Разработка механизма динамического доверия для защиты веб-сессий в архитектуре ZeroTrust

Е.О. Лычева, О.Р. Лапонина

Аннотация **- В работе предложен механизм динамического доверия для защиты веб-сессий, объединяющий криптографический и поведенческий контуры. Криптографический контур отвечает за корректность, свежесть и непереносимость токена, а поведенческий контур оценивает соответствие текущей траектории действий типичному пользовательскому сценарию. Формальная модель механизма включает требования непереносимости токена, устойчивости к повторному использованию и непрерывности доверия.**

Практическая часть реализована в виде прототипа из сервиса аутентификации, сервиса оценки риска и ресурсного сервиса. Для поведенческого контура используется LSTM-модель, обучаемая на нормальных последовательностях действий, реконструированных из датасета CSIC 2010.

Экспериментальная проверка выполняется на двух сценариях злоупотребления установленной сессией: автоматизированной атаке ботами и обходе бизнес-логики. В качестве основной метрики использована ROC-AUC, отражающая способность модели ранжировать аномальные примеры выше нормальных без жёсткой фиксации порога.

Результаты эксперимента показывают, что для предложенного механизма определяющим фактором является объём контекста, доступного модели в момент оценки. В обоих сценариях качество возрастает при увеличении длины истории, однако глубина контекста, необходимая для устойчивого отделения аномалии от нормы, различается: для автоматизированного поведения достаточно более короткой последовательности, тогда как для обхода бизнес-логики требуется более длинная история. При этом влияние размера скрытого состояния оказывается заметно слабее.

Таким образом, работа механизма допуска должна опираться на накопленную историю сессии как на обязательную часть состояния, поскольку оценка изолированного запроса не даёт сопоставимой надёжности. Следовательно, при внедрении предложенной схемы критическим рабочим параметром становится длина анализируемой последовательности, а доверие к сессии должно пересчитываться по мере накопления новых действий.

Ключевые слова **- ZeroTrust, LSTM, аномалии веб-сессий, веб-сессия, динамическое доверие, поведенческий анализ, управление доступом.**

Статья получена 21 апреля 2026.

Е.О. Лычева – МГУ имени М.В. Ломоносова (email: ders.1981@mail.ru).

О.Р. Лапонина – МГУ имени М.В. Ломоносова (email: laponina@oit.cmc.msu.ru).

I. Введение

После успешной аутентификации именно веб-сессия начинает определять, какие действия пользователь может выполнять в системе и в каком состоянии находится взаимодействие с сервисом. Поэтому для задач прикладной безопасности важно не только подтвердить личность субъекта в момент входа, но и контролировать дальнейшее развитие уже установленной сессии. В архитектуре ZeroTrust это особенно применимо, поскольку решение о доступе должно опираться на непрерывную проверку, а не на единичный акт доверия [1].

Современный веб-сервис работает вне классического сетевого периметра: использует облачную инфраструктуру, внешних провайдеров идентификации, мобильных клиентов и распределённые хранилища состояния. В этой постановке злоумышленник может получить преимущество не за счёт взлома криптографии, а за счёт использования уже установленного сессионного контекста: активного браузерного сеанса, похищенного cookie, повторно предъявленного токена или уязвимости в логике обработки состояния [2], [3], [9], [10].

Даже современные схемы с короткоживущими токенами, доказательством владения и ротацией артефактов отвечают прежде всего на вопрос о корректности самого токена. Однако они не позволяют определить, насколько допустимо текущее развитие активной сессии с точки зрения поведения пользователя внутри конкретного сервиса. В результате возникает разрыв между формальной валидностью токена и реальной легитимностью доступа [4], [5], [6].

В статье рассматривается механизм динамического доверия к веб-сессии, в котором криптографическая проверка сочетается с поведенческой аттестацией. Сначала анализируются угрозы безопасности веб-сессий и существующие подходы к выявлению аномалий, затем описываются формальная модель и архитектура прототипа, после чего приводятся результаты экспериментальной оценки поведенческого контура на базе последовательностной LSTM-модели и датасета CSIC 2010 [7], [18], [19].

