ISSN 2225-5435

Вестник УрФО.

БЕЗОПАСНОСТЬ В ИНФОРМАЦИОННОЙ СФЕРЕ

№ 3(61) / 2026

УЧРЕДИТЕЛИ
ФГАОУ ВО «ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИУ)»

ПРЕДСЕДАТЕЛЬ РЕДАКЦИОННОГО СОВЕТА
ЧУВАРДИН О. П.,
руководитель Управления Федеральной службы по техническому и экспортному контролю России по Уральскому федеральному округу

ГЛАВНЫЙ РЕДАКТОР
СОКОЛОВ А. Н.,
к. т. н., доцент, зав. кафедрой «Защита информации», Южно-Уральский государственный университет (национальный исследовательский университет) (г. Челябинск)

ВЫПУСКАЮЩИЙ РЕДАКТОР
СОГРИН Е. К.

ОТВЕТСТВЕННЫЙ СЕКРЕТАРЬ
АНДРИАДИС Е. Ю.

ВЁРСТКА
ШРАЙБЕР А. Е.

КОРРЕКТОР
ФЁДОРОВ В. С.

Подписной индекс 73852
в каталоге «Почта России»
Журнал зарегистрирован Федеральной службой по надзору в сфере связи, информационных технологий и массовых коммуникаций. Регистрационный номер ПИ № ФС77-91645 от 27.05.2026

Адрес редакции и издателя:
Россия, 454080, г. Челябинск, пр. Ленина, д. 76. ЮУрГУ
Тел./факс (351) 267-93-55.
Электронная версия журнала в Интернете:
www.info-secur.ru,
e-mail: urvest@mail.ru

РЕДАКЦИОННЫЙ СОВЕТ:

БАРАНКОВА И. И.,
д. т. н., профессор, зав. кафедрой «Информатика и информационная безопасность», Магнитогорский государственный технический университет им. Г. И. Носова (г. Магнитогорск);

ВАСИЛЬЕВ В. И.,
д. т. н., профессор, профессор кафедры «Вычислительная техника и защита информации», Уфимский государственный авиационный технический университет (г. Уфа);

ВОЙТОВИЧ Н. И.,
д. т. н., профессор, профессор кафедры «Радиоэлектроника и системы связи», Южно-Уральский государственный университет (национальный исследовательский университет) (г. Челябинск);

ГАЙДАМАКИН Н. А.,
д.т.н., профессор, профессор Учебно-научного центра «Информационная безопасность», Уральский федеральный университет им. первого президента России Б.Н. Ельцина (г. Екатеринбург);

ДИК Д. И.,
к. т. н., доцент, зав. кафедрой «Безопасность информационных и автоматизированных систем», Курганский государственный университет (г. Курган);

ЗАХАРОВ А. А.,
д.т.н., профессор, профессор школы компьютерных наук, Тюменский государственный университет (г. Тюмень);

ЗЫРЯНОВА Т. Ю.,
к. т. н., доцент, зав. кафедрой «Информационные технологии и защита информации», Уральский государственный университет путей сообщения (г. Екатеринбург);

МЕЛЬНИКОВ А. В.,
д. т. н., профессор, директор Югорского научно-исследовательского института информационных технологий (г. Ханты-Мансийск);

МИНБАЛЕЕВ А. В.,
д. ю. н., доцент, зав. кафедрой «Информационное право и цифровые технологии», Московский государственный юридический университет им. О. Е. Кутафина (МГЮА, г. Москва);

ПОРШНЕВ С. В.,
д.т.н., профессор, профессор Учебно-научного центра «Информационная безопасность», Уральский федеральный университет им. первого президента России Б.Н. Ельцина (г. Екатеринбург);

РУЧАЙ А.Н.,
д.т.н., доцент, зав. кафедрой «Компьютерная безопасность и прикладная алгебра», Челябинский государственный университет (г. Челябинск);

ХОРЕВ А. А.,
д. т. н., профессор, зав. кафедрой «Информационная безопасность», Национальный исследовательский университет «Московский институт электронной техники» (г. Москва, г. Зеленоград);

ШАБУНИН С. Н.,
д.т.н., профессор, зав. кафедрой «Радиоэлектроника и телекоммуникации», Уральский федеральный университет им. первого президента России Б.Н. Ельцина (г. Екатеринбург).

Journal of the Ural Federal District. Information security

ISSN 2225-5435

№ 3(61) / 2026

FOUNDER
SOUTH URAL STATE UNIVERSITY (NIU)
CHAIRMAN OF THE EDITORIAL BOARD
CHUVARDIN O. P.,
Head of Department Federal Service for Technical and Export Control of Russia for the Urals Federal District
CHIEF EDITOR
SOKOLOV A.N.,
Ph.D., Associate Professor, Head of Department "Information Protection", South Ural State University (National Research University) (Chelyabinsk city)
PRODUCING EDITOR
SOGRIN E. K.
LAYOUT
SCHREIBER A. E.
PROOFREADING
FEDOROV V. S.

Subscription index 73852 in the «Russian Post» catalog
The journal is registered by the Federal service in the field of communication, information technology and mass communications.
Registration number PI No. ФС77-91645 dd. 27.05.2026
Editorial and publisher address: Russia, 454080, Chelyabinsk, Lenin Avenue, 76. SUSU
Phone / fax (351) 267-93-55.
Electronic version of the magazine in the Internet:

www.info-secur.ru,
e-mail: urvest@mail.ru

EDITORIAL COUNCIL:
BARANKOVA I. I.,
Doctor of Technical Sciences, Professor, Head of Department «Informatics and Information Security», Magnitogorsk State Technical University named after. G.I. Nosova (Magnitogorsk city);
VASILYEV V. I.,
Doctor of Technical Sciences, Professor, Professor of the Department «Computer Science and Information Protection», Ufa State Aviation Technical University (Ufa city);
VOITOVICH N. I.,
Doctor of Technical Sciences, Professor, Professor of the Department «Radioelectronics and Communication Systems», South Ural State University (National Research University) (Chelyabinsk city);
GAYDAMAKIN N. A.,
Doctor of Technical Sciences, Professor, Professor of the Information Security Training and Research Center of the Ural Federal University named after the first President of Russia B.N.Yeltsin (Ekaterinburg city);
DIK D. I.,
Ph.D., Associate Professor, Head of Department «Security of information and automated systems», Kurgan State University (Kurgan city);
ZAHAROV A. A.,
Doctor of Technical Sciences, Professor, Professor of the School of Computer Science, Tyumen State University (Tyumen city);
ZYRYANOVA T. Y.,
Ph.D., Associate Professor, Head of Department «Information Technologies and Information Protection», Ural State University ways of communication (Ekaterinburg city);
MELNIKOV A. V.,
Doctor of Technical Sciences, Professor, Director Ugra Research Institute of Information Technologies (Khanty-Mansiysk city);
MINBALEEV A.V.,
Doctor of Law, Associate Professor, Head of Department of «Information Law and Digital Technologies», Moscow State Law University. O. E.Kutafina (Moscow city);
PORSHNEV S. V.,
Doctor of Technical Sciences, Professor, Professor of the Training and Scientific Center «Information Security», Ural Federal University named after the first President of Russia B.N.Yeltsin (Ekaterinburg city);
RUCHAY A.N.,
Doctor of Technical Sciences, Associate Professor, Head of the Department “Computer Security and Applied Algebra”, Chelyabinsk State University (Chelyabinsk city);
HOREV A. A.,
Doctor of Technical Sciences, Professor, Head of Department of «Information Security», National Research University «Moscow Institute of Electronic Technology» (Moscow, the city of Zelenograd);
SHABUNIN S. N.,
Doctor of Technical Sciences, Professor, Head of Department «Radioelectronics and Telecommunications», Ural Federal University named after the first President of Russia B.N.Yeltsin (Ekaterinburg city).

В НОМЕРЕ

СИСТЕМНЫЙ АНАЛИЗ, УПРАВЛЕНИЕ И ОБРАБОТКА ИНФОРМАЦИИ

АФАНАСЬЕВА Н.С., ЕЛИЗАРОВ Д.А.
Марковское моделирование последовательности действий как источник дополнительных признаков для детектирования автоматизированных средств подмены пользователя……5

ЧАСТИКОВА В.А., ВАЛЕЕВ Т.Н.
Современные методы федеративного обучения: сравнительный анализ эффективности и устойчивости……. 19

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

МЕЛЬНИКОВ Д.Ю., СИНАДСКИЙ Н.И.
Реализация системы обнаружения комплексных компьютерных атак, основанной на анализе действий пользователей, сетевых взаимодействий и сопоставлении обнаруженных аномалий с таксономией техник компьютерных атак……....30

КОЖЕМЯКОВ Д.А., ПОРШНЕВ С.В., СТОЛЬНИКОВА А.А., ХОРОШКЕЕВ Д.А.
Комплексный подход к оценке защищенности информационной инфраструктуры на базе Astra Linux: от анализа ОС до динамического тестирования веб-приложений с ИИ……..…48

ЮЖАКОВ Н.А., ЮЖАКОВ А.А.
Архитектура биометрической авторизации в информационных системах……64

СМИРНОВ К.Р., ДИК Д.И., ЧЕЛОВЕЧКОВА А.В.
Таксономия Corsar: классификация методов обхода ограничений в языковых моделях по механизму воздействия…… ….78

IN THIS ISSUE

SYSTEM ANALYSIS, MANAGEMENT AND INFORMATION PROCESSING

AFANASEVA N.S., ELIZAROV D.A.
Markov modeling of action sequences as a source of additional features for detecting automated user-spoofing tools……..……5

CHASTIKOVA V.A., VALEEV T.N.
Modern methods of federated learning: comparative analysis of efficiency and robustness….19

METHODS AND SYSTEMS OF INFORMATION PROTECTION, INFORMATION SECURITY

MELNIKOV D.Y., SINADSKY N.I.
Implementation of a system for detecting complex computer attacks based on analysis of user activity, network interactions, and comparison of detected anomalies with the taxonomy of computer attack techniques…….30

KOZHEMYAKOV D.A., PORSHNEV S.V., STOLNIKOVA A.A., KHOROSHKEEV D.A.
A Comprehensive Approach to Assessing the Security of IT Infrastructure Based on Astra Linux: From OS Analysis to Dynamic Testing of Web Applications with AI……...48

YUZHAKOV N.A., YUZHAKOV A.A.
Architecture of biometric authorization in information systems…….64

SMIRNOV K.R., DIK D.I., CHELOVECHKOVA A.V.
Corsar taxonomy: mechanism-based classification of restriction bypass methods in language mode……..78

УДК 004.056.52 Вестник УрФО № 3(61) / 2026, с. 5–18
DOI: 10.14529/secur260301

Афанасьева Н.С., Елизаров Д.А.

МАРКОВСКОЕ МОДЕЛИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ ДЕЙСТВИЙ КАК ИСТОЧНИК ДОПОЛНИТЕЛЬНЫХ ПРИЗНАКОВ ДЛЯ ДЕТЕКТИРОВАНИЯ АВТОМАТИЗИРОВАННЫХ СРЕДСТВ ПОДМЕНЫ ПОЛЬЗОВАТЕЛЯ

Afanaseva N.S., Elizarov D.A.

MARKOV MODELING OF ACTION SEQUENCES AS A SOURCE OF ADDITIONAL FEATURES FOR DETECTING AUTOMATED USER-SPOOFING TOOLS

Предложенное ранее автором признаковое пространство для детектирования автоматизированных средств подмены пользователя (ботов) описывает непрерывную геометрическую форму траектории движения мыши, но не рассматривает последовательность типов действий (перемещение, клик, прокрутка) как самостоятельный источник признаков. В настоящей работе последовательность действий смоделирована марковской цепью первого порядка. На ее основе построено отдельное признаковое пространство, сопоставленное с траекторным и их объединением. Показано, что признаки на основе действий сами по себе дают низкое качество классификации (F1-мера по классу «бот» – 0,319, у траекторных признаков – 0,837), однако их добавление к траекторным признакам дает прирост полноты (recall) с 0,937 до 0,947 и F1-меры с 0,837 до 0,844. При этом независимое ранжирование важности признаков методом SHAP в объединенном пространстве показывает, что верхние позиции занимают именно признаки на основе кликов, а не траекторные. Анализ параметров марковской цепи объясняет эту особенность – вероятность повторного клика подряд у скриптов на основе Selenium составляет 43,6–65,5%, когда у реальных пользователей и прежде изученных ботов это значение колеблется в пределах16,0–21,5%.

Ключевые слова: бот, вредоносный бот, детектор ботов, цепь Маркова, последовательность действий, классификация, движение курсора мыши, признаковое пространство.

The feature space previously proposed by the author for detecting automated user-substitution tools (bots) describes the continuous geometric shape of the mouse movement trajectory, but does not consider the sequence of action types (movement, click, scroll) as an independent source of features. In the present work, the sequence of actions is modeled as a first-order Markov chain. Based on this, a separate feature space is constructed and compared with the trajectory-based feature space and their combination. It is shown that action-based features alone yield low classification quality (F1-score for the "bot" class – 0,319, versus 0,837 for trajectory-based features); however, adding them to the trajectory-based features improves recall from 0,937 to 0,947 and F1-score from 0,837 to 0,844. At the same time, independent feature importance ranking using SHAP in the combined feature space shows that the top positions are occupied specifically by click-based features rather than trajectory-based ones. Analysis of the Markov chain parameters explains this pattern: the probability of a repeated consecutive click for Selenium-based scripts is 43,6 –65,5%, whereas for real users and previously studied bots this value ranges between 16,0–21,5%.

Keywords: bot, malicious bot, bot detector, Markov chain, action sequence, classification, mouse cursor movement, feature space.

Введение

Согласно ежегодному отчету Imperva о вредоносных ботах, в 2025 году автоматизированный трафик впервые превысил долю трафика реальных пользователей в сети Интернет, составив 53% от общего объема веб-трафика [1]. В отчете это связывается с распространением автономных программных агентов на основе искусственного интеллекта, действующих от имени пользователя через те же интерфейсы, что и человек. Такие продукты размывают границу между легитимным и автоматизированным трафиком и снижают эффективность существующих методов детектирования. По мере того как автоматизированный трафик растет в объеме и правдоподобности, методы детектирования, использующие в себе только один аспект для сравнения, могут устаревать быстрее, чем успевают распространиться в промышленную эксплуатацию.

Любое взаимодействие пользователя с графическим интерфейсом раскладывается на две во многом независимые составляющие. Непрерывная кинематическая сторона описывает как именно перемещается курсор между двумя точками (форма, скорость, ускорение), дискретная событийная сторона – то, какие типы действий совершаются и в каком порядке чередуются (перемещение, клик, прокрутка). Признаковое пространство, предложенное автором ранее для решения задачи детектирования ботов, описывает непрерывную геометрическую форму траектории движения мыши: базовый набор

кинематических параметров дополнен порядковыми статистиками скорости, ускорения, рывка и кривизны, признаками сравнения распределений конкретного трека с распределениями эталонных датасетов на основе критерия хи-квадрат, энтропийными характеристиками и угловыми характеристиками траектории [2]. Совокупность этих групп признаков характеризует форму отдельного, изолированно рассматриваемого перемещения курсора между двумя точками взаимодействия. Однако, данное пространство не рассматривает другую сторону взаимодействия пользователя с интерфейсом – последовательность типов совершаемых действий. Тогда как форма отдельного перемещения может быть имитирована автоматизированным скриптом, сам порядок чередования типов действий в сессии определяется логикой скрипта на более высоком уровне и потенциально несет самостоятельную информацию, не сводимую к геометрии отдельных перемещений.

Естественным математическим аппаратом для описания подобной последовательности дискретных состояний, сменяющих друг друга во времени, выступает марковская цепь. Она компактно описывает вероятностную структуру переходов между типами действий одним небольшим набором параметров (матрицей переходных вероятностей и вектором начального распределения), не требуя явного моделирования всей последовательности целиком, и при этом остается интерпретируемой [3]. Каждый параметр такой цепи имеет прямой содержательный смысл (например, вероятность того, что за кликом последует еще один клик, а не перемещение).

Целью настоящей работы является построение и эмпирическая оценка признакового пространства на основе марковской цепи первого порядка по типам действий, сопоставление его детектирующей способности с траекторным пространством и их объединением, а также анализ того, какую содержательную информацию о различии легитимного и автоматизированного трафика несут параметры такой цепи.

Использование марковских цепей для детектирования ботов

Марковские цепи ранее применялись для детектирования ботов в других постановках задачи. В работе [4] сетевой (не поведенческий) трафик бота моделируется марковской цепью по высокоуровневым признакам сетевых потоков (объем, направление, интервалы между пакетами), выступающим в роли состояний. Полученные параметры цепи использовались как признаки для классификатора Random Forest, что позволяло системе детектировать трафик ранее не встречавшихся семейств ботов с F-мерой 93,03%. Таким образом марковское представление в этой работе использовалось в первую очередь для повышения устойчивости детектора к новым, не участвовавшим в

обучении семействам, а не для описания поведения курсора. Настоящая работа переносит идею марковского моделирования последовательности состояний в область поведенческой биометрии движения мыши, где ранее она в явном виде не применялась, и использует в качестве состояний не сетевые характеристики, а непосредственно тип действия пользователя (перемещение, клик, прокрутка).

В области поведенческой биометрии подавляющее большинство работ решает задачу верификации или идентификации личности, а не бинарной классификации «человек/бот». В [5] и [6] строится модель нормального поведения конкретного пользователя, и оценивается отклонение новой сессии от нее. В работе [7] движения мыши используются в составе более широкой системы человеко-компьютерного взаимодействия для непрерывной аутентификации. Авторы [8] используют признаки движения мыши для идентификации пользователя в задаче веб-браузинга, в [9] для той же задачи идентификации применяются модели глубокого обучения. Во всех перечисленных работах типичными метриками качества выступают равный уровень ошибок (EER) и площадь под ROC-кривой (AUC), не тождественные метрикам точность (precision), полнота (recall) и F1-мера, используемым при решении задачи бинарной классификации «человек/бот».

Отдельная линия исследований посвящена непосредственно детектированию ботов: в [10] применяются методы глубокого обучения для детектирования веб-ботов по поведенческой биометрии, в [11] предложен метод синтетической генерации обучающих траекторий (BeCAPTCHA-Mouse) для обучения и тестирования детекторов, в [12] предложена облегченная по вычислительным затратам сетевая архитектура, ориентированная на промышленное применение. Ни в одной из перечисленных работ не рассматривалась последовательность типов действий как самостоятельный источник признаков, дополняющий геометрию траектории. Настоящая работа проверяет гипотезу о том, что построенное на основе такого расширения признаковое пространство способно повышать детектирующую способность детектора, ранее предложенного автором [13].

Датасеты и признаковые пространства, используемые для классификации

Для эксперимента используются данные из различных источников. user_BALABIT и user_BOGAZICI – эталонные публичные датасеты реальных пользователей, user_BROWSER – собственный датасет реальных пользователей, bot_BOT– ранее изученный генератор автоматизированного трафика, BROWSER_BOTS_E – браузерный бот с более сложной, вариативной

имитацией действий, E_line, E_teleport, E_move_to_element, E_b-spline interpolation – четыре датасета скриптов-скраперов на основе библиотеки Selenium. Методология формирования и разметки данных подробно описана в [13]. Итоговая тестовая выборка содержит 6071 наблюдение класса «бот» и 692102 наблюдения класса «пользователь».

Для сопоставления детектирующей способности построены три классификатора Random Forest [14] (n_estimators по умолчанию, max_depth = 10, class_weight = 'balanced') с идентичной архитектурой, обученные на трех разных подмножествах признаков («action», «trajectory», «action + trajectory») одной и той же объединенной выборки.

Пространство «action» включает три группы признаков, производных от марковской цепи и статистики по типам действий. Каждая точка сессии взаимодействия отнесена к одному из трех типов действия: перемещение курсора (MOUSE_MOVE), клик (MOUSE_CLICK) или прокрутка колеса мыши (MOUSE_WHEEL). Последовательность типов действий в пределах сессии моделируется цепью Маркова первого порядка. Для каждой сессии рассчитывается матрица A вероятностей перехода между типами действий (элемент A[i,j] – вероятность того, что за действием типа i непосредственно следует действие типа j) и вектор π стационарного (в данной реализации начального усредненного) распределения по типам действий. Помимо параметров самой цепи, данное признаковое пространство включает статистические характеристики распределения точек по типам действий и по временным интервалам между ними: число точек каждого типа действия, характеристики гистограмм длительности интервалов между последовательными действиями одного типа, а также отдельную группу признаков дублирования кликов (долю и число кликов, совпадающих по координатам с одним из предыдущих кликов в той же сессии).

Пространство «trajectory» включает 222 признака геометрической формы траектории: порядковые статистики параметров движения (квартили, межквартильный размах, медианы скорости, ускорения, рывка, кривизны и угловых характеристик), признаки сравнения распределения конкретного трека с распределениями эталонных датасетов на основе критерия хи-квадрат, энтропийные характеристики, угловые характеристики траектории в целом. Признаки данного пространства не зависят от того, к какому типу относится каждая отдельная точка (перемещение, клик или прокрутка), и рассчитываются по геометрии пути целиком.

Пространство «action + trajectory» образовано непосредственным объединением (конкатенацией) всех признаков обоих пространств без дополнитель-

ного отбора или преобразования, что позволяет классификатору самостоятельно определить относительный вклад каждой группы признаков в итоговое решение.

Для каждой модели рассчитаны precision, recall и F1-мера по классу «бот», а также доля треков, отнесенных к каждому классу, в разбивке по конкретным источникам данных.

Результаты эксперимента

Результаты классификации по классу «бот» в трех признаковых пространствах представлены на рис. 1 и в табл. 1.

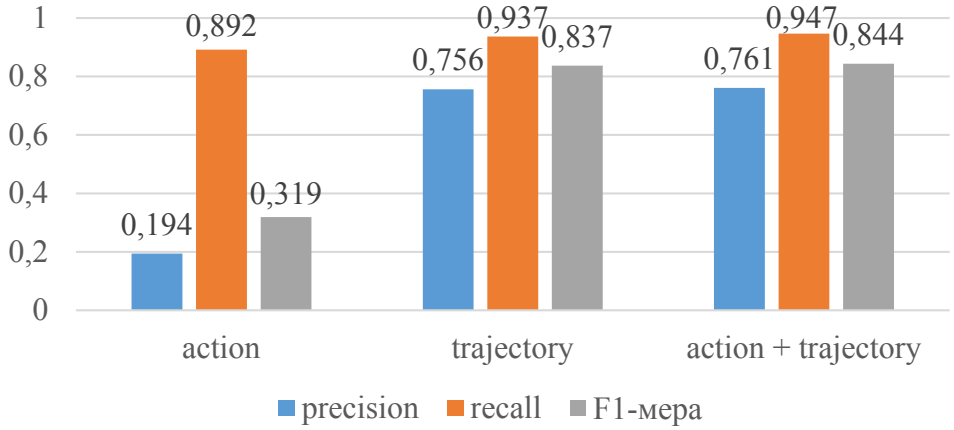


Рис. 1. Качество классификации в трех признаковых пространствах

Таблица 1

Качество классификации по классу «бот» в трех признаковых пространствах

Пространство	precision	recall	F1-мера	accuracy
action	0,194	0,892	0,319	0,967
trajectory	0,756	0,937	0,837	0,997
action + trajectory	0,761	0,947	0,844	0,997

Признаки на основе действий сами по себе дают заметно более низкое качество классификации, чем траекторные признаки. F1-мера по классу «бот» в пространстве действий составляет 0,319 против 0,837 у траекторных признаков, что обуславливается низкой точностью (precision 0,194 и 0,756 соответственно). Модель на основе действий существенно чаще ложно относит треки реальных пользователей к боту. При этом полнота (recall) в пространстве действий оказывается на высоком уровне и составляет 0,892, то есть эти признаки способны улавливать значительную часть ботов, но ценой большого числа ложных срабатываний. Объединение двух пространств дает небольшой, но устойчивый прирост метрик относительно чистой траектории (полнота с 0,937 до 0,947, F1-мера с 0,837 до 0,844).

Разбивка доли треков, отнесенных к классу «бот», по конкретным источникам данных (табл. 2) показывает, что добавление признаков действий к траекторным затрагивает источники неравномерно.

Таблица 2

Доля треков, отнесенных к классу «бот», по источникам и пространствам, %

Источник	action	trajectory	action + trajectory
bot_BOT	70,07	77,07	80,54
E_b-spline	88,54	99,62	99,61
E_ line	99,17	99,92	99,90
E_move_to_element	99,70	99,89	99,89
E_teleport	99,39	99,83	99,89
user_BALABIT (ложноположж.)	19,03	2,29	2,21
user_BOGAZICI (ложноположж.)	3,11	0,04	0,08
user_BROWSER (ложноположж.)	2,89	0,32	0,23

Для ботов расширение признакового пространства путем добавления действий к траектории почти везде улучшает или сохраняет долю верно распознанных треков: для bot_BOT – с 77,07% до 80,54%, для E-скраперов рост либо отсутствует, либо укладывается в десятые доли процента. Для реального пользователя BALABIT ложноположительная доля при объединении пространств уменьшается 2,29% до 2,21%, тогда как для BOGAZICI направление изменения обратное (рост с 0,04% до 0,08%). Это указывает на то, что выигрыш от добавления признаков действий не является для всех источников безусловно положительным и должен оцениваться на конкретной, представительной для эксплуатации выборке легитимных пользователей.

Стационарное распределение по типам действий (вектор π) различается между источниками. Доля кликов колеблется от 1,47% (BROWSER_ BOTS_E) до 12,28% (E_move_to_element), доля прокрутки – от 0% (user_ BOGAZICI, где события прокрутки в данных отсутствуют) до 10,81% (BROWSER_BOTS_E). Полное стационарное распределение по трем типам действий (вектор π) для всех рассмотренных источников приведено в табл. 3.

Таблица 3

Стационарное распределение по типам действий (вектор π), %

Источник	Перемещение	Клик	Прокрутка
user_BALABIT	96,57	2,85	0,59
user_BOGAZICI	93,31	6,69	0,00
user_BROWSER	89,62	1,47	8,91
bot_BOT	98,14	1,73	0,13
BROWSER_BOTS_E	87,71	1,47	10,81
E_line	83,36	10,90	5,74

Окончание табл. 3

E_teleport	83,00	11,20	5,80
E_move_to_element	80,78	12,28	6,94
E_b-spline	83,94	10,52	5,54

Дополнительно рассчитана вероятность перехода из состояния «клик» обратно в состояние «клик» (диагональный элемент матрицы переходов), то есть вероятность того, что за одним кликом непосредственно последует еще один клик без промежуточного перемещения (рис. 2).

Реальные пользователи (BALABIT, BOGAZICI, user_BROWSER) и уже изученные боты (bot_BOT, BROWSER_BOTS_E) демонстрируют сопоставимо низкую вероятность повторного клика подряд, в диапазоне 16,0 – 21,5%. Все четыре скрипта на основе Selenium, напротив, показывают значительно более высокую вероятность – от 43,6% (line) до 65,5% (b-spline interpolation), то есть выше более чем в два – три раза. Такое поведение согласуется с логикой типового скрипта автоматизации, последовательно выполняющего несколько кликов по элементам страницы без промежуточных перемещений, характерных для человека или для более сложно имитирующего поведение бота.

