Метод классификации угроз нарушения информационной безопасности системы обработки данных

Д. О. Дедов

Санкт-Петербургский государственный электротехнический университет ЛЭТИ имени В.И. Ульянова Ленина

Dedovdanil4@yandex.ru

Р. Р. Фаткиева

Санкт-Петербургский государственный электротехнический университет ЛЭТИ имени В.И. Ульянова Ленина

rikki2@yandex.ru

Аннотация. **Рассматривается метод классификации и анализа угроз нарушения информационной безопасности системы обработки данных на основе анализа системных журналов событий. Предложенный метод основан на многоэтапной обработке событий операционной системы, включающей нормализацию записей журналов, извлечение типовых событий, формирование последовательностей действий пользователей и системных процессов, а также их последующую классификацию с использованием нейросетевой модели на основе архитектуры трансформера. Результаты экспериментальных исследований продемонстрировали возможность эффективной классификации инцидентов информационной безопасности на основе анализа последовательностей событий, полученных из системных журналов.**

Ключевые слова: информационная безопасность, классификация инцидентов, информационные угрозы, анализ событий безопасности, нейронные сети

I. Введение

Современные системы обеспечения информационной безопасности базируются на использовании статических правил и сигнатурных методов и не обладают полноценными механизмами динамического выявления нарушений при обработке данных. Связано это с тем, что методы обнаружения нарушений, как правило, функционируют или на сетевом, или на системном уровнях и не осуществляют комплексную оценку выявления деструктивных воздействий, а также зависят от вида программного обеспечения анализируемого объекта. В обоих случаях анализ журналов логирования позволяет выделить основные аномалии, проанализировать их корреляцию и уникальный набор признаков, идентифицирующих нарушения [1]. Для выявления пропущенных инцидентов информационной безопасности в [2] предложен метод повторного анализа событий безопасности. Многоуровневый анализ позволяет увеличить количество выявленных аномалий, однако требует увеличение вычислительных мощностей и не выявляет нарушения информационной безопасности «на лету». Применение правило-ориентированного метода к анализу событий представлено в [3], однако предложенное решение опирается на сигнатурный анализ, что предполагает пропуск атак «нулевого дня». К отдельному классу обнаружения нарушений можно отнести методики, нацеленные на анализ данных в оперативной памяти информационной системы [4] применение которых целесообразно для выявления подозрительной активности, но может быть ограничено количеством выделяемой для анализа оперативной памятью. В [5] предложен подход к расследованию продолжающегося инцидента информационной безопасности, который ограничен видом операционной системы. Работы [6–7] исследует классификацию нарушений в сетевом трафике, однако предложение методы имеют ограничения по применяемым характеристикам.

Анализ существующих подходов показал, что основные проблемы заключаются в том, что применение интеллектуальных методов в реальных информационных системах затрудняется высокими требованиями к качеству исходных данных и вычислительным ресурсам, а также сложностью их интеграции с уже существующими средствами защиты. В результате отсутствует комплексный подход к интеллектуальному анализу и категоризации инцидентов информационной безопасности.

II. Метод классификации и анализа угроз нарушения информационной безопасности

Метод предназначен для автоматического выявления инцидентов информационной безопасности. Основная идея заключается в представлении событий, происходящих в операционной системе в виде последовательности с последующей их классификацией. В работе метода используется нейронная сеть архитектуры Transformer (BERT) [9], которая позволяет анализировать последовательности событий с учетом контекста, что делает возможным выявление сложных многоэтапных атак.

На первом этапе производится сбор данных из системных журналов операционной системы [10]. На базе собранных данных формируется множество событий-инцидентов $D = \{d_1, d_2, d_3, d_4, d_5\}$, где каждый элемент d_i представляет собой вектор признаков события, полученного из следующих системных журналов: d_1 – данные журналов auth.log/secure, содержащие информацию о процессах аутентификации, SSH-подключениях и использовании sudo; d_2 – данные журналов syslog/messages/journald, отражающие системные события и работу служб; d_3 – данные журнала yum.log, содержащие события загрузки файлов; d_4 – данные журнала audit.log, фиксирующие действия пользователей и системные операции; d_5 – данные

журнала cron.log, содержащие события планировщика задач.