Целью работы является разработка механизма динамической оценки доверия к веб-сессии,

позволяющего замечать аномальное развитие пользовательского взаимодействия относительно логики конкретного веб-сервиса и использовать эту оценку при принятии решения о допуске. Для достижения этой цели в работе анализируются угрозы управления веб-сессией, исследуются подходы к обнаружению аномалий в пользовательском поведении, формулируются требования к механизму динамического доверия, разрабатывается архитектура прототипа, подготавливаются данные для последовательностной модели и проводится вычислительный эксперимент на размеченных сценариях отклоняющегося поведения.

В работе используются методы анализа угроз веб-приложений, формального описания свойств безопасности, методы машинного обучения для анализа последовательностей, а также экспериментальные методы оценки качества поведенческой модели. Преимущество предложенного механизма состоит в том, что динамическое доверие к веб-сессии рассматривается как связующее звено между криптографической корректностью токена и наблюдаемым поведением внутри сервиса. Разработанный прототип сервиса оценки риска может использоваться как дополнительный компонент в архитектуре ZeroTrust.

II. Анализ существующих решений

A. Угрозы безопасности веб-сессий в пост-аутентификационном контуре

Веб-сессия после входа в систему выступает основным носителем доверия, а её идентификатор обычно реализуется в виде токена, передаваемого через cookie, заголовки HTTP или локальное хранилище. В классических схемах управления доступом наличие корректного токена считается достаточным основанием для обращения к ресурсу, однако такая логика плохо учитывает случаи, когда действующая сессия уже используется не тем субъектом или не тем способом, который предполагался при проектировании [2], [3].

Угрозы веб-сессии делятся на несколько классов. Первый связан с доступом к доверенной клиентской среде: злоумышленник получает активный браузерный контекст или устройство и действует в рамках уже установленной сессии. Второй связан с утечкой или перехватом сессионного артефакта, включая ошибки обработки cookie и повторное использование токена. Третий класс составляют сценарии, в которых сессия внешне остаётся корректной, но используется аномально: автоматизированным клиентом, который повторяет короткие шаблоны действий, либо субъектом, нарушающим ожидаемый порядок бизнес-процесса [9], [10], [11], [12], [13], [14].

Архитектура ZeroTrust требует рассматривать доступ не как разовое решение после аутентификации, а как непрерывно управляемый процесс. Отсюда следует, что активная сессия должна оставаться наблюдаемой на протяжении всего жизненного цикла. Формальная корректность токена при этом остаётся необходимым условием, но перестаёт быть достаточной гарантией безопасного поведения внутри сервиса [1], [2].

B. ZeroTrust и доступ на уровне сессии

Парадигма ZeroTrust переносит фокус защиты с сетевого периметра на конкретный субъект и конкретный ресурс. В этой модели не предполагается наличие неявного доверия к устройствам, учётным записям или внутренним сегментам сети. Для веб-приложения это означает, что решение о допуске должно зависеть не от самого факта пребывания пользователя внутри доверенной зоны, а от совокупности подтверждённых свойств текущего взаимодействия [1].

Применительно к веб-сессии из этой идеи следуют несколько практических положений. Во-первых, доступ должен предоставляться не сети в целом, а конкретному субъекту по отношению к конкретному ресурсу. Во-вторых, решение о допуске может зависеть от текущего состояния клиента, сервиса и наблюдаемого поведения. В-третьих, сессия должна рассматриваться как управляемый жизненный цикл, а не как статическая сущность, однажды признанная допустимой. Именно поэтому активная веб-сессия должна оставаться контролируемой на протяжении всего времени работы.