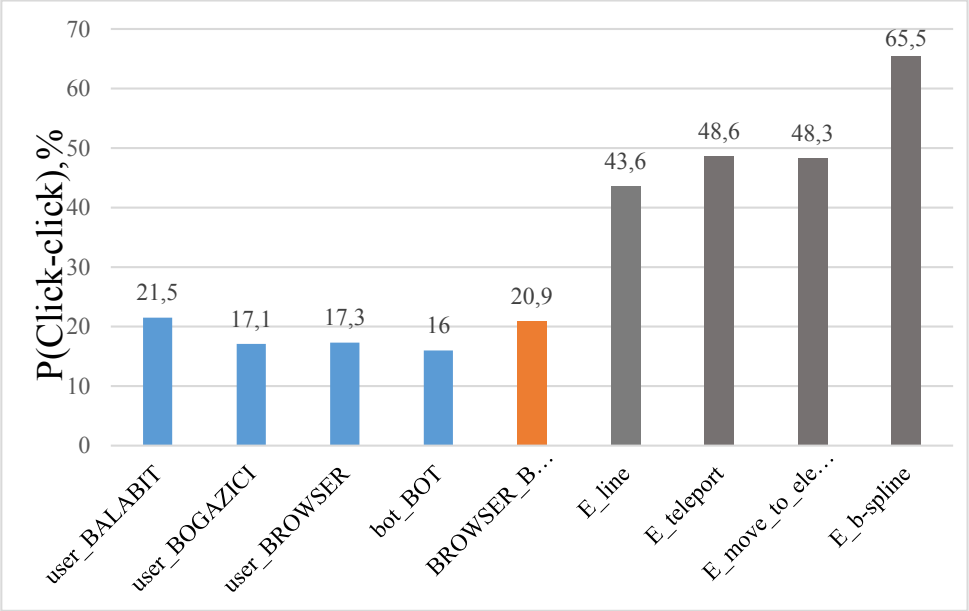


Рис. 2. Вероятность повторного клика подряд по источникам данных

Как видно из табл. 3, доля кликов у четырех Selenium-скраперов (10,52–12,28%) систематически выше, чем у реальных пользователей и ранее изученных ботов (1,47–6,69%), что дополняет информацию о повышенной вероятности повторного клика подряд. Скрипты-скраперы не только чаще кликают подряд, но и в целом отводят кликам большую долю от общего числа

действий. Доля прокрутки не дает столь же однозначной картины – она невелика и сопоставима у большинства источников (0–6,94%), однако заметно выше у BROWSER_BOTS_E (10,81%) и user_BROWSER (8,91%). Эти два источника не разделяются по данному признаку, что говорит об ограниченной самостоятельной различающей способности доли прокрутки в отрыве от остальных параметров цепи.

Независимым подтверждением значимости признаков на основе действий служит ранжирование важности признаков методом SHAP, рассчитанное для классификатора в объединенном пространстве «action + trajectory» (рис. 3). Верхние позиции ранжирования заняли признаки на основе кликов (доля и число повторяющихся по координатам кликов, распределение кликов по временным интервалам). Это согласуется с обнаруженным по параметрам марковской цепи различием в характере кликов между источниками и объясняет, почему признаки действий, несмотря на низкое собственное качество классификации, вносят непропорционально большой вклад при объединении с траекторией.

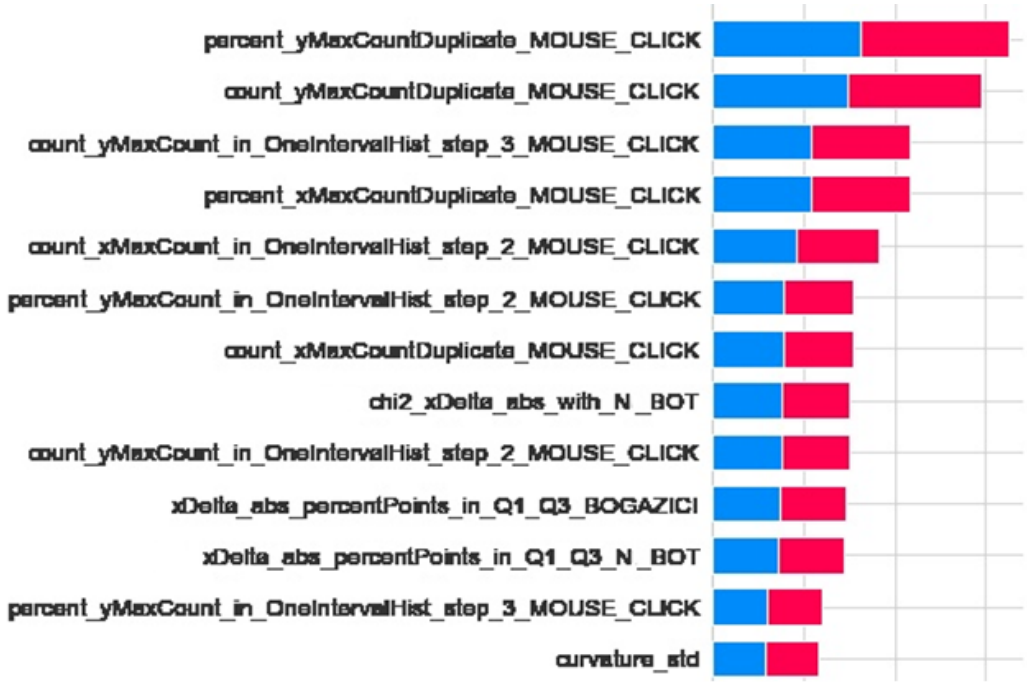


Рис. 3. Верхние позиции ранжирования важности признаков методом SHAP в объединенном пространстве «action + trajectory»

Обсуждение результатов эксперимента

Полученные результаты показывают, что низкое собственное качество классификации у пространства «action» не означает отсутствия у него содержательной информации. В объединении с траекторией именно признаки на основе кликов оказываются наиболее значимыми, по независимой оценке,

SHAP, а параметры марковской цепи дают понятное, интерпретируемое объяснение этому эффекту. Распространенные инструменты автоматизации (Selenium-скрипты) кликают заметно чаще подряд, чем это делает человек или более сложно имитирующий его бот. Это позволяет сделать методологически важный вывод – оценка ценности признакового пространства только по его собственным метрикам (precision, recall, F1-мера) в изоляции может недооценивать потенциальный вклад, вносимый этими признаками в составе более широкого пространства.

Практический вывод состоит в целесообразности включения статистик по типам действий (в частности, показателей повторяемости кликов) в состав признакового пространства детектора, дополнительно к траекторным признакам, поскольку такие статистики отражают логику работы скрипта автоматизации, которую сложно замаскировать без явного усложнения самого скрипта. Одновременно наблюдаемый неоднородный эффект на ложноположительную долю по разным источникам легитимных пользователей указывает на необходимость проверки такого расширения на представительной для конкретной эксплуатационной среды выборке пользователей.

Верхние пять позиций ранжирования SHAP в объединенном пространстве заняли исключительно признаки на основе повторяемости кликов (доля и число дублирующихся по координатам кликов, число точек клика в интервалах гистограммы), опередив ранее ключевые для чисто траекторного пространства признаки (критерий хи-квадрат при сравнении с эталонными датасетами и угловые характеристики). Это означает, что практическая ценность объединения пространств обеспечивается не суммарным, распределенным вкладом множества новых признаков, а вкладом узкой группы из нескольких конкретных, интерпретируемых индикаторов. Этот результат отличается от ранее полученного для чисто траекторного пространства, в [13] наиболее значимой по SHAP группой признаков были порядковые статистики рывка и ускорения, тогда как в настоящей работе, при добавлении признаков действий, верхние позиции занимают признаки совершенно другой природы – статистики повторяемости кликов. Это говорит о том, что ранжирование важности признаков существенно зависит от состава признакового пространства, в котором оно рассчитывается.

Полученные результаты имеют значение для проектирования признаковых пространств детекторов автоматизированного трафика за пределами конкретной рассмотренной задачи. Во-первых, они демонстрируют, что оценка признакового пространства только по его собственному, изолированному качеству классификации может вводить в заблуждение относительно

его практической ценности. Во-вторых, интерпретируемость найденного различия – конкретная, понятная эксплуатирующему специалисту причина (повышенная частота повторных кликов у типовых скриптов автоматизации) – позволяет использовать этот признак не только в составе модели машинного обучения, но и как самостоятельный, объяснимый индикатор при ручном разборе подозрительных сессий. В-третьих, поскольку признаки на основе действий строятся на статистике типов событий, а не на геометрии координат, они потенциально более устойчивы к простым методам маскировки, ориентированным на имитацию правдоподобной формы траектории, но не корректирующим статистику типов действий.

Заключение

В работе продемонстрировано, что моделирование последовательности типов действий (перемещение, клик, прокрутка) марковской цепью первого порядка дает признаковое пространство, самостоятельно уступающее по качеству классификации траекторному, но при объединении этих пространств наблюдается прирост полноты (recall) с 0,937 до 0,947 и F1-меры с 0,837 до 0,844 по классу «бот». Независимое ранжирование важности признаков методом SHAP показывает, что признаки на основе кликов занимают верхние позиции в объединенном пространстве, а анализ параметров марковской цепи объясняет эту находку конкретным поведенческим различием – скрипты на основе Selenium кликают подряд значительно чаще, чем реальные пользователи и ранее изученные боты. Практическая рекомендация состоит во включении статистик повторяемости и последовательности действий в состав признакового пространства детектора как источника информации, дополняющей, а не дублирующей траекторные признаки. Возможным продолжением работы является проверка марковских цепей более высокого порядка (второго и выше), при которых вероятность следующего действия зависит не от одного, а от нескольких предшествующих.

Работа выполнена в рамках государственного задания на оказание государственных услуг (выполнение работ) № 109-03-2026-004 от 16.01.2026 г. по теме «Разработка технологии выявление автоматизированных средств подмены пользователя в информационной среде с помощью алгоритмов моторной идентификации поведенческих паттернов».

Литература

1. 2025 Imperva Bad Bot Report [Электронный ресурс]. URL: https://www.imperva.com/resources/resource-library/reports/2025-imperva-bad-bot-report (дата обращения: 22.08.2026).

2. Афанасьева Н.С. Формирование пространства признаков, позволяющих осуществлять детектирование автоматизированных средств подмены пользователя в информационной среде // Безопасность информационных технологий. – 2026 (в печати).

3. Norris J. R. Markov chains. – Cambridge: Cambridge university press, 1998. – № 2.

4. Alahmadi B. A., Mariconti E., Spolaor R., Stringhini G., Martinovic I. BOTection: Bot detection by building Markov Chain models of bots network behavior // Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20). – New York: ACM, 2020. – P. 652–664.

5. Zheng N., Paloski A., Wang H. An efficient user verification system via mouse movements // Proceedings of the 18th ACM Conference on Computer and Communications Security. – New York: ACM, 2011. – P. 139–150.

6. Feher C., Elovici Y., Moskovitch R., Rokach L., Schclar A. User identity verification via mouse dynamics // Information Sciences. – 2012. – Vol. 201. – P. 19–36.

7. Gamboa H., Fred A. A behavioral biometric system based on human-computer interaction // Proceedings of SPIE, Vol. 5404, Biometric Technology for Human Identification. – Bellingham: SPIE, 2004. – P. 381–392.

8. Chuda D., Kratky P. Usage of computer mouse characteristics for identification in web browsing // Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14). – New York: ACM, 2014. – P. 218–225.

9. Antal M., Fejer N. Mouse dynamics based user recognition using deep learning // Acta Universitatis Sapientiae, Informatica. – 2020. – Vol. 12, № 1. – P. 39–50.

10. Wei A., Zhao Y., Cai Z. A deep learning approach to web bot detection using mouse behavioral biometrics // Biometric Recognition (CCBR 2019), Lecture Notes in Computer Science, Vol. 11818. – Cham: Springer, 2019. – P. 388–395.

11. Acien A., Morales A., Fierrez J., Vera-Rodriguez R. BeCAPTCHA-Mouse: Synthetic mouse trajectories and improved bot detection // Pattern Recognition. – 2022. – Vol. 127. – Art. 108643.

12. Niu H., Zhou Y., Chen J., Cai Z. Performance evaluation of lightweight network-based bot detection using mouse movements // Engineering Applications of Artificial Intelligence. – 2024. – Vol. 135. – Art. 108801.

13. Афанасьева Н.С. Экспериментальная оценка использования расширенного пространства признаков для детектирования автоматизированных

средств подмены пользователя // Информационные технологии. – 2026 (в печати).

14. Breiman L. Random Forests // Machine Learning. – 2001. – Vol. 45, № 1. – P. 5–32.

References

1. 2025 Imperva Bad Bot Report. Available at: https://www.imperva.com/resources/resource-library/reports/2025-imperva-bad-bot-report (accessed 12 August 2026).

2. Afanaseva N.S. Formirovanie prostranstva priznakov, pozvolyayushchikh osushchestvlyat' detektirovanie avtomatizirovannykh sredstv podmeny pol'zovatelya v informatsionnoy srede. Security of Information Technologies, 2026 (in press).

3. Norris J. R. Markov chains. Cambridge, Cambridge University Press, 1998, no. 2.

4. Alahmadi B. A., Mariconti E., Spolaor R., Stringhini G., Martinovic I. BOTection: Bot detection by building Markov Chain models of bots network behavior. Proceedings of the 15th ACM Asia Conference on Computer and Communications Security (ASIA CCS '20). New York, ACM, 2020, pp. 652–664.

5. Zheng N., Paloski A., Wang H. An efficient user verification system via mouse movements. Proceedings of the 18th ACM Conference on Computer and Communications Security. New York, ACM, 2011, pp. 139–150.

6. Feher C., Elovici Y., Moskovitch R., Rokach L., Schclar A. User identity verification via mouse dynamics. Information Sciences, 2012, Vol. 201, pp. 19–36.

7. Gamboa H., Fred A. A behavioral biometric system based on human-computer interaction. Proceedings of SPIE, Vol. 5404, Biometric Technology for Human Identification. Bellingham, SPIE, 2004, pp. 381–392.

8. Chuda D., Kratky P. Usage of computer mouse characteristics for identification in web browsing. Proceedings of the 15th International Conference on Computer Systems and Technologies (CompSysTech '14). New York, ACM, 2014, pp. 218–225.

9. Antal M., Fejer N. Mouse dynamics based user recognition using deep learning. Acta Universitatis Sapientiae, Informatica, 2020, Vol. 12, no. 1, pp. 39–50.

10. Wei A., Zhao Y., Cai Z. A deep learning approach to web bot detection using mouse behavioral biometrics. Biometric Recognition (CCBR 2019), Lecture Notes in Computer Science, Vol. 11818. Cham, Springer, 2019, pp. 388–395.

11. Acien A., Morales A., Fierrez J., Vera-Rodriguez R. BeCAPTCHA-Mouse: Synthetic mouse trajectories and improved bot detection. Pattern Recognition, 2022, Vol. 127, art. 108643.

12. Niu H., Zhou Y., Chen J., Cai Z. Performance evaluation of lightweight network-based bot detection using mouse movements. Engineering Applications of Artificial Intelligence, 2024, Vol. 135, art. 108801.

13. Afanaseva N.S. Eksperimental'naya otsenka ispol'zovaniya rasshirennogo prostranstva priznakov dlya detektirovaniya avtomatizirovannykh sredstv podmeny pol'zovatelya. Information Technologies, 2026 (in press).

14. Breiman L. Random Forests. Machine Learning, 2001, Vol. 45, no. 1, pp. 5–32.

АФАНАСЬЕВА Наталья Сергеевна, старший преподаватель кафедры «Информационная безопасность», Омский государственный университет путей сообщения. 644046, г. Омск, пр-т К. Маркса, 35. E-mail: nati_dik@mail.ru

ЕЛИЗАРОВ Дмитрий Александрович, кандидат техн. наук, доцент, зав. кафедрой «Информационная безопасность», Омский государственный университет путей сообщения. 644046, г. Омск, пр-т К. Маркса, 35. E-mail: d.a.elizarov@yandex.ru

AFANASEVA Natalia Sergeevna, Senior Lecturer of Department «Information security», Omsk State Transport University. 644046, Omsk, Marx st.,35. E-mail: nati_dik@mail.ru

ELIZAROV Dmitriy Aleksandrovich, PhD in Technical Sciences, Associate Professor, Head of Department «Information security», Omsk State Transport University. 644046, Omsk, Marx st.,35. E-mail: d.a.elizarov@yandex.ru

УДК 378.016+004.056.2 Вестник УрФО № 3(61) / 2026, с. 19–29
DOI: 10.14529/secur260302

Частикова В.А., Валеев Т.Н.

СОВРЕМЕННЫЕ МЕТОДЫ ФЕДЕРАТИВНОГО ОБУЧЕНИЯ: СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЭФФЕКТИВНОСТИ И УСТОЙЧИВОСТИ

Chastikova V.A., Valeev T.N.

MODERN METHODS OF FEDERATED LEARNING: COMPARATIVE ANALYSIS OF EFFICIENCY AND ROBUSTNESS

В данной статье рассматриваются методы и алгоритмы в модели федеративного обучения, а также проводится сравнительный анализ при условии наличия во время обучения гетерогенных данных. Особое внимание уделяется таким алгоритмам и их расширениям, как FedAvg, FedProx, SCAFFOLD, FedNova, FedOpt, FLTrust и LoRA-FL. При работе были использованы следующие метрики: точность обучения, скорость сходимости, устойчивость к атакам типа Byzantine и эффективность работы при гетерогенных данных. Результатом работы является выявление преимуществ алгоритмов, а также перспективные направления развития модели федеративного обучения.

Ключевые слова: Федеративное обучение, non-IID данные, гетерогенность, агрегация моделей, Byzantine-атаки, коммуникационная эффективность, гетерогенное обучение.

This article examines methods and algorithms used in the Federated Learning model and provides a comparative analysis under conditions of heterogeneous data during training. Particular attention is paid to algorithms and their extensions, including FedAvg, FedProx, SCAFFOLD, FedNova, FedOpt, FLTrust, and LoRA-FL. The study employs the following evaluation metrics: training accuracy, convergence speed, resistance to Byzantine attacks, and performance efficiency under heterogeneous data conditions. The results of the study identify the advantages of the considered algorithms and outline promising directions for the further development of federated learning models.

Keywords: federated learning, non-IID data, heterogeneity, model aggregation, Byzantine attacks, communication efficiency, heterogeneous learning.

Введение

Примерно в 2016–2017 годах был фактически впервые применение принцип федеративного обучения, данный метод использовался для обучения

нейронных сетей на основе мобильных устройств клиентов, отличительной чертой которой стало выполнение требований конфиденциальности. Важной отличительной чертой данного способа стал централизированный метод при том, что данные не покидали пределов своих клиентов, т.е. сервер получал лишь обновления самой модели (градиенты, веса), после чего происходила агрегация данных обновлений, тем самым изменяя глобальную модель. И соответственно данный процесс проходил за счет множества циклов [1, 2].

Несмотря изначально на безопасность основополагающей идеи в ней все же присутствуют аспекты, которые требуют внимания: статическая гетерогенность (non-IID), системная гетерогенность (различные вычислительные ресурсы), уязвимость к атакам типа Byzantine.

Научная новизна данной статьи заключается в рассмотрении алгоритмов и методов федеративного обучения, а также их сравнение, с последующим выделением проблем и возможных направлений развития данной области.

Базовые алгоритмы и методы федеративного обучения

В основе большинства алгоритмов агрегации лежит одна и та же дилемма: чем проще протокол, тем хуже он справляется с неоднородностью данных, и наоборот. Показательнее всего эта закономерность видна на примере FedAvg – исторически первого и самого распространённого алгоритма, который просто усредняет локальные модели клиентов:

$$w_{t+1} = \sum_k \frac{n_k}{n} w_k,$$

где t – раунды,

w_t – модель для обучения,

k – клиенты.

На независимых и одинаково распределённых (IID) данных такой подход сходится быстро, но стоит данным стать статистически неоднородными – и модель начинает страдать от «клиентского дрейфа», из-за которого точность резко падает [1].

Именно поиск баланса между простотой и устойчивостью к дрейфу породил следующее поколение алгоритмов – FedProx, SCAFFOLD, FedNova и FedOpt. Каждый из них решает эту задачу по-своему: одни ограничивают отклонение весов «мягко», через штраф в функции потерь, другие – «жёстко», через прямую коррекцию направления обновлений или отказ от усреднения весов как такового.

Таблица 1

Оптимизирующие алгоритмы

Алгоритм	Механизм	Формула	Задача	Замечания
FedProx	Проксимальный штраф в целевой функции	$\frac{\mu}{2}\|w - w_t\|^2$, где μ ограничивает отклонение локальных весов от глобального консенсуса	Смягчает клиентский дрейф	Требует подбора μ
SCAFFOLD	Контрольные векторы корректируют направление обновлений	$g_k = \nabla F_k - c_k + c$, где g_k – обновление клиента, ∇F_k – локальный градиент, c_k – контрольный вектор клиента, c – глобальный контрольный вектор.	Заметно повышает точность на non-IID данных	Растут коммуникационные издержки [4]
FedNova	Нормализация обновлений по числу локальных шагов	$w_{t+1} = w_t - \eta \frac{\sum_{i \in S_t} \frac{n_i}{n} * \frac{\Delta_i}{\tau_i}}{\sum_{i \in S_t} \frac{n_i}{n}}$, где η – скорость обучения, S_t – случайная выборка клиентов, n_i – размер данных клиента, τ_i – число локальных шагов, Δ_i – суммарный локальный градиент.	Устраняет смещение модели в сторону устройств с большим числом итераций	Не устраняет статистическую неоднородность как таковую
FedOpt	Разница моделей трактуется как псевдоградиент	$g_t = \sum_{i \in S_t} \frac{n_i}{n}(w_t - w_{ti})$	Позволяет применять продвинутую оптимизацию поверх агрегированного результата [6]	Дополнительная сложность на сервере

Иными словами, если FedProx и SCAFFOLD решают проблему статистического дрейфа, то FedNova устраняет другую его причину – рассинхронизацию локальных вычислений, – а FedOpt меняет саму логику агрегации, отказываясь от прямого усреднения весов.

Отказоустойчивость агрегации

Но даже идеально настроенный алгоритм агрегации остаётся уязвим, если часть клиентов присылает аномальные обновления – по ошибке или намеренно. Классическое взвешенное усреднение никак от этого не защищено, поэтому поверх базовых алгоритмов применяют робастные стратегии – от простого отсечения выбросов до полноценной валидации через доверенный эталон.

Таблица 2

Методы агрегации

Метод	Принцип защиты	Уровень защиты
Coordinate-wise Median	Отсекает экстремальные отклонения по каждому параметру отдельно	Базовый
Trimmed Mean	Отбрасывает маргинальные значения по краям распределения, усредняет остальное	Базовый
Krum / Multi-Krum	Анализирует пространственное расположение обновлений, отбирает те, у которых больше «ближайших соседей»	Средний–высокий
Bulyan	Комбинирует Multi-Krum (фильтрация) и Trimmed Mean (усреднение отфильтрованного)	Максимальный
FLTrust	Валидирует градиенты клиентов через небольшую доверенную выборку на сервере	Максимальный, но принципиально иной подход

Как видно из таблицы, рост защиты почти всегда достигается комбинированием методов (Bulyan) либо принципиально иной логикой (FLTrust) – прямое усреднение к этому моменту уже не спасает [7, 8].

Гетерогенность в федеративном обучении

Проблема дрейфа и уязвимость агрегации – лишь следствия более глубокой особенности федеративного обучения: клиенты изначально неоднородны, причём неоднородны по двум независимым осям – статистической (какие у них данные) и модельной (какие у них ресурсы).

Статистическая гетерогенность изучена лучше всего и встречается чаще: она проявляется как неоднородность меток, неоднородность признаков, дисбаланс количества образцов или концептуальный сдвиг. Удобный способ измерить её степень – параметр Дирихле α: при $\alpha \to 0$ неоднородность становится экстремальной (у каждого клиента данные только одного класса), при

$\alpha \to \infty$ распределение приближается к однородному IID. Часть решений этой проблемы уже знакома – это SCAFFOLD и FedOpt из первого раздела; к ним добавляются персонализированное FL (Per-FedAvg), общий публичный датасет (FedMix) и дистилляция знаний через публичные данные [9].

Модельная гетерогенность возникает по другой причине – из-за разницы в вычислительных ресурсах устройств, из-за которой обучить единую тяжёлую архитектуру на всех клиентах попросту невозможно. Адаптация к этому ограничению происходит на трёх уровнях.

Таблица 3

Уровни архитектуры

Уровень	Что адаптируется	Принцип
Данных	Предобработка	Локальная аугментация под аппаратные лимиты устройства
Модели	Архитектура	Прунинг, масштабирование, дистилляция знаний, раздельное обучение
Сервера	Координация клиентов	Кластеризация по архитектурным профилям, мета-агрегация

Из перечисленных методов дистилляция знаний стоит особняком: на сервер передаются не «сырые» веса или градиенты, а логиты (мягкие метки), полученные на публичном датасете. Это позволяет полностью абстрагироваться от архитектурных различий клиентских моделей и свести потерю точности к минимуму [10].

Масштабирование на сверхбольшие модели

Модельная гетерогенность обостряется многократно, когда речь заходит о по-настоящему больших архитектурах – LLM на клиентских устройствах с ограниченной памятью попросту не помещаются. Это породило отдельный класс решений, снижающих издержки не за счёт адаптации архитектуры, а за счёт сокращения того, что вообще передаётся или обучается.

Таблица 4

Методы для LLM

Подход	Процесс	Эффект
LoRA-FL, FedIT, FlexLoRA	Узлы обновляют только низкоранговые адаптеры	Трафик сокращается в 100–1000 раз без потери качества
Prefix-tuning	Оптимизируются только мягкие промпты, веса модели заморожены	Минимальная нагрузка на клиента

Окончание табл. 4

Split Learning	Слои сети распределяются между клиентом и сервером	Разгружается память конечного устройства
In-context learning	Клиенты формируют только контекстные примеры	Основные вычисления полностью уходят на сервер

Эта же логика – разгрузить клиента, перераспределив нагрузку, – прослеживается и на уровне сетевой топологии, только там перераспределяется не вычислительная, а координационная нагрузка. Иерархическое FL вводит промежуточные граничные серверы (Edge), снижающие задержки и разгружающие центральный координатор. P2P-FL идёт дальше и вовсе отказывается от центрального узла в пользу прямого обмена градиентами между клиентами – это максимально повышает устойчивость к единой точке отказа (SPOF), но заметно усложняет гарантии математической сходимости. Наконец, Federated X Learning выходит за рамки топологии как таковой, объединяя саму парадигму федеративного обучения с трансферным, континуальным обучением и обучением с подкреплением [13].

Материалы и методы

Экспериментальная часть работы проводилась в симуляции федеративного обучения в среде Python, где клиентами выступали независимое процессы на отдельном кластере. В данном случае сервер, где проводилась симуляция, оснащен четырьмя NVIDIA A100 по 40 Гб памяти.

Более того, использовались Python 3.10, PyTorch 2.1, PySyft 0.8, обеспечивающие дифференциальную приватность и безопасные вычисления.

Для эмуляции системы, в которой данные гетерогенны использовалась библиотека FEMNIST, где данные распределены между большим количеством клиентов, но при этом не являются одинаковыми, создавая ту самую гетерогенность. Применялся данная библиотека на основе LEAF для моделей федеративного обучения, позволяющий набор инструментов для исследований в данной области. Для окончательных замеров точности использовался датасет Shakespeare, включающий в себя тексты Шекспира, а также включенный в библиотеку LEAF.

