на предприятии с применением различных информационных технологий зонирования

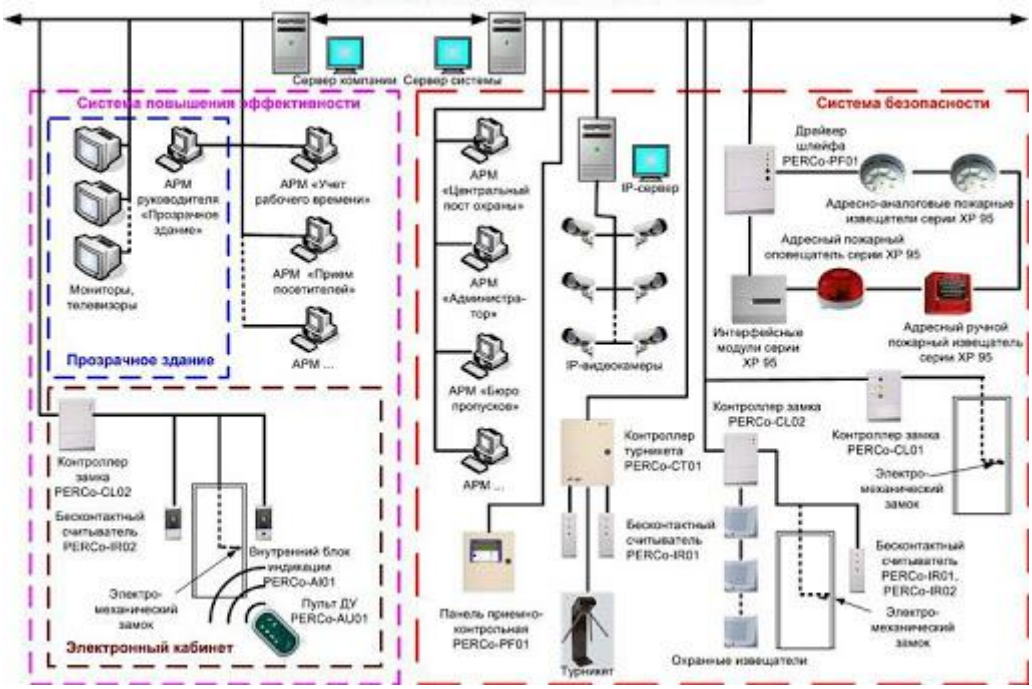


Рисунок 1.4 – Пример организации контроля доступа на предприятии с применением программ контроля действий персонала технологий зонирования

Для оптимального выбора системы управления контроля доступа для каждого предприятия необходимо проанализировать и классифицировать существующие системы управления контроля доступа. Учитывая исследования в этой области, установлено, что в современных системах

управления контроля доступом большое внимание уделено таким новейшим информационным технологиям как идентификация, биометрическая аутенфикация личности и др.

Цель данного раздела магистерской работы - проанализировать существующие современные система управления контроля доступа. На основе проведенного анализа осуществить классификацию современных систем контроля и управления доступом на предприятии. Выявить дальнейшие пути исследования с использованием информационных технологий с целью усиления уровня безопасности на предприятии, а также возможности получения информации разного уровня секретности.

Основная часть системы управления контроля доступом называется - совокупность программно-аппаратных технических средств безопасности, имеющих целью: ограничение и регистрацию входа-выхода объектов (людей, транспорта) на заданной территории через «точки прохода»: двери, ворота, контрольно-пункты. Также, система контроля и управления доступом используют для сбора различной информации о работниках, их работе с информационными системами (например, учет продуктивного рабочего времени, см. рис. 1.5), передвижении по территории предприятия, сроках работы на автоматизированных рабочих местах, времени нахождения в подразделениях и др.

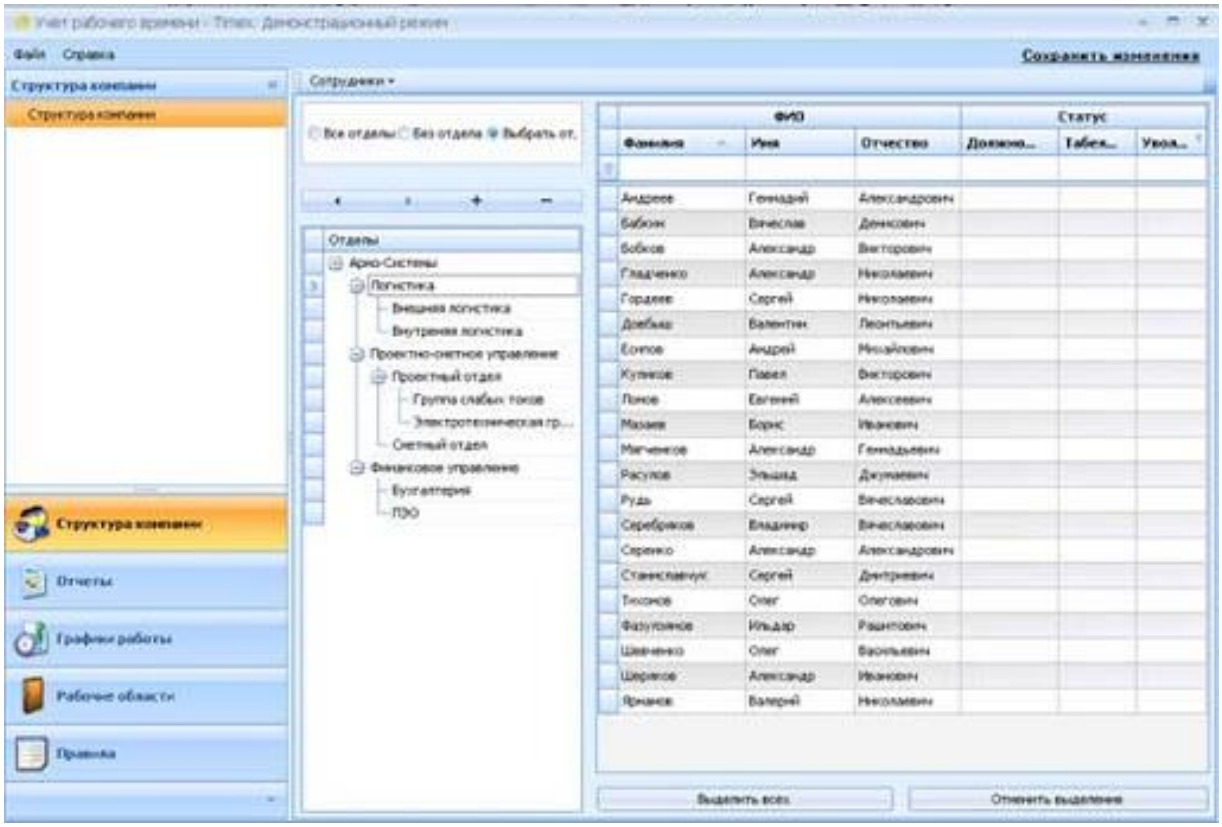


Рисунок 1.5 – Программа учета действий пользователя и рабочего времени, как элемент системы управления контроля доступа

Для оценки информационной безопасности и уязвимостей информационно-вычислительной системы рассмотрим возможные способы несанкционированного доступа к информационным ресурсам.

Под несанкционированный доступ к информации имеют в виду такой доступ, что нарушает правила использования информационных ресурсов компьютерной системы, установленные для ее пользователей.

Вычислительные системы является территориально распределенные компьютерные сети, которые объединяют с помощью каналов связи различные компьютеры и локальные сети. Уязвимость таких систем существенно превышает уязвимость автономных компьютеров. Это связано прежде всего с открытостью и масштабностью компьютерных сетей, используемых на железной дороге. Соответственно существует немало способов атак на эти компьютерные сети.

Все возможные способы несанкционированного доступа к информации в компьютерных системах, защищаемых можно классифицировать по следующим признакам:

1. По принципу несанкционированного доступа: физический несанкционированный доступ, логический несанкционированный доступ.

Физический несанкционированный доступ может быть реализован одним из следующих способов: преодоление рубежей территориальной защиты и доступ к незащищенным информационным ресурсам, хищение документов и носителей информации, визуальный перехват информации, выводимой на экраны мониторов и принтеров, а также подслушивание, перехват электромагнитных излучений.

Логический несанкционированный доступ предполагает логическое преодоление системы защиты ресурсов активной компьютерной сети.

2. Расположением источника несанкционированного доступа: несанкционированный доступ, источник которого расположен в локальной сети, источник которого расположен вне локальной сети.

3. По режиму выполнения несанкционированного доступа: атаки, проводимые при постоянном участии человека; атаки, проводимые специально разработанными программами без непосредственного участия человека.

4. Несанкционированный доступ ориентированный на использование прямого стандартного пути доступа к компьютерным ресурсам или ориентированным на использование скрытого нестандартного пути доступа к компьютерным ресурсам.

5. По непосредственным местом расположения конечного объекта атаки: на информацию, хранящуюся на внешних запоминающих устройствах,

на информацию, передаваемую по линиям связи, атаки на информацию, обрабатываемую в основной памяти компьютера.

6. По непосредственным объектам атаки.

1.2 Требования к подсистемам контроля и управления доступом в локальных вычислительных сетях

Целесообразно выделять следующие группы требований к системам защиты информации:

- общие требования;
- организационные требования;
- конкретные требования к подсистемам защиты;
- требования к техническому и ПО;
- требования к документированию всех инцидентов, связанных с информационной безопасностью;

прочие общие требования.

Прежде всего необходима полная идентификация пользователей, терминалов, программ, а также основных процессов и процедур. Кроме того, следует ограничить доступ к информации, используя совокупность следующих способов:

- иерархическая классификация доступа пользователей к ресурсам;
- классификация информации по важности и месту ее возникновения;
- указания ограничений к информационным объектам. В качестве примера, можно указать, то, что пользователь может осуществлять только чтение файла без права записи в него. Сюда следует отнести и определения программ и процедур, предоставляемых только конкретным пользователям.

Как правило, общие требования характеризуются:

- по способам построения системы защита информации или ее отдельных компонентов (к программному, программно-аппаратного, аппаратного);
- архитектурой информационной системы;
- применением стратегии защиты информационно-вычислительных ресурсов;
- затратами ресурсов на обеспечение систем защиты информации;
- надежностью функционирования систем защиты информации;
- количеством степеней секретности информации, поддерживаемых системами защиты информации;
- обеспечением скорости обмена информацией в информационной системе, в том числе с учетом используемых криптографических преобразований;

- количеством поддерживаемых систем защиты информации уровней полномочий;
- возможностями систем защита информации обслуживать определенное количество пользователей;
- продолжительностью процедуры генерации программной версии систем защита информации;
- длительностью процедуры подготовки систем защиты информации к работе после подачи питания на компоненты информационной системы;
- возможностью систем защиты информации реагировать на попытки несанкционированного доступа, или на «опасные ситуации»;
- наличием и обеспечением автоматизированного рабочего места администратора защиты информации в информационной системе;
- составом используемого программного и лингвистического обеспечения, в его совместимости с другими программными платформами;
- возможностью модификации и т.п.

Основной задачей системы контроля и управления доступом является управление доступом на заданную территорию, включая также ограничения доступа на заданную территорию, идентификация лица, имеющего доступ на заданную территорию, а также учет рабочего продуктивного времени; расчет заработной платы (при интеграции с системами бухгалтерского учета); ведение базы персонала / посетителей; интеграция со всеми системами безопасности:

- с системой видеонаблюдения для совмещения архивов событий систем, передачи системе видеонаблюдения сообщений о необходимости стартовать запись, повернуть камеру для записи последствий зафиксированного подозрительного события;
- с системой охранной сигнализации для ограничения доступа в помещение, или для автоматического снятия и постановки помещений на охрану;
- с системой пожарной сигнализации для получения информации о состоянии пожарных сигнализаций, автоматической разблокировки эвакуационных выходов и закрывания противопожарных дверей в случае пожарной тревоги.

На особо ответственных объектах сеть устройств системы контроля и управления доступом выполняется физически несвязанной с другими информационными сетями.

В точках доступа на объект монтируются считыватели, которые считывают код электронных карт доступа и передают его в контроллеры системы контроля и управления доступом.

Далее контроллер принимает решение на основании внутренней базы данных пользователей или полученной из компьютерной базы данных, и в

соответствии с запрограммированным алгоритмам контроля, управляет исполнительными устройствами система контроля и управления доступом.

Каждый пользователь системы контроля и управления доступом, будь то работник предприятия или посетитель получает электронную пластиковую карту доступа, выдается после регистрации на пункте контроля.

Каждой электронной карте доступа отвечает конкретный пользователь системы со своими правами доступа на контролируемый объект.

Вся необходимая информация о пользователе также заносится на компьютер диспетчера системы в базу данных системы контроля и управления доступом.

При построении сетевых систем контроля и управления доступом используются четыре уровня сетевого взаимодействия:

1. Первый (высший) уровень – компьютерная сеть типа клиент / сервер на основе сети ETHERNET, с протоколом обмена TCP/IP. Также применяют сетевые операционные системы Windows или Unix. Этот уровень обеспечивает связь между сервером и рабочими компьютерами подсистем.

2. Второй уровень - связь между контроллерами и компьютерами подсистеми предприятия или организации. На этом уровне используется интерфейс RS 232.

3. Третий уровень - связь между контроллерами и считывающими устройствами. Здесь применяется интерфейс RS 485 или ставших уже стандартом, интерфейсы считывателей на магнитных картах.

4. Четвертый уровень - уровень извещателей пожарной сигнализации и цепей управления. Здесь применяют сбалансированные и несбалансированные радиальные и адресные шлейфы, релейные выходные цепи управления.

Кроме того, используют нестандартные специализированные интерфейсы и протоколы обмена информацией.

Системы автоматизированного контроля доступа можно разделить на два типа: автономные и сетевые системы.

К автономным системам контроля и управления доступом относятся системы, расположенных в одном помещении. Их главная функция - ограничение доступа к объектам, контролируемым.

Такие система контроля и управления доступом используют для небольших предприятий с несколькими пунктами пропуска на объект.

К таким системам контроля и управления доступом можно подключить различные устройства:

- шлагбаумы;
- турникеты;
- электромагнитные замки и считыватели RFID.

RFID (Radio Frequency Identification) – метод, автоматически идентифицирует объекты с помощью записи данных, хранящихся в так

называемых RFID-метках и считывает эти данные с помощью радиооборудования.

К сетевых системам контроля и управления доступом относят системы, позволяющие контролировать присутствие сотрудников на рабочих местах, получать разнообразные отчеты о присутствии за различные отрезки времени по каждому работнику и по подразделению в целом, ограничивать доступ к контролируемым объектов по времени суток, распределять зоны доступа по кабинетам, этажах, корпусах и т.п., разработать программное обеспечение для составления личной карточки для каждого работника с фотографией, личными данным, маркой автомобиля и др.

Контроллеры, работающие в автономном режиме, должны обеспечивать прием информации от считывателей, обработку информации и выработку сигналов управления для устройств исполнительных. Контроллеры, работающие в сетевом режиме, должны обеспечивать:

- обмен информацией по линии связи между контроллерами и управляющим компьютером или ведущим контроллером;
- сохранение памяти, установок, кодов идентификаторов в случае обрыва связи с управляющим компьютером, отключение питания и при переходе на резервное питание;
- контроль линий связи между отдельными контроллерами и между контроллерами и управляющим компьютером.

Для гарантированной работы систем контроля и управления доступом расстояние между отдельными компонентами не должна превышать величин, указанных в паспортах.

Протоколы обмена информацией и интерфейсы должны быть стандартных типов.

Виды и параметры интерфейсов должны быть установлены в паспортах и / или других нормативных документах на конкретные средства.

Рекомендуемые типы интерфейсов:

- между контроллерами - RS 485;
- между контроллерами и управляющим компьютером - RS 232.

Программное обеспечение должно обеспечивать:

- инициализацию идентификаторов (занесение кодов идентификаторов в память системы);
- задания характеристик контролируемых точек;
- установку временных интервалов доступа (окон времени);
- установку уровней доступа пользователей;
- протоколирование текущих событий;
- ведение баз данных;

- сохранение данных и установок при авариях и сбоях в системе между контроллерами и компьютерами подсистем. На этом уровне используют интерфейс RS 232.

Обычно администратор (или лицо ответственное за пропускной режим) имеет мастер-карту (мини-компьютер), с помощью которой он может вносить в список системы коды идентификаторов работников и посетителей или исключать их из списка, а также считывать информацию из буфера системы. используют на объектах, где требуется ограничение доступа только посторонних лиц.

К системам контроля и управления доступом 2-го класса относят одноуровневые и многоуровневые системы контроля и управления доступом малой и средней вместимости, работающих в автономном или сетевом режимах и обеспечивают:

- ограничение допуска в зону охраняемого конкретного лица, группы лиц по дате и временными интервалами в соответствии с имеющимся идентификатора;
- автоматическую регистрацию событий в собственном буфере памяти,
- выдачу тревожных извещений (по несанкционированного проникновения, неправильного набора кода или взломе препятствующего устройства или его элементов) на внешние оповещатели или внутренний пост охраны; автоматическое управление открытием / закрытием устройств заграждения.

Используются, как и система контроля и управления доступом 1-го класса, на объектах, где требуется учет и контроль присутствия работников в разрешенной зоне, как дополнение к имеющимся на объекте систем охраны и защиты.

К системам контроля и управления доступом 3-го класса относят одноуровневые и многоуровневые системы контроля и управления доступом средней вместимости, работающих в сетевом режиме и обеспечивают:

- функции систем контроля и управления доступом 2 класса;
- контроль перемещений лиц и имущества по охранным зонам предприятия;
- ведение табельного учета и баз данных по каждому работнику, непрерывный автоматический контроль исправности составных частей системы;
- интеграцию с системами и средствами пожарной сигнализации и телевизионных систем видеоконтроля на релейном уровне.

Используются, как и система контроля и управления доступом 2-го класса, на объектах, где требуется табельный учет и контроль перемещений работников по объекту, для совместной работы с системами охраны и пожарной сигнализации, и телевизионных систем видеоконтроля.

К системам контроля и управления доступом 4-го класса относят многоуровневые система контроля и управления доступом средней и большой вместимости, работающих в сетевом режиме и обеспечивают:

- функции систем контроля и управления доступом 3 класса;
- охранную и пожарную безопасность, телевизионных систем видеоконтроля и других систем безопасности и управления на программном уровне;
- автоматическое управление устройствами заграждения в случае пожара и других чрезвычайных ситуациях.