C. Сценарии компрометации активной сессии

Наиболее прямой сценарий компрометации связан не с криптографией, а с доступом к уже доверенной клиентской среде. Пользователь может оставить разблокированное устройство, работать из скомпрометированного браузера или сохранить действующий контекст в локальном хранилище. В таком случае злоумышленник получает готовую активную сессию со всеми правами уже аутентифицированного субъекта. Для такого класса угроз поведенческий контроль особенно важен, поскольку формально токен остаётся корректным, а аномалия проявляется только в способе использования сессии [2], [9].

Отдельный класс угроз связан с перехватом или повторным использованием сессионного артефакта. Источником могут быть XSS-уязвимости, небезопасные журналы, ошибки обработки cookie, дефекты настройки поддоменов и смежные проблемы реализации. Даже при наличии короткоживущих токенов и ротации артефактов остаётся окно, в котором атакующий пытается перенести полученный токен в другой контекст и действовать от имени жертвы [2], [3], [9], [10].

Третий сценарий относится к автоматизированному использованию установленной сессии. Даже если злоумышленник не нарушает форму запросов, сама манера работы с сервисом может стать нетипичной: короткие циклы действий, высокая повторяемость переходов, неестественная регулярность и малый разброс интервалов. Для веб-приложения это означает нарушение геометрии пользовательской траектории [11], [12].

Наиболее тонкий сценарий связан с обходом бизнес-логики внутри активной сессии. В этом случае злоумышленник использует легитимные конечные точки приложения, но пытается перейти к чувствительному действию раньше, чем это допускает

накопленная история. Отдельный запрос при этом может выглядеть правдоподобно, а отклонение проявляется именно на уровне последовательности. Следовательно, обнаружение такого класса атак требует учёта предшествующих шагов и делает последовательностную модель естественным инструментом поведенческого анализа [13], [14].

D. Подходы к выявлению аномалий в пользовательских веб-сессиях

В литературе по обнаружению аномалий применяются несколько классов методов. Один из них опирается на модели восстановления и компактного представления данных, например, автокодировщики. Такие решения удобны тем, что обучаются на модели нормы и не требуют перечисления всех классов атак, однако они лучше описывают необычность набора признаков, чем логику развития пользовательского маршрута [8], [15].

Статистические и одно-классовые методы позволяют строить границу нормы на основе частот событий, временных интервалов и агрегированных характеристик трафика. Они полезны для фонового мониторинга и хорошо выявляют аномальную интенсивность или нетипичное сочетание признаков. Тем не менее в задаче контроля веб-сессии этого часто недостаточно, поскольку атакующий может использовать легитимные конечные точки, а отклонение проявляется именно на уровне последовательности действий [15], [16], [17].

Отдельную группу составляют правила и политико-ориентированные ограничения. Они незаменимы там, где ограничение известно заранее: например, нельзя повторно выполнять одноразовый шаг или переходить к чувствительной операции без обязательного промежуточного этапа. Однако такие правила локальны и плохо обобщают новые сценарии отклоняющегося поведения, особенно если отдельные запросы формально выглядят допустимо [1 3], [14].

Для веб-сессий особенно важны методы, которые рассматривают поведение как траекторию. В этом случае подозрительность действия определяется не только его содержанием, но и положением внутри уже наблюдаемого маршрута. Именно поэтому для рассматриваемой постановки ключевое значение получают последовательностные модели, позволяющие оценивать вероятность следующего шага по истории сессии [7], [15], [16], [17].

Важным ограничением существующих публикаций является то, что многие сравнительные результаты получены на наборах данных для сетевых IDS, таких как KDD, NSL-KDD или UNSW-NB15, где объектом анализа выступает сетевое соединение или агрегированный поток, а не траектория уже аутентифицированного пользователя внутри одного веб-приложения. Для задачи динамического доверия этого недостаточно, поскольку основная аномалия возникает не на уровне межсетевого трафика вообще, а в логике развития сессии в границах одного сервиса [15], [16].