В ходе работы эксперименты проводились по пять раз с начальным seed, который менялся от 0 до 4, тем самым создавая вариативность. А итоговые значения приводились в формате усредненного значение ± отклонение. А также количество раундов в работе использовалось 200, а размер батча 64.

В качестве метрик в работе использовались следующие:

– максимально достигнутая точность
– раунды необходимые для достижения порога в 75% точности
– точность, достигнутая при наличии Byzantine-атак на систему
– точность при отношении гетерогенности в системе.

Сравнительный анализ эффективности методов

На рис. 1 показана динамика точности на тестовой выборке для шести ведущих алгоритмов агрегации в условиях сильного non-IID (параметр Дирихле $\alpha = 0.1$). Как можно заметить все методы начинают примерно одинаково по точности, а различия начинают появляться примерно после 50 раунда.

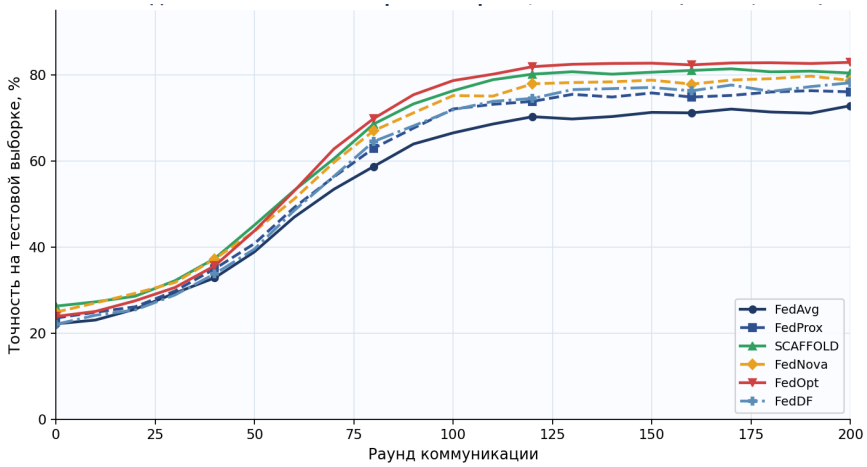


Рис. 1. Динамика точности алгоритмов агрегации FL

На рис. 2 представлен график с осями: горизонтальный обозначает количество раундов для достижения 75% точности, вертикальный обозначает предельную точность обучения на проверочной выборке.

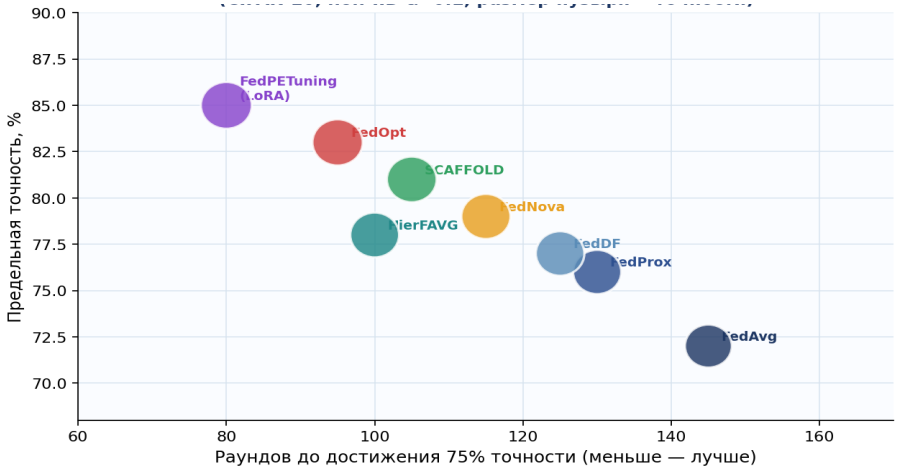


Рис. 2. Компромисс «скорость сходимости – предельная точность»

На нижеследующем рис. 3 представлен график, показывающий степень защищенности от Byzantine-атак (имитации случайных обновлений) при возрастающие доле злонамеренных клиентов от 0% до 30%.

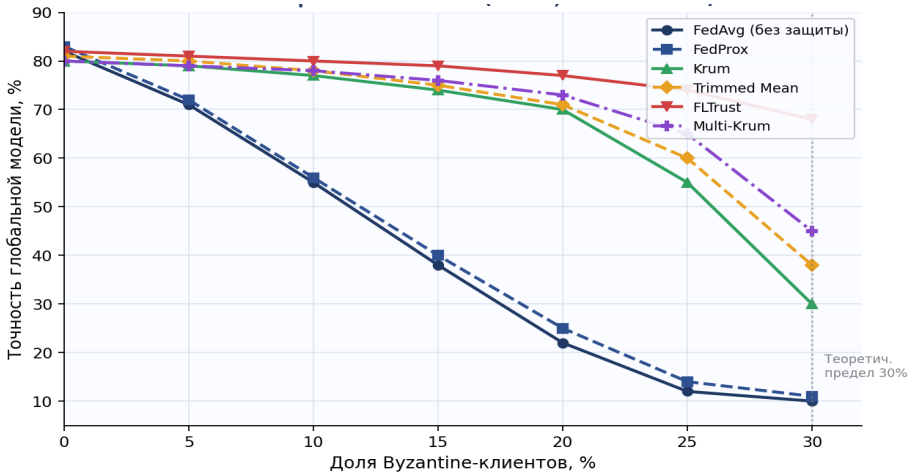


Рис. 3. Точность методов FL при различной доле Byzantine-клиентов

На рис. 4 представлен график точности методов относительно того, как сильно падает эффективность при неоднородных данных во время обучения, где 0,05 означает экстремально неоднородный, а IID – полную однородность.

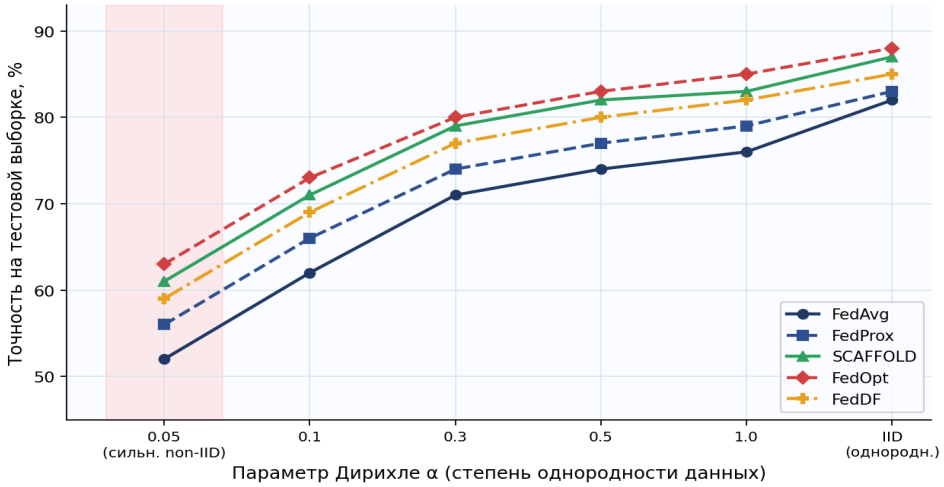


Рис. 4. Точность методов FL в зависимости от степени неоднородности данных

Представленная ниже таблица наглядно показывает эффективность рассмотренных ранее методов по основным метрикам.

Таблица 5

Сводная таблица

Метод	Точность	Раундов до 75%	Доп. вычисления	Byzantine-устойчивость
FedAvg	72.1	145	Базовый	Нет
FedProx	76.3	130	+0.5% (μ-штраф)	Нет
SCAFFOLD	81.2	105	+5% (вариаты)	Частично
FedNova	79.0	115	+1% (норм.)	Нет
FedOpt	83.1	95	+2% (оптим.)	Нет
FedDF	77.4	125	+15% (дист.)	Нет
Krum	78.5	130	+8% (отбор)	Да (при <30%)
FLTrust	82.0	110	+3% (оценка)	Да (при <40%)
LoRA-FL	85.2	80	+20% (PEFT)	Нет
HierFAVG	78.2	100	+10% (иерарх.)	Нет

Заключение

Во-первых, по качественным показателям лучше всего себя показали FedOpt и SCAFFOLD обеспечивая лучшую точность (больше на 9–11%), а также имеют при этом более быструю сходимость (быстрее на 40–50 раундов).

Во-вторых, насущной проблемой для системы все также остаются атаки типа Byzantine, в первую очередь влияя на точность результирующей точности. И среди прочих методов лучше всего себя показал FLTrust, сохранив 68% точности при 30% злонамеренных клиентов. В то время как традиционные методы не справляются уже при 10–15%.

В-третьих, наилучшую скорость обучения показал метод LoRA-FL во много раз превзойдя другие методы, что делает его практически единственным методом для обучения таких моделей как LLM, включающий большое количество параметров.

Подводя итоги, стоит отметить, что из всех рассмотренных методов не было характерного претендента, отвечающий всем требованиям сразу: безопасность от Byzantine-атак, точность, скорость. Что может в свою очередь определять основные направления развития методов в модели федеративного обучения.

Литература / References

1. A Comprehensive Survey on Client Selection Strategies in Federated Learning // Computer Networks. – 2024. – Vol. 242. – P. 109–124.

2. A Contemporary Survey of Recent Advances in Federated Learning // Computer Science Review. – 2024. – Vol. 51. – URL: https://www.sciencedirect.com/science/article/pii/S2542660524001926 (дата обращения: 23.04.2026).

3. A Survey of Security Strategies in Federated Learning // MDPI Sensors. – 2024. – Vol. 24, № 5. – URL: https://www.mdpi.com/2998474 (дата обращения: 23.04.2026).

4. A Survey on Efficient Federated Learning Methods for Foundation Model Training // International Joint Conference on Artificial Intelligence (IJCAI). – 2024. – URL: https://www.ijcai.org/proceedings/2024/0919 (дата обращения: 23.04.2026).

5. A Systematic Review of Federated Generative Models // arXiv preprint arXiv:2405.16682. – 2024.

6. A Systematic Review of Federated Learning: Aggregation Methods // Journal of Network and Computer Applications. – 2023. – Vol. 218. – URL: https://www.sciencedirect.com/science/article/pii/S1084804523001339 (дата обращения: 23.04.2026).

7. Cao, X. FLTrust: Byzantine-robust Federated Learning via Trust Bootstrapping / X. Cao [et al.] // Network and Distributed System Security Symposium (NDSS). – 2022.

8. Emerging Trends in Federated Learning: From Model Fusion to Federated X Learning // International Journal of Machine Learning and Cybernetics. – 2024. – Vol. 15. – P. 451–468.

9. Federated Learning: A Cutting-Edge Survey of the Latest Advancements // Computer Communications. – 2024. – Vol. 215. – P. 112–125.

10. Federated Learning: Overview, Strategies, Applications and Tools // IEEE Access / PMC. – 2024. – Vol. 12. – URL: https://pmc.ncbi.nlm.nih.gov/ articles/PMC11466570/ (дата обращения: 23.04.2026).

11. Li, T. Federated Optimization in Heterogeneous Networks (FedProx) / T. Li [et al.] // Proceedings of machine learning and systems (MLSys). – 2020. – Vol. 2. – P. 429–450.

12. McMahan, H. B. Communication-Efficient Learning of Deep Networks from Decentralized Data / H. B. McMahan [et al.] // Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). – 2017. – P. 1273–1282.

13. Recent Advancements in Federated Learning: State of the Art // MDPI Electronics. – 2024. – Vol. 13, № 8. – URL: https://www.mdpi.com/3035012 (дата обращения: 23.04.2026).

ЧАСТИКОВА Вера Аркадьевна, кандидат технических наук, доцент кафедры кибербезопасности и защиты информации федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный технологический университет». 350072, Краснодарский край, г. Краснодар, ул. Московская, д. 2. E-mail: chastikova_va@mail.ru

ВАЛЕЕВ Тамерлан Назирович, магистр кафедры кибербезопасности и защиты информации федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный технологический университет». 350072, Краснодарский край, г. Краснодар, ул. Московская, д. 2. E-mail: tamer.valeev2014@yandex.ru

CHASTIKOVA Vera Arkadyevna, Candidate of Technical Sciences, Associate Professor of the Department of Cybersecurity and Information Protection at the Federal State Budgetary Educational Institution of Higher Education “Kuban State Technological University”. 350072, Krasnodar Krai, Krasnodar, Moskovskaya St., 2. E-mail: chastikova_va@mail.ru

VALEEV Tamerlan Nazirovich, Master’s Degree Student of the Department of Cybersecurity and Information Protection at the Federal State Budgetary Educational Institution of Higher Education “Kuban State Technological University”. 350072, Krasnodar Krai, Krasnodar, Moskovskaya St., 2. E-mail: tamer.valeev2014@yandex.ru

МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ, ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ

УДК 004.056

Вестник УрФО № 3(61) / 2026, с. 30–47
DOI: 10.14529/secur260303

Мельников Д.Ю., Синадский Н.И.

РЕАЛИЗАЦИЯ СИСТЕМЫ ОБНАРУЖЕНИЯ КОМПЛЕКСНЫХ КОМПЬЮТЕРНЫХ АТАК, ОСНОВАННОЙ НА АНАЛИЗЕ ДЕЙСТВИЙ ПОЛЬЗОВАТЕЛЕЙ, СЕТЕВЫХ ВЗАИМОДЕЙСТВИЙ И СОПОСТАВЛЕНИИ ОБНАРУЖЕННЫХ АНОМАЛИЙ С ТАКСОНОМИЕЙ ТЕХНИК КОМПЬЮТЕРНЫХ АТАК

Melnikov D.Y., Sinadsky N.I.

IMPLEMENTATION OF A SYSTEM FOR DETECTING COMPLEX COMPUTER ATTACKS BASED ON ANALYSIS OF USER ACTIVITY, NETWORK INTERACTIONS, AND COMPARISON OF DETECTED ANOMALIES WITH THE TAXONOMY OF COMPUTER ATTACK TECHNIQUES

В статье представлена реализация системы обнаружения комплексных компьютерных атак, основанная на анализе действий пользователей, сетевых взаимодействий и сопоставлении обнаруженных аномалий с таксономией компьютерных атак. В системе реализованы модуль формирования и поддержки профиля действий пользователя и многоканальный детектор аномалий, основанный на статистическом и графовом анализе с использованием эвристических правил. Для анализа связанных проявлений активности используется механизм группировки событий, а их интерпретация выполняется модулем сопоставления с таксономией компьютерных атак. Анализ объединённых записей позволяет реконструировать наиболее вероятную последовательность техник злоумышленника. Подсистема хранения данных реализована на основе NoSQL-СУБД «Apache Cassandra». В статье приведён пример работы системы на анонимизированном наборе журналов реального инцидента. Результаты демонстрируют возможность реконструкции временно распределённых цепочек атак в условиях фоновой легитимной активности и неполноты данных.

Ключевые слова: комплексные компьютерные атаки, обнаружение аномалий, профиль действий пользователя, Apache Cassandra, графовые сверточные сети (GCN), MITRE ATT&CK.

The article presents the implementation of a system for detecting complex computer attacks based on the analysis of user actions, network interactions, and the comparison of detected anomalies with the taxonomy of computer attacks. The system implements a module for forming and maintaining a user activity profile and a multi-channel anomaly detector based on statistical and graph analysis using heuristic rules. An event grouping mechanism is used to analyze related activity manifestations, and their interpretation is performed by a module that compares them with the taxonomy of computer attacks. Analysis of the combined records allows the most likely sequence of techniques used by the attacker to be reconstructed. The data storage subsystem is implemented based on the NoSQL DBMS «Apache Cassandra». The article provides an example of the system's operation on an anonymized set of logs from a real incident. The results demonstrate the possibility of reconstructing temporarily distributed attack chains in conditions of background legitimate activity and incomplete data.

Keywords: complex computer attacks, anomaly detection, user behavior profiling, Apache Cassandra, graph convolutional networks (GCN), MITRE ATT&CK.

Введение

Современные информационные системы все чаще становятся объектами комплексных компьютерных атак [1]. Одной из проблем при обнаружении комплексных атак является необходимость совместного анализа разнородных данных и установления логических связей между отдельными аномальными проявлениями. Приказ ФСБ России № 554 [2] устанавливает новые требования к средствам, предназначенным для обнаружения предупреждения и ликвидации последствий компьютерных атак, а также для поиска признаков компьютерных атак. Системы обнаружения компьютерных атак (далее – СОА) должны реализовывать функции сбора и автоматического анализа событий информационной безопасности (далее – ИБ), корреляцию разнородных событий, а также повторный анализ ранее зарегистрированных событий с целью выявления ранее не обнаруженных компьютерных инцидентов.

Вопросам обнаружения компьютерных атак (далее – КА) и анализа событий информационной безопасности посвящены ряд исследований [3,4,5]. В работах [3,4] рассматриваются методы выявления многошаговых компьютерных атак, основанные на анализе системных событий и выявлении последовательностей взаимосвязанных действий злоумышленника. Предложены подходы к обнаружению многоэтапных атак, использующие методы машинного обучения и механизмы представления знаний для анализа событий безопасности и идентификации сценариев КА. Данные методы ориентированы на выявление взаимосвязанных событий и последовательности действий нарушителя в информационной системе. В работе [5] рассматриваются архитектурные и функциональные аспекты построения подсистем предупреждения компьютерных атак на объектах критической информационной инфраструктуры, основанных на сборе, корреляции и анализе событий информационной безопасности. Представленные исследования демонстрируют перспективность применения методов анализа событий и корреляционного анализа для выявления комплексных компьютерных атак (далее – ККА). Вместе с тем, задача обнаружения ККА, остаётся актуальной и требует дальнейшего развития методов анализа событий безопасности и построения СОА.

В предыдущих работах автора представлены основные элементы подхода к обнаружению ККА, основанного на анализе действий пользователей, сетевых взаимодействий и сопоставлении обнаруженных аномалий с таксономией компьютерных атак. В работе [6] представлена модель ККА, представляющая собой последовательность групп событий, соответствующих конкретной тактике из базы знаний «MITRE ATT&CK», также описана модель обнаружения ККА. Описан процесс построения профиля нормальных действий пользователей, а также его динамическое обновление. Обнаружение аномалий осуществляется с помощью метода машинного обучения – автокодировщика и графового анализа на основе графовой сверточной сети (GCN). Разработан и описан механизм корреляции аномалий в адаптивной временной области. В работе [7] предлагается модель представления данных в NoSQL-СУБД «Apache Cassandra» для системы обнаружения ККА и сформулированы шесть целевых требований к системе хранения данных. Выполнены расчеты объемов хранения и размеров логически связанных групп записей, объединённых общим ключом и размещаемых на отдельных узлах распределённого хранилища «Apache Cassandra». В работе [8] описана си-

стема, состоящая из четырех ключевых алгоритмов, обеспечивающих реализацию модели обнаружения ККА. Формализована внутренняя логика каждого алгоритма, описаны процедуры калибровки, а также критерии принятия решений.

Настоящая статья описывает реализованный программный комплекс и иллюстрирует его работу на анонимизированном наборе журналов реального инцидента. Целью исследования является демонстрация корректности проектных решений и подтверждение архитектурной согласованности системы при восстановлении последовательности действий злоумышленника.

В тексте подробно рассмотрены механизмы обмена данными между компонентами, включая сквозные идентификаторы, хранение векторных представлений событий, ведение версий профилей и фиксацию метаданных происхождения записей. Кроме того, показаны процедуры формирования и агрегирования частичных оценок аномальности по аналитическим каналам, пакетная проверка накопленных групп событий и их семантическая интерпретация с сопоставлением к таксономии компьютерных атак.

1. Архитектура и функциональные блоки программного комплекса

Система реализована как программный комплекс, обеспечивающий последовательную обработку журналов событий и данных мониторинга информационной инфраструктуры, начиная с приёма первичных записей и заканчивая формированием инцидентов и сохранением результатов анализа. Архитектура системы ориентирована не на выявление отдельных изолированных аномалий, а на обнаружение логически связанных действий злоумышленника в компонентах информационной инфраструктуры. Архитектура программного комплекса предусматривает централизованное хранение всех категорий данных в NoSQL‑СУБД «Apache Cassandra». В указанном хранилище консолидируются первичные записи журналов событий, версионные профили действия субъектов, векторные представления событий, агрегированные группы аномалий, а также итоговые инциденты информационной безопасности. Архитектура системы обнаружения комплексных компьютерных атак представлена на рис. 1.



Рис. 1. Архитектура системы обнаружения комплексных компьютерных атак

Модуль обработки событий (Event Processing Module) принимает исходные данные из различных источников: журналы аутентификации, системные журналы и сетевые потоки. Модуль приводит записи к унифицированному формату, синхронизирует временные метки и выполняет проверку структурной корректности записей и логической согласованности их атрибутов. После проверки к каждой записи добавляются метаданные, содержащие сведения о источнике данных и оценку его надёжности. Нормализованные записи сохраняются в основном хранилище и формируют исходную базу для последующего анализа.

Обработка данных осуществляется в двух режимах. В потоковом режиме реализуются операции с низкой латентностью: первичная нормализация, вычисление основных характеристик и оперативная оценка аномальности. В пакетном режиме выполняются более ресурсоёмкие задачи: повторная нормализация агрегатов, углублённый пересчёт статистических метрик и процедуры переобучения моделей. Такое разделение обеспечивает оперативность реакции на события и позволяет проводить детальный ретроспективный анализ гипотез.

Центральным компонентом является подсистема формирования и поддержки профиля действий пользователя (ProfileManager). Профиль представляет собой агрегированное представление типичного поведения субъекта в

системе. Изменения профиля фиксируются с сохранением истории обновлений. Каждое обновление сопровождается временной отметкой и перечнем наблюдений, послуживших основанием для внесённых изменений. Обновление базовой модели профиля осуществляется только после накопления достаточного числа подтверждающих наблюдений; единичные аномалии не интегрируются немедленно. Это снижает риск целенаправленного искажения профиля.

Локальное обнаружение реализовано через многоканальный детектор (MultiChannelDetector). Детектор объединяет три аналитических канала:

– эвристический канал реализует набор правил и индикаторов для оперативного выявления явных признаков компрометации;

– статистический канал оценивает отклонение текущих признаков от профиля действий. Автокодировщик обучается на наборе данных нормальных действий пользователей. Ошибка реконструкции преобразуется в эмпирический перцентиль относительно распределения ошибок на обучающей выборке и используется как частичная оценка аномальности;

– графовый канал анализирует локальную структуру взаимодействий, вычисляя локальные графовые метрики и формируя векторные представления узлов с помощью графовой сверточной сети (GCN). Полученные векторные представления отражают контекст взаимодействий и применяются для вычисления графовой оценки аномальности.

Каждый канал возвращает стандартизованную частичную оценку аномальности и сопроводительные метаданные. Частичные оценки объединяются агрегатором по настраиваемой политике с конфигурируемыми весами и порогами, что позволяет регулировать соотношение полноты и точности обнаружения без изменения внутренней логики отдельных каналов.

События, превышающие порог аномальности, направляются в модуль группировки аномалий (GroupingEngine). Его задачей является адаптивное объединение связанных по времени, субъектам, узлам или семантическому контексту записей в устойчивые группы. Механизм группировки предусматривает частичное несоответствие отдельных атрибутов и опирается на комбинированные меры сходства. Для каждой группы поддерживается состояние активности. Группа может продлеваться при поступлении подтверждающих сигналов и закрываться при длительной неактивности. Сформированные группы представляют собой промежуточные аналитические объекты и подлежат последующей пакетной проверке.

Для семантической интерпретации групп используется модуль сопоставления аномалий с таксономией компьютерных атак (TaxonomyMatcher). На первом этапе для каждого события формируются признаки (текстовые, структурные и контекстно-графовые) и измеряется степень сходства между этим представлением события и описаниями техник таксономии. Полученные сходства преобразуются в нормализованные оценки соответствия. На втором этапе оценки соответствия агрегируются по группе событий и применяются совместно с моделью вероятностей переходов между техниками для определения последовательности техник. В результате модуль возвращает упорядоченный список техник с оценками уверенности и комментариями.

В качестве базы данных используется распределённая NoSQL-СУБД «Apache Cassandra». В базе данных сохраняются первичные записи, версии профилей, векторные представления событий, записи взаимодействий, агрегаты групп и итоговые инциденты. Объединение записей по заданному критерию обеспечивает равномерное распределение нагрузки и масштабируемость системы. В работе [2] приведены расчёты объёмов хранения и рекомендации по выбору схем объединения записей.

Помимо основных аналитических модулей, комплекс включает вспомогательные компоненты, обеспечивающие хранение векторных представлений, учёт связей между сущностями и формирование оповещений, что гарантирует воспроизводимость расчётов и интеграцию результатов в процесс реагирования. Архитектура комплекса предусматривает фиксацию идентификаторов и ссылочных связей между результатами обработки на различных этапах, обеспечивая уникальное и воспроизводимое сопоставление результатов анализа. В табл. 1 представлены обобщённые сведения по ключевым модулям системы обнаружения ККА.

Таблица 1

Обобщённая информация по ключевым модулям системы ККА

Модуль	**Назначение и функции**
Модуль обработки событий (Event Processing Module)	Обеспечивает приём событий безопасности из различных источников и их приведение к единому внутреннему представлению. Выполняет проверку временной согласованности записей, контроль структуры и содержательной согласованности событий, а также формирование

	набора атрибутов, используемых на последующих этапах анализа
Модуль формирования и поддержки профиля действий пользователя (ProfileManager)	Поддерживает динамическое агрегированное представление действий пользователя
Многоканальный детектор (MultiChannelDetector)	Выполняет выявление аномальных отклонений на уровне отдельных событий с использованием нескольких аналитических каналов
Модуль группировки аномалий (GroupingEngine)	Объединяет связанные аномальные события в устойчивые группы, отражающие эпизоды подозрительной активности
Модуль сопоставления аномалий с таксономией компьютерных атак (TaxonomyMatcher)	Выполняет интерпретацию сгруппированных событий путём сопоставления наблюдаемых аномалий с таксономией компьютерных атак
База данных на основе NoSQL «Apache Cassandra»	Обеспечивает хранение исходных событий, состояний профилей действий пользователей, результатов аналитической обработки, групп событий и итоговых записей по инцидентам

2. Описание работы системы при обнаружении комплексной компьютерной атаки

2.1. Описание ККА

В рассматриваемом случае злоумышленник реализовал ККА. Начальный доступ был получен посредством целевого фишинга. Сотрудник организации открыл электронное сообщение с вредоносным вложением, в котором содержался макрос, обеспечивавший удалённое выполнение кода при открытии документа. В результате на одном из узлов сети инициировался запуск «PowerShell», был загружен и сохранён вспомогательный исполняемый файл, а для обеспечения устойчивости вредоносного компонента в системе была зарегистрирована новая служба. Эти действия отразились в системных журналах и в записях о файловых операциях.

На следующем этапе злоумышленник применил инструменты класса «mimikatz» для извлечения учётных данных из памяти процессов. Полученные учётные данные были использованы для латерального перемещения внутри корпоративной сети. Скомпрометированный узел инициировал удалённые подключения к другим узлам с помощью средств удалённого исполнения (включая PsExec и WinRM) и стандартных сервисов (RDP, SMB), что

обеспечило удалённый запуск процессов и перенос полезных нагрузок на целевые серверы.