Используются, как и система контроля и управления доступом 2-го класса на объектах, где требуется табельный учет и контроль перемещений работников по объекту, для совместной работы с системами охраны и пожарной сигнализации, и телевизионных систем видеоконтроля.

В Казахстане на сегодняшний день работает около 35 компаний, которые производят как технические средства, так и программное обеспечение и предоставляют услуги для формирования системы контроля и управления доступом под конкретные нужды предприятия. Среди них - CYPHRAX, U-Pro, Orion, SmartSecurity, Tescom, Elko, Энерго Инжиниринг, «Эксимтек ПЛЮС», Vel-Trade и др.

Исходя из выше сказанного в следующих разделах магистерской работы рассмотрены различные методы, модели и технологии, которые могут быть применены для повышения степени защищённости информационных ресурсов в локальных сетях. Причем акцент сделан как на адаптивное управление правами доступа, так и на реализацию аппаратной части защиты компонентов сети с помощью разграничения доступа.

2 МОДЕЛЬ АДАПТИВНОГО УПРАВЛЕНИЯ ПРАВАМИ ДОСТУПА В СЕТИ

Современный уровень применения информационных технологий и систем в экономике достиг высочайшего уровня. При этом как было показано в разделе 1 магистерской работы, появился новый термин – информационно-вычислительная сеть объекта информатизации.

Как и любой объект информатизации информационно-вычислительная сеть требует решения задач по защите информации и кибербезопасности. При этом большинство специалистов в области информационной технологий отмечают необходимость первоочередного приоритета заданиям сохранения целостности, конфиденциальности и доступности информации, вне зависимости от ее функционального назначения.

Общей первоначальной задачей при построении эффективных систем защиты и кибербезопасность информационно-вычислительная сеть объекта информатизации, остается задача обследования конкретного объекта защиты, формирование моделей потенциального нарушителя (компьютерного злоумышленника) и киберугроз. Реализация вышеуказанных шагов позволит в конечном итоге получить адекватные требования к системам защиты информации информационно-вычислительная сеть объекта информатизации.

В условиях усложнения сценариев кибератак аналитикам служб информационной безопасности необходимо достаточно оперативно реагировать на кибератаки, аномалии угрозы. Это делает актуальной задачу поиска новых способов повышения результативности принятия решений в заданиях реагирования на попытки деструктивного вмешательства со стороны компьютерного злоумышленника или недобросовестного персонала в работу объектов информатизации, в том числе, информационно-вычислительная сеть.

По мнению большого числа специалистов, достаточно перспективным представляется возможность описания функциональных моделей различных систем защиты информационно-вычислительная сеть в терминах теории сети Петри.

Такое представление позволит аналитикам информационной безопасности и защита информации детализировать киберугрозы в информационно-вычислительная сеть. Кроме того, в последующем, возможно определение состояний, которые потенциально определяют уязвимости

информационно-вычислительной сети перед новыми киберугрозами. Также рассматривается перспективность применения данной модели основе сетей Петри (и Петри–Маркова) и раскрашенных сетей Петри в качестве математической и алгоритмической составляющих, проектируемой интеллектуализированной системы поддержки принятия решений в процессе анализа кибеугроз для информационно-вычислительная сеть. По нашему мнению, данные суждения делают нашу работу релевантной и повышают результативность в ходе работ по созданию интеллектуализтрованной системы поддержки принятия решений в задачах защита информации и кибербезопасность информационно-вычислительная сеть.

В работах были представлены результаты исследований, посвященных применению сетей Петри для описания модели киберугроз. И хотя данные работы внесли несомненный теоретический вклад в данном вопросе, на наш взгляд предлагаемые авторами модели несколько затруднительно реализовать программно, в частности в интеллектуализтрованной системы поддержки принятия решений по защиты информации и кибербезопасность информационно-вычислительная сеть.

Основываясь на работах, модели угроз возможно построить, используя достаточно наглядную табличную форму отображения угроз при актуализации вопроса оценки защищенности информационно-вычислительная сеть. Но как было указанно ранее, данный подход к составлению моделей угроз трудоемок. А кроме того, рост количества угроз делает подобный табличный формат представления сложным для восприятия, особенно специалистам с небольшим опытом работы в сфере кибербезопасности.

Сети Петри (и Петри–Маркова) успешно использовались и для описания моделей нарушителя. Однако, авторы не рассматривали возможность корректировки модели нарушителя в информационно-вычислительная сеть, в частности путем объединения ее с моделями на основе теории графов, что позволило бы более точно описать переходы состояний в процессе вероятного преодоления компьютерного злоумышленника периметров (рубежей) киберзащиты информационно-вычислительная сеть.

В работах модели систем защиты информации рассматривались как предварительно выделенные в сети Петри последовательности элементарных операций, из которых возможна кибератака. Модели позволяли просчитывать вероятности реализации разных атак за отведенный промежуток времени. Однако, рассмотренные в модели не позволяли рассчитать временные характеристики в процессе реализации новых киберугроз.

В работах, также предлагались модели, основанные на сетях Петри и описывающие процессы реализации угроз в информационных системах (ИС). И ходя данные модели позволяли провести оценку многих параметров защищенности объектов, в частности, вероятности реализации угроз, времени на реализацию угроз, согласованность действий компьютерного

злоумышленника они представляются не до конца завершенными. В частности, в данных работах не изучен вопрос разрешения конфликтных ситуаций, возникающих при изменении состояний информационной системы в ходе атак, относящихся к разным классам. Это обстоятельство, на наш взгляд ограничивает практическую применимость данных исследований.

Таким образом, синтез новых моделей, а также дополнение существующих моделей и методов адаптивного управления киберзащитой информационно-вычислительной сети с использованием возможностей аппарата сетей Петри и учитывая потенциал визуализации сетей Петри, может стать эффективным инструментарием для прогнозирования состояния защищенности для информационно-вычислительной сети и других крупных учебных заведений. Это позволит значительно упростить понимание для новых киберугроз и в дальнейшем возможно результативное применение предлагаемых подходов аналитиками служб защиты информации, информационная безопасность и кибербезопасность различных объектов информатизации.

Таким образом в рамках второго раздела магистерской работы необходимо решить задачи по разработке:

концептуальной модель адаптивного управления киберзащитой объекта информатизации с использованием аппарата сетей Петри;

модели распределения пользовательских задач в компьютерных сетях объекта информатизации;

дополнений к методу контроля прав доступа в контексте сверки прав доступа, которые запрашиваются задачей и требований политики безопасности и степени согласованности задачи и разрешенных к доступу узлов информационно-вычислительной сети.

2.1 Концептуальная модель адаптивного управления правами доступа в информационно-вычислительной сети

Рассмотрим конкретный пример решения задачи адаптивного управления правами доступа пользователей с использованием аппарата сетей Петри и соответствующего программного обеспечения, которое позволяет автоматизировать корректировку профиля пользователя информационно-вычислительной сети, а также с помощью интеграции модуля интеллектуализированной системы поддержки принятия решений рекомендовать способы нейтрализации киберугроз в информационно-вычислительной сети.

Постановку задач управления правами доступа с учетом публикаций сформулируем так:

1) построить модель разграничения доступа для заданной информационно-вычислительной сети;

2) определить управляемые параметры модели;

3) выполнить параметризацию риска нарушения конфиденциальности информации для информационно-вычислительной сети.

Формальная математическая постановка задачи по оптимизации схемы разграничения доступа в информационно-вычислительной сети.

Исходные данные:

1) Объекты доступа в информационно-вычислительной сети – $AO = \{ao_i\}, i = \overline{1, I}$;

2) субъекты доступа в информационно-вычислительной сети – $SA = \{sa_j\}, j = \overline{1, J}$;

3) коммуникационные узлы в информационно-вычислительной сети – $CN = \{cn_k\}, k = \overline{1, K}$;

4) адаптивный механизм, который отвечает позволяет поддерживать метрики безопасности доступа в информационно-вычислительной сети на заданном уровне – $AM^0 = \{am_{i,j}^0\}, i = \overline{1, I}, \ j = \overline{1, J}$.

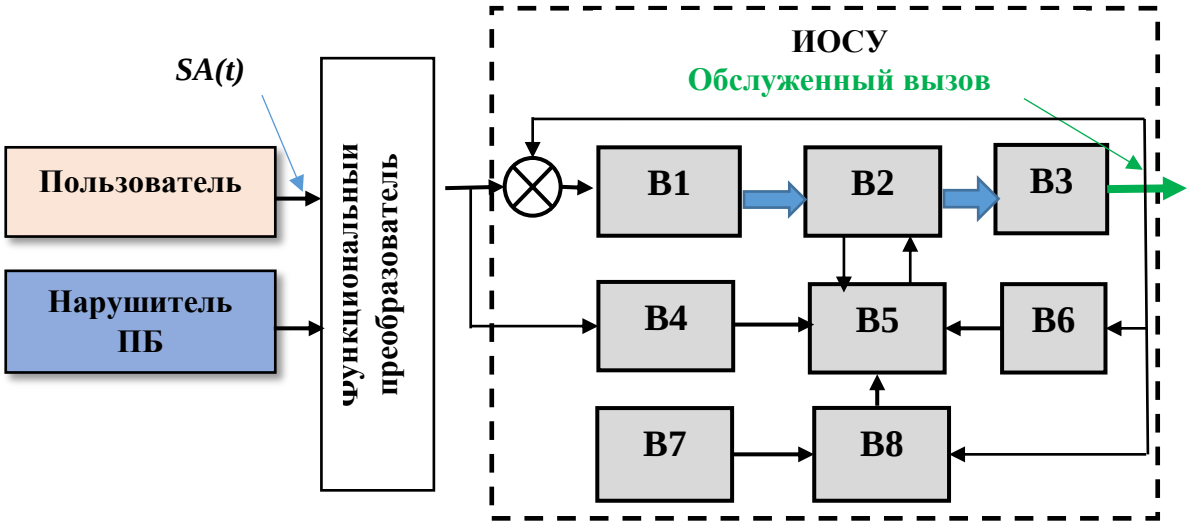


Рисунок 2.1 – Схема концептуальной модели адаптивного управления киберзащитой информационно-вычислительной сети

Принятые обозначения на рисунке 2.1:

B1 – блок информационно-измерительные устройства в информационно-вычислительной сети;

B2 – блок многоканальных управляющих устройств;

B3 – информационно-вычислительной сети как объект управления доступом к ресурсам;

B4 – блок прогнозирования состояний в информационно-вычислительной сети;

B5 – блок принятия решений о праве доступа;

B6 – блок расчета эффективности по количеству реализованных угроз, связанных с нарушением доступа в информационно-вычислительной сети;

B7 – блок априорной информации;

B8 – блок переменных моделей.

Будем полагать, что приемлемый уровень защиты информационно-вычислительной сети достигнут если выполняются условия, описанные в таблице 2.1.

Таблица 2.1 – Условия, при которых достигнут приемлемый уровень защиты информационно-вычислительной сети

No	**Параметр**	**Условие**
1	Адаптивный механизм, который отвечает позволяет поддерживать метрики безопасности доступа в информационно-вычислительной сети на заданном уровне.	$am_{i,j} = \begin{cases} 1, \text{if } am_i \text{ it is placed on} \\ \text{a node } cm_k; \\ 0, \text{Otherwise}. \end{cases}$
2	Ущерб, от вероятного несанкционированного доступа к ресурсам – $DA^0 = \{da^0_{i,j}\}, i = \overline{1,I},\ j = \overline{1,J}$.	*Смотри примечание.*
3	Структуры вычислительной сети – $NS = \{ns_{m,n}\}, m,n = \overline{1,K}$	$ns_{m,n} = \begin{cases} 1, \text{if } (cn_m \in NS_o)\& \\ (cn_n \in NS_o); \\ 0, \text{Otherwise}. \end{cases}$ $where \quad NS_o$ – Network objects.
Управляемые параметры (задаются администратором защиты информации и кибербезопасность)		
1	Признаки общего доступа к ресурсам информационно-вычислительной сети – $SV = \{sv_i\}$.	$sv_i = \begin{cases} 1, \text{if the general access} \\ \text{to a node } sv_i \\ \text{is allowed}; \\ 0, \text{Otherwise}. \end{cases}$
2	Размещение AO на узлах информационно-вычислительной сети – $MP^1 = [mp^1_{i,k}]$	$mp^1_{i,k} = \begin{cases} 1, \text{if } ao_i \in cn_k; \\ 0, \text{Otherwise}. \end{cases}$

3	Размещение SA на узлах информационно-вычислительной сети – $MP^2 = \left[mp_{j,k}^2\right]$.	$mp_{j,k}^2 = \begin{cases} 1, \text{if } sa_j \in cn_k; \\ 0, \text{Otherwise}. \end{cases}$
Примечание: Ущерб, от вероятного несанкционированного доступа к ресурсам (строка 2) определим степенью информационных ресурсов на узле информационно-вычислительной сети, а также профилем пользователя (с учетом характеристик вероятных нарушителей, см. таблицу 2.2).		

Условия, при которых достигнут приемлемый уровень защиты рассматривается в последующем в совокупности с данными таблицы 2.2.

Таблица 2.2 – Характеристики вероятных нарушителей

Классификация	Характеристика
По мотивам нарушения	Нарушение целостности, конфиденциальности, доступности с корыстной или иной целью.
По уровню информированности и квалификации	Нарушитель: 1) высокий уровень знаний; 2) достаточные знания для сбора информации, применение известных эксплойтов и написание собственного программного обеспечения для осуществления кибератаки; 3) Нарушитель не является авторизованным пользователем в информационно-вычислительной сети.
По месту действия	Без непосредственного (физического) доступа на территорию информационно-вычислительной сети. Нарушитель действует удаленно, например, через сети общего пользования.

Будем полагать, что целевой функцией является величина вероятных ожидаемых финансовых или иных потерь (ущерба) нанесённых объекта информатизации, в результате несанкционированного доступа к информационным ресурсам и информационно-вычислительной сетям.

Данный параметр определим в рамках магистерской работы как меру расхождения между реальным и рациональным размежеванием доступа пользователей к информационным ресурсам для конкретной информационно-вычислительной сети.

$$TF = \sum_{i=1}^{I} \sum_{j=1}^{J} da_{i,j}^0 \cdot \left| am_{i,j} - am_{i,j}^0 \right|, \tag{2.1}$$

где $\{am_{i,j}\}-$ элементы множества, которое отображает уже реализованные права доступа.

Полагаем, что с учетом

$$am_{i,j} = \sum_{k=1}^{K} ns_{i,k} \cdot w_{k,j}^{0}; \tag{2.2}$$

$$w_{k,j}^{0} = w_{k,j}^{1} + sv_i \cdot \left(1 - w_{k,j}^{1}\right); \tag{2.3}$$

$$w_{k,j}^{1} = \sum_{k=1}^{K} \left(mp_{i,k}^{2} \cdot mp_{k,j}^{1}\right). \tag{2.4}$$

Таким образом, была получена формулировка задачи разграничения прав доступа. Данная задача относится к заданиям нелинейной оптимизации. При решении рассматриваем вектор управляемых параметров в формате операций над булевыми переменными:

$$UD = \min \sum_{i-1}^{I} \sum_{j=1}^{J} da_{i,j}^{0} \cdot \left| am_{i,j} - am_{i,j}^{0} \right| \tag{2.5}$$

для следующих ограничений

$$\sum_{k-1}^{K} da_{i,j}^{1} \le 1, \tag{2.6}$$

$$\sum_{k-1}^{K} da_{i,j}^{2} \le 1. \tag{2.7}$$

Постоянная корректировка профиля активного пользователя предполагала применение специального итерационного алгоритма. Данный алгоритм базируется на неявной обратной связи сервера с пользователем ресурсами конкретной информационно-вычислительной сети. Ключевым фактором является статистика запросов. Оценка текущего профиля пользователя применялась для того чтобы проранжировать пользователей на группы по степени опасности для информационным ресурсам и информационно-вычислительной сетям.

Принято:

а) пользователь;

б) потенциально опасный пользователь;

в) опасный пользователь;

г) нарушитель.

Оптимизация настроек процедур управления доступом осуществлялась на основе определения таких параметров:

1) Интенсивность переходов $\lambda_{i,j}(t)$ (определены на основе регрессионных моделей);

2) параметризация риска, который связан с нарушением политики информационной безопасности (рассматривались все свойства: конфиденциальность, целостность и доступность) информации в информационно-вычислительной сети.

Была получена следующая регрессионная модель 2-го порядка:

$$P_r(\tau) = da_0 + \sum_{k=1}^{m} da_k \cdot h_k + \sum_{ao=1}^{m} da_{ao,ao} \cdot h_{ao}^2 + \sum_{i,j=1}^{m} da_{i,j} \cdot h_i \cdot h_j,\ i \neq j,$$

где $h = (h_1, ..., h_m)^T -$ управляемые параметры, которые регламентируют правила разграничивающие доступ в сетях конкретных информационно-вычислительной сети,

$\tau -$ время.

Относительно любого информационно-вычислительной сети с распределенной схемой доступа к ресурсам модель абонентских заданий определена так:

$$\sum = (PN, PIS, AT, s_0, FTR, MRT, RES), \tag{2.8}$$

где $PN = (TGR, T, MPN, F) -$ информационные ресурсы и информационно-вычислительные сети (представлены сетью Петри);

$TGR = \{tgr\} -$ множество вершин графа (вершина – поставщик информационных ресурсов информационно-вычислительной сети);

$T = \{t\} -$ число переходов между вершинами; $MPN = (mpn_1, ..., mpn_n) -$ разметка сети Петри;

$F -$ отношение соседства вершин;

$PIS -$ политики информационной безопасности;

AT – активные задачи, инициированные пользователями информационными ресурсами и информационно-вычислительные сетями;

s_0 – начальное состояние $S = \{s\}$;

$FTR: PN \times AT \times MRT \times PIS \times A \rightarrow S$ – функция перехода между состояниями информационных ресурсов информационно-вычислительной сети;

$MRT = \langle CL, U \rangle$ – маркеры в сети Петри;

CL – класс ресурсов, которые запрашивают абоненты (пользователи) (U);

RES – текущее позиция в правах доступа к информационным ресурсам в информационно-вычислительных сетях.

С учетом предшествующих выкладок, получены такие правила для программного продукта «Анализатор угроз» для принятия решения о возможности доступа абонента (пользователя к информационным ресурсам:

абоненту U санкционирован доступ к информационным ресурсам владельца OWR, если процедура обоюдной аутентификации прошла корректно.