Смежные направления, например, UEBA (User and Entity Behavior Analytics), постепенно смещают акцент именно к повторяемости действий, временной структуре маршрутов и различию между нормальным и отклоняющимся поведением субъекта. Для рассматриваемой постановки это особенно существенно: после успешной аутентификации злоумышленник может использовать те же конечные точки, что и легитимный пользователь. Следовательно, задача обнаружения становится задачей распознавания нетипичного развития уже установленной сессии, а не только задачей обнаружения вредоносного запроса как такового.

E. Последовательностные модели для анализа поведения

Одним из примеров такого подхода является DeepLog, в котором аномалия определяется как отклонение от ожидаемой последовательности событий. Несмотря на то, что система разрабатывалась для журналов, сама идея естественно переносится на веб-сессию: вместо сигнатур атак строится модель нормального маршрута, а нарушение выявляется как неожиданное продолжение последовательности [7].

Среди прикладных последовательностных архитектур особый интерес представляют LSTM-модели. Их ключевое свойство состоит в способности удерживать историю нескольких предыдущих шагов и использовать её при прогнозировании следующего действия. Для задачи контроля веб-сессии это принципиально, поскольку аномалия может проявляться не в одном редком запросе, а в неправильном положении вполне допустимого действия относительно уже пройденных стадий.

Более простые модели, основанные на n-граммах, марковских переходах или локальных правилах, удобны для коротких и устойчивых сценариев, однако с ростом длины маршрута либо чрезмерно разрастаются, либо теряют дальний контекст. Трансформерные архитектуры потенциально сильнее на длинных последовательностях, но для умеренного словаря действий и сравнительно компактного набора данных их вычислительная стоимость выглядит избыточной. Поэтому в практической части работы LSTM используется как последовательностная модель для поведенческого контура [15], [18].

Дополнительным преимуществом LSTM является управляемая память. За счёт механизмов записи, сохранения и забывания сеть может накапливать историю нескольких предыдущих стадий сценария и использовать её как компактное представление контекста. Именно это свойство делает LSTM предпочтительной там, где подозрительность очередного шага определяется не по форме текущего запроса, а по тому, насколько естественно он появляется внутри уже развивающейся траектории сессии [18].

III. Механизм динамического доверия для защиты веб-сессий

A. Формальная модель и свойства безопасности

В рассматриваемой постановке веб-сессия рассматривается как сочетание действительного сессионного идентификатора, серверного состояния и

текущей последовательности действий пользователя. Такое представление позволяет связать два уровня допустимости: протокольный, отвечающий за корректность токена, и поведенческий, отвечающий за то, развивается ли активная сессия в пределах нормального сценария сервиса.

Первым базовым требованием механизма выступает непереносимость токена. Наличие похищенного токена не должно само по себе давать злоумышленнику право действовать от имени пользователя без владения секретом или ключом, к которому привязан токен. В практической реализации это свойство обеспечивается средствами подтверждения владения [5], [6].

$$P(Accept(I, proof) = I \text{ and not } Owns(A, k) \leq \varepsilon(\lambda) \quad (1)$$

Здесь Accept(I, proof) = 1 означает, что сервер принял предъявленное доказательство; I - идентификатор сессии или субъекта доступа; proof - доказательство владения токеном; A - атакующий; k - секрет или ключ, с которым токен связан; Owns(A, k) - факт владения этим секретом; $\epsilon(\lambda)$ - пренебрежимо малая функция параметра безопасности λ.

Вторым требованием является устойчивость к повторному использованию сообщений и токенов. Ранее корректный артефакт не должен приниматься повторно за пределами допустимого окна времени. На практике это достигается комбинацией короткоживущих токенов доступа, ротации токенов обновления, контроля идентификаторов и проверки свежести [4], [5], [6].

$$P[Accept(I, M, t') = 1 \; for \; t' >= t + \Delta t_{max}] \leq \varepsilon(\lambda) \quad (2)$$

Здесь Accept(I, M, t') = 1 величина M обозначает сообщение или токен, предъявляемый в момент t'; t - момент его формирования; Δ_{t_max} - допустимое окно времени, в пределах которого артефакт ещё считается свежим; ϵ(λ) - пренебрежимо малая функция параметра безопасности.