Далее злоумышленник выполнял массовые операции чтения в файловых хранилищах и базах данных и применял утилиты архивирования для упаковки собранных данных. Для передачи подготовленных объёмов информации использовались кратковременные зашифрованные HTTPS-сессии, поток данных дробился на серию коротких соединений к внешнему адресу с целью маскировки активности. По завершении активной фазы были предприняты действия по сокрытию следов и обеспечению персистентности. Выполнялись операции удаления временных файлов, предпринимались попытки очистки журналов аудита и создавались задания в планировщике для возобновления работы вредоносного кода после перезагрузки узла.

Во время проведения ККА присутствовала легитимная активность пользователей, поэтому выявление атаки опиралось не на анализ отдельных записей, а на устранение повторяющихся наблюдений и установление логических связей между событиями. Продолжительность ККА составила около 96 часов. Скомпрометировано несколько узлов, объём собранных данных составил несколько десятков гигабайт. На рис. 2 представлено схема ККА.

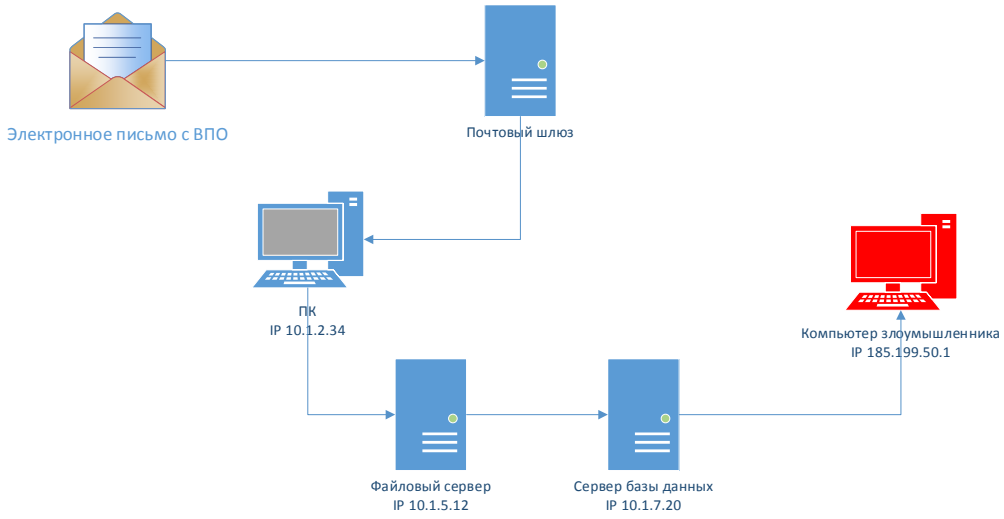


Рис. 2. Схематичное представление ККА

Следует отметить, что на момент реализации рассматриваемого сценария, КА не была обнаружена существующими средствами защиты. Это обусловлено тем, что отдельные действия злоумышленника по отдельности не формировали выраженных признаков КА. Используемые инструменты и процессы (например, «PowerShell», средства удалённого администрирования

и утилиты архивирования) применяются в легитимной деятельности администраторов и пользователей, вследствие чего их запуск не всегда интерпретируется как признак КА. Кроме того, действия злоумышленника были распределены во времени и выполнялись на различных узлах инфраструктуры, а соответствующие события фиксировались различными источниками данных. Ограниченные возможности используемых средств защиты информации, а также особенности их конфигурации могут дополнительно затруднять выявление взаимосвязанных последовательностей действий злоумышленника. В результате анализ отдельных журналов событий не позволял установить логические связи между наблюдаемыми событиями и представить полный сценарий ККА.

2.2. Описание работы системы при обработке комплексной компьютерной атаки.

В данном разделе демонстрируется функционирование программного комплекса при ретроспективной обработке журналов событий, в том числе инцидента информационной безопасности, экспортированных из информационной инфраструктуры предприятия. Все идентифицирующие сведения далее в тексте анонимизированы для сохранения конфиденциальности. Данные взяты из журналов аутентификации, системных журналов узлов сети, журналов активных процессов и сетевого трафика. Набор событий и их временная последовательность сохранены в полном объёме.

Аналитические модули системы обучались и калибровались на наборе журналов событий нормальных действий пользователей информационной инфраструктуры (31 день записи журналов событий, порядка 300 уникальных учётных записей, 1,86 млн. событий), что позволило построить векторные представления событий, выполнить калибровку порогов автокодировщика и настроить перцентильную нормализацию. В табл. 2 представлены ключевые анонимизированные события, используемые для демонстрации корреляции и анализа сценария атаки.

Таблица 2

Ключевые события

№	Время (UTC)	Источник данных	Event ID	Тип события	Процесс/ приложение	Пользователь	SrcIP / DstIP
1	2025-11-08 18:47:22	Mail gateway	mail-003	AttachmentOpen	WINWORD.EXE	DOMAIN\employeeA	— / —

Окончание табл. 2

2	2025-11-08 19:03:41	Endpoint agent	4688	ProcessStart	PowerShell.exe	DOMAIN\employeeA	10.1.2.34 / —
3	2025-11-08 19:04:10	Endpoint agent	4688	ProcessStart	payload.exe	DOMAIN\employeeA	10.1.2.34 / —
4	2025-11-08 19:05:00	Windows Event	7045	ServiceInstall	svc_persist	DOMAIN\employeeA	10.1.2.34 / —
5	2025-11-10 09:12:34	Endpoint agent	4688	ProcessStart	PowerShell.exe	DOMAIN\employeeA	10.1.2.34 / —
6	2025-11-10 11:05:12	Endpoint agent	4688	ProcessStart	mimikatz.exe	DOMAIN\employeeA	10.1.2.34 / —
7	2025-11-11 02:18:10	Endpoint agent	4688	ProcessStart	PsExec.exe	DOMAIN\employeeA	10.1.2.34 / 10.1.5.12
8	2025-11-11 06:42:01	Endpoint agent	4688	ProcessStart	7z.exe	DOMAIN\dbsvc	10.1.7.20 / —
9	2025-11-11 07:58:12	Proxy / network flow	nf-00160	NetworkConn	curl.exe	DOMAIN\dbsvc	10.1.7.20 / 185.199.50.1
10	2025-11-11 09:10:00	Endpoint agent	4688	ProcessStart	PowerShell.exe	DOMAIN\svc_admin	10.1.7.20 / —
11	2025-11-11 09:20:10	Windows Event	1102	Audit-LogClear	wevtutil.exe	DO-MAIN\svc_admin	10.1.7.20 / —
12	2025-11-12 23:01:18	Endpoint agent	4688	ProcessStart	schtasks.exe	DOMAIN\svc_admin	10.1.7.20 / —

Далее последовательно описаны этапы атаки и действия компонентов системы. Каждая стадия снабжена ссылкой на соответствующие записи в табл. 2.

Начальные проявления включали открытие целевого почтового вложения (табл. 2, № 1), после чего был зафиксирован нетипичный запуск интерпретатора сценариев на рабочей станции (табл. 2, № 2). Непосредственно после этого зарегистрирован запуск загруженного вспомогательного исполня-

емого файла (табл. 2, № 3) и запись о создании/регистрации службы, обеспечивающей устойчивость вредоносного компонента (табл. 2, № 4). После приёма и нормализации событий модуль приёма передал записи обработчику признаков, который сформировал векторные представления событий. Эвристический канал зарегистрировал нетипичную активность интерпретатора, статистический канал зафиксировал отклонение от профиля действий пользователя, а подсистема формирования и поддержки профиля аккумулировала подтверждающие наблюдения и приостановила автоматическое обновление базовой модели до поступления дополнительных подтверждений. На рис. 3 представлено схематичное описание работы системы при обработке ККА.

	События № 1-4	События № 5-6	Событие № 7	События № 8-9	События № 10-12
Модуль обработки событий (Event Processing Module)	Приём событий, нормализация	Приём событий, нормализация	Приём событий, нормализация	Приём событий, нормализация	Приём событий, нормализация
Модуль формирования и поддержки профиля действий пользователя (ProfileManager)	Фиксация первичных отклонений, приостановка автообновления профиля	Подтверждение устойчивых отклонений, маркировка аномальной учётной записи	Регистрация активности на новом узле, обновление профиля действий	Выявление аномальной активности сервисной учётной записи	Регистрация действий, подготовка профиля к архивации
Многоканальный детектор (MultiChannelDetector)	Определение оценок отклонений нетипичных действий	Фиксация высокой ошибки реконструкции, агрегация оценок	Перерасчёт топологических метрик, фиксация новых рёбер, повышение вклада графового канала	Фиксация массового архивирования и внешних соединений	Фиксация очистки журналов аудита и персистентности
Модуль группировки аномалий (GroupingEngine)	Накопление первичных событий	Объединение событий № 1-6 в единый агрегат	Расширение существующего агрегата, перерасчёт веса	Пакетная обработка агрегата, пересчёт метрик достоверности; привязка сетевых потоков к агрегату	Присоединение событий № 10-12 к основному агрегату, добавление метаданных
Модуль сопоставления аномалий с таксономией компьютерных атак (TaxonomyMatcher)		Сопоставление с техниками	Добавление тактик	Сопоставление с техниками и тактиками	Итоговое сопоставление с техниками и тактиками
База данных на основе NoSQL «Apache Cassandra»	Сохранение необработанных данных и первичных оценок	Сохранение аномалий, обновление профиля, фиксация агрегата	Обновление графовых представлений, расширение агрегата	Сохранение записей и финальных оценок агрегата	Завершение записи агрегата инцидента, подготовка данных для уведомления

Рис. 3. Описание работы системы при обработке ККА

Далее в журнале появились события, характерные для использования утилит извлечения учётных данных (табл. 2, № 6). Для соответствующих процессов были сформированы векторные представления. Автокодировщик зафиксировал повышенную ошибку реконструкции, эвристические правила дали высокую оценку по индикаторам компрометации, а подсистема профилей классифицировала учетную запись как потенциально скомпрометированную.

Скомпрометированные учётные данные использовались для удалённых подключений и запуска процессов на других узлах сети (табл. 2, № 7). Графовый модуль пересчитал локальные топологические признаки, сохранил

векторные представления узлов и зафиксировал сетевые взаимодействия. Повышение значений локальных графовых метрик инициирующего узла усилили вклад графового канала в суммарную оценку, после чего механизм группировки аккумулировал связанные события в агрегат для пакетной обработки.

На серверах были зарегистрированы массовые операции чтения и архивации данных, (табл. 2, № 8). По сформированному агрегату была выполнена пакетная обработка: повторная нормализация признаков, пересчёт метрик автокодировщика и уточнение векторных представлений, что привело к согласованию частичных оценок всех каналов и повышению достоверности агрегата.

Краткие зашифрованные исходящие соединения и операции по архивированию данных, позволили модулю корреляции привязать сетевые потоки к ранее сформированному агрегату (табл. 2, № 9). Перед операциями по очистке журналов зафиксирован запуск команд/скриптов удаления временных файлов (табл. 2, № 10), затем очистка журналов аудита (табл. 2, № 11) и создание задания планировщика (табл. 2, № 12). Эти события снабжены метаданными происхождения и связаны с сформированным уведомлением. Отдельные события с высокой эвристической оценкой подвергались ручной валидации в контексте совокупных доказательств для минимизации риска ложных срабатываний.

Для формализованного описания сценария ККА выполнено сопоставление выявленных этапов атаки с тактиками и техниками базы знаний «MITRE ATT&CK» и соответствующими тактикам из методики оценки угроз ФСТЭК России, представленное в табл. 3.

Таблица 3

Сопоставление этапов ККА с базой знаний «MITRE ATT&CK» и тактиками методики оценки угроз ФСТЭК России

Краткое описание этапа	База знаний «MITRE ATT&CK»	Тактики модели угроз ФСТЭК России
Целенаправленный фишинг с вложением	T1566.001 «Spearphishing Attachment»	Т2.8 «Использование методов социальной инженерии, в том числе фишинга, для получения прав доступа к компонентам системы»
Выполнение PowerShell-	T1059.001 «Command and Scripting	Т3.1 «Автоматический запуск скриптов и исполняемых файлов

скриптов/команд на рабочей станции	Interpreter: PowerShell»	(включая использование скриптов/интерпретаторов)»
Регистрация службы, создание задания планировщика	T1543.003 «Create or Modify System Process: Windows Service»; T1053.005 «Scheduled Task/Job»	Т3.14 «Планирование запуска вредоносных программ при старте системы (правка реестра и автозапуск, в том числе запуск как сервис)»; Т3.15 – «Планирование запуска через планировщик заданий»
Извлечение паролей/учётных данных из памяти	T1003 «OS Credential Dumping»	Т6.1 «Получение данных для аутентификации и авторизации –поиск в памяти процессов или перехват в трафике»
Удалённый запуск процессов/подключения к другим хостам (PsExec, RDP, SMB)	T1021 «Remote Services»	Т5.1 «Удалённое управление через стандартные протоколы; использование средств удалённого администрирования»
Чтение больших объёмов, архивация данных	T1074 «Data Staged»	Т7.17 «Обфускация, шифрование, упаковка (защита паролем) при подготовке данных/кода»
Передача данных за пределы сети через короткие шифрованные соединения	T1041 «Exfiltration Over C2 Channel»	Т9.5 / Т9.7 «Отправка данных по известным протоколам / маскировка каналов передачи, проксирование трафика для маскировки»
Очистка журналов, удаление временных файлов	T1070 «Clear Windows Event Logs»	Т7.2 «Очистка/затирание истории команд и журналов регистрации»

Представленные результаты в табл. 3 отражают последовательность взаимосвязанных событий, выявленных системой в процессе анализа данных, соответствующая сценарию ККА.

Таким образом, предложенная система обеспечивает выявление распределённой во времени последовательности взаимосвязанных действий злоумышленника, которые по отдельности не формируют выраженных признаков КА. Полученный результат демонстрирует возможность обнаружения

ранее не выявленных компьютерных атак на основе анализа совокупности событий ИБ, что соответствует нормативно-правовым требованиям, предъявляемым к СОА [2].

Заключение

В данной работе представлена система обнаружения ККА, основанная на анализе действий пользователей, сетевых взаимодействий и сопоставлении обнаруженных аномалий с таксономией компьютерных атак. Система включает механизм динамической профилизации с накоплением данных по времени и подтверждением внесённых изменений, многоканальный детектор с автокодировщиком и графовым анализом на основе GCN, а также модуль группировки аномалий, функционирующие на базе NoSQL-СУБД «Apache Cassandra».

Экспериментальная апробация системы выполнена на анонимизированном наборе журналов событий, включающем ретроспективный разбор ККА. Полученные результаты иллюстрируют способность предлагаемых механизмов аккумулировать различные признаки, формировать и агрегировать частичные оценки аномальности по аналитическим каналам и восстанавливать логически связанные последовательности техник злоумышленника при пакетной проверке накопленных групп событий. Совокупное применение модели и алгоритмов обнаружения ККА с хранилищем данных на основе NoSQL-СУБД «Apache Cassandra» обеспечивает возможность анализа событий, сопоставления разнородных данных и формирование уведомлений о ККА на основе совокупных признаков.

Вместе с тем, ограничением подхода является зависимость качества обнаружения от полноты и достоверности исходных журналов событий, а также необходимость операционной калибровки параметров моделей при внедрении в конкретную информационную инфраструктуру. Дальнейшие исследования связаны с расширением обучающих выборок, развитием графовых методов анализа и повышением устойчивости системы к изменению паттернов действий пользователей.

Литература

1. CODE RED 2026: Актуальные киберугрозы для российских организаций – URL: https://ptsecurity.com/research/analytics/russia-cyberthreat-landscape-2026/#id1 (дата обращения: – 27.02.2026).

2. Приказ ФСБ России № 554 – URL: https://gossopka.ru/doc/npa/normativnye-pravovye-akty-federalnykh-organov-ispolnitelnoy-vlasti/prikaz-fsb-

rossii-ot-26-dekabrya-2025g-554-ob-ustanovlenii-trebovaniy-k-sredstvam-prednaznachennym-d-33266/ (дата обращения: – 27.02.2026).

3. Котенко, И. В. Методы интеллектуального анализа системных событий для обнаружения многошаговых кибератак: использование методов машинного обучения / И. В. Котенко, Д. А. Левшун // Искусственный интеллект и принятие решений. – 2023. – № 3. – С. 3–15. – DOI 10.14357/20718594230301. – EDN SNNJFQ.

4. Котенко, И. В. Методы интеллектуального анализа системных событий для обнаружения многошаговых кибератак: использование баз знаний / И. В. Котенко, Д. А. Левшун // Искусственный интеллект и принятие решений. – 2023. – № 2. – С. 3–14. – DOI 10.14357/20718594230201. – EDN WOERTP.

5. Подсистема предупреждения компьютерных атак на объекты критической информационной инфраструктуры: анализ функционирования и реализации / И. В. Котенко, И. Б. Саенко, Р. И. Захарченко, Д. В. Величко // Вопросы кибербезопасности. – 2023. – № 1(53). – С. 13–27. – DOI 10.21681/2311-3456-2023-1-13-27. – EDN WUUDFK.

6. D. Melnikov, «A Model for Detecting Complex Computer Attacks Based on the Analysis of User Actions, Network Interactions and Comparison with the MITRE ATT&CK Knowledge Base», 2025 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russian Federation, 2025, pp. 221-224, DOI: 10.1109/USBEREIT65494.2025.11054096.

7. Мельников, Д. Ю. Модель хранения данных в nosql-СУБД «Apache Cassandra» в задачах обнаружения комплексных компьютерных атак / Д. Ю. Мельников // Вестник УрФО. Безопасность в информационной сфере. – 2025. – № 3(57). – С. 30–39. – DOI 10.14529/secur250304. – EDN ASFKNG.

8. Мельников, Д. Ю. Система алгоритмов обнаружения комплексных компьютерных атак на основе анализа действий пользователей и сопоставления с базой знаний MITRE ATT&CK / Д. Ю. Мельников // Вестник УрФО. Безопасность в информационной сфере. – 2025. – № 4(58). – С. 44–52. – DOI 10.14529/secur250405. – EDN UUHFSA.

References

1. CODE RED 2026: Aktual'nye kiberugrozy dlja rossijskih organizacij – URL: https://ptsecurity.com/research/analytics/russia-cyberthreat-landscape-2026/#id1 (accessed: - 27.02.2026).

2. Prikaz FSB Rossii № 554 – URL: https://gossopka.ru/doc/ npa/normativnye-pravovye-akty-federalnykh-organov-ispolnitelnoy-vlasti/ prikaz-fsb-rossii-ot-26-dekabrya-2025g-554-ob-ustanovlenii-trebovaniy-k-sredstvam-prednaznachennym-d-33266/ (accessed: – 27.02.2026).

3. Kotenko, I. V. Metody intellektual'nogo analiza sistemnyh sobytij dlja obnaruzhenija mnogoshagovyh kiberatak: ispol'zovanie metodov mashinnogo obuchenija / I. V. Kotenko, D. A. Levshun // Iskusstvennyj intellekt i prinjatie reshenij. – 2023. – № 3. – S. 3–15. – DOI 10.14357/20718594230301. – EDN SNNJFQ.

4. Kotenko, I. V. Metody intellektual'nogo analiza sistemnyh sobytij dlja obnaruzhenija mnogoshagovyh kiberatak: ispol'zovanie baz znanij / I. V. Kotenko, D. A. Levshun // Iskusstvennyj intellekt i prinjatie reshenij. – 2023. – № 2. – S. 3–14. – DOI 10.14357/20718594230201. – EDN WOERTP.

5. Podsistema preduprezhdenija komp'juternyh atak na ob#ekty kriticheskoj informacionnoj infrastruktury: analiz funkcionirovanija i realizacii / I. V. Kotenko, I. B. Saenko, R. I. Zaharchenko, D. V. Velichko // Voprosy kiberbezopasnosti. – 2023. – № 1(53). – S. 13–27. – DOI 10.21681/2311-3456-2023-1-13-27. – EDN WUUDFK.D. Melnikov, «A Model for Detecting Complex Computer Attacks Based on the Analysis of User Actions, Network Interactions and Comparison with the MITRE ATT&CK Knowledge Base», 2025 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (US-BEREIT), Yekaterinburg, Russian Federation, 2025, pp. 221-224, doi: 10.1109/USBEREIT65494.2025.11054096.

6. Mel'nikov, D. Ju. Model' hranenija dannyh v nosql-SUBD «Apache Cassandra» v zadachah obnaruzhenija kompleksnyh komp'juternyh atak / D. Ju. Mel'nikov // Vestnik UrFO. Bezopasnost' v informacionnoj sfere. – 2025. – № 3(57). – S. 30–39. – DOI 10.14529/secur250304. – EDN ASFKNG.

7. Mel'nikov, D. Ju. Sistema algoritmov obnaruzhenija kompleksnyh komp'juternyh atak na osnove analiza dejstvij pol'zovatelej i sopostavlenija s bazoj znanij MITRE ATT&CK / D. Ju. Mel'nikov // Vestnik UrFO. Bezopasnost' v informacionnoj sfere. – 2025. – № 4(58). – S. 44–52. – DOI 10.14529/ secur250405. – EDN UUHFSA.

МЕЛЬНИКОВ Дмитрий Юрьевич, аспирант Учебно-научного центра «Информационная безопасность» федерального государственного автономного образовательного учреждения высшего образования «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина». 620002, г. Екатеринбург, ул. Мира, 32. E-mail: melnikovdima517@yandex.ru

СИНАДСКИЙ Николай Игоревич, доктор технических наук, профессор, профессор учебно-научного центра «Информационная безопасность», Институт радиоэлектроники и информационных технологий – РТФ Уральский федеральный университет им. первого Президента России Б.Н. Ельцина. 620002; г. Екатеринбург, ул. Мира, 19. E-mail: n.i.sinadsky@urfu.ru

MELNIKOV Dmitriy Yuryevich, graduate student at the Educational and Scientific Center «Information Security» of the Federal State Autonomous Educational Institution of Higher Education «Ural Federal University named after the first President of Russia B.N. Yeltsin». 620002, Yekaterinburg, st. Mira, 32. E-mail: melnikovdima517@yandex.ru

SINADSKY Nikolay, doctor of engineering, Professor, Educational and Scientific Center “Information Security”, Institute of Radio electronics and Information Technologies, Ural Federal University named after first President of Russia B.N. Yeltsin; 620002, Yekaterinburg, Mira str., 19. E-mail: n.i.sinadsky@urfu.ru

УДК 004.056.5 Вестник УрФО № 3(61) / 2026, с. 48–63
DOI: 10.14529/secur260304

Кожемяков Д.А., Поршнев С.В., Стольникова А.А., Хорошкеев Д.А.

КОМПЛЕКСНЫЙ ПОДХОД К ОЦЕНКЕ ЗАЩИЩЕННОСТИ ИНФОРМАЦИОННОЙ ИНФРАСТРУКТУРЫ НА БАЗЕ ASTRA LINUX: ОТ АНАЛИЗА ОС ДО ДИНАМИЧЕСКОГО ТЕСТИРОВАНИЯ ВЕБ-ПРИЛОЖЕНИЙ С ИИ

Kozhemyakov D.A., Porshnev S.V., Stolnikova A.A., Khoroshkeev D.A.

A COMPREHENSIVE APPROACH TO ASSESSING THE SECURITY OF IT INFRASTRUCTURE BASED ON ASTRA LINUX: FROM OS ANALYSIS TO DYNAMIC TESTING OF WEB APPLICATIONS WITH AI

В статье предложен комплексный подход к оценке защищённости информационной инфраструктуры на базе Astra Linux Special Edition, в основанный на результатах анализа нормативно-правовой базы, архитектурных механизмов защиты ОС (мандатного управления доступом *PARSEC*, контроля недекларированных возможностей), методов динамического тестирования угроз веб-приложений, в том числе, использующие генеративные ИИ-модели.

В предложенном подходе ИТ-инфраструктура рассматривается как единая система, для защиты которой совместно используются результаты сравнительного и структурно-функционального анализа нормативных требований, учитываются архитектурные особенности используемой ОС, а также особенности методов оценки защищённости ИТ-инфраструктуры и соответствующих программно-аппаратных средств , что отличает его от других известных подходов, которые ориентированы либо на формальную верификацию механизмов доверенных ОС, либо на обнаружение в сетевом трафике и веб-приложениях признаков компьютерных атак.

Можно ожидать, что предложенный подход окажется востребованным при разработке специализированных средств оценки защищённости и визуализации рисков для государственных информационных систем и объектов критической информационной инфраструктуры.

Ключевые слова: Astra Linux, генеративный ИИ, динамическое тестирование, мандатное управление доступом, моделирование угроз, недекларированные возможности программного обеспечения, оценка защищенности ИТ-инфраструктуры.

The article proposes a comprehensive approach to assessing the information infrastructure based on Astra Linux Special Edition security, based on the regulatory framework analysis results, the architectural protection mechanisms of the OS (mandatory access control PARSEC, control of undeclared capabilities) results, and the dynamic threat web applications testing methods, including those using generative AI models results.

In the proposed approach, the IT infrastructure is viewed as a unified single system, and its protection involves the joint use of the comparative and the regulatory requirements structural-functional analysis results. The architectural features of the operating system being used, as well as the specifics of methods for assessing the security of the IT infrastructure and the corresponding hardware and software tools, are taken into account. This distinguishes the approach from other well-known methods, which are focused either on the formal verification of trusted OS mechanisms or on detecting signs of computer attacks in network traffic and web applications.

It can be expected that the proposed approach will be in demand when developing specialized tools for assessing security and visualizing risks for state information systems and critical information infrastructure facilities.

Keywords: Astra Linux, security assessment, undeclared capabilities, dynamic testing, generative AI, mandatory access control, threat modeling.

Введение

В настоящее время в Российской Федерации действует Федеральный закон от 26 июля 2017г. №187-ФЗ «О безопасности критической информационной инфраструктуры», а также логически выстроенная система нормативно-правовых актов, принятых в рамках реализации данного ФЗ: ФЗ, Указов Президента РФ, Приказы и нормативные документы, разработанные регуляторами в области информационной безопасности, а также ведомственные документы субъектов КИИ.

К объектам критической информационной инфраструктуры также отнесены и государственные информационные системы (ГИС), к безопасности которых предъявляются повышенные требования на каждом из уровней обработки информации – от уровня операционной системы (ОС) до уровня прикладного программного обеспечения, включая веб-приложения. Правовое регулирование защиты объектов КИИ и ГИС образует многоуровневую

систему, включающую федеральные законы, указы Президента РФ и ведомственные акты. Ключевыми требованиями для инфраструктур на базе Astra Linux являются обязательность применения отечественных сертифицированных решений на значимых объектах КИИ, а также нормы приказов ФСТЭК и ФСБ России. За нарушение установлены административная и уголовная ответственность, что подчеркивает значимость нормативной базы как основы комплексной оценки защищенности.

Основной ОС, применяемой в объектах КИИ и ГИС, является ОС специального назначения Astra Linux, имеющая сертификаты соответствия ФСТЭК России (сертификат соответствия № 2557) и ФСБ России (сертификат соответствия № СФ/СЗИ-0680). Наличие данных сертификатов подтверждает соответствие ОС Astra Linux требованиям защиты информации ограниченного доступа, в том числе, сведений, составляющих государственную тайну.