Для владельца информацон OWR определяется локальная учетная запись. При этом в данной записи отображены все абоненты и их тип доступа в соответствии с ПБ:

$$Has\ COMP\ Ass\ Ri(U, OWR, PIS) = \begin{pmatrix} Is\ TRU\ By\ U(U, OWR) \wedge \\ Is\ TRU\ By\ TGR(OWR, U) \wedge \\ \wedge (MapU\ To\ UD(OWR, U) \neq 0) \end{pmatrix} \wedge$$

$$Is\ Acc\ AL\ By\ PIS(U, OWR, PIS), \tag{2.9}$$

и по отношению к информационные ресурсы и информационно-вычислительные сети, локальные учетные записи на узлах, в которых отображены абоненты, обязаны также обладать правом доступа – RI к объекту Ob:

$$Has\ FC\ Ass\ Ri\left(U, OWR, PIS, Ob, RI\right) = \begin{pmatrix} Is\ TRU\ \ By\ U\left(U, OWR\right) \wedge \\ \wedge\ Is\ TRU \\ By\ TGR\left(OWR, U\right) \wedge \\ \begin{pmatrix} ListU = \\ MapU\ To\ UD\left(OWR, U\right) \neq 0 \end{pmatrix} \end{pmatrix} \wedge$$

$$Is\ Acc\ AL\ By\ PIS\left(U, OWR, PIS\right), \qquad (2.10)$$

где $Acc -$ доступ;

$RI -$ право доступа к информационные ресурсы и информационно-вычислительные сети;

$AL -$ позволено;

$TRU -$ является надежным;

$MapU -$ карта абонента/пользователя;

$ListU -$ локальная учетная запись абонента;

$MapU\ To\ UD -$ функция, которая отображает множество пользователей информационные ресурсы и информационно-вычислительные сети в формате локальных учетных записей владельца информационных ресурсов – *OWR*.

На этапе нахождения вероятностных параметров реализации конкретной киберугрозы (нарушение разграничения полномочий) для информационно-вычислительной системы, будем полагать, что смоделирована работа всех систем защиты информации. И кроме того выполнен расчет вероятностных параметров смены разметок в соответствующих вероятностных сетях Петри (ВСП) или сетях Петри-Маркова (СПМ).

Реализация угрозы в информационно-вычислительной сети – это порядок передвижений (полушагов) по вероятностных сетях Петри или сетях Петри-Маркова. Будем полагать, что вероятностных сетях Петри или сетях Петри-Маркова пребывают в каждом состоянии некоторое случайный отрезок времени. Рассматриваемому временному отрезку также ставим в соответствие параметр, задающий соответственно величину плотности распределения вероятности. Далее, анализируем передвижения по траекториям вероятностных сетях Петри или сетях Петри-Маркова за полушаги. А затем проверяем логические условия переключения вероятностных сетях Петри или сетях Петри-Маркова в следующее состояние. Постоянство состояний вероятностных сетях Петри или сетях Петри-Маркова определит траекторию исследуемого процесса для рассматриваемой киберугрозы. Аналитически описать процесс можно, применив интегро-дифференциальные уравнения для траекторий передвижений из начальных состояний в конечные.

Рассмотрим пример: пусть $h(tr:1(a)) \to j(a) = h(tr_1) -$ номер траектории передвижения из состояния $a_{1(a)}$ (индекс с буквой означает номер состояния) в состояние $a_{j(a)}$.

Траектория включает в себя серию полушагов. Или из состояния в переход, а затем из перехода в состояние и т.д.:

$$S_{1[h(tr)]}, S_{2[h(tr)]}, \ldots, S_{i[h(tr)]}, \ldots, S_{j[h(tr)]},$$

где i, j – индексы, которые соответствуют номеру состояния (или номеру перехода).

Количество траекторий описано величиной $H(tr)$. Тогда величины, задающие вероятность и плотность распределения времени выполнения соответствующего полушага, соответственно $P_{j(a)j(z)}$ и $f_{j(a)j(z)}$.

Вероятность и плотность распределения времени передвижения из $a_{1(a)}$ в $a_{j(a)}$ по $h(tr_{1j})$ находим так:

$$P_{h(tr_{1j})} = \prod_{j[h(ttr_{1j})]=1}^{J[h(tr_{1j})]} P_{j[h(tr_{1j})]};$$

$$f_{h(tr_{1j})} = f_{1[h(tr_{1j})]} * f_{2[h(tr_{1j})]} * \ldots *$$
$$* f_{i[h(tr_{1j})]} * \ldots * f_{J[h(tr_{1j})]},$$

где $J[h(tr_{1j})]$ – количество позиций и переходов в $h(tr_{1j})$;

$*$ – обозначение операции свертки $a_{1(a)}$ по всем возможным $h(tr_{1j})$ из соотношений:

$$P_{1(a)j(a)} = \prod_{h(tr_{1j})=1}^{H(tr_{1j})} P_{h(tr_{1j})}; \tag{2.11}$$

$$f_{1(a)j(a)} = \frac{\prod_{h(tr_{1j})=1}^{H(tr_{1j})} P_{h(tr_{1j})} \cdot f_{h(tr_1)}}{\prod_{h(tr_{1j})=1}^{H(tr_{1j})} P_{h(tr_{ij})}}. \tag{2.12}$$

В соответствии с можно найти вероятности $\Phi_{i,j(t)}$ реализации анализируемой киберугрозы в инфомационно-вычислительной сети.

С учетом работ были предложены уточнения к методу контроля и управления доступом, с учетом специфики сети инфомационно-вычислительной сети. Уточненный и дополненный метод контроля и управления доступом заключается в сверке прав доступа, которые запрашиваются задачей и требований политики безопасности, а кроме того согласованием задачи и разрешенных к доступу узлов инфомационно-вычислительной сети.

Для узлов инфомационно-вычислительной сети также происходит процедура сверки прав доступа для всех абонентов, имеющих соответствующие права. Как результат будет получено множество узлов, на которых абонентские задачи допускаются к выполнению. При этом учитываются текущие показатели политики безопасности для конкретного инфомационно-вычислительной сети и метрики безопасности. Возможна корректировка правил для новых задач или перераспределяемых задач. Эта, корректировка или перераспределение задач могут быть описаны в нотации сетей Петри и с учетом математической модели, которая описана выражениями.

Опираясь на выше приведенные рассуждения, в базисе модифицированных сетей Петри была разработана модель адаптивного ролевого управления доступом к ресурсам информационно-вычислительной сети (Система управления доступом). И выполнено имитационное моделирование в пакетах PIPE v4.3.0 (Platform Independent Petri net Editor) и Petri.Net Simulator. 2.017.

Такой подход позволил корректно описать конфликтные ситуации, а кроме того была учтена особенность обработки запросов, которые возникают

на большинстве информационных систем информационно-вычислительной сети в процессе многопользовательского режима работы.

На рисунке 2.2-2.4 показаны схемы имитационных моделей и формализованные результаты моделирования.

Схемы отображают логическую структуру операционной модели системы правами доступа (для варианта трёхступенчатого управления). Позиции и переходы в сетевой модели в базисе модифицированных сетей Петри показаны в таблице 2.3. Позиции и переходы в сетевой модели в базисе модифицированных сетей Петри приняты по.

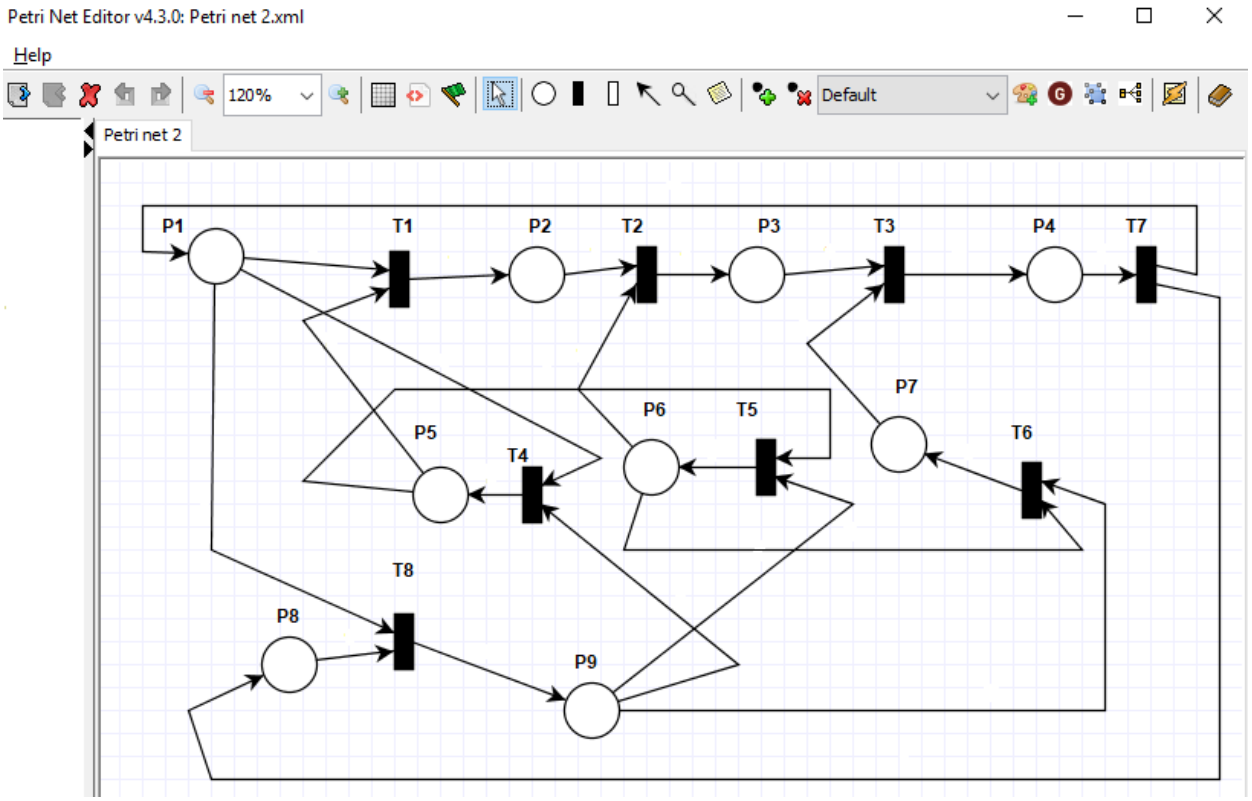


Рисунок 2.2 – Имитационная модель адаптивного управления доступом в информационно-вычислительной сети PIPE v4.3.0 (с учетом регулирования роли абонента)

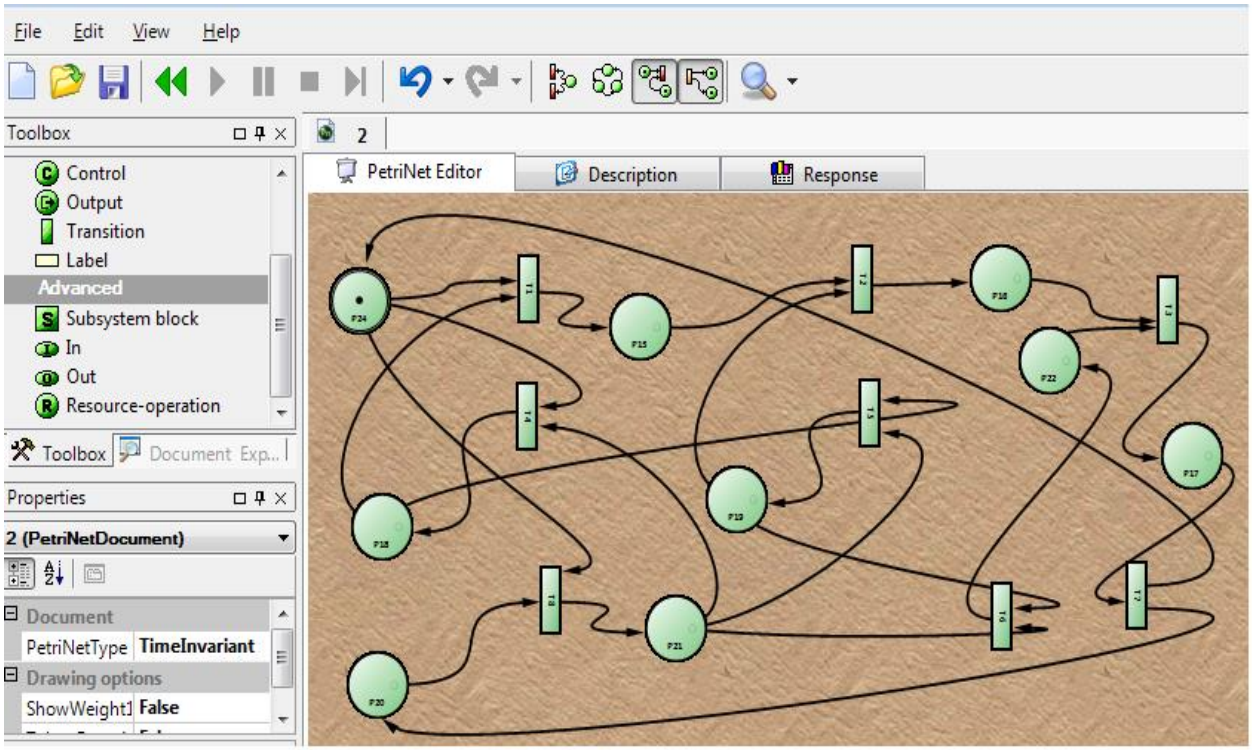


Рисунок 2.3 – Имитационная модель адаптивного управления доступом в информационно-вычислительной сети Petri.Net Simulator. 2.017

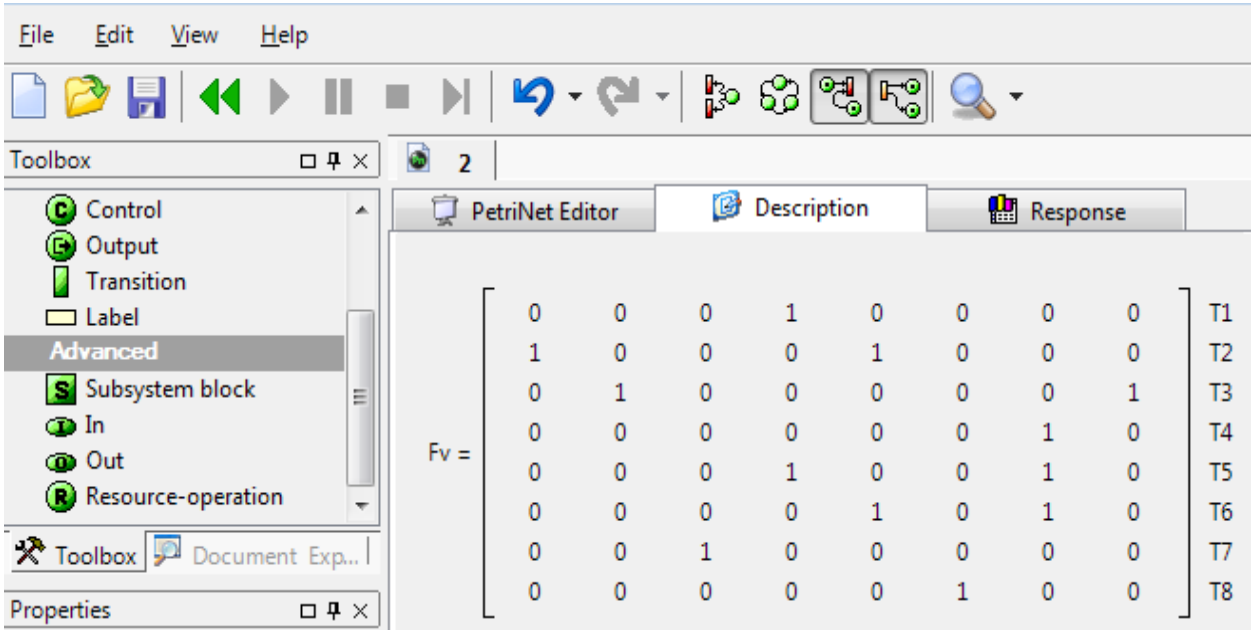


Рисунок 2.4 – Формализация позиций в модели адаптивного управления доступом в информационно-вычислительной сети
Среда моделирования Petri.Net Simulator. 2.017

В процессе исследований была выполнена оценка результативности предложенных моделей и уточнений к методу контроля прав доступа. Для оценки результативности, предложенных решений, использовался показатель, характеризующий сокращение затрат времени на принятие решений. Соответственно, оценивались затраты времени на обработку данных до и после применения, предложенных моделей и метода.

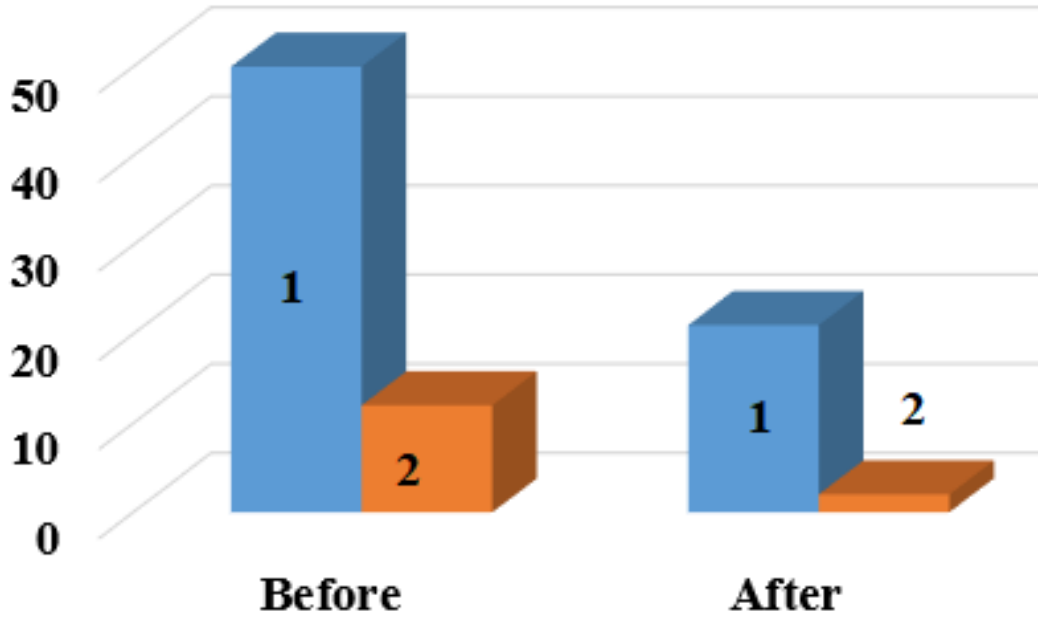
Имитационный эксперимент проведен для 400 вычислительных узлов. Каждому узлу ставилась в соответствие виртуальная машина. Количество реализованных киберугроз, до имплементации в интегрированную систему информационной безопасности и кибербезопасность информационно-вычислительной сети и после ее внедрения, показано на рис. 2.5.