Третьим требованием выступает непрерывность доверия. В отличие от бинарной схемы авторизации, здесь вводится функция доверия ρ(t), которая пересчитывается по мере поступления новых действий пользователя и новых признаков контекста. Решение о доступе принимается только при одновременном выполнении криптографической проверки и достаточном значении текущего доверия.

$$Access(t) = 1\{v(t) = 1 \; and \; \rho(t) \geq \rho_{min}\} \quad (3)$$

Здесь величина v(t) обозначает результат криптографической проверки в момент времени t, ρ(t) - текущий уровень доверия к сессии, а ρ_{min} - минимально допустимый порог доверия.

В этой цепочке удобно различать три уровня величин: локальную оценку неожиданности очередного действия, агрегированную оценку риска и собственно доверие ρ(t), участвующее в политике допуска. Модель машинного обучения отвечает только за оценку отклонения; окончательное решение о разрешении, усиленной аутентификации или отказе остаётся на уровне политики доступа. Такое разделение делает механизм совместимым с архитектурой ZeroTrust и позволяет использовать поведенческий сигнал без отказа от протокольной строгости [1], [2].

B. Сопоставление с существующими механизмами управления сессией

Для проверки содержательности предлагаемого механизма полезно сопоставить его с распространёнными практиками управления сессией. Обычная cookie-модель обеспечивает простоту и совместимость, однако её безопасность в значительной степени зависит от корректной настройки атрибутов Secure, HttpOnly, SameSite и доменной конфигурации. Даже при грамотной реализации cookie остаётся статическим маркером сессии: если он украден, дальнейшая защита держится главным образом на ограничении срока жизни и внешнем мониторинге [2], [3].

Схемы OAuth с короткоживущими токенами, ротацией токенов обновления и подтверждением владения существенно усиливают криптографический контур. Они повышают непереносимость токена и уменьшают риск прямого повторного использования. Однако такие механизмы, как правило, не дают непрерывной поведенческой переоценки доверия. В результате остаётся окно, в котором формально корректный, но поведенчески подозрительный контекст продолжает считаться допустимым [4], [5], [6].

Предлагаемый механизм не пытается заменить протокольные средства защиты. Напротив, он дополняет их и разделяет роли двух контуров: криптографический отвечает за доказуемую корректность и свежесть артефакта, а поведенческий – за то, сохраняется ли доверие к сессии по мере её развития. Именно такое разделение позволяет одновременно сохранить формальную строгость и уменьшить слепоту по отношению к аномалиям, не нарушающим проверку токена.

C. Связь доверия, риска и допуска

Рассмотрим роли каждого из контуров. Криптографическая проверка отвечает на вопрос, допустимо ли вообще продолжать рассматривать сессию как кандидата на доступ. Поведенческий контур отвечает на другой вопрос: насколько текущее развитие этой сессии согласуется с нормальным сценарием. Следовательно, доверие – это отдельная управляющая величина, которая обновляется под влиянием риска, но используется на уровне решения о допуске.

В этой цепочке оценка неожиданности отдельного действия выступает низкоуровневым сигналом. Далее из неё строится агрегированная характеристика поведения в окне сессии, а уже затем обновляется доверие ρ. Такая декомпозиция важна по двум причинам. Во-первых, она позволяет разделить роль модели машинного обучения и роль политики доступа: модель оценивает отклонение, но не принимает окончательное решение. Во-вторых, она не требует, чтобы каждое подозрительное действие автоматически приводило к отказу. Для реального веб-

сервиса это существенно, поскольку отдельные отклонения могут быть вызваны и нетипичным, но добросовестным поведением пользователя.