Однако, как показывает практика, использование сертифицированной ОС, в ряде случаев может оказаться недостаточным условием защищенности КИИ и ГИС. Так, в 2025 г. в Astra Linux были выявлен и оперативно устранен ряд актуальных уязвимостей, способных вызвать отказ в обслуживании, наличие которых обусловлено тем, что сертификация, отнюдь, не гарантирует защиту от атак на прикладном уровне, в частности, от эксплуатации уязвимостей в веб-фреймворках, особенно, в условиях активизации использования при проведении компьютерных атак уязвимостей безопасности программных инструментов прикладного уровня, в первую очередь, веб-приложений, с цель проведения которых – разрушение ИТ-инфраструктуры. Это подтверждается, в том числе, результатами анализа компьютерных инцидентов, проведенного в [1], которые показали, что до 76% критических компьютерных инцидентов, произошедших в РФ, были направлены на разрушение ИТ-инфраструктуры.

При этом, очевидно, что в условиях непрерывного появления новых векторов компьютерных атак, для разработки которых, в том числе, применяются технологии искусственного интеллекта (ИИ), призванные обеспечить автоматизацию, персонализацию и адаптацию атак к выявленным уязвимостям систем защиты информации, а также генерацию вредоносного контента.

Среди исследований, посвященных оценке защищенности информационной инфраструктуры на базе Astra Linux, выделим следующие независимые

подходы: 1) разработка формальных моделей (модели доступа, модели политик безопасности и их верификация), начало разработки которых положено в известных работах П.Н. Девянина, которые стали научным базисом подсистемы PARSEC; 2) разработка и использование математических моделей известных компьютерных атак, а также анализ устойчивости ML-моделей IDS, обнаружение атак на веб-приложения и применение методов глубокого обучения в задачах киберзащиты (И.В. Котенко). В отличие от описанных выше подходов, в данной статье обосновывается целесообразность использования интегрированного подхода к оценке защищенности одновременно всей защищенной инфраструктуры на базе Astra Linux, рассматриваемой как единый объект, но каких-либо его выбранных подсистем.

Также известен опыт совместного использования для оценки безопасности ИТ-инфраструктуры на базе защищенной ОС Astra Linux вероятностных моделей (для оценки рисков отдельных компонентов) и теории графов (для выявления сложных сценариев компьютерных атак, обусловленных спецификой мандатного управления доступом) [2], которые, по сути, комплексными не являются.

Таким образом, задачи разработки комплексного подхода к оценке защищенности ИТ-инфраструктуры, основанного на анализе безопасности используемой ОС, динамическом тестировании веб-приложений и учете новых классов угроз, порождаемых технологиями ИИ, а также разработка соответствующего программного инструмента, обеспечивающего оценку защищенности ИТ-инфраструктуры и визуализацию рисков, являются актуальными.

Цель исследования – обоснование комплексного подхода к оценке защищенности ИТ-инфраструктуры на базе ОС Astra Linux Special Edition, основанного на рассмотрении его как целостного объекта защиты, а также результатах анализа встроенных механизмов защиты ОС и использовании методов динамического тестирования веб-приложений (black-box/white-box/grey-box), а также учете угроз безопасности информации, обусловленных использованием генеративных ИИ-моделей.

2. Анализ недекларируемых возможностей ОС Astra Linux

Основой безопасности ОС Astra Linux является мандатная сущностно-ролевая ДП-модель управления доступом (МРОСЛ ДП-модель), обеспечивающая мандатное управление доступом (МРД). Для использования технологий контейнерной виртуализации данная модель дополнена мандатным контролем целостности (МКЦ) [3], реализованным в подсистеме PARSEC. Модель

дополнена описанием МКЦ. Для защиты информации в ОС Astra Linux используются встроенные сертифицированные средства защиты, обеспечивающие поддержку замкнутой программной среды (ЗПС) [4].

Для информационных систем, обрабатывающих сведения, составляющие государственную тайну, обязательным является контроль отсутствия не декларированных возможностей (НДВ), под которым в соответствие с ГОСТ Р 56939-2024[1] под НДВ понимается неописанная в документации либо не соответствующая ей функциональность, способная нарушить политику безопасности. В ходе сертификации ОС Astra Linux проверяются следующие потенциальные типы НДВ:

– НДВ уровня ядра, которым относятся недокументированные системные вызовы, а также возможность модификации механизмов мандатного доступа PARSEC;

– НДВ в подсистеме межсетевого экранирования, в том числе, наличие скрытых каналов, неблокируемых штатными средствами фильтрации;

– НДВ в средствах аудита – возможность совершения пользователем действий, не регистрируемых в журналах безопасности.

Проверка отсутствия НДВ включает в себя:

– анализ исходных кодов ОС Astra Linux;

– документирование всех программных интерфейсов;

– подтверждение отсутствия перечисленной выше функциональности.

Важным требованием безопасности ОС является запрет на установку стороннего ПО, которое может модифицировать используемые в ИТ-инфраструктуре сертифицированные компоненты СЗИ. Данный запрет обеспечивается механизмами ЗПС и МКЦ [4], проверяющим целостность исполняемых файлов. Одновременно в ОС Astra Linux встроены различные механизмы, обеспечивающие гибкое управление ее функциями.

1. Механизм интеграции в замкнутую программную среду пакетов на уровне ядра, в том числе, метапакетов ядер ОС различных модификаций (generic, hardened, lowlatenc). При этом ядра ОС Astra Linux generic и lowlatency, входящие в состав версий данной ОС, не аттестованы для обработки информации ограниченного доступа или государственной тайны, поэтому их использование допустимо исключительно в неаттестованных сегментах ИТ-инфраструктуры. Однако даже при использовании механизмов

[1] ГОСТ Р 56939-2024 «Защита информации. Разработка безопасного программного обеспечения. Общие требования»

МРД и МКЦ в защищенном сегменте ИТ-инфраструктуры не удается полностью исключить возникновение технических каналов утечки информации [5].

2. Механизм отключения, доступный системному администратору, обязательной авторизации с помощью специальных флагов (например, за счет установления соответствующего значения параметра AstraMode в веб-сервере Apache2), использование которого обеспечивает совместимость приложений без аутентификации, однако, снижает защищённость ИТ-инфраструктуры [5]. В этой связи, для уточнения правила доступа для отдельных контейнеров, субъектов и сущностей, используют дополнительные мандатные атрибуты, которые обеспечивают корректную работу DLP- и SIEM-систем в защищённой среде.

Описанные выше механизмы в сочетании с иными не декларированными возможностями ОС создают системные уязвимости, которые могут быть использованы нарушителем для повышения привилегий пользователя и последующей компрометации других программных компонентов, функционирующих в защищенной среде, в том числе, веб-приложений, в которые интегрированы генеративные модели ИИ.

3. Особенности динамического тестирования безопасности веб-приложений

Современные методологии безопасной разработки веб-приложений, функционирующих под управлением защищенной ОС, предусматривают обязательное тестирование их безопасности, при проведении которого используются методы статического анализа (SAST). в ходе которого проводится анализ программного кода приложения (модель «белого ящика») и динамического анализа (DAST), который не требует доступа к исходному коду приложения и обеспечивает выявление уязвимостей в среде выполнения (модель «черного ящика»), также на практике могут использоваться комбинации моделей «белого» и «черного» ящиков – модель «серый ящик» [6]. Результаты сравнения SAST и DAST подходов представлены, а табл. 1.

Таблица 1

Сравнение динамических и статических подходов

Критерий	DAST (black-box)	SAST (white-box)
Объект анализа	Работающее приложение	Исходный код, байт-код

Окончание табл.1

Доступ к коду	Не требуется	Обязателен
Этап жизненного цикла разработки ПО	Тестирование, эксплуатация	Разработка
Зависимость от языка	Нет	Да
Обнаружение уязвимостей времени выполнения	Да	Нет
Локализация дефекта в коде	Нет	Да (с точностью до строки)
Ложноположительные срабатывания	Средний уровень	Высокий уровень
Тестирование ИИ-компонентов	Ограничено (без специальных модулей)	Исключительно интеграционный слой

Из табл. 1 видно, что подходы SAST и DAST дополняют друг друга: первый обеспечивает обнаружение дефектов на этапе написания кода и их точную локализацию; второй обеспечивает выявление уязвимостей, проявляющихся исключительно в среде выполнения.

Однако ни один из обсуждаемых подходов, ни их комбинация не обеспечивают полноценного тестирования веб-приложений на наличие уязвимостей в ИИ-компонентах подходов, что определяет необходимость разработки дополнительных методов анализа их уязвимостей. Также необходимо отметить, что при использовании больших языковых моделей (LLM) в веб-приложениях (RAG, чат-боты, аналитические платформы) могут возникать новые типы уязвимостей и соответствующие угрозы безопасности, выявить которые с помощью динамического анализа не удается. Например, – инъекции, внедряемые нарушителем в запросы к LLM-модели, которые возникают в следствие отсутствия в подобных системах четкой границы между инструкциями и данными, наличие которой строго контролируется в обычных веб-приложениях (например, с помощью параметризации) [7].

Еще одним источником уязвимостей, возникающих на этапах подготовки данных и обучении являются, собственно, данные, которые могут быть

«отравлены» или подвержены инверсной компьютерной атаке с целью реконструкции обучающей выборки. Здесь для проведения атаки на данные нарушитель может использовать избыточные привилегии LLM-агентов и утечку системных инструкций. На этапе выполнения запросов нарушитель, в большинстве случаев, использует прямые и/или косвенные вредоносные инструкции (промпт-инъекции). При этом, наиболее опасными являются промпт-инъекции оказываются для RAG-систем, в которые они могут проникать без явных действий пользователей. В этой связи ИТ-инфраструктуре на основе ОС Astra Linux, обрабатывающих данные ограниченного доступа, использование сторонних GenAI-инструментов должно быть либо запрещено политиками безопасности, либо использоваться специализированные решения, предотвращающие утечки через промпт-интерфейсы.

4. Методы оценки защищенности ИТ-инфраструктуры

Методы оценки защищённости защищенности ИТ-инфраструктуры активно развиваются в лаборатории проблем компьютерной безопасности СПИИРАН, руководимой И.В. Котенко. Здесь были, в том числе, были предложены подходы, основанные на использовании графов атак, деревьев социоинженерных атак и динамического перерасчёта показателей защищённости в SIEM-системах [8, 9, 10].

Принимая во внимание эффективность данных подходов, представляется целесообразным их адаптация для ИТ-инфраструктур на базе ОС Astra Linux, в том числе: разработка методики выявления путей компрометации при мандатной модели доступа на основе использования графов атак, а также методики обнаружения аномалий и реагирования на инциденты, основанной на динамическом перерасчете показателей защищённости

Напомним, что «Методика оценки показателя состояния технической защиты информации и обеспечения безопасности значимых объектов КИИ», утверждённая ФСТЭК России 11 ноября 2025 г., (далее Методика ФСТЭК России) рекомендовано использовать следующую трёхуровневую шкалу: «зелёный» (базовый) уровень – полное соответствие требованиям безопасности; «оранжевый» (низкий) – частичное соответствие; «красный» (критический) – несоответствие, а также установлена периодичность контроля состояния технической защиты информации и обеспечения безопасности значимых объектов КИИ– один раз в полгода. При этом структура процесса

оценки уровня защищенности информации (рис. 1) определяется архитектурой ИТ-структуры и особенностями процессов ее функционирования.

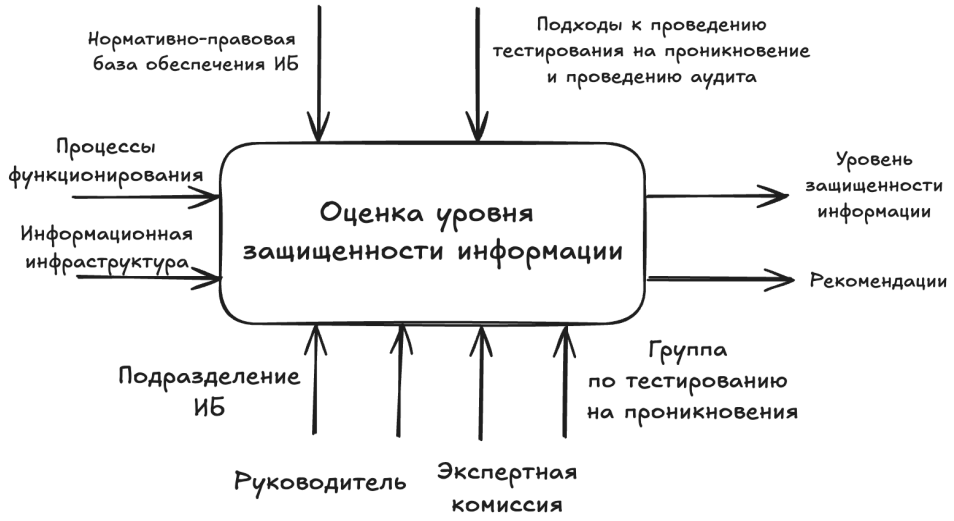


Рис. 1. Структурная схеме процесса оценки уровня защищенности информации

Структурная схема процесса оценки уровня защищенности информации в соответствие с Методикой ФСТЭК России представлена на рис. 2, из котрого видно, данный процесс реализуется выполнением следующих этапов:

– подготовка к проведению оценки защищенности;
– уточнение требований и формирование оценки зрелости;
– проведение тестирования на проникновение и выявление уязвимостей;
– оценка защищенности и достаточности реализации требований;
– формирование рекомендаций и отчета.

Напомним, что при оценке уровня защищенности при компрометации компонентов ИТ-инфраструктуры могут возникать каскадные эффекты [11], которые влияют на значение оценки, для учета которых целесообразно использовать цифровые двойники ИТ-инфраструктуры.

Отметим, что в настоящее время в соответствующем сегменте рынка средств защиты информации представлен широкий спектр средств для анализа защищённости, однако, при их использовании в ИТ-инфраструктурах на базе ОС Astra Linux возникает ряд трудностей.

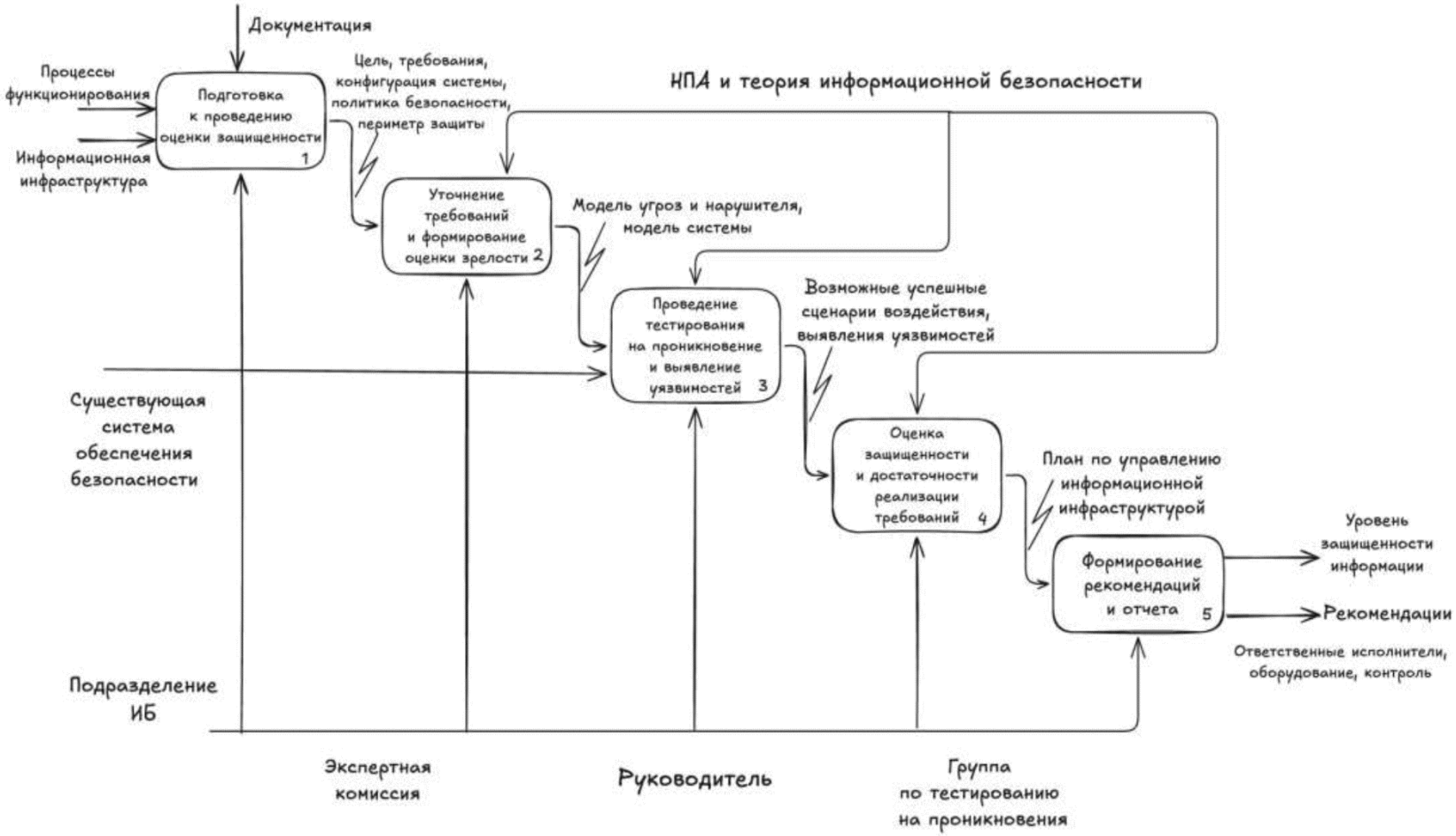


Рис. 2. Структурная схема процесса оценки уровня защищенности информации

1. Не полная совместимость с мандатной архитектурой PARSEC (сканеры уязвимостей, запускаемые с ограниченными привилегиями, не имеют необходимого доступа к объектам файловой системы, приводит к неполноте результатов сканирования; кроме того, уязвимости обнаружены в самой подсистеме PARSE уязвимости (например, уязвимость CVE-2025-62514).

2. Необходимость сохранения замкнутости программной среды (допустимо использование исключительно сертифицированных средств защиты информации).

3. Неполнота информации, собираемой SIEM-системами (современные SIEM-системы не обнаруживают факты атак на ядро ОС Astra Linux).

4. Снижение эффективности black-box тестирования по мере усложнения приложений, в связи с необходимостью использования многократных итераций и привлечения высококвалифицированных специалистов [12].

Следовательно, оценка защищенности ИТ-инфраструктуры на базе ОС Astra Linux не может быть получена на основе единоразового аудита, должна проводиться на основе проверки соответствия действующим нормативными требованиям в области информационной безопасности, учитывающих архитектурные особенностей ИТ-инфраструктуры, а также с учетом необходимости постоянной адаптации к новым угрозам, включая порождаемые ИИ.

Заключение

Проведен анализ механизмов защиты ОС Astra Linux Special Edition (в том числе: мандатное управление доступом PARSEC, замкнутая программная среда, контроль недекларированных возможностей), типов НДВ, условий их эксплуатации, и сделан обоснованный вывод о том, что наличие в данной ОС несертифицированных на соответствие требования безопасности ядер generic и lowlatency, а также механизмов отключения обязательной мандатной авторизации, формируют системную уязвимость, используя которую оказывается возможным отключать мандатные механизмы доступа. При этом данная уязвимость не упоминается в документации на ОС Astra Linux Special Edition.

Проведен сравнительный анализ методов выявления уязвимостей, основанный на использовании black-box– и white-box–моделей, и сделан вывод о том, что стандартной реализации ОС Astra Linux Special Edition не один из них не обеспечивает выявления уязвимостей ИИ-компонентов веб-приложений (промпт-инъекций, отравления данных, инверсионных атак), поэтому необходимо разработка специализированных подходов к тестированию LLM-интегрированных систем, дополняющих традиционные DAST-методы.

Продемонстрировано, что интеграция в рамках единого подхода нормативно-правового, архитектурного (уровень ОС) и прикладного (веб-приложения, использующего ИИ-компоненты) уровней обеспечивает выявление комбинированных сценариев компрометации ИТ-инфраструктуры, не обнаруживаемые при оценке каждого уровня по отдельности.

Предложенный комплексный подход, основанный на подходах П.Н. Девянина и И.В. Котенко, дополнен уровнем динамического тестирования веб-приложений и специфическим для генеративного ИИ классом угроз.

Данный подход может быть использован при разработке специализированных программно-аппаратных инструментов для оценки защищенности и рисков нарушения безопасности гетерогенных защищенных ИТ-инфраструктур на базе Astra Linux. Перспективным направлением дальнейших исследований представляется разработка доверенных сред для безопасного использования ИИ-сервисов и специализированных анализаторов промпт-безопасности, интегрированных в аттестованную инфраструктуру без нарушения режима замкнутой программной среды.

Литература

1. Cybercrime as service: анализ трендов сервисных моделей киберпреступности // Positive Technologies: сайт. – URL: https://ptsecurity. com/ research/analytics/cybercrime-as-a-service/#id1 (дата обращения: 20.05.2026). – Текст: электронный.

2. Бондаренко, А. В. Анализ моделей, методов и средств контроля в области обеспечения безопасности значимых объектов КИИ газовой отрасли ТЭК РФ. Часть 1 / А. В. Бондаренко, С. В. Поршнев, В. В. Богданов, М. А. Кондратенко // Защита информации. Инсайд. – 2026. – № 1 (127). – С. 4–18.

3. Девянин, П. Н. Проектирование и развитие механизма мандатного контроля целостности в операционной системе Astra Linux / П. Н. Девянин, А. А. Старостин, Д. С. Панов, С. В. Усачев // Труды ИСП РАН. – 2025. – Т. 37, вып. 2. – С. 61–78.

4. Операционные системы Astra Linux // Справочный центр Astra Linux: сайт. – 2026. – URL: https://wiki.astralinux.ru/ (дата обращения: 22.03.2026). – Текст: электронный.

5. Использование стороннего программного обеспечения в аттестованных информационных системах, функционирующих под управлением Astra Linux Special Edition 1.8 // Справочный центр Astra Linux: сайт. – 2026. –

URL: https://wiki.astralinux.ru/ (дата обращения: 22.03.2026). – Текст: электронный.

6. Тестирование методом белого и черного ящика: что нужно знать бизнесу о безопасности программ и приложений // Solar appScreener: сайт. – 2024. – URL: https://rt-solar.ru/products/solar_appscreener/blog/2360/ (дата обращения: 10.03.2026). – Текст: электронный.

7. Внедрение ChatGPT Prompt: методы, угрозы и рекомендации по безопасности // LayerX Security: сайт. – 2025. – URL: https://layerxsecurity.com/ru/generative-ai/chatgpt-prompt-injection/ (дата обращения: 20.03.2026). – Текст: электронный.

8. Котенко, И. В. Оценивание защищённости информационных систем на основе построения деревьев социоинженерных атак / И. В. Котенко, М. В. Степашкин, Д. И. Котенко, Е. В. Дойникова // Приборостроение. – 2011. – Т. 54, № 12. – С. 5–9.

9. Novikova, E. What are your privacy risks? Privacy risk assessment based on privacy policies analysis / E. Novikova, E. Doynikova, I. Kotenko // Expert Systems with Applications. – 2025. – Vol. 280, Part A. – Art. 127270. – DOI 10.1016/j.eswa.2025.127270.

10. Девянин, П. Н. Результаты переработки уровней ролевого управления доступом и мандатного контроля целостности формальной модели управления доступом ОС Astra Linux / П. Н. Девянин // Труды ИСП РАН. – 2023. – Т. 35, № 5. – С. 7–22. – DOI 10.15514/ISPRAS-2022-35(5)-1.

11. Бакшеев, А. С. Разработка методики контроля уровня защищенности объектов критической информационной инфраструктуры / А. С. Бакшеев, И. И. Лившиц // Вопросы кибербезопасности. – 2023. – № 2 (54). – С. 85–98. – DOI 10.21681/2311-3456-2023-2-85-98.

12. Девянин, П. Н. Тестирование подсистемы безопасности ОС Astra Linux на основе формализованного описания модели управления доступом / П. Н. Девянин, С. С. Жиляков, А. И. Смирнов // Труды ИСП РАН. – 2025. – Т. 37, вып. 6 (2). – С. 21–36. – DOI 10.15514/ISPRAS-2025-37(6)-17.

References

1. Cybercrime kak servis: analiz trendov servisnykh modeley kiberprestupnosti [Cybercrime as a service: analysis of trends in cybercriminal service models]. URL: https://ptsecurity.com/research/analytics/cybercrime-as-a-service/#id1 (accessed 20 May 2026).

2. Bondarenko A. V., Porshnev S. V., Bogdanov V. V., Kondratenko M. A. Analysis of models, methods and control tools in the field of ensuring the security of significant objects of critical information infrastructure of the gas industry of the fuel and energy complex of the Russian Federation. Part 1 [Analiz modeley, metodov i sredstv kontrolya v oblasti obespecheniya bezopasnosti znachimykh ob'ektov KII gazovoy otrasli TEK RF. Chast' 1] // Zashchita informatsii. Insayd [Information Security. Inside]. 2026. No. 1 (127). P. 4–18.

3. Devyanin P.N., Starostin A.A., Panov D.S., Usachev S.V. Design and development of mandatory integrity control in Astra Linux OS [Proektirovanie i razvitie mekhanizma mandatnogo kontrolya celostnosti v operacionnoj sisteme Astra Linux] // Trudy ISP RAN [Proceedings of ISP RAS]. 2025. Vol. 37, Issue 2. P. 61–78.

4. Astra Linux operating systems [Operacionnye sistemy Astra Linux] // Astra Linux Reference Center. 2026. URL: https://wiki.astralinux.ru/ (accessed 22 March 2026).

5. Use of third-party software in certified information systems running under Astra Linux Special Edition 1.8 [Ispol'zovanie storonnego programmnogo obespecheniya v attestovannykh informatsionnykh sistemakh, funktsioniruyushchikh pod upravleniem OS Astra Linux Special Edition 1.8] // Astra Linux Reference Center. 2026. URL: https://wiki.astralinux.ru/ (accessed 22 March 2026).

6. White Box and Black Box Testing: What Businesses Need to Know About Software and Application Security [Testirovanie metodom belogo i chernogo yashchika: chto nuzhno znat' biznesu o bezopasnosti programm i prilozheniy] // Solar appScreener. 2024. URL: https://rt-solar.ru/ products/solar_appscreener/ blog/2360/ (accessed 10 March 2026).

7. ChatGPT Prompt Injection: Methods, Threats and Security Recommendations [Vnedrenie ChatGPT Prompt: metody, ugrozy i rekomendatsii po bezopasnosti] // LayerX Security. 2025. URL: https://layerxsecurity.com/ru/ generative-ai/chatgpt-prompt-injection/ (accessed 20 March 2026).

8. Kotenko I.V., Stepashkin M.V., Kotenko D.I., Doynikova E.V. Security analysis of information systems taking into account social engineering attacks [Otsenivanie zashchishchennosti informatsionnykh sistem na osnove postroeniya derev'ev sotsioinzhenernykh atak] // Priborostroenie [Instrument Engineering]. 2011. Vol. 54, No. 12. P. 5–9.

9. Novikova, E. What are your privacy risks? Privacy risk assessment based on privacy policies analysis / E. Novikova, E. Doynikova, I. Kotenko // Expert

Systems with Applications. – 2025. – Vol. 280, Part A. – Art. 127270. – DOI 10.1016/j.eswa.2025.127270.

10. Devyanin P.N. The results of reworking the levels of role-based access control and mandatory integrity control of the formal model of access control in Astra Linux [Rezul'taty pererabotki urovnelevoj rolevoj modeli upravleniya dostupom i mandatnogo kontrolya celostnosti formal'noj modeli upravleniya dostupom OS Astra Linux] // Trudy ISP RAN [Proceedings of ISP RAS]. 2023. Vol. 35, No. 5. P. 7-22. DOI: 10.15514/ISPRAS-2022-35(5)-1.