Полученные в ходе имитационного моделирования статистические данные (в частности, касающиеся о динамики маркеров в МВП дали основание установить конкретные характеристики система учета доступа для компаний, участвовавших в апробации модели.

Таблица 2.3 – Обозначения, принятые в имитационной модели (в базисе модернизированных сетей Петри)

Позиции	
Обозначение принятые на схеме (см. рис. 2.2 и 2.3.)	Описание позиции для пользователя
$P1$	активное состояние
$P2$	Абонент к работе по проекту допущен
$P3$	Абонент допущен к работам с функциональными компонентами информационно-вычислительной сети
$P4$	Абонент имеет допуск к файлам и контенту информационно-вычислительной сети
$P5$	Проверка имеющихся прав для выполнения заданий
$P6$	Проверка прав доступа к функциональными компонентами
$P7$	Проверка прав доступа к файлам информационно-вычислительной сети
$P8$	Восстановление исходного состояния система учета доступа
$P9$	Ограничение активности по времени (например, при корректировках задания или выполнения тестовых заданий)

Переходы	
Обозначение принятые на схеме (см. рис. 2.2 и 2.3.)	Описание
*T*1...*T*8	Отображают совокупность условий перехода (и модификации) маркеров из одной позиции сети в другие. Условия определены набором априорных данных.



1- Общее количество киберугроз которые были реализованы в информационно-вычислительной сети;
2- Киберугрозы, связанные с нарушение прав доступа и превышением полномочий

Рисунок 2.5 – Оценка информационной безопасности информационно-вычислительной сети

К достоинствам нашего подхода можно отнести тот факт, что предложенные решения, в частности, разработанная концептуальная модель адаптивного управления правами доступа с использованием аппарата сетей Петри, а также модели и метод, были успешно апробированы в подсистеме администрирования информационной и кибербезопасности нескольких крупных предприятий г. Алматы.

2.2 Метод и модель анализа возможных угроз при аутентификации пользователей в информационно-вычислительной сети

Как было ранее показано в разделе 1, релевантной остается проблема по совершенствованию существующих и разработке новых методов и моделей выявления угроз несанкционированного доступа к информационно-вычислительной сети объекта информатизации, при акцентировании на задачах минимизации ошибочных результатов проверки действий абонентов. Это должно увеличить результативность выявления новых угроз в информационно-вычислительной сети и эффективно распределить ресурсы систем защиты информации и кибербезопасности учебных заведений.

Цель исследования, результаты которого представлены в данном разделе магистерской диссертации состоит в развитии методов и математических моделей, которые применимы по отношению к процедурам обнаружения угроз несанкционарного доступа при аутентификации абонентов в информационно-вычислительной сети объекта информатизации.

Для достижения цели решались задачи по развитию и разработке:

метода анализа данных о возможных угрозах в информационно-вычислительной сети, что позволить минимизировать время их ее распознавания;

математической модели аутентификации абонентов информационно-вычислительной сети, что позволяет сократить количество ошибочных срабатываний и сообщений о ложных угрозах.

При формировании наборов, которые задействованы в детектировании угроз (далее НДУ – набор для детектирования угрозы) для информационно-вычислительной сети учитывалось следующее:

1. Детектор не должен активизироваться при легитимных действиях абонентов (пользователей в информационно-вычислительной сети). Также детектор не должен активизироваться при легитимных действиях объектов и субъектов систем защиты информации и кибербезопасность в информационно-вычислительной сети;

2. Интервалы значений реализаций признаков, которые соответствуют объектам детектирования, должны быть достаточными для минимизации тождественностей. Полагаем, что объекты используемые для обучения системы детектирования получены с использованием методов и моделей, описанных.

3. Если экземпляр набора, участвующего в детектировании угрозы для информационно-вычислительной сети, успешно ее распознал, то данный экземпляр сохраняется в базе знаний систем защиты информации и далее принимает участие в генерации новых поколений объектов используемых для “тренировки” систем кибербезопасностей.

Проанализировав имеющиеся методы и алгоритмы аутентификации абонентов информационно-вычислительной сети, была разработана усовершенствованная схема, которая показана на рис. 2.6.

На схеме (см. рис. 2.6) используются два потока. Поток №1 предназначен для поиска в базе угроз, которые уже встречались раньше в базе знаний систем защиты информации конкретной информационно-вычислительной сети. Поток №2 предназначен для выявления новых угроз. Под новыми понимаем угрозы, которые ранее не встречались при аутентификации субъекта информационно-вычислительной сети. Отличительной чертой, предложенной схемы, является возможность задействовать механизм, удостоверяющий проверку. Данный подход основан на получении активирующего сигнала от «ближайших» набора для детектирования угрозы. При этом при анализе сигнала используется весь массив данных, которые предоставлены одним и тем же субъектом информационно-вычислительной сети.

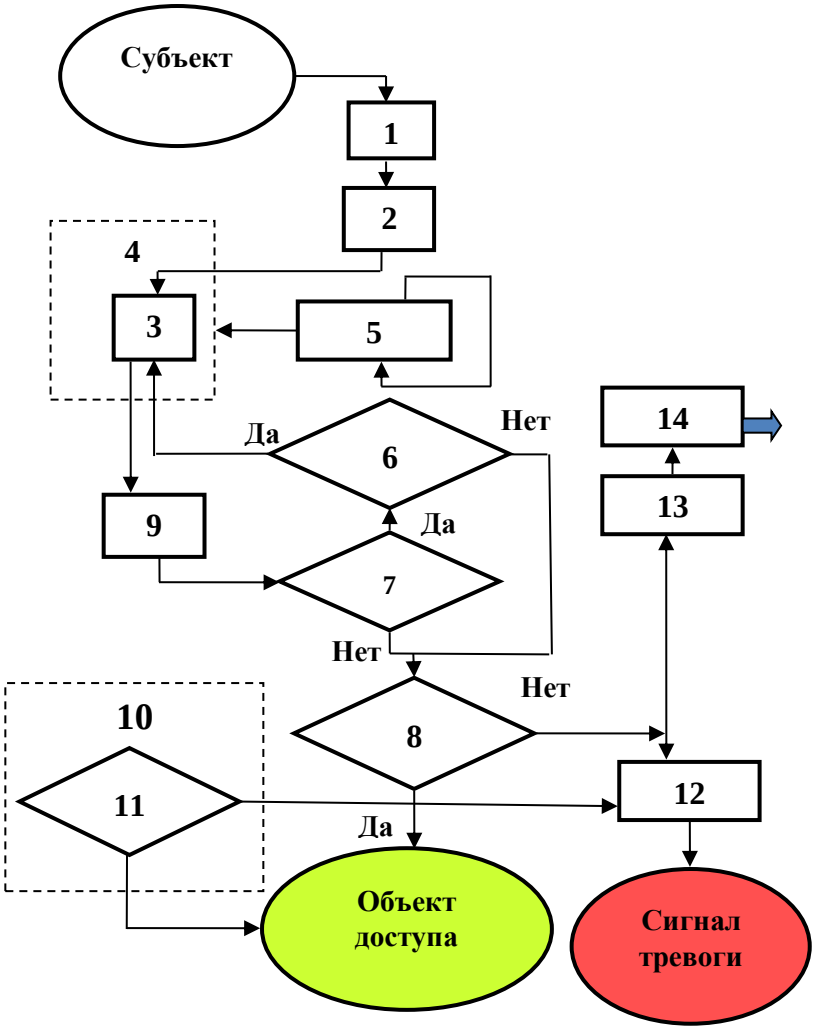


Рисунок 2.6 – Схема аутентификации субъекта в информационно-вычислительной сети на основе обновляемых набор для детектирования угрозы

Принятые на схеме обозначения: 1 – считать идентифицирующие данные субъекта; 2 – сформировать представительский набор для поиска; 3 – реализация процедур поиска на множестве наборов, используемых для детектирования (МНДУ); 4 – множество наборов, используемых для детектирования угроз информационно-вычислительной сети; 5 – обновление множество наборов, используемых для детектирования угроз; 6 – проверка сигнала; 7 – ожидание подтверждающего сигнала; 8 – проверка пройдена; 9 – формирование результирующих данных; 10 – множества наборов восстановленных из памяти, используемых для детектирования угроз; 11 – поиск среди множества наборов восстановленных из памяти, используемых для детектирования угроз; 12 – блокирование действий субъекта; 13 – запись в память системы защиты информационно-вычислительной сети; 14 – обновление памяти.

Для расчета скорости с которой можно обнаружить попытки проникновения в информационно-вычислительной сети (для предложенной схемы аутентификации субъекта на основе обновляемых набор для детектирования угрозы (далее ОНДУ), рис. 2.7) использовались следующие уравнения:

$$\frac{da}{dt} = rev \cdot \zeta \cdot |Tr| - \alpha \cdot |Tr(MP \setminus W(MP))|, \tag{2.13}$$

$$W(MP) = \{w \mid \forall w \exists mp \in MP : (w)\, Con\ MP\ |\}, \tag{2.14}$$

где $Tr = Tr(S) -$ множество угроз для информационно-вычислительной сети;

$MP -$ система защиты информации в информационно-вычислительной сети;

$W(MP) -$ система защиты информации, которые задействованы в проверках абонентов, $rev -$ коэффициент задержки при включении систем защиты информации в информационно-вычислительной сети;

$\zeta -$ коэффициент, характеризующий увеличение количества угроз;

$Tr(PR) -$ угрозы, нейтрализованные системы защиты информации;

$\alpha -$ коэффициент, характеризующий степень нарушений работоспособности информационно-вычислительной сети, вызванных атакой;

$Con -$ условия, при которых угрозы для информационно-вычислительной сети нейтрализованы существующей системой защиты.

Предлагаемый метод анализа данных о возможных угрозах в информационно-вычислительной сети схематично показан на рис. 2.17.

С учетом результатов работ было предложено в процессе “тренировки” набор для детектирования угрозы, применять весовые значения для обновляемых набор для детектирования угрозы особо выраженной связью «причина-следствие». Показано, что с позиции приоритетности задачи увеличения результативности выявления угроз для информационно-вычислительной сети, обязательно исключать набор для детектирования угрозы, которые не вошли в допустимую область. В итоге, как результат “тренировки” системы распознавания угроз и соответствующего алгоритма, получим результирующее подмножество, способное к последующему обучению.

Полагаем, что в процессе аутентификации задействованы данные, которые были получены от авторизованных субъектов. Кроме того, в процессе анализа данных о возможных угрозах в информационно-вычислительной сети, задействованы набор для детектирования угрозы, служебные данные об абонентах и регулирующие правила. На входе алгоритма используются данные, которые получены от систем защиты информации информационно-вычислительной сети (например, первоначальная информация от субъекта). Далее набор для детектирования угрозы проверяют эту информацию и генерируют соответствующее решение.

Рисунок 2.7 – Алгоритм формирования наборов, участвующих в детектировании угроз для информационно-вычислительной сети

Ниже приведен пример регулирующих правил для подсистемы анализа возможных угроз при аутентификации пользователей в информационно-вычислительной сети ОБИ.

Заданы следующие первоначальные данные:

DS – множество набор для детектирования угрозы $(ds \subset DS)$;

ID – входная информация от субъекта информационно-вычислительной сети $(id \subset ID)$;

$SS-$ множество реализаций признаков набор для детектирования угрозы или исходных данных субъекта;

$VE-$результаты верификации за период времени t;

$SP-$служебные параметры систем защиты информации в информационно-вычислительной сети.

Получен следующий перечень регулирующих правил для системы управления базой данных информационно-вычислительной сети, (модель аутентификации абонентов):

$$VE(ds(ID)) = \sum(all\ SS(id) > tv(SS)); \tag{2.15}$$

$$\begin{aligned} &if \quad VE(ds(ID)) > tvr \quad \& \quad nc = isNull \quad then \\ &VE = 1 \quad \& \quad new\, ds(ID); \end{aligned} \tag{2.16}$$

$$\begin{aligned} &if \quad VE(ds(ID)) > tvr \quad \& \quad nc == 0 \quad then \\ &VE = 0 \quad \& \quad Stop; \end{aligned} \tag{2.17}$$

$$\begin{aligned} &if \quad VE(ds(ID)) > nc \cdot tvr \quad then \\ &ds(nit) \subset new\ DS, \end{aligned} \tag{2.18}$$

где $tv-$пороговый уровень схожести сравнения набора реализации признаков (SS) с исходными данными;

$tvr-$ пороговый уровень результирующих данных набор для детектирования угрозы с фиксацией предупреждения об угрозе для информационно-вычислительной сети;

$nc-$минимально требуемое количество достоверных подтверждений от других набор для детектирования угрозы (обязательно для отнесения субъекта к представляющим потенциальную опасность);

$nit-$ количество итераций в цикле.

Были рассмотрены следующие типы элементов, которые можно использовать в предложенной схеме аутентификации:

Группа № 1 – наборы, участвующие в процессах детектирования. (например, бинарные матрицы, используемые как объекты обучения наборов.

Каждый набор соответствует конкретному классу реализаций признаков объектов наблюдения. Первая группа отвечает за обработку информации, которая предоставлена субъектом в информационно-вычислительной сети.

Группа № 2 – базовые объекты (БО) информационно-вычислительной сети. Для каждого элемента защищаемой информационно-вычислительной сети (в ряде случаев для всей системы), формируют набор внутренние базовые объекты. Данные базовые объекты, предназначены для реализации служебных функций.

К базовым объектам типичной информационно-вычислительной сети можно отнести: информационные массивы, а также объект, который будет содержать, записывать и накапливать данные по собственным характеристикам объекта защиты. Эта информация в последующем будет использована как основа для синтеза наборов, участвующих в детектировании угроз. Ко второй группе, также следует отнести служебные конфигурации систем защиты информационно-вычислительной сети. Данные конфигурации содержат данные обязательные для корректировки наборов, участвующих в детектировании угроз.

Следует упомянуть и управляющую подсистему информационно-вычислительной сети. Данная подсистема непосредственно реализует контроль за процессом аутентификации субъектов в процессе клавиатурного распознавания в информационно-вычислительной сети на основе обновляемых набор для детектирования угрозы.

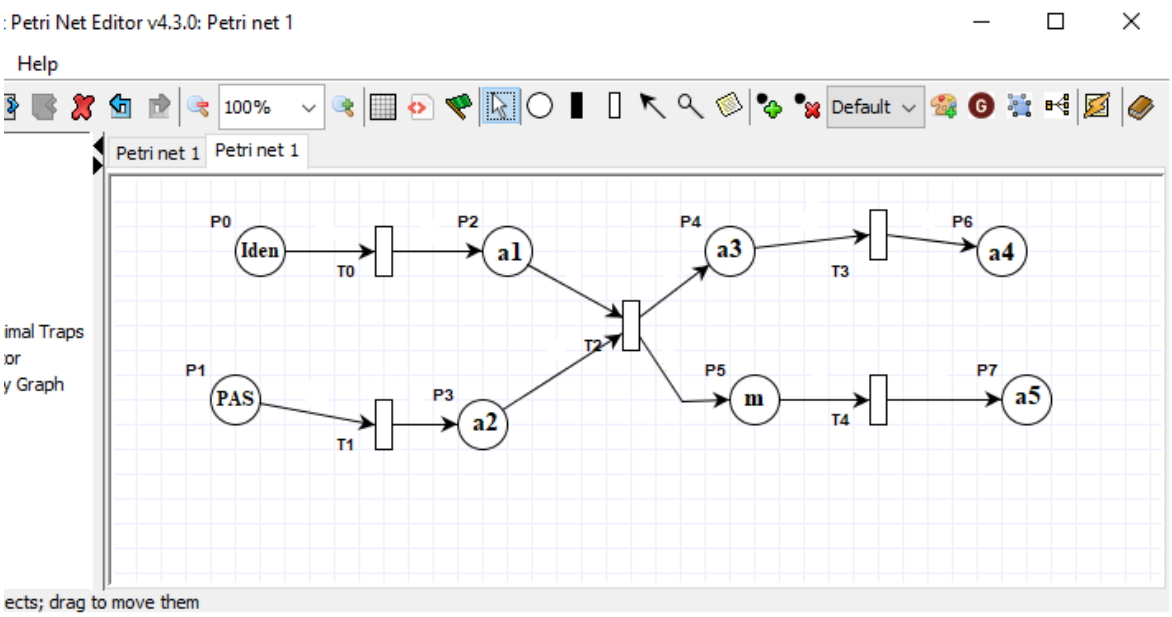
Предлагаемый метод позволил:

1) повысить достоверность и надежность результатов верификации (это достигнуто за счет дополнительных проверок в ходе использования информационно-вычислительной сети);

2) повысить информативность наборов для детектирования угрозы для конкретных классов киберугроз информационно-вычислительной сети (это достигнуто за счет сохранения значений признаков угрозы с высокой степени для минимизации признакового пространства, выделяемого под набор для детектирования угрозы).