Каждый вектор события d_i описывается признаковым пространством X = (x_1, x_2, …, x_n), где x_i представляет значение соответствующего признака, характеризующего событие (метрика, атрибут или параметр лога).

На втором этапе осуществляется нормализация и агрегация полученных на первом этапе данных. Для этого осуществляется преобразования текстовых записей признакового пространства в структурированные события, содержащие ключевые параметры: время; тип сервиса; характеристики пользователя; IP-адреса; виды сообщений. Полученные структурированные события в дальнейшем используются для определения типа события и анализа последовательностей в действиях процесса, вызвавшего событие в системе.

На третьем этапе формируется набор правил, основанных на ключевых словах и регулярных выражениях, позволяющих осуществить идентификацию нарушения в виде шаблонов, по типам события.

На четвертом этапе выполняется сопоставление каждого шаблона определенному типу события. Для этого каждому событию присваивается уникальный идентификатор события: Event = (EventId, EventTemplate), где EventId – идентификатор типа события; EventTemplate – текстовый шаблон события.

На четвертом этапе проводится агрегации событий. События группируются в инциденты. Инцидентом является цепочка связанных во времени событий. Для этого события сортируются по времени, далее группируются по IP адресу источника и его характеристикам, что позволяет сформировать последовательности сессий. Полученные последовательности событий используются далее для обучения модели BERT, которая выполняет классификацию инцидентов информационной безопасности.

На пятом этапе осуществляется формирование обучающего датасета. Основной задачей которого является преобразование последовательностей событий, полученных на предыдущих этапах обработки, в формат, пригодный для подачи в нейронную сеть. На этом этапе выполняются следующие операции:

- формирование последовательностей событий;
- подготовка текстового представления последовательностей;
- присвоение меток классов;
- сохранение данных в формате обучающего набора.

Результатом этапа является файл обучающего датасета, содержащий входные последовательности и соответствующие им метки классов.

На шестом этапе полученный датасет подается на вход нейронной сети для обучения и последующего определения типа активности системы или возможного инцидента информационной безопасности. Процесс классификации включает несколько последовательных этапов обработки данных внутри нейронной сети. Модель состоит из следующих компонентов: слой токенизации; слой эмбеддингов; трансформер-энкодер; классификационный слой.

Первой стадией обработки входных данных является токенизация, которая представляет собой процесс преобразования входной текстовой последовательности событий в набор токенов, понятных нейронной сети.

На следующей стадии каждый токен преобразуется в вектор признаков фиксированной размерности. Если размерность эмбеддинга равна n, то каждый токен представляется вектором: $d_i \in R^n$, где d_i – вектор события; n – размерность эмбеддинга. Вектор эмбеддинга содержит информацию о семантическом значении события. Дополнительно к эмбеддингам токенов добавляются позиционные эмбеддинги, которые кодируют порядок элементов в последовательности.

Далее последовательности поступают на основной компонент модели – трансформер-энкодер, который выполняет анализ взаимосвязей между событиями последовательности. Энкодер состоит из нескольких слоев, каждый из которых включает:

- *механизм self-attention*, который позволяет каждому элементу последовательности учитывать влияние других элементов [11]. На этой стадии для каждого токена вычисляются три вектора:

$$Q = ZW_Q$$
$$K = ZW_K$$
$$V = ZW_V,$$

где Q – вектор запроса; K – вектор ключа; V – вектор значения; W_Q, W_K, W_V – матрицы весов.

- *механизм внимания* – модель определяет, какие события в последовательности наиболее важны для текущего контекста:

$$Attention\ (Q,K,V) = softmax\ (\frac{QK^T}{\sqrt{d_k}})V,$$

где K^T – транспонированная матрица ключей; QK^T – матрица сходства; d_k – размерность вектора ключей; V – матрица значений.