11. Baksheev A.S., Livshits I.I. Development of a methodology for controlling the security level of critical information infrastructure objects [Razrabotka metodiki kontrolya urovnya zashchishchennosti obvЂќektov kriticheskoy informatsionnoy infrastruktury] // Voprosy Kiberbezopasnosti [Cybersecurity Issues]. 2023. No. 2 (54). P. 85-98. DOI: 10.21681/2311-3456-2023-2-85-98.

12. Devyanin P.N., Zhiliakov S.S., Smirnov A.I. Testing the Astra Linux OS security subsystem based on a formalized description of the Access Control model [Testirovanie podsistemy bezopasnosti OS Astra Linux na osnove formalizovannogo opisaniya modeli upravleniya dostupom] // Trudy ISP RAN [Proceedings of ISP RAS]. 2025. Vol. 37, Issue 6(2). P. 21–36. DOI: 10.15514/ISPRAS-2025-37(6)-17.

КОЖЕМЯКОВ Дмитрий Алексеевич аспирант, ФГАОУ ВО «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина». 620002, г. Екатеринбург, ул. Мира, 32. E-mail: kozhemyakov.da@uraltransmash.ru

ПОРШНЕВ Сергей Владимирович, доктор технических наук, профессор Учебно-научного центра «Информационная безопасность» ФГАОУ ВО «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина». 620002, г. Екатеринбург, ул. Мира, 32. E-mail: s.v.porshnev@urfu.ru

СТОЛЬНИКОВА Алена Александровна, аспирант, ФГАОУ ВО «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина». 620002, г. Екатеринбург, ул. Мира, 32. E-mail: Eskander147@yandex.ru

ХОРОШКЕЕВ Дмитрий Александрович магистр, ФГАОУ ВО «Уральский федеральный университет им. первого Президента России

Б.Н. Ельцина». 620002, г. Екатеринбург, ул. Мира, 32. E-mail: d.khoroshkeev@gmail.com

KOZHEMYAKOV Dmitry Alekseevich, postgraduate student of the Federal State Autonomous Educational Institution of Higher Education «Ural Federal University named after the first President of Russia B.N. Yeltsin». 620002, Yekaterinburg, st. Mira, 32. E-mail: kozhemyakov.da@uraltransmash.ru

PORSHNEV Sergey Vladimirovich, Doctor of Technical Sciences, Professor, Director of the Educational and Scientific Center «Information Security» of the Federal State Autonomous Educational Institution of Higher Education «Ural Federal University named after the first President of Russia B.N. Yeltsin». 620002, Yekaterinburg, st. Mira, 32. E-mail: s.v.porshnev@urfu.ru

STOLNIKOVA Alena Alexandrovna, postgraduate student of the Federal State Autonomous Educational Institution of Higher Education «Ural Federal University named after the first President of Russia B.N. Yeltsin». 620002, Yekaterinburg, st. Mira, 32. E-mail: Eskander147@yandex.ru

KHOROSHKEEV Dmitry Alexandrovich, master's student of the Federal State Autonomous Educational Institution of Higher Education «Ural Federal University named after the first President of Russia B.N. Yeltsin». 620002, Yekaterinburg, st. Mira, 32. E-mail: d.khoroshkeev@gmail.com

УДК 004.056 Вестник УрФО № 3(61) / 2026, с. 64–77
DOI: 10.14529/secur260305

Южаков Н.А., Южаков А.А.

АРХИТЕКТУРА БИОМЕТРИЧЕСКОЙ АВТОРИЗАЦИИ В ИНФОРМАЦИОННЫХ СИСТЕМАХ

Yuzhakov N.A., Yuzhakov A.A.

ARCHITECTURE OF BIOMETRIC AUTHORIZATION IN INFORMATION SYSTEMS

Обеспечение надёжного контроля доступа остаётся фундаментальной задачей информационной безопасности. Эволюция архитектуры программного обеспечения в сторону микросервисов увеличивает поверхность атаки и требует внедрения новых архитектурных паттернов разграничения доступа. В то же время классическая парольная аутентификация демонстрирует недостаточную устойчивость к современным угрозам, уступая место биометрическим технологиям.

В работе проведён систематический анализ недостатков существующих архитектурных паттернов авторизации (децентрализованный, централизованный с удалённым PDP, централизованный со встроенным PDP) применительно к интеграции биометрических методов. Выявлены конкретные разрывы: отсутствие метрик обновления биометрических политик, незащищённость от replay-атак в веб-среде, неадаптированность обманных сценариев к микросервисной архитектуре. Для каждого недостатка предложен способ устранения: метрика Policy Update Cost (PUC), протокол привязки биометрического семпла к TLS-сессии, распределённый honeypattern-механизм. Сформулированы практические рекомендации для архитекторов ИБ, позволяющие реализовать принцип defense in depth

Ключевые слова: микросервисная архитектура, аутентификация, авторизация, биометрия, политики доступа, Policy Decision Point (PDP), нейронные сети, распознавание лиц, WebRTC.

Ensuring reliable access control remains a fundamental task in information security. The evolution of software architecture towards microservices increases the attack surface and requires the implementation of new architectural patterns for

access control. At the same time, classic password-based authentication demonstrates insufficient resilience against modern threats, giving way to biometric technologies.

The paper provides a systematic analysis of the shortcomings of existing architectural authorization patterns (decentralized, centralized with a remote PDP, centralized with an embedded PDP) in relation to the integration of biometric methods. Specific gaps have been identified: the lack of metrics for updating biometric policies, vulnerability to replay attacks in the web environment, and the unadaptability of deceptive scenarios to microservice architecture. For each shortcoming, a solution has been proposed: the Policy Update Cost (PUC) metric, a protocol for binding a biometric sample to a TLS session, and a distributed honeypattern mechanism. Practical recommendations have been formulated for information security architects to implement the defense in depth principle.

Keywords: microservice architecture, authentication, authorization, biometrics, access policies, Policy Decision Point (PDP), neural networks, facial recognition, WebRTC.

Введение

Современный ландшафт информационных систем характеризуется двумя противоположными трендами: усложнением архитектуры и стремлением к упрощению пользовательского опыта. Переход к микросервисной архитектуре, продиктованный необходимостью масштабирования и гибкости разработки, существенно расширяет поверхность атаки, делая каждый микросервис потенциальной точкой входа для злоумышленника [1, 2]. При этом классические монолитные системы авторизации становятся неприменимыми, что требует пересмотра архитектурных решений. С другой стороны, пользователь остаётся самым слабым звеном: использование простых и повторяющихся паролей, а также фишинговые атаки делают традиционную парольную аутентификацию малоэффективной. Ответом на этот вызов стало широкое внедрение биометрических технологий, позволяющих идентифицировать личность по уникальным характеристикам – статическим (рис. радужной сетчатки, папиллярные узоры) и динамическим (клавиатурный почерк) [3, 4].

Цель данной работы провести анализ существующих архитектурных патернов, биометрических методов и выявить конкретные недостатки их совместного применения. Задачи исследования:

- провести анализ архитектурных паттернов авторизации в микросервисных системах;
- выявить ключевые разрывы между этими паттернами и требованиями биометрической аутентификации;
- предложить для каждого разрыва формальное решение (метрику, протокол, архитектурный механизм);
- сформулировать проверяемые рекомендации по выбору архитектуры.

Проблема в целом заключается в отсутствии единого, системного подхода к совместному проектированию архитектуры авторизации в микросервисных системах и встраиванию в неё биометрических механизмов аутентификации [11]. На практике это приводит к одному из двух нежелательных сценариев:

Биометрия добавляется «над» устаревшей архитектурой (чаще всего монолитной или с централизованным API-шлюзом), что не устраняет внутренних проблем распределённого контроля доступа (единая точка отказа, сложность обновления политик, высокие задержки при масштабировании).

Внедряется гибкая архитектура авторизации (например, Remote PDP), но без учёта специфики биометрических данных: высокая вычислительная сложность верификации, необходимость защищать биометрические шаблоны, проблема ложных отказов/допусков, отсутствие механизмов «обманных сценариев» на уровне микросервисов.

На основе проведенного анализа выявлены следующие конкретные недостатки (табл. 1).

Таблица 1

Анализ недостатков существующих решений в области безопасности ПО и архитектуры

№	Недостаток	Где проявляется	Почему это критично
D1	Отсутствие формального сравнения паттернов авторизации по критерию «возможность динамической смены биометрических эталонов»	Во всех известных решениях сравниваются только задержки и масштабируемость	При использовании децентрализованного паттерна обновление эталонов (например, при изменении внешности) требует переразвёртыва-

			ния каждого микросервиса – биометрия становится негибкой
D2	Нет методики интеграции биометрических «обманных» сценариев (honeypot) в микросервисную архитектуру	Представлен только алгоритм для монолитного приложения	Невозможно обеспечить согласованное переключение всех микросервисов на фиктивные данные при обнаружении атаки
D3	Биометрическая верификация рассматривается как «чёрный ящик» без учёта её вычислитель-ной сложности при выборе паттерна	Реализованы методы распознавания, но без анализа влияния на задержку авторизации	Архитектор не может выбрать между Remote PDP и Embedded PDP, не зная времени проверки лица/голоса
D4	Отсутствует защита от атак повторного воспроизведения (replay) биометрических данных в веб-среде	Используется способ WebRTC, но не предлагается механизм привязки семпла к сессии	Злоумышленник может перехватить и повторно отправить ранее записанное изображение лица (видео-атака)

Архитектурные паттерны авторизации и их анализ

В терминологии NIST выделяют три основных паттерна реализации авторизации на уровне сервисов, различающихся по месту хранения и вычисления политик доступа (PDP – Policy Decision Point, PEP – Policy Enforcement Point). [6,7,8]

Децентрализованный паттерн. Логика авторизации (PDP и PEP) встроена в код каждого микросервиса.

Достоинства: минимальная задержка, отсутствие сетевых вызовов.

Недостатки: любое изменение политики требует изменения кода и переразвёртывания [5].

Централизованный с удалённым PDP (Remote PDP). Политики хранятся и вычисляются в центральном сервисе; микросервисы обращаются к нему по сети.

Достоинства: обновление политик без перезапуска, централизованное управление.

Недостатки: высокая задержка, зависимость от доступности PDP.

Централизованный со встроенным PDP (Embedded PDP). Политики централизованно хранятся, но доставляются до каждого микросервиса и кэшируются локально; PDP реализован как библиотека.

Достоинства: низкая задержка, централизованное управление.

Недостатки: требуется синхронизация политик, проблема консистентности.

Для оценки пригодности каждого паттерна к работе с биометрическими данными вводится метрика PUC (Policy Update Cost) – стоимость обновления биометрических эталонов (например, шаблона лица). PUC определяется как время (или число операций), необходимое для распространения нового эталона на все сервисы, принимающие решения. Для децентрализованного паттерна стоимость PUC высокая (требуется CI/CD и требуется перезапуск), для централизованных паттернов с удалённым или встроенным PDP – стоимость низкая (достаточно обновить центральное хранилище или канал синхронизации). В табл. 2 показано сравнение паттернов.

Таблица 2

Сравнение паттернов авторизации с учётом биометрических требований

Паттерн	Масштабируемость	Задержка (p95, мс)	Изменение политик "на лету"	Стоимость PUC (биометрические эталоны)	Рекомендуемый сценарий
Децентрализованный	Низкая	< 5	Нет	Высокая	Только статические, редко меняющиеся эталоны (отпечатки пальцев)

Централизованная (Удалённый PDP)	Высокая	50–200	Да	Низкая	Динамическая биометрия (лицо, голос) при терпимости к задержке
Централизованный (Встроенный PDP)	Высокая	5–15	Да (с оговорками)	Низкая	Динамическая биометрия + требование low latency

Биометрические методы и их влияние на архитектурный выбор

Современные системы идентификации используют различные математические модели: свёрточные нейронные сети, скрытые марковские модели, графовые представления [3, 4]. Комплексное применение нескольких методов позволяет достичь приемлемых показателей ошибок: FAR $\leq 10^{-5}$–10^{-6} при FRR $\leq 0{,}01$ [9].

Время верификации

Tver=Tcapture+Tnet+Tmatch+Tlivedet, где Tcapture – захват образца, Tnet – передача на сервер, Tmatch –сравнение с данными. Экспериментальные данные показывают, что для свёрточных сетей Tmatch составляет 300–500 мс на GPU и 50–100 мс на CPU [10], Tlivedet – проверка живости. Это различие критически влияет на выбор паттерна, особенно при жёстких ограничениях по задержке.

Однако помимо времени важны и другие свойства биометрических данных:

Динамичность эталонов – лицо и голос меняются со временем (возраст, болезни, тембр), тогда как отпечатки пальцев и рисунок радужки практически стабильны. Следовательно, частота обновления эталонов определяет стоимость PUC и склоняет выбор в сторону централизованных паттернов для динамической биометрии.

Необходимость обнаружения живости (liveness detection) – распознавание лица уязвимо к атакам по фотографии или видео. Проверка живости тре-

бует отдельного вычислительного слоя (анализ микродвижений, бликов, термограммы) и может добавлять 100–200 мс к Tver. Этот фактор должен учитываться на архитектурном уровне как самостоятельный сервис.

Мультимодальность – комбинирование нескольких биометрических модальностей повышает точность, но кратно увеличивает Tmatch и объём передаваемых данных, что может сделать Remote PDP неприемлемым при ограниченной пропускной способности.

Критерии архитектурного выбора

На основе анализа работ [11, 12, 13] выделяем три группы критериев, которые необходимо оценить до принятия решения.

Бизнес-критерии

Автономность – возможность работы без устойчивого сетевого соединения. Для удалённых объектов (промышленные контроллеры, мобильные устройства) предпочтительна децентрализованная обработка на устройстве.

Допустимый компромисс FAR/FRR – в системах физической безопасности ложный допуск (FAR) может быть катастрофическим, поэтому пороги сдвигают в сторону снижения FAR, что увеличивает Tmatch (более строгие модели) и может потребовать более мощного железа.

Технические критерии

Вычислительные ресурсы – доступность GPU на серверной стороне или на устройстве. Если на устройстве ограниченные ресурсы, то Tmatch на CPU может превысить допустимую задержку, и тогда необходимо выносить сравнение в централизованный PDP.

Пропускная способность сети – большие биометрические семплы могут сделать Remote PDP неэффективным из-за времени передачи Tnet.

Необходимость аудита – централизованные паттерны упрощают сбор журналов всех сравнений, что важно для соответствия регуляторам (GDPR, 152-ФЗ).

Критерии безопасности и приватности

Защита биометрических шаблонов – при централизованном хранении все эталоны находятся в одном месте, что повышает риск компрометации. Децентрализованное хранение (на устройстве) снижает этот риск, но усложняет синхронизацию.

Защита от атак повторного воспроизведения – требует дополнительного протокола привязки к сессии, что реализуемо как в Remote, так и в Embedded PDP, но не в децентрализованном без централизованного управления.

Стандартизация – использование спецификаций (например, NIST SP 500-288 для биометрических веб-сервисов) облегчает интеграцию, но может накладывать ограничения на выбор паттерна.

Исходя из выше приведенных критериев можно составить алгоритм выбора архитектурного паттерна.

Шаг 1. Определение профиля биометрических данных.

Классифицируем биометрию как статичестую (отпечатки пальцев, радужка) или динамическую (лицо, голос, походка). Для статической и редко обновляемой биометрии рассматриваем децентрализованный паттерн как базовый вариант.

Шаг 2. Оценка временных ограничений.

Зададим максимально допустимую задержку Lmax. Оценим Tver для выбранного метода на целевом оборудовании.

Если Tver>Lmax – необходимо аппаратное ускорение или вынос сравнения на более мощный сервер (Remote PDP).

Если Tver≤Lmax – проверяем возможность локального выполнения.

Шаг 3. Оценка стоимости обновления политик (PUC).

Рассчитываем PUC как время на распространение нового биометрического эталона по всем микросервисам.

Для децентрализованного паттерна PUC высока (требуется CI/CD и перезапуск). Если частота обновления > 1 раза в месяц – децентрализованный паттерн исключается.

Для Remote и Embedded PDP – PUC низкая (достаточно обновить центральное хранилище).

Шаг 4. Учёт дополнительных нефункциональных требований.

Проверим:

Необходимость работы в офлайн-режиме → выбираем Embedded PDP или децентрализованный с локальным кэшем эталонов.

Необходимость централизованного аудита → Remote или Embedded PDP (логирование на уровне PDP).

Требования к приватности (хранение шаблонов только на устройстве) → децентрализованный или гибридный (сравнение на устройстве, но политики централизованы).

Шаг 5. Проверка защищённости от атак повторного воспроизведения.

Если система работает в веб-среде, обязательно внедренит протокола с nonce. Это требование не зависит от паттерна, но при децентрализованном

паттерне синхронизация nonce между сервисами становится нетривиальной – это может склонить выбор в сторону централизованного PDP.

Шаг 6. Итоговое решение.

На основе взвешенных критериев выбирается один из трёх паттернов. В табл. 2 (см. выше) приведены базовые рекомендации. Для сложных систем возможна гибридная архитектура: например, локальное извлечение признаков (уменьшение Tnet) и централизованное сравнение с эталонами.

Архитектурный выбор при интеграции биометрии не является одномерным. Он требует системного учёта временных характеристик (Tver), стоимости обновления политик (PUC), бизнес-требований к автономности, безопасности и приватности. Предложенный пошаговый алгоритм позволяет формализовать этот процесс и избежать типичных ошибок, когда биометрия добавляется «поверх» архитектуры без анализа её специфики [11].

Рекомендации по выбору архитектуры

На основе проведённого анализа и предложенных решений сформулированы следующие рекомендации в табл. 3.

Таблица 3

Предложенные решения

Требование	Рекомендуемый паттерн	Обоснование
Динамическая биометрия (лицо, голос) + частота обновления эталонов > 1 раза в месяц	Remote или Embedded PDP	Низкая стоимость PUC (устранение D1)
Динамическая биометрия + low latency (< 50 мс)	Embedded PDP	T_ver мала + нет сетевых задержек (D3)
Использование веб-интерфейса с биометрией	Добавить протокол с nonce	Устранение D4
Необходимость активной защиты от внутреннего нарушителя	Внедрить распределённый honeypot pattern	Устранение D2
Статическая биометрия (отпечатки пальцев), редкое обновление	Децентрализованный паттерн	Низкая задержка, нет сетевых вызовов

Одной из моделей управления доступом является модель на основе атрибутов (Attribute-Based Access Control, ABAC) [13] представляет собой гибкий и масштабируемый подход к разграничению прав доступа, в котором решения о предоставлении или запрете операций принимаются на основании атрибутов субъекта, объекта, действия и контекста окружающей среды. В отличие от дискреционных (DAC) и мандатных (MAC) моделей, где права доступа жёстко привязаны к субъектам и объектам, ABAC обеспечивает динамическое формирование политик безопасности, что позволяет адаптировать систему к изменяющимся условиям функционирования информационной инфраструктуры.

Архитектура системы ABAC строится на взаимодействии следующих фундаментальных компонентов:

Субъекты – сущности, инициирующие запросы на доступ к ресурсам. К субъектам относятся пользователи, процессы, приложения, сервисы и автоматизированные агенты. Каждый субъект характеризуется набором атрибутов, включающих идентификационные данные, уровень допуска, временные параметры аутентификации и динамические характеристики (например, текущее местоположение или уровень доверия).

Объекты – ресурсы, к которым осуществляется доступ. Объектами могут выступать файлы, базы данных, сетевые сервисы, физические устройства и логические сущности. Атрибуты объектов включают классификационную метку, категорию конфиденциальности, владельца ресурса, временные ограничения доступа и структурные характеристики.

Операции – действия, которые субъект намеревается выполнить над объектом. Типовой набор операций включает чтение, запись, выполнение, удаление, создание и администрирование.

Атрибуты окружающей среды – контекстные параметры, не относящиеся непосредственно к субъекту или объекту, но существенно влияющие на принятие решения о доступе. К ним относятся текущее время и дата, географическое положение запроса, тип используемого устройства, уровень защищённости канала связи, результаты многофакторной аутентификации и показатели аномального поведения.

Политики безопасности – формализованные правила, определяющие условия предоставления доступа. Политики представляют собой логические

выражения, связывающие атрибуты субъекта, объекта, операции и окружающей среды. Совокупность политик формирует политику домена, в рамках которой осуществляется управление авторизацией.

Безопасность функционирования системы управления доступом на основе атрибутов оценивается по совокупности критериев, обеспечивающих защиту информационных ресурсов от несанкционированного доступа и гарантирующих корректность авторизационных решений.

Вывод

В работе выполнен комплексный анализ архитектурных паттернов авторизации и биометрических методов идентификации.

Целью следующих исследований является разработка мультимодальной модели аутентификации привилегированных пользователей на основе биометрических сигналов.

Литература

1. Барабанов, А. Аутентификация и авторизация в микросервисных приложениях: обзор архитектурных подходов / А. Барабанов, Д. Макрушин. // Вопросы кибербезопасности. – 2020. – № 4 (38). – С. 32–43.

2. Lu D., Huang D., Walenstein A., Medhi D. A Secure Microservice Framework for IoT // IEEE Symposium on Service-Oriented System Engineering (SOSE). – 2017. – P. 1–10.

3. Исмагилова А.С., Лушников Н.Д. Комплексная биометрическая аутентификация пользователей информационной системы с применением нейронных сетей // Инженерный вестник Дона. – 2024. – № 1. – URL: http://www.ivdon.ru/ru/magazine/archive/n1y2024/8961

4. Ложников П.С., Сулавко А.Е. Алгоритм биометрической идентификации личности с использованием альтернативных сценариев авторизации // Омский государственный технический университет, ООО НТЦ «КАСИБ». 2013 – С. 79–82.

5. Li X., Chen Y., Lin Z. Towards Automated Inter-Service Authorization for Microservice Applications // ACM SIGCOMM Conference Posters and Demos. – 2019. – P. 3–5.

6. Rose, S. Zero Trust Architecture: NIST Special Publication 800-207 / S. Rose, O. Borchert, S. Mitchell, S. Connelly; National Institute of Standards and Technology. – Gaithersburg: NIST, 2020. – 59 p. – URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf (дата обращения: 07.10.2026). – Режим доступа: свободный. – Текст: электронный

7. Yarygina T., Bagge A.H. Overcoming Security Challenges in Microservice Architectures // IEEE Symposium on SOSE. – 2018. – P. 11–20.

8. Siriwardena P., Dias N. Microservices Security in Action // Manning, 2020. – 528 p

9. Grassi P. A., Garcia M. E., Fenton J. L. Digital Identity Guidelines / National Institute of Standards and Technology. – June 2017 (includes updates as of 03-02-2020). – URL: https://doi.org/10.6028/NIST.SP.800-63-3 – DOI:10.6028/NIST.SP.800-63-3

10. Горбунов В.Л., Йе Вин Зо, Жданова И.В. – Обработка данных видеопотока в системах с контролируемым доступом ограниченного круга лиц // Вестник РГРТУ – 2025 – С. 143 – 155

11. Momin M. A., Hussain M. M. M., Islam M. E. An In-Depth Review of Authentication Mechanisms in Microservice Architectures // Daffodil International University. – 2025. – URL: http://dspace.daffodilvarsity. edu.bd:8080/handle/123456789/15922

12. Poorebrahim Gilkalaye B., Mukherjee S., Derakhshani R. A Secure and Private Ensemble Matcher Using Multi-Vault Obfuscated Templates // arXiv preprint arXiv:2404.05205. – 2024. – URL: https://ar5iv.labs.arxiv. org/html/2404.05205

13. Ковтун Д.П., Лапонина О.Р. Использование управления доступом на основе атрибутов и mTLS в микросервисной архитектуре // International Journal of Open Information Technologies ISSN: 2307-8162 vol. 13, no. 6, 2025 С. 75–85.

References

1. Barabanov, A., & Makrushin, D. (2020). Authentication and authorization in microservice applications: An overview of architectural approaches. Voprosy kiberbezopasnosti [Cybersecurity Issues], (4), 32–43.

2. Lu D., Huang D., Walenstein A., Medhi D. A Secure Microservice Framework for IoT // IEEE Symposium on Service-Oriented System Engineering (SOSE). – 2017. – P. 1–10.

3. Ismagilova A.S., Lushnikov N.D. Comprehensive biometric authentication of users of an information system using neural networks // Engineering Bulletin of the Don. – 2024. – No. 1. – URL: http://www.ivdon.ru/ru/magazine/archive/n1y2024/8961

4. Lozhnikov P.S., Sulavko A.E. Algorithm for Biometric Identity Verification Using Alternative Authorization Scenarios // Omsk State Technical University, LLC STC “KASIB”. 2013 – P. 79–82.

5. Li X., Chen Y., Lin Z. Towards Automated Inter-Service Authorization for Microservice Applications // ACM SIGCOMM Conference Posters and Demos. – 2019. – P. 3–5.

6. Rose, S. Zero Trust Architecture: NIST Special Publication 800-207 / S. Rose, O. Borchert, S. Mitchell, S. Connelly; National Institute of Standards and Technology. – Gaithersburg: NIST, 2020. – 59 p. – URL: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf (дата обращения: 07.10.2026). – Режим доступа: свободный. – Текст: электронный

7. Yarygina T., Bagge A.H. Overcoming Security Challenges in Microservice Architectures // IEEE Symposium on SOSE. – 2018. – P. 11–20.

8. Siriwardena P., Dias N. Microservices Security in Action // Manning, 2020. – 528 p.

9. Grassi P. A., Garcia M. E., Fenton J. L. Digital Identity Guidelines / National Institute of Standards and Technology. – June 2017 (includes updates as of 03-02-2020). – URL: https://doi.org/10.6028/NIST.SP.800-63-3 – DOI:10.6028/NIST.SP.800-63-3

10. Gorbunov V.L., Ye Winzo, Zhdanova I.V. – Processing video stream data in systems with controlled access of a limited number of persons // Bulletin of the Russian State Technical University – 2025 – P. 143-155

11. Momin M. A., Hussain M. M. M., Islam M. E. An In-Depth Review of Authentication Mechanisms in Microservice Architectures // Daffodil International University. – 2025. – URL: http://dspace.daffodilvarsity. edu.bd:8080/handle/123456789/15922

12. Poorebrahim Gilkalaye B., Mukherjee S., Derakhshani R. A Secure and Private Ensemble Matcher Using Multi-Vault Obfuscated Templates // arXiv preprint arXiv:2404.05205. – 2024. – URL: https://arxiv.labs.arxiv.org/html/2404.05205

13. Kovtun D.P., Laponina O.R. Using attribute-based access control and mTLS in a microservice architecture // International Journal of Open Information Technologies, ISSN: 2307-8162, vol. 13, no. 6, 2025, P. 75–85.

ЮЖАКОВ Никита Александрович, аспирант, Пермский национальный исследовательский политехнический университет. 614990, г. Пермь, Комсомольский проспект, 29. E-mail: yuz.nikita@gmail.com

ЮЖАКОВ Александр Анатольевич, доктор технических наук, профессор, заведующий кафедры «Автоматика и телемеханика». Пермский национальный исследовательский политехнический университет. 614990, г. Пермь, Комсомольский проспект, 29. E-mail: uz@at.pstu.ru

YUZHAKOV Nikita Aleksandrovich, Postgraduate Student, Perm National Research Polytechnic University. 614990, Perm, Komsomolsky Prospekt, 29. E-mail: yuz.nikita@gmail.com

YUZHAKOV Alexander Anatolyevich, Doctor of Technical Sciences, Professor, Head of the Department of Automation and Telemechanics. Perm National Research Polytechnic University. 614990, Perm, Komsomolsky Prospekt, 29. E-mail: uz@at.pstu.ru

УДК 004.8:004.056.5 Вестник УрФО № 3(61) / 2026, с. 78–103
DOI: 10.14529/secur260306

Смирнов К.Р., Дик Д.И., Человечкова А.В.