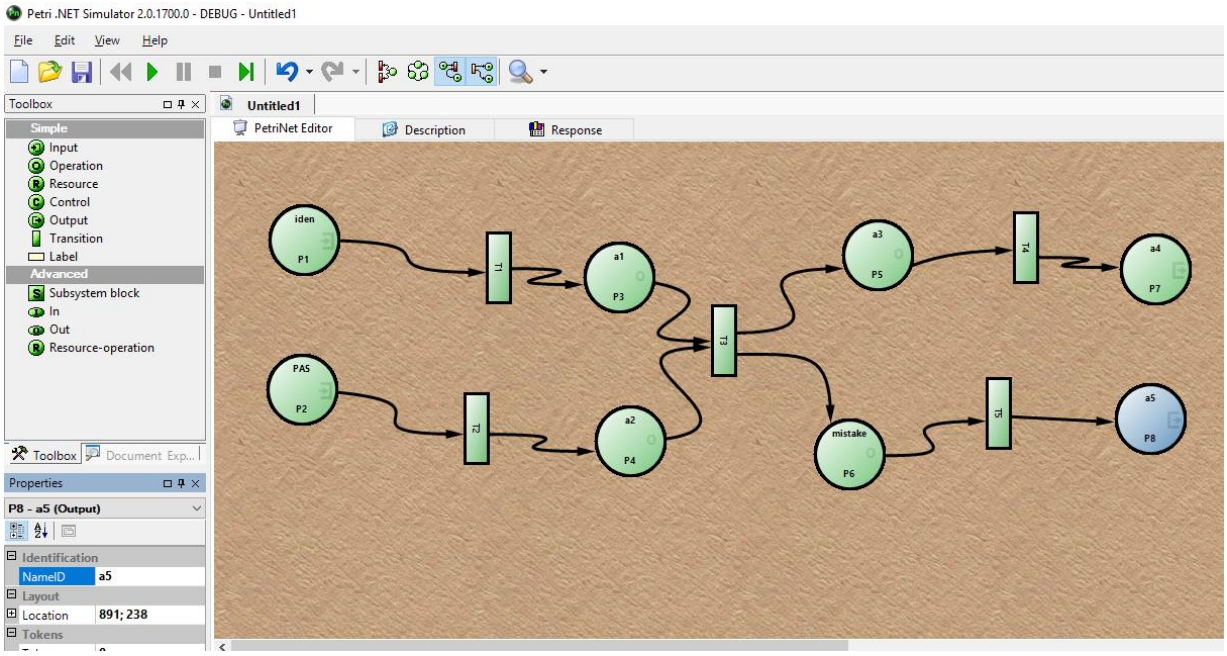
Классический вариант идентификации пользователей при входе в информационно-вычислительной сети производится по паролю. Пользователи ресурсами информационно-вычислительной сети осуществляют ввод/изменение данных, например, вовремя загрузки ответов на учебные задания или иных случаях.

Выполним имитационное моделирование в среде PIPE v4.3.0 для классической схемы идентификации (аутентификации). На рис. 2.8 представлена блок-схема алгоритма и сеть Петри входа пользователя в информационно-вычислительной сети для классической схемы. Аналогичная модель для среды Petri.Net Simulator. 2.017. показана на рис. 2.9.

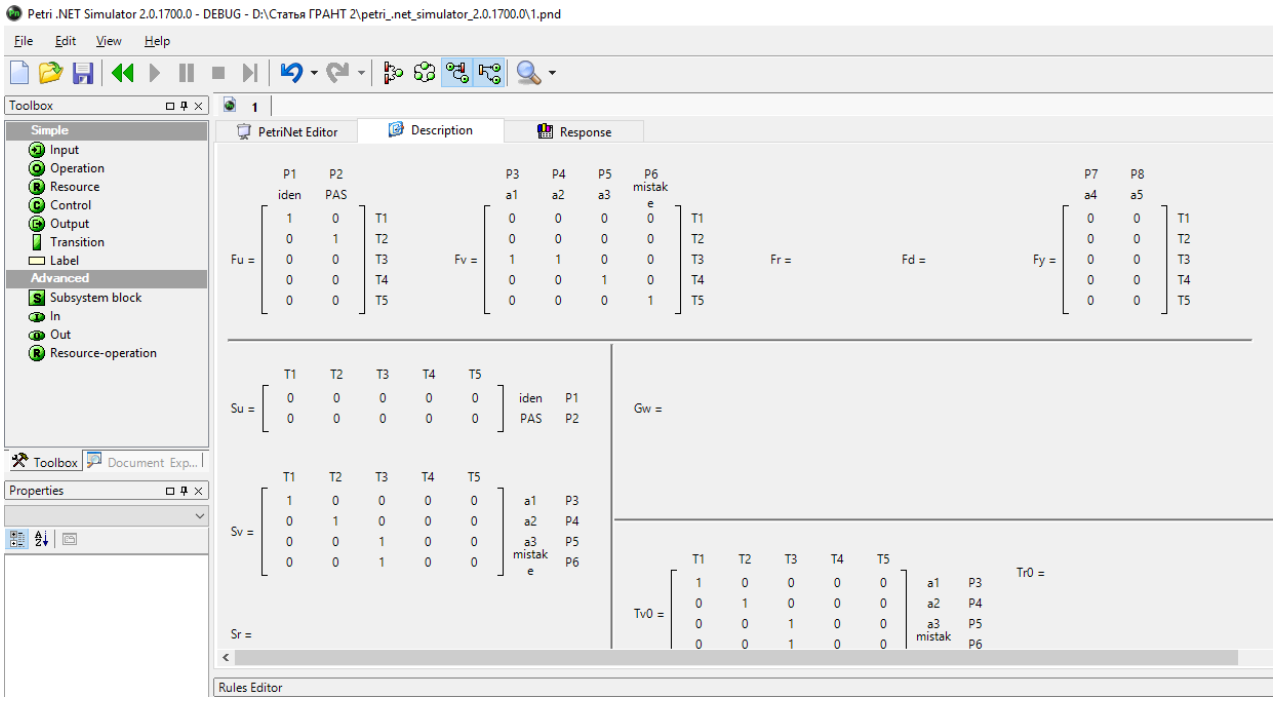


Принятые обозначения: $P0(PAS)-$пароль, вводимый абонентом в сети информационно-вычислительной сети для аутентификации (1 – пароль, соответствующий требованиям; фишки другого типа – неправильные и (или) некорректные пароли); $P1(Iden)-$идентификатор, используемый пользователем для идентификации в информационно-вычислительной сети; $a1-$ ввод пароля при запросе информационно-вычислительной сети; $a2-$ идентификатор введённый абонентом (пользователем); $a3-$ идентификатор прошел проверку паролем, и аутентификация успешно завершена; $a4-$ санкционированный вход пользователя информационно-вычислительной сети; $a5-$ право доступа пользователя информационно-вычислительной сети не предоставлено; $T0-T4-$отображают совокупность условий перехода.

Рисунок 2.8 – Сеть Петри входа пользователя в информационно-вычислительной сети для классической схемы идентификации (аутентификации) PIPE v4.3.0



а)

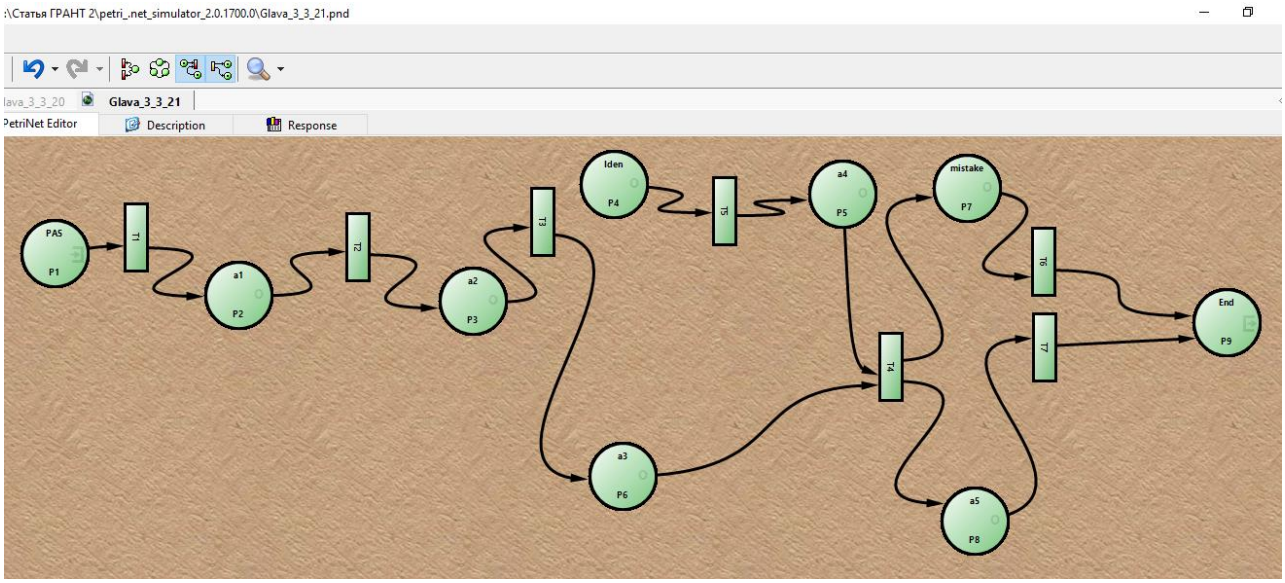


б)

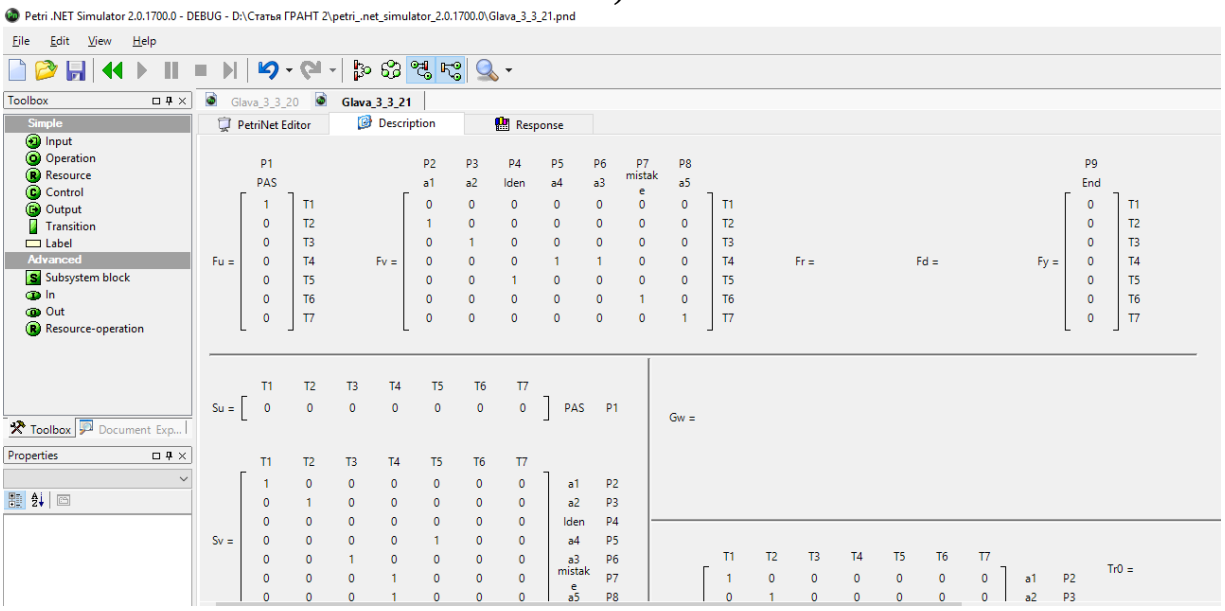
а) модель сети; б) фрагмент результатов моделирования

Рисунок 2.9 – Сеть Петри входа пользователя в информационно-вычислительной сети для классической схемы идентификации (аутентификации) Petri.Net Simulator. 2.017.

Руководствуясь ранее разработанной схемой (см. рис. 2.16–2.17), выполним имитационное моделирование для схемы аутентификации субъекта в информационно-вычислительной сети на основе обновляемых наборов для детектирования угрозы, см. рис. 2.10.



а)



а) модель сети; б) фрагмент результатов моделирования

Рисунок 2.10 – Имитационное моделирование для схемы аутентификации субъекта в информационно-вычислительной сети на основе обновляемых набор для детектирования угрозы с использованием нотации сетей Петри (Petri.Net Simulator. 2.017.)

В таблице 2.4 показаны обозначения, принятые в имитационной модели для схемы аутентификации субъекта в информационно-вычислительной сети на основе обновляемых набор для детектирования угрозы, представленной на рис. 2.10 (в базисе сетей Петри).

Таблица 2.4 – Обозначения, принятые в имитационной модели для схемы аутентификации субъекта в информационно-вычислительной сети на основе обновляемых набор для детектирования угрозы, представленной на рис. 2.10 (в базисе сетей Петри)

Позиции	
Обозначение принятые на схеме см. рис. 2.20	Описание позиции для пользователя
PAS	пароль, введенный абонентом информационно-вычислительной сети для аутентификации
Iden	проверка представительского набора
*a*1	ввод пароля при запросе информационно-вычислительной сети
*a*2	идентификатор введённый абонентом (пользователем) совпал с «эталоном» детектирующего набора (ДеН)
*a*3	набор для детектирования угрозы определил, что идентификатор введённый абонентом совпал с «эталоном» ДеН
*a*4	набор для детектирования угрозы определил, что пароль введённый абонентом совпал с «эталоном» ДеН
*a*5	проверка абонента информационно-вычислительной сети пройдена
mistake	проверка не пройдена, формирование результирующих данных и корректирование набор для детектирования угрозы в информационно-вычислительной сети
Переходы	
*T*1...*T*7	Отображают совокупность условий перехода (и модификации) маркеров из одной позиции сети в другие. Условия определены набором априорных данных.

В таблице 2.5 показан фрагмент для набора выходных данных, которые были получены в ходе сравнения вычислительных экспериментов и опытной проверки, предложенной схемы аутентификации для случая задачи анализа поведения субъекта в информационно-вычислительной сети.

На рис. 2.10 б) показаны результаты тестирования, предложенной схемы

аутентификации для случая задачи анализа поведения субъекта в информационно-вычислительной сети.

На графиках рис. 2.11 показаны зависимости для модуля системы парольной защиты информационно-вычислительной сети. При этом анализируется скорость распознавания субъекта в информационно-вычислительной сети от количества введенных субъектом символов.

В экспериментах было принято, что длина пароля равна 8 символам, а количество возможных попыток ввода пароля 3. Количество наборов, задействованных в детектировании составляло 250.

Таблица 2.5 – Фрагмент для набора выходных данных в ходе экспериментов

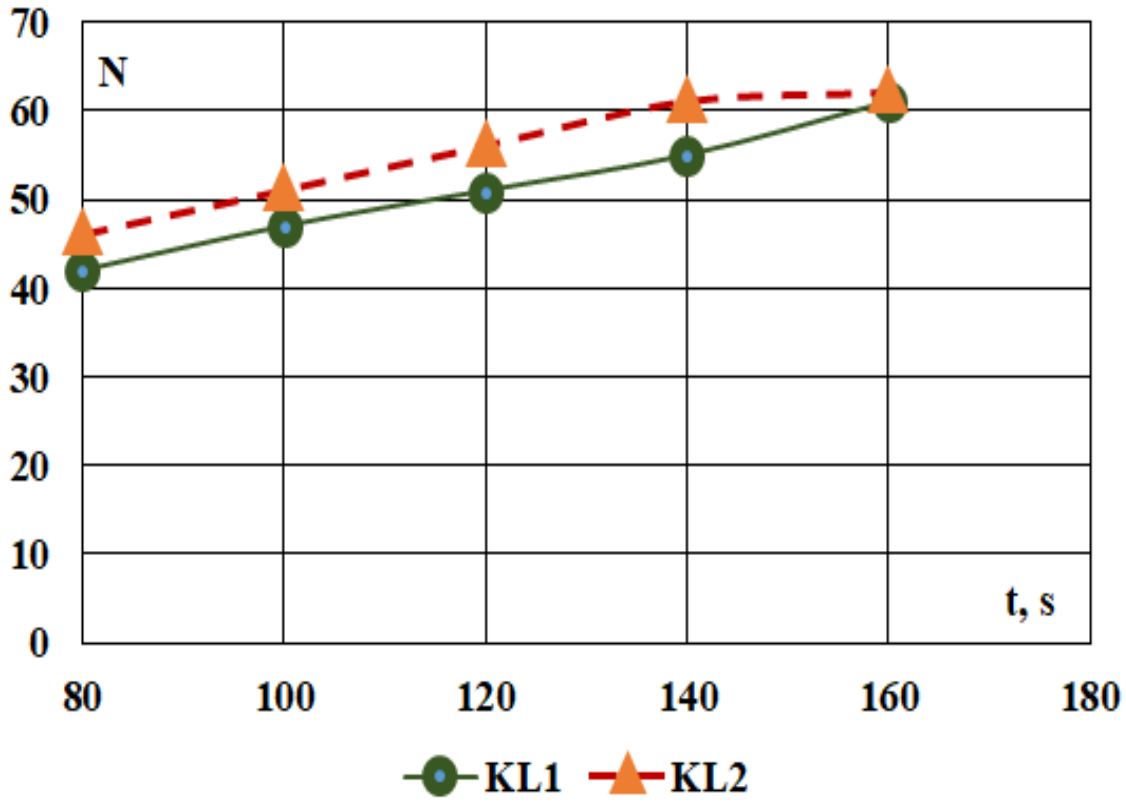
$rev \cdot \cdot \zeta \cdot \cdot \lvert Tr \rvert$	t, s	Вероятность обнаружения потенциально опасных субъектов информационно-вычислительной сети									
		Стандартная аутентификация субъекта в информационно-вычислительной сети, $\cdot 100\%$		Аутентификации субъекта в информационно-вычислительной сети на основе обновляемых набор для детектирования угрозы, $\cdot 100\%$							
		$\alpha = 0{,}1$	$\alpha = 0{,}9$	$\alpha = 0{,}1$				$\alpha = 0{,}9$			
				$P_m = 0{,}7$		$P_m = 0{,}98$		$P_m = 0{,}7$		$P_m = 0{,}98$	
				ξ							
				0,1	0,3	0,1	0,3	0,1	0,3	0,1	0,3
1	50	59,2	0,2	100	99,9	99,3	99,2	77,2	71	50,3	45,4
	160	59,2	0,1	60,1	59,9	59,1	59,1	0,8	0,4	0,2	0,1
10	50	61,3	0,4	99,9	98,1	98,4	96	77,3	70,9	50,3	49,8
	160	60,1	0,4	60, 2	59,1	60,2	59,7	0,6	0,3	0,3	0,1

Примечания: P_m – вероятность того, что ошибочно будут отождествляться данные набор для детектирования угрозы и данные, которые представлены субъектом. (пороговый предел схожести наборов определен заранее); ξ – коэффициент, характеризующий возможность применять результаты проверки на основе конкретного набор для детектирования угрозы к нескольким подвидам угроз.

Таким образом, было показано, что даже если злоумышленник узнал пароль, ему сложно подделать машинную манеру авторизованного абонента, обязательно вести за ним долгое время наблюдение и затем тренироваться информационно-вычислительной сети. Следовательно, предложенная схема аутентификации субъекта в информационно-вычислительной сети на основе

обновляемых набор для детектирования угрозы, достаточно эффективна для задач идентификации абонента в системе.