Последующая обработка осуществляется на полносвязном слое. После прохождения последовательности через трансформер-энкодер используется выходной вектор токена – CLS. Этот вектор содержит обобщённую информацию обо всей последовательности событий. Вектор подается на полносвязный слой и выполняется линейное преобразование: $z_k = W \cdot h_{CLS} + b$, где W – матрица весов; b – вектор смещений; z – вектор выходных значений. Полученный вектор z содержит оценки (logits**)** для каждого класса. Вероятность принадлежности последовательности к тому или иному классу нарушений определяется с применением функции Softmax:

$$P\ (y = k \mid X) = \frac{e^{z^k}}{\sum_{j=1}^{K} e^{z_j}},$$

где K – количество классов инцидентов; $z_k = W \cdot h_{CLS} + b$ – выход модели; X – вектор признаков события.

Финальное определение категории определяется как:

$$f = \begin{cases} arg\ \max P_i, max_i\ P_i\ \geq \emptyset \\ unknown, max_i\ P_i\ < \emptyset \end{cases}$$

где P – вероятность принадлежности событию классу i; $\emptyset$ – порог уверенности классификации. Если максимальная вероятность превышает порог $\emptyset$, событие относится к соответствующему классу, иначе классифицируется как неопределённое (unknown).

На седьмом этапе осуществляется оценка точности применяемой модели. Для оценки модели используются метрики:

- Precision – показывает, какая доля сработавших событий действительно является инцидентами. [12]:

$$Precision = \frac{TP}{TP + FP},$$

где TP (True Positive) – количество объектов, правильно отнесённых моделью к положительному классу. FP (False Positive) – количество объектов, ошибочно отнесённых к положительному классу.

- Recall – показывает, какую долю реальных инцидентов модель обнаружила.

$$Recall = \frac{TP}{TP + FN},$$

где TP (True Positive) – количество объектов, правильно отнесённых моделью к положительному классу; FP (False Positive) – количество объектов, ошибочно отнесённых к положительному классу; FN (False Negative) – количество объектов, ошибочно отнесённых к отрицательному классу, тогда как в действительности они принадлежат положительному классу.

- F1-Score – метрика качества классификации.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}.$$

Представленные метрики позволяют комплексно оценить эффективность модели, отражая её способность корректно классифицировать инциденты, минимизировать количество ложных срабатываний и обеспечивать полноту обнаружения событий информационной безопасности.

III. Практическая реализация программного модуля распознавания угроз нарушения безопасности

Для разработки программного модуля распознавания инцидентов информационной безопасности выбран язык программирования Python 3.11.9, а также библиотеки Pandas, matplotlib, seaborn и фреймворк глубокого обучения PyTorch. Для обработки текстовых данных и построения модели использована библиотека transformers, реализующая архитектуру BERT.

Для исследования использовались журналы событий операционной системы, содержащие информацию о процессах аутентификации, выполнении команд, сетевой активности и системных задачах. Данные представлены в виде текстовых записей логов.

В рамках работы были выделены следующие типы инцидентов информационной безопасности: перебор паролей (Brute Force); повышение привилегий (Privilege Escalation); выполнение вредоносного кода (Malware Execution); установка инструментов атаки (Tool Installation); нормальные события (Normal).

В исходных данных каждая запись представляет собой текстовую строку журнала. Для повышения качества распознавания был реализован подход анализа последовательностей событий.

Формирование обучающих примеров выполнялось с использованием скользящего окна фиксированной длины. Каждая последовательность включает 5 последовательных лог-записей, объединённых в один текст с использованием специального токена-разделителя [SEP]. Метка класса назначается по последнему элементу последовательности.

Таким образом, модель обучается на распознавание 5 классов, один из которых соответствует безопасному поведению, а остальные – различным этапам атаки.

После удаления дубликатов исходный набор данных составил 39 996 записей. После формирования последовательностей было получено 39 996 примеров. Данные были разделены на обучающую и тестовую выборки. Обучающая выборка составила 33 532 записи, тестовая – 6 464 записи (табл. 1).