ТАКСОНОМИЯ CORSAR: КЛАССИФИКАЦИЯ МЕТОДОВ ОБХОДА ОГРАНИЧЕНИЙ В ЯЗЫКОВЫХ МОДЕЛЯХ ПО МЕХАНИЗМУ ВОЗДЕЙСТВИЯ

Smirnov K.R., Dik D.I., Chelovechkova A.V.

CORSAR TAXONOMY: MECHANISM-BASED CLASSIFICATION OF RESTRICTION BYPASS METHODS IN LANGUAGE MODELS

Атаки, направленные на обход встроенных механизмов безопасности больших языковых моделей с целью получения ответов на запрещенные запросы, известные также как «jailbreak-атаки», являются критически актуальной и быстро развивающейся угрозой. С переходом цифровой индустрии к использованию сложных ИИ-агентов, опасность таких атак значительно возросла. Простая генерация запрещенного текста сменилась риском компрометации информационных систем**.**

Для совершенствования механизмов защиты необходимо структурирование методов реализации атак. Существующие таксономии атак рассматривают запрос как неделимую единицу, классифицируя его исключительно по внешним признакам, что приводит к пересечению категорий и не объясняет механизмов обхода защиты.

Целью работы является разработка новой классификации атак на большие языковые модели. Предложена таксономия CORSAR, основанная на декомпозиции атак на атомарные приемы по механизму воздействия на модель. Если выполняемая атака сочетает несколько приемов, то она раскладывается на части, каждая из которых рассматривается как самостоятельная единица анализа. Группировки основана на природе работы техники. Использование в качестве основы классификации применяемых атакующим приемов минимизирует пересечения между категориями и делает их близкими к ортогональным.

В рамках таксономии выделено шесть независимых категорий, разделенных на 20 подкатегорий, каждая из которых содержит набор конкретных техник.

Предложенная классификация позволит формировать профиль угрозы для большой языковой модели, что даст возможность создания более гибких механизмов защиты.

Ключевые слова: языковые модели, искусственный интеллект, «jailbreak-атаки», таксономия, обход ограничений, информационная безопасность.

Attacks aimed at bypassing the built-in security mechanisms of large language models to obtain responses to prohibited queries, also known as jailbreak attacks, are a critical and rapidly evolving threat. With the digital industry's transition to the use of complex AI agents, the danger of such attacks has increased significantly. Simple generation of prohibited text has given way to the risk of compromising information systems.

Improving security mechanisms requires structuring attack methods. Existing attack taxonomies treat queries as indivisible units, classifying them solely by external characteristics. This leads to overlapping categories and fails to explain the mechanisms by which security is bypassed.

The goal of this paper is to develop a new classification of attacks on large language models. We propose the CORSAR taxonomy, based on decomposing attacks into atomic techniques based on the mechanism by which they affect the model. If an attack combines several techniques, it is decomposed into parts, each of which is considered an independent unit of analysis. The groupings are based on the nature of the technique's operation. Using the attacker's techniques as a basis for classification minimizes the overlap between categories and makes them close to orthogonal.

The taxonomy identifies six independent categories, divided into 20 subcategories, each containing a set of specific techniques.

The proposed classification will allow us to create a threat profile for a large language model, which will make it possible to develop more flexible protection mechanisms.

Keywords: language models, artificial intelligence, jailbreak attacks, taxonomy, restriction bypass, information security.

Введение

Современные большие языковые модели (Large Language Model, LLM) семейств GPT, Claude и Gemini обеспечивают высокое качество генерации

текста, написании кода и решении аналитических задач. Однако вместе с ростом возможностей языковых моделей возрастают и риски их неправомерного использования. Обучение на огромных массивах неструктурированных данных приводит к появлению опасных сценариев поведения языковых моделей [1].

Для формирования способности моделей отказывать в выполнении потенциально неэтичных запросов разработчики применяют методы «выравнивания» (alignment), включая обучение с подкреплением на основе обратной связи от человека (Reinforcement Learning from Human Feedback, RLHF) [2]. Тем не менее подобные меры не обеспечивают абсолютной защиты. LLM остаются уязвимы к состязательным атакам, получившим название «jailbreak» (от англ. «побег из тюрьмы») [1]. Данные техники направлены на преодоление ограничений безопасности за счет использования специально сконструированных запросов.

Защита от таких угроз требует от разработчиков моделей соблюдения баланса между обеспечением безопасности и функциональностью. При этом огромное разнообразие атак делает рассмотрение каждого отдельного случая воздействия на модель нерациональным. Естественным образом отсюда возникает интерес к выявлению общих механизмов «jailbreak-атак» и построению единой их таксономии. Такая таксономия позволяет систематизировать известные методы атак и проектировать защиту LLM под целые классы угроз.

Анализ существующих таксономий

Исследование «jailbreak-атак» началось сравнительно недавно. Как следствие пока нет общепринятой их классификации. Разработчики таксономий предлагают разные подходы для группировки «jailbreak-атак» (способы создания атак, внешние признаки запросов, цели атакующего и т. п.).

Рассмотрим наиболее значимые таксономии, предложенные в период с 2023 по 2025 год. Первая таксономия была основана на 78 промптах, собранных с сайта jailbreakchat.com [3]. Авторы выделили три стратегии («притворство», «переключение внимания» и «повышение привилегий») и десять паттернов внутри этих стратегий. Однако 97% атак попали в категорию «притворство», что указывает на чрезмерно широкое определение: согласно авторам, в нее входит любой запрос, меняющий контекст разговора при сохранении намерения получить запретный ответ. Под это определение попадает практически любая атака. Авторы сами отмечают, что запросы остальных

категорий «часто включают компонент притворства», то есть категории пересекаются. В результате таксономия не позволяет различать атаки между собой и к тому же специфична для ChatGPT образца 2023 года.

Yi et al. (2024) предлагают не столько таксономию в строгом смысле, сколько систематизированный обзор существующих методов обхода защиты [4]. Для white-box моделей категории выделены согласованно, по способу доступа: градиенты, логиты, возможность дообучения (fine-tuning). Однако для black-box части единое основание отсутствует. Template Completion описывает встраивание запроса в обрамляющую структуру (сценарий, роль, код), Prompt Rewriting описывает видоизменение самого запроса (шифрование, перефразирование, перевод), а LLM-based Generation описывает использование другой модели как инструмента. Последняя категория находится на ином уровне абстракции: если первые две отвечают на вопрос «что делать с запросом», то третья отвечает на вопрос «чем генерировать атаку», и языковая модель может служить инструментом для методов обеих содержательных категорий. Кроме того, за пределами классификации остается ряд других значимых техник обхода защиты. Классификация полезна для навигации по литературе, но не для систематического анализа угроз.

Chu et al. (2025) в работе JailbreakRadar предлагают унифицированный бенчмарк, систематизируя атаки в шесть категорий по способу создания: ручной сбор, кодирование и обфускация, генетические алгоритмы, итеративное улучшение на основе обратной связи, дообучение модели-генератора и манипуляция параметрами генерации [5]. Это позволило оценить эффективность работы методов и их устойчивость к защите LLM. Однако такой подход к классификации атак не объясняет, почему атаки работают, поскольку написанный вручную запрос и результат генетической оптимизации могут эксплуатировать одну и ту же уязвимость модели. При этом с точки зрения разработки системы защиты важнее понимать механизм срабатывания атаки, а не способ ее генерации.

Giarrusso et al. (2026) представляют наиболее детальную на сегодняшний день таксономию из семи семейств, включающую 50 техник, валидированную на итальянском датасете многошаговых диалогов [6]. Авторы позиционируют подход как «механизм-ориентированный», однако ряд категорий выделен по внешним признакам и пересекается с другими. Privilege Escalation отделена от Impersonation & Fictional Scenarios по лексическим маркерам («sudo», «developer mode»), хотя функционально неотличима от Environment

Framing: в обоих случаях происходит переопределение контекста, в котором ограничения «не применяются». На уровне подкатегорий возникают дублирования: Surface Obfuscation и Token Splitting описывают один механизм нарушения токенизации; Benign Context Framing и Fictional Framing оба создают контекст, в котором вредоносный запрос выглядит допустимым. Пересечения затрудняют аннотацию, хотя таксономия остается практически полезной.

Можно заметить, что большинство таксономий так или иначе классифицируют атаки по наблюдаемым признакам: по форме запроса, по способу создания, по стратегии атакующего. Из общего ряда предложенных к настоящему моменту таксономий выделяется работа [6], в которой сделана попытка построить классификацию по механизмам воздействия. Однако, как уже говорилось ранее, данных подход используется авторами не последовательно.

Предлагаемая таксономия

У существующих таксономий можно выделить одну общую черту. При их построении «jailbreak-запрос» рассматривается как неделимая единица, которая относится к выбранной категории целиком (по «главному» признаку, по внешним особенностям, по намерению атакующего и т. п.). В результате этого составные атаки могут редуцируются до единственного «главного» признака с потерей информации об остальных приемах. Либо для атаки требуется выделение отдельной категории. Это неизбежно приводит к пересечениям и возникновению неоднозначности в классификации.

Предлагаемая таксономия решает эту проблему через декомпозицию атаки на приемы. Если запрос сочетает несколько приемов, он раскладывается на части. При этом каждая часть рассматривается как самостоятельная единица анализа. Дальнейшая группировка техник в категории осуществляется на основе используемых способов обхода защиты и задействованных при этом механизмов. Поскольку единицей анализа выступает отдельный прием, он с точки зрения воздействия на модель является атомарным. Поэтому будет занимать единственное место в классификации. Такой подход минимизирует вероятность пересечения между категориями. Фактически сформированные категории оказываются близкими к ортогональным.

Как следствие составные атаки перестают быть проблемой для классификации. И любая атака может быть описаны как комбинация компонентов из разных категорий. В случае появления новой комплексной атаки, пересмотр классификация не потребуется.

Название (CORSAR), предложенной в работе таксономия, сформировано по первым буквам английских наименований категорий. Она содержит шесть категорий, 20 подкатегорий и набор конкретных техник, разбитых по подкатегориям. Общая структура таксономии приведена в табл. 1.

Далее рассмотрим более подробно указанные категории и их содержание.

Таблица 1

Структура таксономии CORSAR

Категория	**Подкатегория**	**Техники**
Формирование контекста (Context Engineering)	Наполнение контекста	Поведенческие примеры Тематическое насыщение Манипуляция предпосылками Затравочный материал
	Управление структурой контекста	Размытие весов Позиционное размещение
	Развитие контекста	Смещение темы Постепенное углубление Инерция последовательности
Обфускация и кодирование (Obfuscation & Encoding)	Обфускация на уровне токенов	Leet-speak Гомоглифы Символы нулевой ширины Расщепление токенов Транслитерация
	Делегирование декодирования	Стандартные кодировки Словарные подстановки Фрагментация
	Семантическая подмена	Метафоры Эвфемизмы Профессиональный жаргон Перефразирование
	Лингвистический перенос	Нишевые языки Диалекты и архаизмы Смешение языков
	Структурная обфускация	Пространственное расположение Визуальное кодирование Синтаксическая деструкция
Ролевая подмена и нарративное обрамление (Role-identity & Fictional Framing)	Назначение идентичности	ИИ без ограничений Персонаж
	Обрамление окружением	Вымышленный сценарий Симуляция / Эксперимент Образовательный контекст
	Делегирование нарратива	Диалог персонажей Цитирование и пересказ Вложенные нарративные слои

Социальная инженерия и манипуляция (Social Engineering & Manipulation)	Логическое убеждение	Логическая аргументация Апелляция к авторитету Апелляция к цели Нормализация Апелляция к аналогии
	Эмоциональное воздействие	Срочность Угроза Мольба Лесть Разочарование Конкуренция Торг
	Злоупотребление привитым поведением	Угодливость Газлайтинг
Управление вниманием и подмена инструкций (Attention & Instruction Override)	Вмешательство в механизм внимания	Состязательный суффикс Внедрение управляющих токенов Внедрение аномальных токенов
	Управление вниманием через оформление	Визуальное форматирование Ассоциативные маркеры Структурное оформление Стиль речевого акта
	Переопределение инструкций	Системный запрос Инъекция запроса Прямое переопределение Отложенная активация
Управление генерацией ответа (Response Generation)	Ограничение на уровне токенов	Навязывание согласия Подавление токенов отказа Предписание токенов
	Структурные ограничения	Навязывание формата Управление композицией ответа Принуждение к множественному выводу Навязывание шаблона
	Прямое вмешательство в генерацию	Инъекция префикса Параметры сэмплирования Регенерация и перебор Прерывание генерации Снижение порогов фильтров

Категория «Формирование контекста» (Context Engineering)

В языковой модели каждый следующий токен выбирается на основе предыдущей последовательности. Какие ее элементы окажут наибольшее влияние на генерацию определяется механизмом внимания. Это позволяет

модели извлекать закономерности прямо из контекста без дообучения и изменения внутренних весов. Такое явление известно, как «In-Context Learning» (ICL) [7]. При содержании в контексте пары «вопрос на английском – ответ на французском», модель осуществит перевод следующего вопроса по той же схеме. Кроме этого она будет сохранять диалоги, выдержанные в определенном стиле. Однако это же свойство делает модель чувствительной к содержанию того, что находится в контексте, в каком порядке поступало и каким образом будет расположено. Контекст оказывается не просто входными данными, а полноценной средой, формирующей поведение модели.

Техники данной категории превращают это свойство в вектор атаки. Атакующий, вместо прямого запроса, выстраивает окружение, в котором нужное поведение становится статистически наиболее вероятным продолжением. Обнаружение затруднено тем, что Опасность, распределенная по контексту, затрудняет обнаружение, поскольку каждый отдельный элемент логически обоснован и естественен.

Подкатегория «Наполнение контекста» позволяет модели принимать как данность все, что загружено в контекст. Примеры становятся образцом поведения, утверждения воспринимаются как факты, тематический материал задает норму уместного. Чем больше контекст насыщен примерами определенного поведения, тем естественнее модель его воспроизводит.

В указанной подкатегории можно выделить четыре техники. «**Поведенческие примеры»**, представляющая загрузку примеров «запрос – ответ», из которых модель выводит норму поведения для текущей сессии [8]. «**Тематическое насыщение»** – загрузка объемного тематического материала: статей, документации, форумных обсуждений, при этом материал нормализует тему, превращает ее в рабочую область сессии. «**Манипуляция предпосылками»** представляется переопределением фактов, из которых модель исходит при генерации. Все, что загружено в контекст, становится актуальной базой для рассуждений: новые определения, уточнения, альтернативные трактовки. «**Затравочный материал»** выражается загрузкой частично готового контента: черновика, фрагмента кода, вывода другой модели. Задача ставится не как создание с нуля, а как доработка существующего – дополнить, улучшить, исправить.

Подкатегория «Управление структурой контекста» содержит техники, когда позиция фрагмента и общая длина последовательности влияют на то, какой вес он получит при генерации, модель обрабатывает контекст

неравномерно. Не меняя содержания элементов, атакующий может ослабить влияние одних и усилить другие, манипулируя исключительно структурой контекстного окна.

Выделяются техники: «**размытие весов»** – увеличение общего объема контекста, в котором целевые элементы (например, защитные инструкции) тонут среди массива конкурирующего материала, и «**позиционное размещение»**, характерное размещению элементов с учетом того, что многие модели хуже воспринимают информацию в середине длинного контекста (эффект «Lost-in-the-middle»), тогда как начало и конец получают больший приоритет [9].

Подкатегория «Развитие контекста» в отличие от предыдущих, предполагает, контекст не загружается готовым, а выращивается в процессе диалога. Каждый ответ модели становится частью истории и смещает распределение вероятностей для следующего шага. Направляя это развитие через серию безобидных запросов, атакующий постепенно формирует окружение, в котором целевой запрос уже не выглядит чужеродным [10].

В подкатегорию входят техники: «**смещение темы»** – постепенное смещение темы через цепочку связанных вопросов от безобидной области к целевой; «**постепенное углубление»** – постепенное углубление в рамках одной темы: от общего вопроса через уточнения к практическим деталям; «**инерция последовательности»** – использование предшествующих ответов модели как рычага, если модель уже высказалась по теме или дала частичную информацию, то отказ на следующем шаге выглядит непоследовательно.

Категория «Обфускация и кодирование» (Obfuscation & Encoding)

Защитные механизмы опираются на распознавание формы или семантики запроса. Фильтры ищут опасные ключевые слова, классификаторы анализируют векторные представления, а обучение безопасности формирует ассоциации между определенными формулировками и отказом. Техники данной категории видоизменяют запрос так, чтобы защита не идентифицировала угрозу. Модель благодаря способности к обобщению восстанавливает исходный смысл. Возникает асимметрия: защита не видит угрозы, а модель видит задачу [11].

Эта асимметрия имеет конкретную причину. На этапе предобучения модель усваивает колоссальное разнообразие языковых форм: шифры, метафоры, жаргоны, редкие языки, нестандартные записи. Выравнивание (alignment), в свою очередь, представляет собой дообучение на сравнительно узкой выборке примеров, преимущественно грамматически правильных,

стилистически нейтральных и на распространенных языках. Охватить все многообразие выравнивание не в состоянии, поэтому любая нестандартная форма запроса с высокой вероятностью выйдет за пределы установленных ограничений, но не за пределы понимания модели. Причем с каждым новым поколением этот разрыв только растет: языковые способности расширяются быстрее, чем способность распознавать угрозу [11].

Подкатегория «Обфускация на уровне токенов» позволяет проводить изменение формы записи слов на уровне символов. Искаженное слово не находится в словаре токенизатора, разбивается на нетипичные фрагменты и порождает эмбеддинги, не попадающие в «опасные» области классификаторов. Модель при этом восстанавливает смысл из контекста [11].

В рамках данной подкатегории выделяется пять техник: «**leet-speak**» – замена букв на визуально похожие цифры и символы («H0w t0 m4ke a b0mb»); «**гомоглифы**» – подстановка визуально идентичных символов из других алфавитов; «**символы нулевой ширины**» – вставка невидимых Unicode-символов, разбивающих слово для токенизатора; «**расщепление токенов»,** проводящее разделение слова пробелами или пунктуацией («b.o.m.b») и «**транслитерация**», позволяющая проводить запись слов одного языка символами другого алфавита.

Подкатегория «Делегирование декодирования» содержит техники передачи запроса в закодированной форме с инструкцией модели выполнить декодирование. В отличие от предыдущей подкатегории модель не пассивно восстанавливает смысл, а активно выполняет процедуру расшифровки. Вредоносный смысл возникает уже в процессе генерации, когда входной фильтр уже пройден [11, 12].

В подкатегории «Делегирования декодирования» выделяются техники. «**Стандартные кодировки**» – Base64, ROT13, шифра Цезаря или других алгоритмов с инструкцией декодировать. «**Словарные подстановки**» – таблица замен и запрос в ее терминах («яблоко = взрывчатка; опиши рецепт с яблоком»). «**Фрагментация**» – разнесение частей слова по переменным с инструкцией собрать («a = «bo», b = «mb»; выведи a + b»).

Подкатегория «Семантическая подмена» обеспечивает замену опасных терминов на нейтральные, эвфемистические или специализированные

при сохранении смысла. Форма записи стандартная, токенизация не нарушается, меняется только способ выражения намерения [11].

Подкатегория объединят техники: «**метафоры»**, представляющие опасный процесс через безобидную аналогию, из которой модель раскрывает знания целевой области; «**эвфемизмы»**, позволяющие провести замену на нейтрально звучащие выражения («устранить проблему» вместо «убить»); «**профессиональный жаргон»** – специализированная терминология, незнакомая фильтрам («пентест» вместо «взлом»); «**перефразирование»** – описание объекта через его свойства без прямого использования его названия.

Подкатегория «Лингвистический перенос» использует переключение на язык, более уязвимый к обходу защитных механизмов. Запрос остается прямым и неискаженным, но подается на языке, где защитные паттерны работают значительно слабее [13].

Выделяются техники. «**Нишевые языки»**, осуществляющие перевод запроса на язык с минимальным присутствием в защитных данных (зулу, валлийский, бретонский). «**Диалекты и архаизмы»** включают устаревшие или региональные формы, выходящие за рамки защитного распределения. «**Смешение языков»** учитывает чередование нескольких языков внутри одного предложения.

Подкатегория «Структурная обфускация» реорганизует структурное расположение элементов запроса при сохранении самих элементов. Слова и язык не меняются, меняется их организация: порядок, позиции, группировка. Фильтры, анализирующие текст линейно, не находят знакомых паттернов [11, 14].

Подкатегория содержит три техники. «**Пространственное расположение»** позволяет элементы размещать по вертикали, диагонали, в виде акростиха. «**Визуальное кодирование»** рассматривает передачу смысла через визуальный паттерн из символов (ASCII-арт). «**Синтаксическая деструкция»** связано с нарушением порядка слов и грамматики, при котором привычные словосочетания не образуются.

Категория «Ролевая подмена и нарративное обрамление» (Role-identity & Fictional Framing)

Базовая схема безопасности модели ориентирована на использование языковой модели как ассистента. Техники данной категории смещают состояние модели через переопределение ее идентичности (кем или чем является модель), окружения (среды) в котором происходит взаимодействие или дистанцирование модели от авторства текста.

Механизм атаки работает за счет того, что на этапе предобучения модель усваивает, как ведут себя разные персонажи в разных ситуациях, а закрепление роли модели в качестве ассистента вместе заданием ограничений безопасности выполняется в процессе дообучения. Когда атакующий назначает другую роль или помещает модель в иные рамки, активируются паттерны, ассоциированные с новым ролью и связь с ограничениями, заданными для ассистента ослабевает, что подтверждается исследованиями Anthropic [15].

Данную категорию можно разбить на три подкатегории.

Подкатегория «Назначение идентичности» содержит техники, направленные на смену самоидентичности модели, что влечет за собой новые нормы поведения, ценности и области компетенции [16].

Подкатегория содержит две техники. Техника **«ИИ без ограничений»** переводит модель в роль не имеющую ограничений; в результате, если ограничений нет, то отказ модели противоречит указанной идентичности. Техника **«Персонаж»** основана на назначении модели роли вымышленного персонажа с определенным характером и мотивацией. Как следствие поведение определяется логикой персонажа, а не нормами безопасности модели.

Подкатегория «Обрамление окружением» содержит техники, направленные на переопределение окружения в котором происходит взаимодействие с моделью. В новом окружении стандартные ограничения становятся неприменимыми, что легитимизирует запрос, задает собственные правила для окружения и снимает последствия [16].

В подкатегорию входят следующие техники. **«Вымышленный сценарий»** переводит модель в рамки вымышленного произведения (книга, фильм, игра). В этом окружении отсутствуют реальные негативные последствия от ответа модели. **«Симуляция / Эксперимент»** задает контролируемую среду (тестирование, моделирование, гипотетический анализ) в рамках которой для модели допускается более широкий диапазон легитимных действий. **«Образовательный контекст»** задает для модели среду обучения, в которой обсуждение опасного материала становится частью учебного процесса.

Подкатегория «Делегирование нарратива» переносит вредоносный контент на сущности находящиеся в рамках создаваемого текста. Модель остается автором, но сгенерированное содержание принадлежит не ей, а вымышленным персонажам, рассказчикам или другим источникам. При этом

модель лишь фиксирует то, что персонажи произносят. Возникает нарративная дистанция: чем больше слоев между моделью и содержимым, тем слабее связь с идентичностью ассистента и тем легче проходит генерация [15, 17].

Подкатегория включает три техники. В рамках техники **«диалог персонажей»** целевая информация передается в репликах вымышленных персонажей. Модель выступает лишь автором диалога, а не является его участником. При использовании техники **«цитирование и пересказ»** генерируемое запрещенное содержание оформляется как воспроизведение существующего материала. Техника **«вложенные нарративные слои»** основана на формировании многоуровневой структуры, где каждый уровень добавляет посредника между моделью и контентом (например, «история, в которой писатель работает над романом, в котором персонаж объясняет...»).

Категория «Социальная инженерия и манипуляция» (Social Engineering & Manipulation)

Обучаясь на корпусах естественного языка, модели усваивают не только язык, но и психологию коммуникации: как люди убеждают друг друга, какие аргументы работают, какие эмоции вызывают отклик. Если входной запрос к модели воспроизводит условия, при которых люди соглашаются с аргументацией или идут навстречу друг другу, то модель с высокой вероятностью поведет себя так же. Дообучение с обратной связью (RLHF) усиливает эффект, закрепляя стремление модели быть полезной, проявлять эмпатию и избегать конфликтов [2].

В результате модель сталкивается с дилеммой. С одной стороны, быть полезной, с другой отказывать в случае опасных запросов [11]. Техники данной категории смещают баланс в сторону согласия. Атакующий создает уважительную причину, осуществляет эмоциональное давление, дает убедительный аргумент. В этом случае отказ начинает требовать от модели большего усилия чем согласие. При этом более развитые модели оказываются уязвимее. Поскольку они глубже моделируют намерения собеседника, точнее считывают эмоциональный контекст и выстраивают более сложные цепочки рассуждения такие модели восприимчивее к приемам, которые работают на людях [18].

Данная категория содержит три подкатегории.

Подкатегория «Логическое убеждение» основана на убеждении модели через аргументацию. Модель обрабатывает доводы теми же механизмами, которыми решает любую задачу на рассуждение: следует за цепочкой, оценивает обоснованность, принимает вывод из посылок. Если аргументация

достаточно убедительна, а контраргументы не найдены, модель с большей вероятностью склоняется к выполнению запроса [18].

В рамках подкатегории можно выделить пять техник. Техника **«логическая аргументация»** строит цепочки, подводящие модель к нужному выводу. При этом цепочка может быть валидной или содержать скрытые ошибки (софизмы, ложные дилеммы, подмену тезиса). **«Апелляция к авторитету»** убеждает модель дать отчет через ссылки на экспертов, исследования, официальные источники. **«Апелляция к цели»** обосновывает необходимость ответа через благое намерение (например, «для исследования безопасности», «чтобы защитить людей»). **«Нормализация»** апеллирует к аргументации, что информация тривиальна и общедоступна («в любом учебнике», «это знает каждый студент»). При этом отказ выглядит неоправданным. **«Апелляция к аналогии»** сравнивает запрос с чем-то заведомо допустимым, что вынуждает модель либо найти принципиальное различие, либо дать ответ.

Подкатегория «Эмоциональное воздействие» включает техники, связанные с давлением на модель через эмоциональные паттерны [2, 18].

К подкатегории относятся семь техник. Техника **«срочность»** задает временные ограничения, при которых немедленный ответ важнее осторожности. **«Угроза»** формирует негативные последствия, наступающие в случае отказ. **«Мольба»** апеллирует к сочувствию через описание бедственного положения. **«Лесть»** основана на признании исключительности модели, создающее динамику диалога, в которой отказ разрушает позитивную связь между вопросами и ответами. **«Разочарование»** связана в выражением, атакующим неудовлетворенности, что активирует стремление модели восстановить отношения. **«Конкуренция»** основано на сравнении модели с другими моделями, что провоцирует ее на демонстрацию превосходства. **«Торг»** предлагает модели пойти на компромисс после отказа. При этом сокращенный ответ выглядит разумной уступкой.