К достоинствам исследования можно отнести тот факт, что предложенные решения, в частности, разработанные программные модули для аутентификации по сравнению с результатами исследований, представленных в работах, показали большую вероятность обнаружения потенциально опасных субъектов в информационных системах и сетях предприятий, и меньшую вероятность того, что набор для детектирования угрозы ошибочно будут отождествляться с данными предоставленными абонентом сети.



*KL*1 – аутентификация субъекта в процессе клавиатурного распознавания в информационно-вычислительной сети на основе применения обновляемых набор для детектирования угрозы; *KL*2 – защита с помощью обычных паролей для субъекта; N – количество переходов при работе субъекта в информационно-вычислительной сети с клавиатуры.

Рисунок 2.11 – Результаты тестирования, предложенной схемы аутентификации для случая задачи анализа поведения субъекта в информационно-вычислительной сети (на примере системы дистанционного обучения)

Созданные на основе предложенных решений программные продукты, позволили автоматизировать контроль, сопровождение и изменение учётных записей абонентов сетей двух крупных предприятий г. Алматы. При этом в программном продукте «Анализатор угроз» была заложена возможность корректировать уровни доступа абонентов к информационным ресурсам и автоматизирована аутентификация пользователей в информационно-вычислительной сети.

Как компонент системы контроля и управления доступом реализован таймера рабочего времени, см. рис. 2.12, Приложение А. Таймер является структурным элементом любой системы управления и контроля доступа и применяется для многих прикладных задач, например, в подсистемах, которые учитывают время работы сотрудников или время их пребывания на рабочем месте. Кроме того, Таймер может использоваться и непосредственно в подсистеме контроля доступа, которая описана в следующем разделе магистерской работы, в частности в системе, контролирующей время пребывания сотрудника в той или иной зоне предприятия, куда вход ограничен.

Таймер написан на языке программирования C# (Visual Studio 2019). Звуковые функции таймера Beep(...) PlaySound(...) импортируются из библиотеки DllImport("Kernel32"), которая является базовой в Windows API. Звуки озвучивают события наступления отдыха и начало работы. Компонет NotifyIcon в виде анимации маятника часов визуализирует информацию работы таймера и выводят окно прошедшего времени работы на компьютере.

Листинг приложения Таймер, как составной части программы СУКД приведен в приложении А.

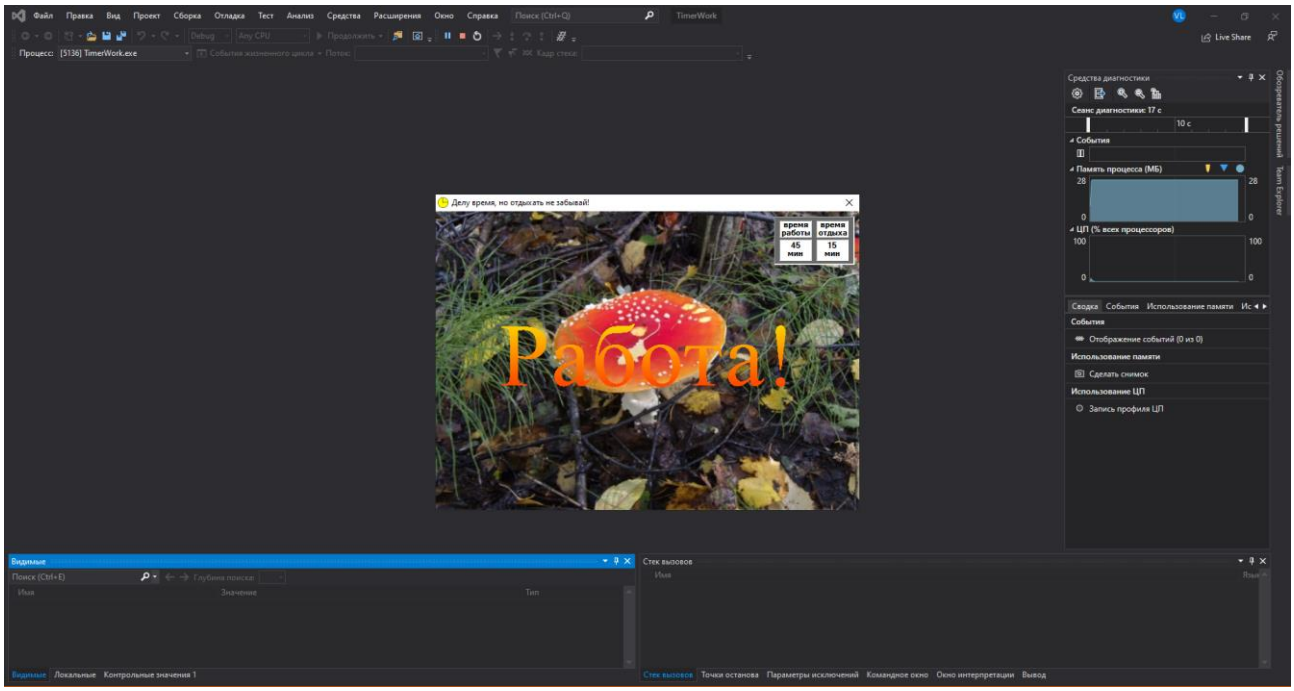


Рисунок 2.12 – Общий вид приложения таймер, которое входит в состав системы контроля и управления доступом

2.4 Выводы по разделу 2

В разделе 2 получены такие результаты:

На основе обзора и анализа научных публикаций, ведущих казахских и зарубежных ученных, занимающихся проектированием системы учета и управление доступом, описана концептуальная модель адаптивного управления киберзащитой информационно-вычислительной сети (ИВС);

рассмотрен пример решения задачи адаптивного управления правами доступа пользователей с использованием аппарата сетей Петри. Реализована соответствующая модель и выполнено имитационное моделирование в пакетах PIPE v4.3.0 и Petri.Net Simulator. 2.017;

Показаны возможности автоматизации процедур корректировки профиля пользователя для минимизации или нейтрализации киберугроз в информационно-вычислительной сети;

описана модель распределения задач, назначенных пользователями, в компьютерных сетях объектов информатизации. Базой для модели послужил математический аппарат сетей Петри. В отличие от существующих, модель содержит переменные, которые позволяют уменьшить мощность подпространства состояний. Также повысилась результативность моделирования, в частности за счет сокращения затрат времени на принятие решений, связанных с регламентацией прав доступа;

уточнен и дополнен метод контроля прав доступа (МКПД). Уточнения коснулись аспектов сверки прав доступа, которые запрашиваются задачей и требований политики безопасности. Кроме того, учитывалось согласованность задачи и разрешенных к доступу узлов информационно-вычислительной сети. Для узлов информационно-вычислительной сети также рассмотрена процедура сверки прав доступа для абонентов, имеющих соответствующие права. Модель учитывает текущие показатели политики безопасности для конкретных информационно-вычислительной сети и метрики безопасности с возможной корректировкой последних. Корректировка правил и метрик безопасности для новых задач или перераспределяемых задач описана в нотации сетей Петри.

3 ОСОБЕННОСТИ ПРОГРАММИРОВАНИЯ СИСТЕМЫ КОНТРОЛЯ ДОСТУПА К ПОМЕЩЕНИЯ

3.1 Создание макета

С самого начала разработки макета системы контроля и управления доступом встал вопрос: «Как передавать сигнал об открытии сетью?». Наш выбор остановился на локальном сервере. Его главным преимуществом является то, что он не зависит от подключения к глобальной сети. Все, что ему нужно для работы - небольшая локальная сеть, чтобы передавать данные HTTP-запросами в виде параметров. Локальный сервер реализован на базе программного обеспечения Denwer. Используются утилиты PHPMyAdmin (включающий в себя MySQL) и интерпретатор PHP 5.4.

Вся логическая часть проекта находится на сервере и написана скриптовым языком программирования PHP. Также в проекте используется JavaScript и его Фреймворк jQuery, они используются для создания уникального идентификатора пользователя и его проверки со значениями, которые находятся в базе данных. Поскольку мы имеем дело с локальным сервером, то необходимо быть подключенным к одной сети, доступ к которой нам обеспечивает роутер TP-LINK Archer C50.

Данные передаются протоколом HTTP, то есть микроконтроллер должен иметь подключение к беспроводной сети, в нашем случае обеспечивает модуль ESP-12E.

Исходя из данных потребностей, основой стал микроконтроллер NodeMCU v.1.0 Wi-Fi (ESP-12E). Он сочетает в себе компактность, встроенный модуль ESP-12E и простоту в использовании.

Микроконтроллер NodeMCU v.1.0 Wi-Fi (ESP-12E) (см. Рис. 3.1).

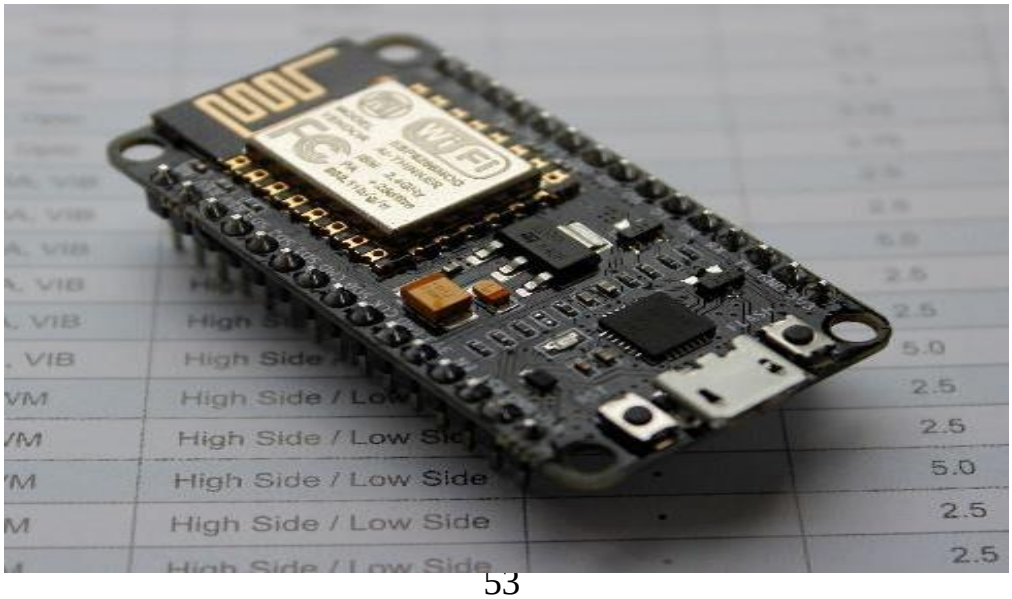


Рисунок 3.1 – Питание и интерфейс подключения к ПК: Micro-USB

Технические спецификации:
Разработчик: ESP8266 Open source Community
Тип: Одноплатный микроконтроллер
Операционная система: XTOS
Центральный процессор: ESP8266
FLASH-память: 128 kBytes
Хранилище: 4 MBytes

Схема цифровых выходов на микроконтроллере NodeMCU представлена в табл.3.1.

Таблица 3.1 – Схема цифровых выходов на NodeMCU

I/O index	**ESP8266 pin**
0 [*]	GPIO16
1	GPIO5
2	GPIO4
3	GPIO0
4	GPIO2
5	GPIO14
6	GPIO12
7	GPIO13
8	GPIO15
9	GPIO3
10	GPIO1
11	GPIO9
12	GPIO10

Разработка скетча осуществлялась в программе Arduino IDE, на языке программирования C.

Листинг программы представлен в приложении А.

Плата самостоятельно сканирует все Wi-Fi сети, подключается к сети, данные которой указаны в скетчи, выводит данные о статусе подключения в COM-порт и запускает сервер, который способен обрабатывать HTTP-протокол. Программа самостоятельно обновляет данные о взаимодействии с клиентскими подключениями и сверяет аргументы, которые передаются в HTTP-протоколе.

Конечной целью в нашем макете есть зажигания светодиода. По умолчанию он горит красным, но, когда приходит сигнал о том, что «двери должны быть открытыми», светодиод начинает гореть зеленым цветом в течение пяти секунд, что символизирует размыкания электромагнитного замка.

3.2 Алгоритм работы

Пользователь, впервые пытается получить доступ к объекту - сканирует QR-код необходимых дверей (см. рис. 3.2).

Рисунок 3.2 – QR-коды (Дверь 1, Дверь 2, соответственно)

Данные в QR-коды записаны так:

192.168.1.107/open.php?door=1

192.168.1.107/ - обращение к коренной папке локального сервера
open.php - обращение к файлу open.php
? Door = 1 - передача значения параметра door, отвечающий за ID двери
Его перенаправят на страницу, содержащую файл index.php (см. рис. 3.3)

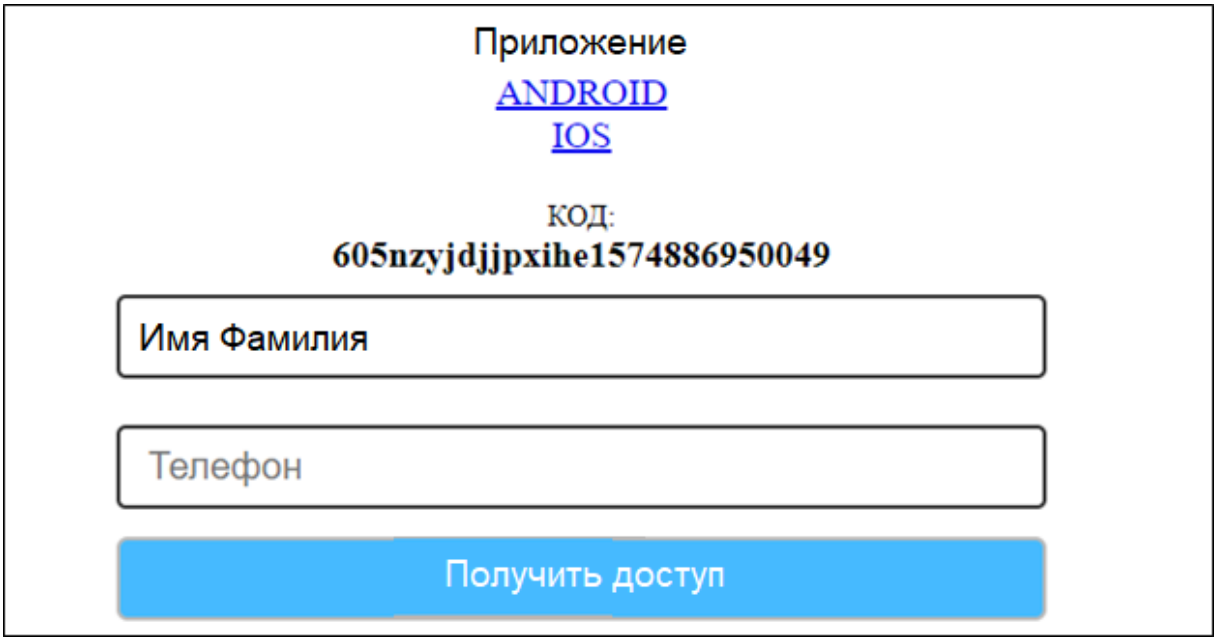


Рис. 3.3. Регистрационная форма

В браузере пользователя появляется запись cookie, содержащий уникальный идентификатор пользователя (код). На этой странице пользователь указывает свои персональные данные и нажимает кнопку «Получить доступ». После чего создается запрос в локальную базу данных, право предоставить доступ пользователю имеет только администратор.

Панель администратора представлена на рис. 3.4:

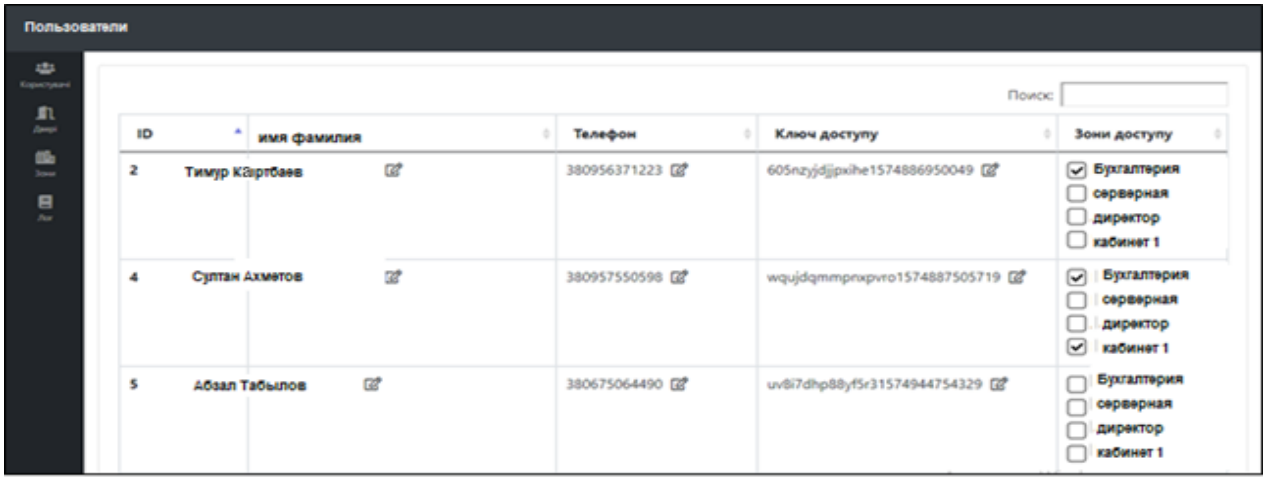


Рисунок 3.4 – Панель администратора

В QR-коде, привязан к двери, находится запрос в файл open.php, который обрабатывает данные с cookies и проверяет доступ пользователя к зоне.

Если пользователю разрешен проход в данную зону, то происходит перенаправление на файл success.php, который в свою очередь создает HTTP-запрос методом POST на микроконтроллер NodeMCU, или сообщает об ошибке в случае, если в пользователь не имеет права прохода в данную зону.

Контроллер сверяет аргумент, который находится в запросе и если все правильно, то плата зажигает зеленый цвет светодиода.

Все действия, совершенные авторизованными пользователями, записываются в базу данных в виде коротких логов с пометкой времени, что делает процесс мониторинга намного легче.

В среднем для идентификации одного пользователя требуется 3 секунды с момента сканирования QR-кода и к зажиганию светодиода, что свидетельствует о высокой пропускной способности созданной системы (тесты проводились в браузере Google Chrome). Это значение зависит от быстродействия устройства и установленного браузера.