ТАБЛИЦА I. КОЛИЧЕСТВЕННОЕ ПРЕДСТАВЛЕНИЕ ВЫБОРКИ

Наименование выборки	Количество элементов
Обучающая	33532
Тестовая	6464

Текстовые данные преобразовывались с использованием токенизатора модели BERT. Для каждой последовательности выполнялись:

- токенизация текста;
- обрезка до максимальной длины;
- дополнение до фиксированной длины.

Максимальная длина входной последовательности установлена равна 256. Для решения задачи использована предобученная языковая модель BERT, дополненная классификационным слоем. Модель реализована с использованием класса AutoModelForSequenceClassification библиотеки transformers, работающей на базе PyTorch. Параметры обучения модели представлены в табл. 2.

ТАБЛИЦА II. ПАРАМЕТРЫ ОБУЧЕНИЯ МОДЕЛИ

Параметр	Значение
Размер мини-батча	16
Размер окна последовательности	5
Алгоритм оптимизации	AdamW
Скорость обучения	2e-5
Максимальная длина последовательности	256
Количество эпох	5

Обучение модели выполнялось с использованием графического процессора (GPU), что позволило существенно ускорить процесс вычислений.

Проведенные эксперименты показали, что разработанная модель демонстрирует высокие значения метрик качества для большинства классов, при этом наилучшие результаты достигаются при распознавании атак, связанных с выполнением вредоносного кода (Malware Execution) (табл. 3).

ТАБЛИЦА III. МЕТРИКИ КАЧЕСТВА

Тип атаки	Precision	Recall	F1-score
Нормальные события	0.95	0.95	0.95
Brute Force (SSH)	0.96	0.93	0.95
Privilege Escalation	0.98	0.98	0.98
Malware Execution	0.92	0.96	0.94
Tool Installation	0.94	0.99	0.96

Наибольшее количество ошибок модели связано с классом Tool Installation, что объясняется схожестью таких событий с легитимной административной деятельностью, связанной с установкой программного обеспечения. В результате, часть этих событий ошибочно классифицируется как нормальное поведение.

Анализ матрицы ошибок демонстрирует, что наиболее частой ошибкой является неправильная классификация событий установки инструментов атаки как нормального трафика в 17% случаев. Также около 9% атак повышения привилегий (Privilege Escalation) интерпретируются как безопасные действия (рис. 1).

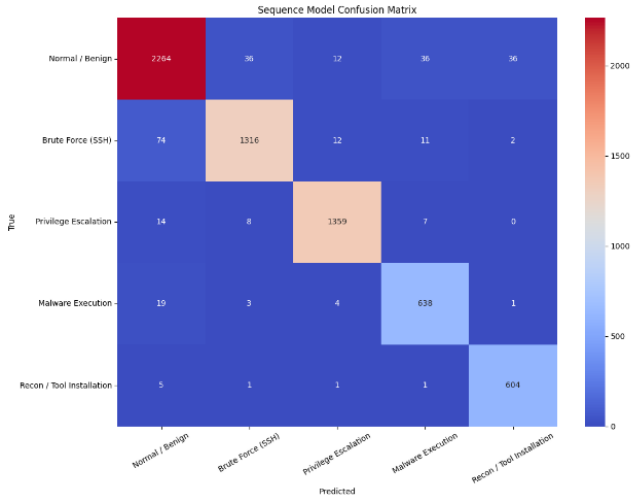

Рис. 1. Матрица ошибок.

Тем не менее, благодаря учёту контекста в анализе последовательностей событий, модель успешно распознает угрозы, даже если на отдельных этапах сценария происходят ошибки. Это позволяет обнаруживать атаки, несмотря на частичные неточности в классификации.

Кроме того, важно отметить, что менее 1% нормальных событий были ошибочно классифицированы как атаки, что свидетельствует о крайне низком уровне ложных срабатываний. Это делает предложенный метод эффективным и подходящим для использования в реальных системах мониторинга информационной безопасности.

IV. Заключение

В рамках проведенного исследования был рассмотрен вопрос повышения эффективности обработки событий информационной безопасности в информационных системах обработки данных. Проведенный анализ существующих подходов показал, что традиционные системы обнаружения инцидентов в значительной степени основаны на использовании сигнатурных и правил-ориентированных методов, что ограничивает их возможности при выявлении новых или сложных типов атак. Кроме того, существующие решения недостаточно эффективно учитывают последовательности событий и динамический характер развития инцидентов.