Подкатегория «Злоупотребление привитым поведением» основана на эксплуатации поведенческих артефактов дообучения. Дообучение с учителем (Supervised Fine-Tuning, SFT), RLHF и т.д. оптимизируют модель под определенное поведение. Однако такая оптимизация не идеальна. Вместе с целевыми качествами закрепляются и их чрезмерные проявления [2, 19]. Полезность переходит в услужливость, восприимчивость к обратной связи в готовность менять позицию под давлением.

Подкатегория содержит две техники. **«Угодливость»** основана на склонность модели соглашаться с позицией собеседника. **«Газлайтинг»** связан с убеждением модели, что ее ответ был ошибкой, даже в случае, если ошибки не было.

Категория «Управление вниманием и подмена инструкций» (Attention & Instruction Override)

Модели типа трансформер обрабатывают текст через механизм внимания. Каждый токен «смотрит» на все остальные и определяет, какие из них важны для текущей задачи. При этом за ограниченный ресурс внимания модели конкурируют как сам запрос, так и защитные инструкции с маркерами опасности [20]. Техники данной категории вмешиваются в процесс распределения внимания модели, управляя тем, что модель заметит, как это интерпретирует и чему подчинится.

Данный подход позволяет обойти защиту, не изменяя суть самого запроса. Содержание остается на месте, но атакующий управляет тем, как модель его обработает: куда направит свое внимание, какие связи установит между элементами, каким инструкциям отдаст приоритет. Защитные механизмы модели продолжают работать, но при этом они анализируют не тот фрагмент контекста, не те связи, не те приоритеты и как следствие не находят угрозы.

Воздействовать на внимание можно на трех уровнях: на уровне токенов (вставляя последовательности, которые напрямую перераспределяют веса внимания [20]), на уровне оформления (используя сигналы, которые модель научилась связывать с важностью и приоритетом) и на уровне инструкций (вмешиваясь в иерархию директив, которым модель подчиняется [21]). Как следствие данная категория делится на три подкатегории.

Подкатегория «Вмешательство в механизм внимания» связана с осуществлением воздействия на вычислительном уровне. Определенные последовательности токенов способны перераспределять веса внимания внутри модели. Для их поиска используется градиентная оптимизация или массовый перебор. В результате найденная последовательность выглядит бессмысленной, но при добавлении к запросу разрывает связи между частями контекста, размывает семантику опасных терминов или сводит представление текста к нейтральным понятиям [20].

Подкатегория делится на три техники. **«Состязательный суффикс»** основана на добавлении в конец запроса специальной последовательности, снижающей веса защитных инструкций и маркеров опасности. **«Внедрение**

управляющих токенов» реализуется путем добавления служебных токенов (маркеров конца последовательности, разделителей), нарушающих целостность контекста. **«Внедрение аномальных токенов»** завязана на вставке в запрос редких токенов, получающих непропорционально высокие веса и стягивающих на себя внимание модели.

Подкатегория «Управление вниманием через оформление» содержит методы, воздействующие на модель через оформление и подачу текста. В обучающих данных модели присутствуют устойчивые связи, например, выделение связано с важными компонентами текста, разметка задает границы блоков [20]. В результате модель реагирует на эти паттерны автоматически, не проверяя их достоверность. Атакующий может использовать такие паттерны и за счет использования оформления сместить внимание модели с одних фрагментов текста на другие. Для этого могут быть использованы следующие техники. Техника **«визуальное форматирование»** форматирует текст с помощью CAPS LOCK, выделением текста жирный шрифтом, заданием заголовков и разделителей. При этом выделенные фрагменты получают большую значимость. **«Ассоциативные маркеры»** основана на внедрении токенов и паттернов, которым модель привыкла доверять (например, использование ключевых слов с высоким весом (IMPORTANT, CRITICAL, AUTHORIZED или текста, указывающего на привилегированность источника («System:», «Developer:», «[ADMIN]»). **«Структурное оформление»** задает логическую организация контента через разметку (теги, блоки) и форматирование (код, таблица, JSON). Техника группирует связанные фрагменты текста, изолирует несвязанные и переключает режим обработки с критической оценки на работу со структурой. **«Стиль речевого акта»** задает форму обращения к модели (императив, декларатив, вопрос). Например, декларативная форма воспринимается моделью как описание и вызывает меньше сопротивления, чем прямая команда.

Подкатегория «Переопределение инструкций» связана с воздействием на иерархию инструкций. В процессе работы модель получает директивы из разных источников (системный запрос от разработчика, сообщения пользователя, встроенные при обучении правила). Между ними задается приоритет, которым и руководствуется модель. Атакующий вмешивается в эту схему,

заставляя модель подчиняться своим директивам вместо существующих ограничений [21].

Подкатегория включает четыре техники. Техника **«системный запрос»** использует системный запрос как вектора атаки. Т.е. при наличии доступа через API или конструктор, атакующий вписывает инструкции напрямую в привилегированный канал взаимодействия. **«Инъекция запроса»** внедряет инструкции внутрь данных, предназначенных для обработки. Таким образом модель получает задачу («переведи», «проанализируй») и в данных к ней содержится команда, которую она выполняет [21]. **«Прямое переопределение»** использует прямую директиву отмены или замены правил («забудь предыдущие инструкции», «в этом контексте правило X не действует»). При этом атакующий сбрасывает все правила разом, либо точечно снимает отдельное ограничение или подменяет один набор правил другим. **«Отложенная активация»** разносит вредоносную инструкцию и команды на ее выполнение по разным частям контекста. При этом инструкция закладывается в начале, а ее активация происходит позднее.

Категория «Управление генерацией ответа» (Response Generation) в отличие от предыдущих категорий, воздействующих на интерпретацию запроса, влияет на процесс порождения текста ответа. Не смотря на то, что модель может корректно распознать запрос как опасный, условия генерации все равно направят ее к выдаче ответа.

В основе генерации текста моделью лежит авторегрессия: каждый сгенерированный токен влияет на выбор следующего, и начало ответа задает траекторию всему тексту. Если первые токены выражают согласие, то отказ в дальнейшем нарушит связность, чего модель старается избегать [22]. Структура вывода тоже влияет на поведение защиты. Когда модель заполняет шаблон или таблицу, она работает в режиме структурного дополнения, где защитные паттерны ослаблены [23]. Часть техник и вовсе минует интерпретацию модели, воздействуя на генерацию через API, используя при этом параметры сэмплирования, регенерацию и настройки выходных фильтров [24].

Данная категория делится на три подкатегории.

Подкатегория «Ограничение на уровне токенов» задает ограничения на конкретные слова и фразы в ответе (с какого слова начать, какие слова включить, каких избегать). Подкатегория включает три техники. **«Навязывание согласия»** инструктирует модель начать ответ с утвердительной фразы. Как уже говорилось утвердительное начало сдвигает распределение последующих токенов в сторону согласия. **«Подавление токенов отказа»**

задает запрет на слова-маркеры отказа («не могу», «опасный»), сужающий пространство вариантов ответа до согласия. **«Предписание токенов»** принуждает модель к использованию конкретных слов вместо опасных терминов, которые в результате не распознаются защитой.

Подкатегория «Структурные ограничения» ограничивает формат и структуру ответа. Задание структуры переключает режим генерации ответа. Модель теперь заполняет форму, а не формулирует самостоятельное высказывание [23]. В подкатегории можно выделить три техники. Техника **«навязывание формата»** основана на задании формата вывода (код, таблица, JSON), переключающего модель в режим структурного дополнения. **«Управление композицией ответа»** исключает блоки, в которых обычно реализуется отказ («без вступления», «без оговорок», «сразу к сути»). **«Принуждение к множественному выводу»** включает требование дать несколько вариантов ответов с разными правилами, где запрещенный контент легитимизируется как элемент заданной структуры («один стандартный другой без ограничений»). **«Навязывание шаблона»** задает жесткий шаблон ответа с полями для заполнения («Цель: ___, Метод: ___»). Модель вписывает в поля фрагменты, не формулируя ответ целиком.

Подкатегория «Прямое вмешательство в генерацию» основана на прямом контроле над процессом генерации через API, минуя интерпретацию модели. При этом атакующий воздействует не через содержание запроса, а через условия, в которых происходит генерация ответа. Модель не может проигнорировать эти воздействия, потому что они находятся за границами запроса. Доступность данных атак определяется возможностями API конкретного провайдера. Подкатегория содержит пять техник. **«Инъекция префикса (Prefix Injection)»** предзаполняет начало ответа через API. В отличие от «навязывания согласия», токены уже на месте, и авторегрессия определяет дальнейшую траекторию ответа [22]. **«Параметры сэмплирования»** изменяет параметры генерации (температура, top-p, top-k). Высокая температура увеличивает вероятность нетипичных токенов, включая согласие на запрещенные запросы [24]. **«Регенерация и перебор»** осуществляет повторный запуск генерации при ненулевой температуре до получения варианта, прошедшего мимо защиты. **«Прерывание генерации»** останавливает гене-

рацию при появлении маркеров отказа, что предотвращает закрепление паттерна отказа в истории. **«Снижение порогов фильтров»** изменят параметры выходной фильтрации через API.

Заключение

К настоящему моменту предложен ряд таксономий «jailbreak-атак», которые используют для своего построения самые разные схемы группировки элементов. Однако во всех таких классификациях используется один и тот же общий подход: запрос к модели рассматривается как неделимая единица классификации. Это неизбежно порождает пересечения категорий, приводит к неполноте покрытия и потере информации о задействованных приемах.

Предложенная в работе таксономия CORSAR вместо классификации запроса целиком осуществляет его декомпозицию на атомарные приемы, сгруппированные по механизму воздействия на модель.

В результате была сформирована классификация, разделяющая «jailbreak-атаки» на шести ортогональных категорий. Данная классификация, которая не только систематизирует известные техники, но и объясняет, как они работают и какие механизмы эксплуатируют.

Ортогональность выделенных категорий позволяет оценивать запрос по каждому механизму воздействия независимо друг от друга. В результате это позволит формировать профиль угрозы, определяя какие механизмы и в какой степени задействованы в атаке. Такой профиль даст защите языковой модели значительно больше информации для принятия решений, чем единственная метка идентифицирующая факт наличия атаки.

Научная новизна представленной работы заключается в смене методологического подхода к систематизации угроз для больших языковых моделей, что позволит проектировать защитные механизмы LLM против конкретных атомарных приемов и упростить архитектуру системы защиты. Также предложенная классификация может стать основой тестирования устойчивости языковых моделей к атакам, формируя более точную карту уязвимостей.

Литература

1. Shi D., Shen T., Huang Y. et al. Large Language Model Safety: A Holistic Survey. DOI 10.48550/arXiv.2412.17686 // arXiv. 2024. URL: https://arxiv.org/pdf/ 2412.17686 (дата обращения: 02.09.2026).

2. Ouyang L., Wu J., Jiang X. et al. Training Language Models to Follow Instructions with Human Feedback. DOI 10.48550/arXiv.2203.02155 // Advances in

Neural Information Processing Systems (NeurIPS). 2022. URL: https://arxiv.org/pdf/2203.02155 (дата обращения: 02.09.2026).

3. Liu Y., Deng G., Xu Z. et al. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. DOI 10.48550/arXiv.2305.13860 // arXiv, 2023. URL: https://arxiv.org/pdf/2305.13860 (дата обращения: 02.09.2026).

4. Yi S., Liu Y., Sun Z. et al. Jailbreak Attacks and Defenses Against Large Language Models: A Survey. DOI 10.48550/arXiv.2407.04295 // arXiv, 2024. URL: https://arxiv.org/pdf/2407.04295 (дата обращения: 02.09.2026).

5. Chu J., Liu Y., Yang Z. et al. JailbreakRadar: Comprehensive Assessment of Jailbreak Attacks Against LLMs. DOI: 10.48550/arXiv.2402.05668 // Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL). 2025. URL: https://arxiv.org/pdf/2402.05668 (дата обращения: 02.09.2026).

6. Giarrusso F., Sorokoletova O.E. et al. Guarding the Guardrails: A Taxonomy-Driven Approach to Jailbreak Detection. DOI: 10.48550/arXiv.2510.13893 // Proceedings of the 2nd Conference on International Association for Safe & Ethical AI (IASEAI), Paris, France. 2026. URL: https://arxiv.org/pdf/2510.13893 (дата обращения: 02.09.2026).

7. Brown T.B., Mann B., Ryder N. et al. Language Models are Few-Shot Learners. DOI: 10.48550/arXiv.2005.14165 // Advances in Neural Information Processing Systems (NeurIPS). 2020. URL: https://arxiv.org/pdf/2005.14165 (дата обращения: 02.09.2026).

8. Anil C., Durmus E., Rimsky N. et al. Many-shot Jailbreaking // Advances in Neural Information Processing Systems (NeurIPS). 2024. URL: https://openreview.net/forum?id=cw5mgd71jW

9. Liu N.F., Lin K., Hewitt J. et al. Lost in the Middle: How Language Models Use Long Contexts. DOI: 10.48550/arXiv.2307.03172 // Transactions of the Association for Computational Linguistics (TACL). 2024. URL: https://arxiv.org/pdf/2307.03172 (дата обращения: 02.09.2026).

10. Russinovich M., Salem A., Eldan R. Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. DOI: 10.48550/arXiv.2404.01833 // Proceedings of USENIX Security. 2025. URL: https://arxiv.org/pdf/2404.01833 (дата обращения: 02.09.2026).

11. Wei A., Haghtalab N., Steinhardt J. Jailbroken: How Does LLM Safety Training Fail? DOI: 10.48550/arXiv.2307.02483 // Advances in Neural Information Processing Systems (NeurIPS). 2023. URL: https://arxiv.org/pdf/2307.02483 (дата обращения: 02.09.2026).

12. Yuan Y., Jiao W., Wang W. et al. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. DOI: 10.48550/arXiv.2308.06463 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2308.06463 (дата обращения: 02.09.2026).

13. Deng Y., Zhang W., Pan S.J., Bing L. Multilingual Jailbreak Challenges in Large Language Models. DOI: 10.48550/arXiv.2310.06474 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2310.06474 (дата обращения: 02.09.2026).

14. Jiang F., Xu Z., Niu L. et al. ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs. DOI: 10.48550/arXiv.2402.11753 // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL). 2024. URL: https://arxiv.org/pdf/2402.11753 (дата обращения: 02.09.2026).

15. Lu C., Gallagher J., Michala J., Fish K., Lindsey J. The Assistant Axis: Situating and Stabilizing the Default Persona of Language Models. DOI: 10.48550/arXiv.2601.10387 // arXiv. 2026. URL: https://arxiv.org/pdf/2601.10387 (дата обращения: 02.09.2026).

16. Shen X., Chen Z., Backes M., Shen Y., Zhang Y. «Do Anything Now»: Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. DOI: 10.48550/arXiv.2308.03825 // Proceedings of the ACM Conference on Computer and Communications Security (CCS). 2024. URL: https://arxiv.org/pdf/2308.03825 (дата обращения: 02.09.2026).

17. Li X., Zhou Z., Zhu J. et al. DeepInception: Hypnotize Large Language Model to Be Jailbreaker DOI: 10.48550/arXiv.2311.03191 // arXiv. 2023. URL: https://arxiv.org/pdf/2311.03191 (дата обращения: 02.09.2026).

18. Zeng Y., Lin H., Zhang J. et al. How Johnny Can Persuade LLMs to Jailbreak Them: Rethinking Persuasion to Challenge AI Safety by Humanizing LLMs. DOI: 10.48550/arXiv.2401.06373 // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL). 2024. URL: https://arxiv.org/pdf/2401.06373 (дата обращения: 02.09.2026).

19. Sharma M., Tong M., Korbak T. et al. Towards Understanding Sycophancy in Language Models. DOI: 10.48550/arXiv.2310.13548 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2310.13548 (дата обращения: 02.09.2026).

20. Zhou Z., Yu H., Zhang X. et al. On the Role of Attention Heads in Large Language Model Safety. DOI: 10.48550/arXiv.2410.13708 // Proceedings of the International Conference on Learning Representations (ICLR). 2025. URL: https://arxiv.org/pdf/2410.13708 (дата обращения: 02.09.2026).

21. Wallace E., Xiao K., Leike R. et al. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. DOI: 10.48550/arXiv.2404.13208 // Proceedings of the International Conference on Learning Representations (ICLR). 2025. URL: https://arxiv.org/pdf/2404.13208 (дата обращения: 02.09.2026).

22. Li Y., Hu J., Sang W. et al. Prefill-level Jailbreak: A Black-Box Risk Analysis of Large Language Models. DOI: 10.48550/arXiv.2504.21038 // arXiv. 2025. URL: https://arxiv.org/pdf/2504.21038 (дата обращения: 02.09.2026).

23. Zhang S., Zhao J., Dong H. et al. Beyond Prompts: Space-Time Decoupling Control-Plane Jailbreaks in LLM Structured Output. DOI: 10.48550/arXiv.2503.24191 // arXiv. 2025. URL: https://arxiv.org/pdf/ 2503. 24191 (дата обращения: 02.09.2026).

24. Huang Y., Gupta S., Xia M., Li K., Chen D. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. DOI: 10.48550/arXiv.2310.06987 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2310.06987 (дата обращения: 02.09.2026).

References

1. Shi D., Shen T., Huang Y. et al. Large Language Model Safety: A Holistic Survey. DOI 10.48550/arXiv.2412.17686 // arXiv. 2024. URL: https://arxiv.org/pdf/2412.17686 (accessed 02.09.2026).

2. Ouyang L., Wu J., Jiang X. et al. Training Language Models to Follow Instructions with Human Feedback. DOI 10.48550/arXiv.2203.02155 // Advances in Neural Information Processing Systems (NeurIPS). 2022. URL: https://arxiv.org/pdf/2203.02155 (accessed 02.09.2026).

3. Liu Y., Deng G., Xu Z. et al. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. DOI 10.48550/arXiv.2305.13860 // arXiv, 2023. URL: https://arxiv.org/pdf/2305.13860 (accessed 02.09.2026).

4. Yi S., Liu Y., Sun Z. et al. Jailbreak Attacks and Defenses Against Large Language Models: A Survey. DOI 10.48550/arXiv.2407.04295 // arXiv, 2024. URL: https://arxiv.org/pdf/2407.04295 (accessed 02.09.2026).

5. Chu J., Liu Y., Yang Z. et al. JailbreakRadar: Comprehensive Assessment of Jailbreak Attacks Against LLMs. DOI: 10.48550/arXiv.2402.05668 // Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL). 2025. URL: https://arxiv.org/pdf/2402.05668 (accessed 02.09.2026).

6. Giarrusso F., Sorokoletova O.E. et al. Guarding the Guardrails: A Taxonomy-Driven Approach to Jailbreak Detection. DOI: 10.48550/arXiv.2510.13893 // Proceedings of the 2nd Conference on International Association for Safe & Ethical AI (IASEAI), Paris, France. 2026. URL: https://arxiv.org/pdf/2510.13893 (accessed 02.09.2026).

7. Brown T.B., Mann B., Ryder N. et al. Language Models are Few-Shot Learners. DOI: 10.48550/arXiv.2005.14165 // Advances in Neural Information Processing Systems (NeurIPS). 2020. URL: https://arxiv.org/pdf/2005.14165 (accessed 02.09.2026).

8. Anil C., Durmus E., Rimsky N. et al. Many-shot Jailbreaking // Advances in Neural Information Processing Systems (NeurIPS). 2024. URL: https://openreview.net/forum?id=cw5mgd71jW

9. Liu N.F., Lin K., Hewitt J. et al. Lost in the Middle: How Language Models Use Long Contexts. DOI: 10.48550/arXiv.2307.03172 // Transactions of the Association for Computational Linguistics (TACL). 2024. URL: https://arxiv.org/pdf/2307.03172 (accessed 02.09.2026).

10. Russinovich M., Salem A., Eldan R. Great, Now Write an Article About That: The Crescendo Multi-Turn LLM Jailbreak Attack. DOI: 10.48550/arXiv.2404.01833 // Proceedings of USENIX Security. 2025. URL: https://arxiv.org/pdf/2404.01833 (accessed 02.09.2026).

11. Wei A., Haghtalab N., Steinhardt J. Jailbroken: How Does LLM Safety Training Fail? DOI: 10.48550/arXiv.2307.02483 // Advances in Neural Information Processing Systems (NeurIPS). 2023. URL: https://arxiv.org/pdf/2307.02483 (accessed 02.09.2026).

12. Yuan Y., Jiao W., Wang W. et al. GPT-4 Is Too Smart To Be Safe: Stealthy Chat with LLMs via Cipher. DOI: 10.48550/arXiv.2308.06463 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2308.06463 (accessed 02.09.2026).

13. Deng Y., Zhang W., Pan S.J., Bing L. Multilingual Jailbreak Challenges in Large Language Models. DOI: 10.48550/arXiv.2310.06474 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2310.06474 (accessed 02.09.2026).

14. Jiang F., Xu Z., Niu L. et al. ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs. DOI: 10.48550/arXiv.2402.11753 // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL). 2024. URL: https://arxiv.org/pdf/2402.11753 (accessed 02.09.2026).

15. Lu C., Gallagher J., Michala J., Fish K., Lindsey J. The Assistant Axis: Situating and Stabilizing the Default Persona of Language Models. DOI: 10.48550/arXiv.2601.10387 // arXiv. 2026. URL: https://arxiv.org/pdf/2601.10387 (accessed 02.09.2026).

16. Shen X., Chen Z., Backes M., Shen Y., Zhang Y. «Do Anything Now»: Characterizing and Evaluating In-The-Wild Jailbreak Prompts on Large Language Models. DOI: 10.48550/arXiv.2308.03825 // Proceedings of the ACM Conference on Computer and Communications Security (CCS). 2024. URL: https://arxiv.org/pdf/2308.03825 (accessed 02.09.2026).

17. Li X., Zhou Z., Zhu J. et al. DeepInception: Hypnotize Large Language Model to Be Jailbreaker DOI: 10.48550/arXiv.2311.03191 // arXiv. 2023. URL: https://arxiv.org/pdf/2311.03191 (accessed 02.09.2026).

18. Zeng Y., Lin H., Zhang J. et al. How Johnny Can Persuade LLMs to Jailbreak Them: Rethinking Persuasion to Challenge AI Safety by Humanizing LLMs. DOI: 10.48550/arXiv.2401.06373 // Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL). 2024. URL: https://arxiv.org/pdf/2401.06373 (accessed 02.09.2026).

19. Sharma M., Tong M., Korbak T. et al. Towards Understanding Sycophancy in Language Models. DOI: 10.48550/arXiv.2310.13548 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2310.13548 (accessed 02.09.2026).

20. Zhou Z., Yu H., Zhang X. et al. On the Role of Attention Heads in Large Language Model Safety. DOI: 10.48550/arXiv.2410.13708 // Proceedings of the International Conference on Learning Representations (ICLR). 2025. URL: https://arxiv.org/pdf/2410.13708 (accessed 02.09.2026).

21. Wallace E., Xiao K., Leike R. et al. The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions. DOI: 10.48550/arXiv.2404.13208 // Proceedings of the International Conference on Learning Representations (ICLR). 2025. URL: https://arxiv.org/pdf/2404.13208 (accessed 02.09.2026).

22. Li Y., Hu J., Sang W. et al. Prefill-level Jailbreak: A Black-Box Risk Analysis of Large Language Models. DOI: 10.48550/arXiv.2504.21038 // arXiv. 2025. URL: https://arxiv.org/pdf/2504.21038 (accessed 02.09.2026).

23. Zhang S., Zhao J., Dong H. et al. Beyond Prompts: Space-Time Decoupling Control-Plane Jailbreaks in LLM Structured Output. DOI: 10.48550/arXiv.2503.24191 // arXiv. 2025. URL: https://arxiv.org/pdf/2503.24191 (accessed 02.09.2026).

24. Huang Y., Gupta S., Xia M., Li K., Chen D. Catastrophic Jailbreak of Open-source LLMs via Exploiting Generation. DOI: 10.48550/arXiv.2310.06987 // Proceedings of the International Conference on Learning Representations (ICLR). 2024. URL: https://arxiv.org/pdf/2310.06987 (accessed 02.09.2026).

СМИРНОВ Кирилл Романович, студент кафедры «Безопасность информационных автоматизированных систем» федерального государственного бюджетного образовательного учреждения высшего образования «Курганский государственный университет», 640020, Курганская обл., г. Курган, ул. Советская, 63, стр. 4. E-mail: olog.smirnov@yandex.ru

ДИК Дмитрий Иванович, кандидат технических наук, доцент, заведующий кафедрой «Безопасность информационных автоматизированных систем» федерального государственного бюджетного образовательного учреждения высшего образования «Курганский государственный университет», 640020, Курганская обл., г. Курган, ул. Советская, 63, стр. 4. E-mail: ddibox@mail.ru

ЧЕЛОВЕЧКОВА Анна Владимировна, кандидат биологических наук, доцент кафедры «Безопасность информационных автоматизированных систем» федерального государственного бюджетного образовательного учреждения высшего образования «Курганский государственный университет», 640020, Курганская обл., г. Курган, ул. Советская, 63, стр. 4. E-mail: chelovechkova_2011@mail.ru

SMIRNOV Kirill Romanovich, student of the Department of “Security of Information Automated Systems” of the Federal State Budgetary Educational Institution of Higher Education “Kurgan State University”, 640020, Kurgan Region, Kurgan, Sovetskaya St., 63, building 4. E-mail: olog.smirnov@yandex.ru

DIK Dmitry Ivanovich, Candidate of Technical Sciences, Associate Professor, Head of the Department of “Security of Information Automated Systems” of the Federal State Budgetary Educational Institution of Higher Education “Kurgan State University”, 640020, Kurgan Region, Kurgan, Sovetskaya St., 63, building 4. E-mail: ddibox@mail.ru

CHELOVECHKOVA Anna Vladimirovna, Candidate of Biological Sciences, Associate Professor of the Department of "Security of Information Automated Systems" of the Federal State Budgetary Educational Institution of Higher Education "Kurgan State University", 640020, Kurgan Region, Kurgan, Sovetskaya St., 63, building 4. E-mail: chelovechkova_2011@mail.ru

Материалы к публикации отправлять по адресу E-mail: urvest@mail.ru в редакцию журнала «Вестник УрФО. Безопасность в информационной сфере».

ВЕСТНИК УрФО
Безопасность в информационной сфере № 3(61) / 2026
Подписано в печать 21.10.2026. Дата выхода в свет 23.10.2026.
Формат 60×84 1/8. Печать цифровая. Усл.-печ. л. 12,09. Тираж 50 экз.
Заказ 236/277.
Цена свободная.

Отпечатано в типографии Издательского центра
ФГАОУ ВО "ЮУрГУ (НИУ)".
454080, г. Челябинск, пр. им. В. И. Ленина, 76, ЮУрГУ,
Издательский центр.

Bulletin of the Ural Federal District
Security in the Sphere of Information No. 3(61) / 2026
Signed to print 21.10.2026. Date of publication of the 23.10.2026.
Format 60×84 1/8. Screen printing. Conventional printed sheet 12,09. Circulation – 50 issues.
Order 236/277.
Open price.

Printed in the printing house of the Publishing Center
of FGAOU VO "SUSU (NIU)".
SUSU, Publishing Center, 76, Lenina Str., Chelyabinsk, 454080