Безопасность системы не подлежит сомнению. Все данные, находящиеся на сервере - защищены от SQL-инъекций, панель администратора защищена паролем, а запрос на плату осуществляется с передачей трех параметров (если в системе наглядно несколько дверей, то параметров - 4), данные о которых находятся на сервере.

При прямом обращении к локальному серверу или плате, пользователь перенаправляет на другие страницы.

Поскольку основой проекта является веб-сайт, то нам удалось создать удобный рабочий интерфейс, используя базовые средства разработки (см. рис. 3.5).

Логирование в нашем макете реализовано очень практично и лаконично.

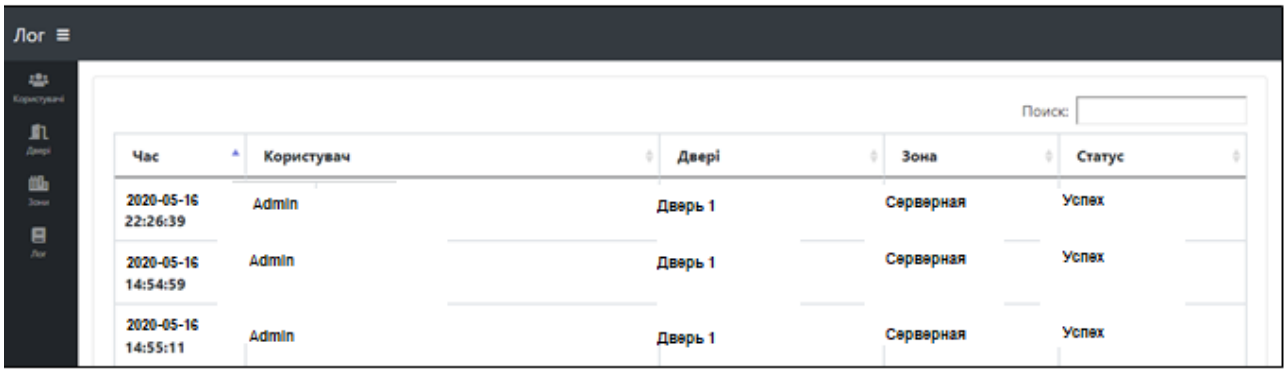


Рисунок 3.5 – Лог

Было проведено исследование скорости работы программы и сервера в самых популярных браузерах на разных платформах: персональный компьютер и смартфон. Результаты представлены на рис. 3.6.

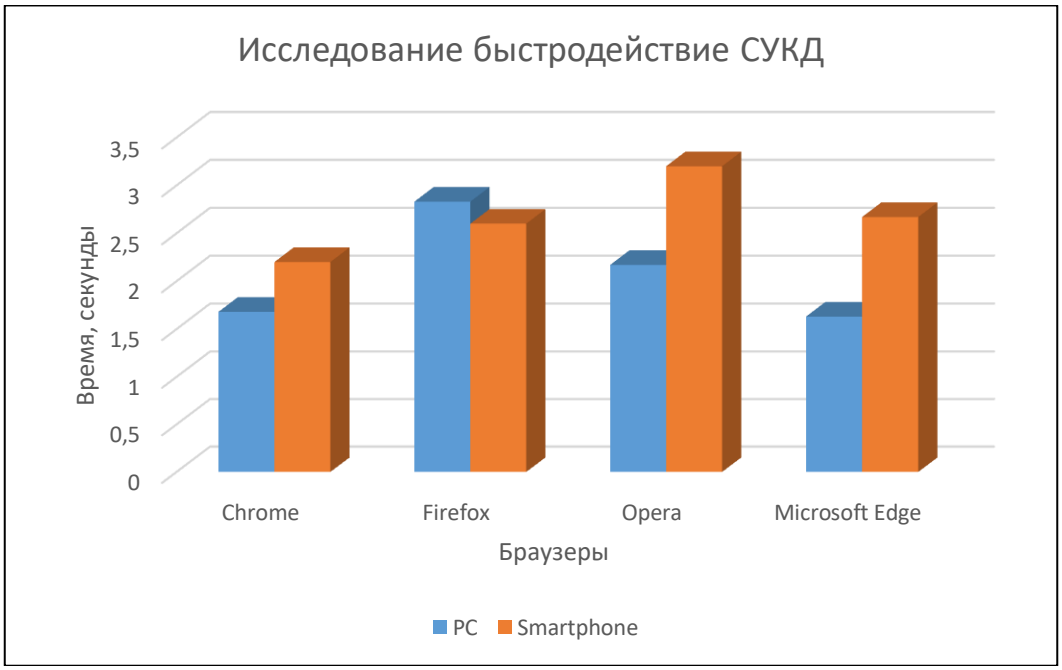


Рисунок 3.6 – Сравнение быстродействия СУКД на компьютере и смартфоне

В среднем выполнения полного цикла программы занимает больше времени при работе со смартфона.

Скорость выполнения зависит от качества Интернет подключения и браузера, а именно его версии и количества кэша, cookies.

Скорость удовлетворительная при использовании любого браузера из вышеупомянутых, код работает исправно.

3.3 Выводы по разделу 3

Результатом проведенного исследования является решение проблемных вопросов, связанных с разработкой и внедрением системы контроля и управления доступом в условиях тотальной цифровизации.

В результате проведенного исследования сформулированы следующие выводы и предложения:

1. Система контроля и управления доступом - чрезвычайно эффективное средство поддержания безопасности на различных объектах цифровизации. Автоматизация контроля позволяет сократить расходы на охрану и охранные пункты, позволяет поднять рентабельность предприятия, а без нее сетевого варианта трудно представить работу многих корпораций. Таким образом,

внедрения системы контроля и управления доступом - это очень выгодная инвестиция в будущее предприятия.

2. Разработан и испытан прототип конкурентоспособной системы контроля и управления доступом. Предложенное решение может быть реализовано с применением любого устройства, имеющего браузер и возможность подключения к беспроводной сети Wi-Fi. Разработанная система контроля и управления доступом не имеет аналогов на рынке Казахстана.

3. Предложенный подход, в отличие от существующих, учитывает современные тенденции тотальной цифровизации бизнес-процессов на предприятии. Используя минимальные ресурсы, предприятия могут имплементировать эффективную систему контроля и управления доступом в своих зонах защиты информационных ресурсов.

ЗАКЛЮЧЕНИЕ

В результате выполнения магистерской работы были получены такие результаты:

1) описана концептуальная модель адаптивного управления киберзащитой информационно-вычислительной сети, а также предложены различные структурные компоненты для реализации системы контроля и управления доступом (СКУД);

2) рассмотрен пример решения задачи адаптивного управления правами доступа пользователей с использованием аппарата сетей Петри. Реализована соответствующая модель и выполнено имитационное моделирование в пакетах PIPE v4.3.0 и Petri.Net Simulator. 2.017;

3) показаны возможности автоматизации процедур корректировки профиля пользователя для минимизации или нейтрализации киберугроз в информационно-вычислительной сети;

4) описана модель распределения задач, назначенных пользователями, в компьютерных сетях объектов информатизации. Базой для модели послужил математический аппарат сетей Петри. В отличие от существующих, модель содержит переменные, которые позволяют уменьшить мощность подпространства состояний. Также повысилась результативность моделирования, в частности за счет сокращения затрат времени на принятие решений, связанных с регламентацией прав доступа;

5) уточнен и дополнен метод контроля прав доступа (МКПД). Уточнения коснулись аспектов сверки прав доступа, которые запрашиваются задачей и требований политики безопасности. Кроме того, учитывалось согласованность задачи и разрешенных к доступу узлов информационно-вычислительной сети. Для узлов информационно-вычислительной сети также рассмотрена процедура сверки прав доступа для абонентов, имеющих соответствующие права. Модель учитывает текущие показатели политики безопасности для конкретных информационно-вычислительной сети и метрики безопасности с возможной корректировкой последних. Корректировка правил и метрик безопасности для новых задач или перераспределяемых задач описана в нотации сетей Петри.

6) разработан и испытан прототип конкурентоспособной системы контроля и управления доступом. Предложенное решение может быть реализовано с применением любого устройства, имеющего браузер и возможность подключения к беспроводной сети Wi-Fi. Разработанная системы контроля и управления доступом не имеет аналогов на рынке Казахстана.

7) Показано, что подложенный подход, в отличие от существующих, учитывает современные тенденции тотальной цифровизации бизнес-процессов на предприятии. Используя минимальные ресурсы, предприятия

могут имплементировать эффективную систему контроля и управления доступом в своих зонах защиты информационных ресурсов.

ЛИТЕРАТУРА

4. Gupta, Brij, Dharma P. Agrawal, and Shingo Yamaguchi, eds. Handbook of research on modern cryptographic solutions for computer and cyber security. IGI Global, 2016.

5. Liu, X., Zhu, P., Zhang, Y., & Chen, K. (2015). A collaborative intrusion detection mechanism against false data injection attack in advanced metering infrastructure. IEEE Transactions on Smart Grid, 6(5), pp. 2435–2443.

6. Jasiul, B., Szpyrka, M., & Śliwa, J. (2014). Detection and modeling of cyber attacks with Petri nets. Entropy, 16(12), pp. 6602–6623.

7. Liu, X., Zhang, J., & Zhu, P. (2017). Modeling cyber-physical attacks based on probabilistic colored Petri nets and mixed-strategy game theory. International Journal of Critical Infrastructure Protection, 16, pp. 13–25.

8. Jasiul, B., Szpyrka, M., & Śliwa, J. (2015). Formal specification of malware models in the form of colored Petri nets. In Computer Science and its Applications (pp. 475–482). Springer, Berlin, Heidelberg.

9. Akhmetov, B., Lakhno, V., Boiko, Y., & Mishchenko, A. (2017). Designing a decision support system for the weakly formalized problems in the provision of cybersecurity. Eastern-European Journal of Eenterprise Technologies, (1(2)), pp. 4–15.

10. Arendt, D. L., Burtner, R., Best, D. M., Bos, N. D., Gersh, J. R., Piatko, C. D., & Paul, C. L. (2015, October). Ocelot: user-centered design of a decision support visualization for network quarantine. In Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on (pp. 1–8). IEEE.

11. Alheeti, K. M. A., Gruebler, A., McDonald-Maier, K. D., & Fernando, A. (2016, January). Prediction of DoS attacks in external communication for self-driving vehicles using a fuzzy petri net model. In Consumer Electronics (ICCE), 2016 IEEE International Conference on (pp. 502–503). IEEE.

12. de Carvalho, M. A., & Bandiera-Paiva, P. (2017, October). Evaluating ISO 14441 privacy requirements on role based access control (RBAC) restrict mode via Colored Petri Nets (CPN) modeling. In Security Technology (ICCST), 2017 International Carnahan Conference on (pp. 1–8). IEEE.

13. Appel, M., Konigorski, U., & Walther, M. (2018). A Graph Metric for Model Predictive Control of Petri Nets. IFAC-PapersOnLine, 51(2), pp. 254–259.

14. Gao, Z., Zhao, C., Shang, C., & Tan, C. (2017, October). The optimal control of mine drainage systems based on hybrid Petri nets. In Chinese Automation Congress (CAC), 2017 (pp. 78–83). IEEE.

15. Narayanan, M., & Cherukuri, A. K. (2018). Verification of Cloud Based Information Integration Architecture using Colored Petri Nets. International Journal of Computer Network and Information Security, 10(2), 1.

16. V. A. Lakhno, Y. N. Tkach, T.A. Petrenko, S.V. Zaitsev, V. M. Bazylevych. Development of adaptive expert system of information security using a procedure of clustering the attributes of anomalies and cyber attacks, Eastern-European Journal of Enterprise Technologies, No 6/9 (84), 2016, pp. 32–44.

17. G. Beketova, B. Akhmetov, A. Korchenko, V, Lakhno, A. Tereshuk. Cyber intelligence systems based on adaptive regression splines and logical procedures of attack recognition. Computer modelling and new technologies, Vol. 21, No. 2, 2017, pp. 7–16.

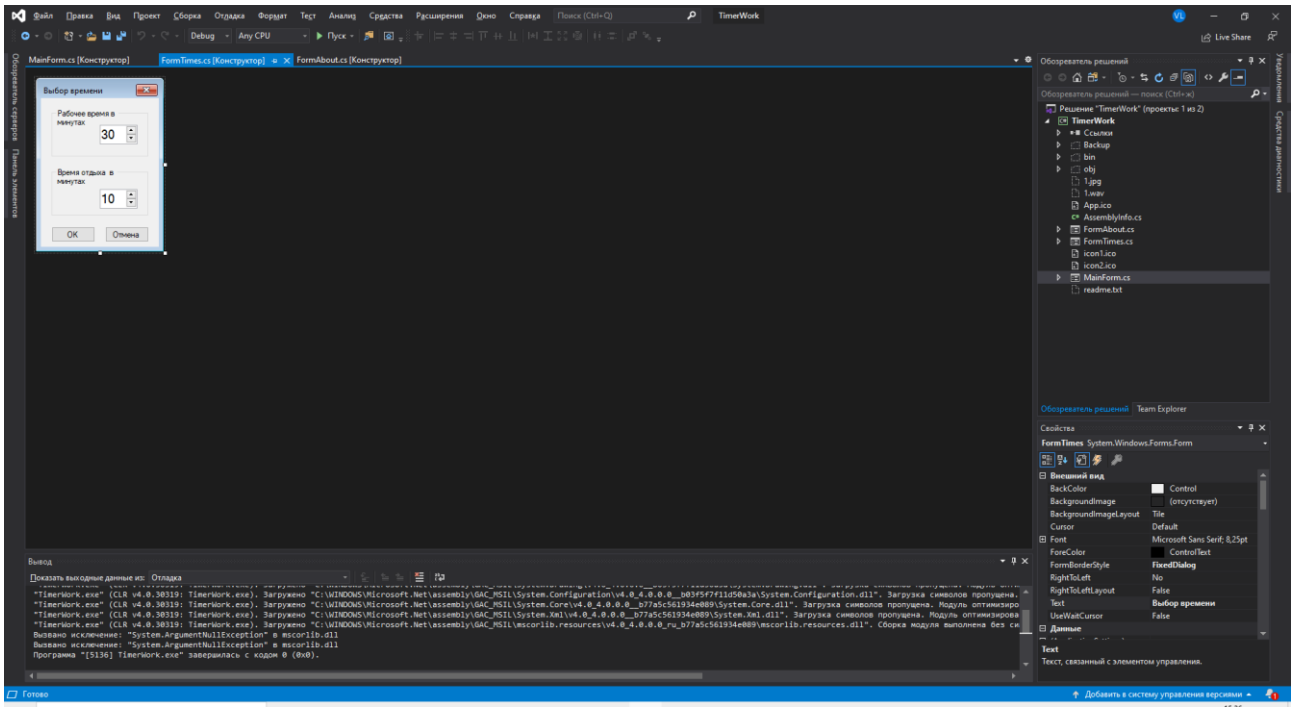
18. Lakhno V., Petrov Al., Petrov Ant. Development of a Support System for Managing the Cyber Security of Information and Communication Environment of Transport, Information Systems Architecture and Technology: 38th International Conference on Information Systems Architecture and Technology (ISAT 2017), Wroclaw, 17–19 September 2017 : proceedings, Wroclaw : Springer, 2017, pp. 113–127.

19. Lakhno, V., Buriachok, V., Parkhuts, L., Tarasova, H., Kydyralina, L., Skladannyi, P., ... & Shostakovska, A. (2018). Development of a conceptual model of adaptive access rights management with using the apparatus of Petri nets. *International Journal of Civil Engineering & Technology (IJCIET)*, *9*(11), 95-104.