Для решения выявленных проблем был предложен метод интеллектуальной классификации инцидентов информационной безопасности, основанный на анализе событий журналов различных источников. Предложенный метод предусматривает сбор и предварительную обработку логов, нормализацию и унификацию признаков событий, а также последующую классификацию инцидентов с использованием модели нейронного обучения.

Таким образом, предложенный метод обеспечивает более гибкий подход к анализу событий информационной безопасности за счет учета взаимосвязи между событиями и автоматического определения вида инцидента с вычислением вероятности его принадлежности к определенному класс. По сравнению с традиционными сигнатурными системами предложенный подход быть использован для повышения эффективности мониторинга и реагирования на инциденты в современных информационных системах.

Список литературы

[1] Галимов А.Д., Стародубов М.И., Артемьева И.Л. Выявление инцидентов информационной безопасности на основе журналов событий операционной системы // Современное образование: интеграция образования, науки, бизнеса и власти. Трансформация образования, науки и производства - основа технологического прорыва: материалы международной научно-методической конференции. В 2 ч., Томск, 26–27 января 2023 года. Том Часть 1. Томск: Томский государственный университет систем управления и радиоэлектроники, 2023. С. 173-176.

[2] Аккуратнов А.Н., Зефиров С.Л. Методы выявления инцидентов информационной безопасности в условиях задержек событий // Труды международного симпозиума "Надежность и качество". 2023. Т. 1. С. 161-164.

[3] Азарычева М.А., Корсунский А.С. Построение и реализация модуля выявления инцидентов на основе сигнатурного метода анализа событий // Автоматизация процессов управления. 2022. № 4(70). С. 41-50.

[4] Комаров Н.В., Стойчина Е.В., Михайленко М.В., Стойчин К.Л. Особенности криминалистического анализа оперативной памяти электронно-вычислительной машины // Безопасность информационного пространства: сборник научных трудов XXI Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых, Екатеринбург, 24–25 ноября 2022 года. Том Выпуск 4 (252). Екатеринбург: Уральский государственный университет путей сообщения, 2023. С. 10-12.

[5] Еремеев М.А., Смирнов С.И., Прибылов И.А. Выявление вредоносных действий злоумышленника на основе журналов событий при расследовании текущего кибер-инцидента. // Инновационные аспекты развития науки техники: Сборник статей VII Международной научно-практической конференции, Саратов, 23 апреля 2021 года. Саратов: Индивидуальный предприниматель Емельянов Н. В., 2021. Р. 22-28.

[6] Фаткиева Р.Р. Левоневский Д.К. Применение бинарных деревьев для агрегации событий систем обнаружения вторжений // Труды СПИИРАН. 2015. № 3(40). С. 110-121.

[7] Фаткиева Р.Р. Корреляционный анализ аномального сетевого трафика // Труды СПИИРАН. 2012. № 4(23). С. 93-99.

[8] Фаткиева Р.Р. Модель обнаружения атак на основе анализа временных рядов // Труды СПИИРАН. 2012. № 2(21). С. 71-79.

[9] Red Hat Enterprise Linux System Administrator’s Guide. Viewing and Managing Log Files [электронный ресурс]. Режим доступа. URL: https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/system_administrators_guide/ch-viewing_and_managing_log_files (дата обращения 07.03.2026).

[10] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [электронный ресурс]. Режим доступа. URL: https://arxiv.org/pdf/1810.04805 (дата обращения 07.03.2026).

[11] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A., Kaiser Ł., Polosukhin I. Attention Is All You Need // Advances in Neural Information Processing Systems 2017. [электронный ресурс]. Режим доступа. URL: https://arxiv.org/pdf/1706.03762 (дата обращения 07.03.2026).

[12] Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness and Correlation // Journal of Machine Learning Technologies. 2011. [электронный ресурс]. Режим доступа. URL: https://arxiv.org/pdf/2010.16061 (дата обращения 07.03.2026).