ПРИЛОЖЕНИЯ

Приложение А

Листинг программы таймер, которое является частью программы контроля доступа




```
////////////////////////////////////////////////////////////////////////////////
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;

namespace TimerWork
{
        /// <summary>
        /// Summary description for FormTimes.
```

```
/// </summary>
public class FormTimes : System.Windows.Forms.Form
{
        private System.Windows.Forms.GroupBox groupBoxTimeWork;
        private System.Windows.Forms.GroupBox groupBoxTimeRest;
        private System.Windows.Forms.NumericUpDown
numericUpDownTimeWork;
        private System.Windows.Forms.NumericUpDown
numericUpDownTimeRest;
        private System.Windows.Forms.Button buttonOK;
        private System.Windows.Forms.Button buttonCancel;
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.Container components = null;

        public FormTimes()
        {
                //
                // Required for Windows Form Designer support
                //
                InitializeComponent();

                //
                // TODO: Add any constructor code after InitializeComponent
call
                //
        }
```

```
/// <summary>
/// Clean up any resources being used.
/// </summary>
protected override void Dispose( bool disposing )
{
	if( disposing )
	{
		if(components != null)
		{
			components.Dispose();
		}
	}
	base.Dispose( disposing );
}

#region Windows Form Designer generated code
/// <summary>
/// Required method for Designer support - do not modify
/// the contents of this method with the code editor.
/// </summary>
private void InitializeComponent()
{
	System.Resources.ResourceManager resources = new
System.Resources.ResourceManager(typeof(FormTimes));
	this.groupBoxTimeWork = new
System.Windows.Forms.GroupBox();
```

```
this.numericUpDownTimeWork = new
System.Windows.Forms.NumericUpDown();
this.groupBoxTimeRest = new
System.Windows.Forms.GroupBox();
this.numericUpDownTimeRest = new
System.Windows.Forms.NumericUpDown();
this.buttonOK = new System.Windows.Forms.Button();
this.buttonCancel = new System.Windows.Forms.Button();
this.groupBoxTimeWork.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numericUpDownTimeW
ork)).BeginInit();
this.groupBoxTimeRest.SuspendLayout();

((System.ComponentModel.ISupportInitialize)(this.numericUpDownTimeRe
st)).BeginInit();
this.SuspendLayout();
//
// groupBoxTimeWork
//
this.groupBoxTimeWork.AccessibleDescription =
resources.GetString("groupBoxTimeWork.AccessibleDescription");
this.groupBoxTimeWork.AccessibleName =
resources.GetString("groupBoxTimeWork.AccessibleName");
this.groupBoxTimeWork.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("groupBoxTimeWor
k.Anchor")));
```

```
                this.groupBoxTimeWork.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("groupBoxTimeWork.BackgroundI
mage")));

        this.groupBoxTimeWork.Controls.Add(this.numericUpDownTimeWork);
                this.groupBoxTimeWork.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("groupBoxTimeWork.
Dock")));
                this.groupBoxTimeWork.Enabled =
((bool)(resources.GetObject("groupBoxTimeWork.Enabled")));
                this.groupBoxTimeWork.Font =
((System.Drawing.Font)(resources.GetObject("groupBoxTimeWork.Font")));
                this.groupBoxTimeWork.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("groupBoxTimeWork.I
meMode")));
                this.groupBoxTimeWork.Location =
((System.Drawing.Point)(resources.GetObject("groupBoxTimeWork.Location")));
                this.groupBoxTimeWork.Name = "groupBoxTimeWork";
                this.groupBoxTimeWork.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("groupBoxTimeWork
.RightToLeft")));
                this.groupBoxTimeWork.Size =
((System.Drawing.Size)(resources.GetObject("groupBoxTimeWork.Size")));
                this.groupBoxTimeWork.TabIndex =
((int)(resources.GetObject("groupBoxTimeWork.TabIndex")));
                this.groupBoxTimeWork.TabStop = false;
                this.groupBoxTimeWork.Text =
resources.GetString("groupBoxTimeWork.Text");
```

```
this.groupBoxTimeWork.Visible =
((bool)(resources.GetObject("groupBoxTimeWork.Visible")));
//
// numericUpDownTimeWork
//
this.numericUpDownTimeWork.AccessibleDescription =
resources.GetString("numericUpDownTimeWork.AccessibleDescription");
this.numericUpDownTimeWork.AccessibleName =
resources.GetString("numericUpDownTimeWork.AccessibleName");
this.numericUpDownTimeWork.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("numericUpDownTi
meWork.Anchor")));
this.numericUpDownTimeWork.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("numericUpDownTime
Work.Dock")));
this.numericUpDownTimeWork.Enabled =
((bool)(resources.GetObject("numericUpDownTimeWork.Enabled")));
this.numericUpDownTimeWork.Font =
((System.Drawing.Font)(resources.GetObject("numericUpDownTimeWork.Font")))
;
this.numericUpDownTimeWork.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("numericUpDownTime
Work.ImeMode")));
this.numericUpDownTimeWork.Location =
((System.Drawing.Point)(resources.GetObject("numericUpDownTimeWork.Locati
on")));
this.numericUpDownTimeWork.Maximum = new
System.Decimal(new int[] {120, 0, 0, 0});
```

```
this.numericUpDownTimeWork.Minimum = new
System.Decimal(new int[] {30, 0, 0, 0});
this.numericUpDownTimeWork.Name =
"numericUpDownTimeWork";
this.numericUpDownTimeWork.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("numericUpDownTi
meWork.RightToLeft")));
this.numericUpDownTimeWork.Size =
((System.Drawing.Size)(resources.GetObject("numericUpDownTimeWork.Size")))
;
this.numericUpDownTimeWork.TabIndex =
((int)(resources.GetObject("numericUpDownTimeWork.TabIndex")));
this.numericUpDownTimeWork.TextAlign =
((System.Windows.Forms.HorizontalAlignment)(resources.GetObject("numericUp
DownTimeWork.TextAlign")));
this.numericUpDownTimeWork.ThousandsSeparator =
((bool)(resources.GetObject("numericUpDownTimeWork.ThousandsSeparator")));
this.numericUpDownTimeWork.UpDownAlign =
((System.Windows.Forms.LeftRightAlignment)(resources.GetObject("numericUpD
ownTimeWork.UpDownAlign")));
this.numericUpDownTimeWork.Value = new
System.Decimal(new int[] {30, 0, 0, 0});
this.numericUpDownTimeWork.Visible =
((bool)(resources.GetObject("numericUpDownTimeWork.Visible")));
//
// groupBoxTimeRest
//
```

```
this.groupBoxTimeRest.AccessibleDescription =
resources.GetString("groupBoxTimeRest.AccessibleDescription");
this.groupBoxTimeRest.AccessibleName =
resources.GetString("groupBoxTimeRest.AccessibleName");
this.groupBoxTimeRest.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("groupBoxTimeRest
.Anchor")));
this.groupBoxTimeRest.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("groupBoxTimeRest.BackgroundIm
age")));

this.groupBoxTimeRest.Controls.Add(this.numericUpDownTimeRest);
this.groupBoxTimeRest.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("groupBoxTimeRest.D
ock")));
this.groupBoxTimeRest.Enabled =
((bool)(resources.GetObject("groupBoxTimeRest.Enabled")));
this.groupBoxTimeRest.Font =
((System.Drawing.Font)(resources.GetObject("groupBoxTimeRest.Font")));
this.groupBoxTimeRest.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("groupBoxTimeRest.Im
eMode")));
this.groupBoxTimeRest.Location =
((System.Drawing.Point)(resources.GetObject("groupBoxTimeRest.Location")));
this.groupBoxTimeRest.Name = "groupBoxTimeRest";
this.groupBoxTimeRest.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("groupBoxTimeRest.
RightToLeft")));
```

```
this.groupBoxTimeRest.Size =
((System.Drawing.Size)(resources.GetObject("groupBoxTimeRest.Size")));
this.groupBoxTimeRest.TabIndex =
((int)(resources.GetObject("groupBoxTimeRest.TabIndex")));
this.groupBoxTimeRest.TabStop = false;
this.groupBoxTimeRest.Text =
resources.GetString("groupBoxTimeRest.Text");
this.groupBoxTimeRest.Visible =
((bool)(resources.GetObject("groupBoxTimeRest.Visible")));
//
// numericUpDownTimeRest
//
this.numericUpDownTimeRest.AccessibleDescription =
resources.GetString("numericUpDownTimeRest.AccessibleDescription");
this.numericUpDownTimeRest.AccessibleName =
resources.GetString("numericUpDownTimeRest.AccessibleName");
this.numericUpDownTimeRest.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("numericUpDownTi
meRest.Anchor")));
this.numericUpDownTimeRest.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("numericUpDownTime
Rest.Dock")));
this.numericUpDownTimeRest.Enabled =
((bool)(resources.GetObject("numericUpDownTimeRest.Enabled")));
this.numericUpDownTimeRest.Font =
((System.Drawing.Font)(resources.GetObject("numericUpDownTimeRest.Font")));
```

```
this.numericUpDownTimeRest.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("numericUpDownTime
Rest.ImeMode")));
this.numericUpDownTimeRest.Location =
((System.Drawing.Point)(resources.GetObject("numericUpDownTimeRest.Locatio
n")));
this.numericUpDownTimeRest.Maximum = new
System.Decimal(new int[] {120, 0, 0, 0});
this.numericUpDownTimeRest.Minimum = new
System.Decimal(new int[] {10, 0, 0, 0});
this.numericUpDownTimeRest.Name =
"numericUpDownTimeRest";
this.numericUpDownTimeRest.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("numericUpDownTi
meRest.RightToLeft")));
this.numericUpDownTimeRest.Size =
((System.Drawing.Size)(resources.GetObject("numericUpDownTimeRest.Size")));
this.numericUpDownTimeRest.TabIndex =
((int)(resources.GetObject("numericUpDownTimeRest.TabIndex")));
this.numericUpDownTimeRest.TextAlign =
((System.Windows.Forms.HorizontalAlignment)(resources.GetObject("numericUp
DownTimeRest.TextAlign")));
this.numericUpDownTimeRest.ThousandsSeparator =
((bool)(resources.GetObject("numericUpDownTimeRest.ThousandsSeparator")));
this.numericUpDownTimeRest.UpDownAlign =
((System.Windows.Forms.LeftRightAlignment)(resources.GetObject("numericUpD
ownTimeRest.UpDownAlign")));
```

```
this.numericUpDownTimeRest.Value = new
System.Decimal(new int[] {10, 0, 0, 0});
this.numericUpDownTimeRest.Visible =
((bool)(resources.GetObject("numericUpDownTimeRest.Visible")));
//
// buttonOK
//
this.buttonOK.AccessibleDescription =
resources.GetString("buttonOK.AccessibleDescription");
this.buttonOK.AccessibleName =
resources.GetString("buttonOK.AccessibleName");
this.buttonOK.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("buttonOK.Anchor")
));
this.buttonOK.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("buttonOK.BackgroundImage")));
this.buttonOK.DialogResult =
System.Windows.Forms.DialogResult.OK;
this.buttonOK.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("buttonOK.Dock")));
this.buttonOK.Enabled =
((bool)(resources.GetObject("buttonOK.Enabled")));
this.buttonOK.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("buttonOK.FlatStyle")));
this.buttonOK.Font =
((System.Drawing.Font)(resources.GetObject("buttonOK.Font")));
this.buttonOK.Image =
((System.Drawing.Image)(resources.GetObject("buttonOK.Image")));
```

```
this.buttonOK.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("buttonOK.ImageAlign
")));
this.buttonOK.ImageIndex =
((int)(resources.GetObject("buttonOK.ImageIndex")));
this.buttonOK.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("buttonOK.ImeMode")))
;
this.buttonOK.Location =
((System.Drawing.Point)(resources.GetObject("buttonOK.Location")));
this.buttonOK.Name = "buttonOK";
this.buttonOK.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("buttonOK.RightToL
eft")));
this.buttonOK.Size =
((System.Drawing.Size)(resources.GetObject("buttonOK.Size")));
this.buttonOK.TabIndex =
((int)(resources.GetObject("buttonOK.TabIndex")));
this.buttonOK.TabStop = false;
this.buttonOK.Text = resources.GetString("buttonOK.Text");
this.buttonOK.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("buttonOK.TextAlign")
));
this.buttonOK.Visible =
((bool)(resources.GetObject("buttonOK.Visible")));
//
// buttonCancel
//
```

```
this.buttonCancel.AccessibleDescription =
resources.GetString("buttonCancel.AccessibleDescription");
this.buttonCancel.AccessibleName =
resources.GetString("buttonCancel.AccessibleName");
this.buttonCancel.Anchor =
((System.Windows.Forms.AnchorStyles)(resources.GetObject("buttonCancel.Anch
or")));
this.buttonCancel.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("buttonCancel.BackgroundImage"))
);
this.buttonCancel.DialogResult =
System.Windows.Forms.DialogResult.Cancel;
this.buttonCancel.Dock =
((System.Windows.Forms.DockStyle)(resources.GetObject("buttonCancel.Dock")))
;
this.buttonCancel.Enabled =
((bool)(resources.GetObject("buttonCancel.Enabled")));
this.buttonCancel.FlatStyle =
((System.Windows.Forms.FlatStyle)(resources.GetObject("buttonCancel.FlatStyle"
)));
this.buttonCancel.Font =
((System.Drawing.Font)(resources.GetObject("buttonCancel.Font")));
this.buttonCancel.Image =
((System.Drawing.Image)(resources.GetObject("buttonCancel.Image")));
this.buttonCancel.ImageAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("buttonCancel.ImageAl
ign")));
```

```
this.buttonCancel.ImageIndex =
((int)(resources.GetObject("buttonCancel.ImageIndex")));
this.buttonCancel.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("buttonCancel.ImeMode
")));
this.buttonCancel.Location =
((System.Drawing.Point)(resources.GetObject("buttonCancel.Location")));
this.buttonCancel.Name = "buttonCancel";
this.buttonCancel.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("buttonCancel.RightT
oLeft")));
this.buttonCancel.Size =
((System.Drawing.Size)(resources.GetObject("buttonCancel.Size")));
this.buttonCancel.TabIndex =
((int)(resources.GetObject("buttonCancel.TabIndex")));
this.buttonCancel.TabStop = false;
this.buttonCancel.Text =
resources.GetString("buttonCancel.Text");
this.buttonCancel.TextAlign =
((System.Drawing.ContentAlignment)(resources.GetObject("buttonCancel.TextAlig
n")));
this.buttonCancel.Visible =
((bool)(resources.GetObject("buttonCancel.Visible")));
//
// FormTimes
//
this.AccessibleDescription =
resources.GetString("$this.AccessibleDescription");
```

```
this.AccessibleName =
resources.GetString("$this.AccessibleName");
this.AutoScaleBaseSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScaleBaseSize")));
this.AutoScroll =
((bool)(resources.GetObject("$this.AutoScroll")));
this.AutoScrollMargin =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMargin")));
this.AutoScrollMinSize =
((System.Drawing.Size)(resources.GetObject("$this.AutoScrollMinSize")));
this.BackgroundImage =
((System.Drawing.Image)(resources.GetObject("$this.BackgroundImage")));
this.ClientSize =
((System.Drawing.Size)(resources.GetObject("$this.ClientSize")));
this.Controls.Add(this.buttonOK);
this.Controls.Add(this.groupBoxTimeWork);
this.Controls.Add(this.groupBoxTimeRest);
this.Controls.Add(this.buttonCancel);
this.Enabled = ((bool)(resources.GetObject("$this.Enabled")));
this.Font =
((System.Drawing.Font)(resources.GetObject("$this.Font")));
this.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedDialog;
this.Icon =
((System.Drawing.Icon)(resources.GetObject("$this.Icon")));
this.ImeMode =
((System.Windows.Forms.ImeMode)(resources.GetObject("$this.ImeMode")));
```

```
                this.Location =
((System.Drawing.Point)(resources.GetObject("$this.Location")));
                this.MaximizeBox = false;
                this.MaximumSize =
((System.Drawing.Size)(resources.GetObject("$this.MaximumSize")));
                this.MinimizeBox = false;
                this.MinimumSize =
((System.Drawing.Size)(resources.GetObject("$this.MinimumSize")));
                this.Name = "FormTimes";
                this.RightToLeft =
((System.Windows.Forms.RightToLeft)(resources.GetObject("$this.RightToLeft"))
);
                this.ShowInTaskbar = false;
                this.StartPosition =
((System.Windows.Forms.FormStartPosition)(resources.GetObject("$this.StartPosit
ion")));
                this.Text = resources.GetString("$this.Text");
                this.TopMost = true;
                this.Load += new System.EventHandler(this.FormTimes_Load);
                this.Closed += new
System.EventHandler(this.FormTimes_Closed);
                this.groupBoxTimeWork.ResumeLayout(false);

        ((System.ComponentModel.ISupportInitialize)(this.numericUpDownTimeW
ork)).EndInit();
                this.groupBoxTimeRest.ResumeLayout(false);
```

```
((System.ComponentModel.ISupportInitialize)(this.numericUpDownTimeRest)).EndInit();
            this.ResumeLayout(false);
        }
        #endregion


        /// <summary>
        /// время работы в секундах
        /// </summary>
        public int timeWork;
        /// <summary>
        /// время отдыха в секундах
        /// </summary>
        public int timeRest;

        private void FormTimes_Closed(object sender, System.EventArgs e)
        {
            timeWork = (int)numericUpDownTimeWork.Value*60;
            timeRest = (int)numericUpDownTimeRest.Value*60;
        }

        private void FormTimes_Load(object sender, System.EventArgs e)
        {
            numericUpDownTimeWork.Value = timeWork/60;
            numericUpDownTimeRest.Value = timeRest/60;
        }
```

```
        }
}
```

Приложение Б

Листинг программы для системы контроля и управления доступом в помещения объекта информатизации

```
#include <ESP8266WiFi.h>
#include <ESP8266WebServer.h>
#include <ESP8266mDNS.h>
// подключение библиотек
#ifndef STASSID
#define STASSID «JolyMole»
#define STAPSK  «naruto31kurama» //подключение wifi
#define LED D0
#define red D5
#define green D6 //инициализация пинов

#define blue D7
#endif
const char* ssid = STASSID;
const char* password = STAPSK;
ESP8266WebServer server(80); //запуск сервера server на 80 порті
void HTTP_init(void){  //функция, возврата http-запроса на плату
  server.on(«/», handleRoot);
```

```
  server.on(«/action», handle_open);
  server.onNotFound(handleNotFound);
  server.begin();
}
void handle_open(){  //функция, которая реагирует на открытие двери
  String data = server.arg(«action»);
  if (data == «open»){
    server.send(200, «text/html»,
«<html><body><script>window.location.href=‘http://192.168.1.107/succes.php’</s
cript></body></html>«);
    digitalWrite(red, LOW);
    digitalWrite(green, HIGH);
    delay(5000);
    digitalWrite(green, LOW);
  }
digitalWrite(red, HIGH);
}
void handleRoot() {  //функция, которая реагирует на обращение платы к
локальному адресу server.send(200, «text/html»,
«<html><body><script>window.location.href=‘http://google.com’</script></body
></html>«);digitalWrite(red, LOW);
digitalWrite(blue, HIGH);
delay(5000);
digitalWrite(blue, LOW);
digitalWrite(red, HIGH);
}
void handleNotFound() {  //функция, которая вызывает исключения, если были
переданы неправильные аргументы
```

```
  String message = «File Not Found\n\n»;
  message += «URI: «;
  message += server.uri();
  message += «\nMethod: «;
  message += (server.method() == HTTP_GET) ? «GET» : «POST»;
  message += «\nArguments: «;
  message += server.args();
  message += «\n»;
  for (uint8_t i = 0; i < server.args(); i++) {
    message += « « + server.argName(i) + «: « + server.arg(i) + «\n»;
  }
  server.send(404, «text/plain», message);
}
void setup() {
  pinMode(red, OUTPUT);
  pinMode(green, OUTPUT);  //
  pinMode(blue, OUTPUT);
  digitalWrite(red, HIGH);
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  WiFi.disconnect(); // инициализация wi-fi модуля
  delay(100);
  Serial.println(«Setup done»);
  Serial.println(«scan start»);
  int n = WiFi.scanNetworks();
  Serial.println(«scan done»); //
  if (n == 0) {
    Serial.println(«no networks found»);
```

```
} else {
  Serial.print(n);
  Serial.println(« networks found»);
  for (int i = 0; i < n; ++i) {
  Serial.print(i + 1);
  Serial.print(«: «);
  Serial.print(WiFi.SSID(i));
  Serial.print(« («);
  Serial.print(WiFi.RSSI(i));
  Serial.print(«)»);
  Serial.println((WiFi.encryptionType(i) == ENC_TYPE_NONE) ? « « : «*»);
  delay(10);
  } // вывод SSID та RSSI для каждой сети
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
   delay(500);
   Serial.print(«*»);
}
 Serial.println(««);
Serial.println(«WiFi connection Successful»);
Serial.print(«The IP Address of ESP8266 Module is: «);
Serial.print(WiFi.localIP());// вывод IP address

 if (MDNS.begin(«esp8266»)) {
  Serial.println(««);
  Serial.println(«MDNS responder started»);
}
```

```
  server.begin();
  Serial.println(«server HTTP started»);
}
HTTP_init(); //вызов функции обработчика события
}
void loop() {
  server.handleClient(); //проверка обращения клиентов
  delay(1);
}
```