Практическое следствие этой схемы состоит в возможности многоступенчатой политики. При высоком доверии операция разрешается сразу, при промежуточном уровне инициируется усиленная аутентификация, а при низком доверии запрос отклоняется. Такой переход от статического допуска к пересчитываемой оценке особенно важен для сценариев, в которых злоумышленник не ломает криптографию, а использует уже существующую сессию.

D. Архитектура прототипа

Практическая часть оформлена как прототип из трёх взаимодействующих сервисов: сервиса аутентификации, сервиса оценки риска и ресурсного сервиса. Такая структура позволяет разделить проверку токена, поведенческую оценку и применение политики доступа.

Работа прототипа организована следующим образом. Пользователь инициирует действие и предъявляет токен доступа ресурсному сервису. Далее ресурсный сервис обращается к сервису аутентификации для проверки токена, его свежести и подтверждения владения, а в сервис оценки риска передаёт идентификатор сессии, текущее нормализованное действие и краткую историю предыдущих шагов. Сервис оценки риска вычисляет оценку неожиданности очередного действия, преобразует её в обновлённое значение доверия ρ и возвращает результат в ресурсный сервис.

После этого ресурсный сервис сопоставляет результат криптографической проверки и текущий уровень доверия и применяет политику допуска. При высоком доверии запрос разрешается, при промежуточном уровне может инициироваться усиленная аутентификация, а при низком доверии обращение отклоняется. При такой архитектуре каждый контур выполняет свою роль: криптографический контур отвечает за корректность и свежесть сессионного артефакта, а поведенческий - за соответствие текущей траектории действий нормальному сценарию сервиса.

Важно, что между сервисами передаётся не полный HTTP-запрос, а сокращённое представление состояния. Сервис аутентификации работает с токеном и признаками его жизненного цикла, сервис оценки риска - с историей нормализованных действий, а ресурсный сервис получает только итоговые результаты. Это снижает связанность компонентов и не смешивает логику выпуска токенов с последовательностным анализом поведения.

Такая архитектура делает поведенческий анализ частью контура допуска. За счёт этого доверие к активной сессии перестаёт быть статической характеристикой и начинает пересчитываться по мере развития взаимодействия пользователя с сервисом.

E. Данные, которыми обмениваются сервисы

Сервис аутентификации работает с токеном, идентификатором клиента и признаками жизненного цикла сессионного артефакта. Сервис оценки риска использует идентификатор сессии, текущее нормализованное действие и краткую историю предыдущих токенов действий. Ресурсный сервис, в свою очередь, получает только итоговые результаты: формально валиден ли токен, и достаточно ли высоко текущее доверие. Такой обмен данными ограничивает связанность компонентов и позволяет развивать их независимо друг от друга.

F. Подготовка данных и поведенческая модель

Для вычислительного эксперимента выбран датасет CSIC 2010, описывающий обращения к одному учебному веб-приложению. Его существенное преимущество заключается в том, что данные относятся к поведению внутри конкретного сервиса, а не к произвольному сетевому трафику. Это делает возможной реконструкцию осмысленных пользовательских маршрутов, связанных с просмотром, аутентификацией, корзиной и оформлением действий [19].

Исходный поток содержит HTTP-запросы, а не готовые сессии, поэтому в работе выполняется реконструкция последовательностей действий. Для каждого запроса строится токен вида METHOD_ACTION, абстрагирующий метод и нормализованный маршрут. Далее весь поток делится на сессии по точкам возврата на домашнюю страницу.

Из реконструированных сессий формируются окна фиксированной длины k: в качестве входа берутся k предыдущих действий, а следующая операция используется как целевая метка. Обучающая и тестовая выборки формируются по схеме 70/30, а из обучающего набора дополнительно выделяется небольшая валидационная часть.

Для обучения модели используются пакеты по 128 примеров, также применяется оптимизатор Adam с шагом обучения 10^{-3}. Для ограничения переобучения применяется ранняя остановка с терпением пять эпох. В экспериментальной части исследуется влияние двух параметров: длины истории и размера скрытого состояния.

Поведенческий сигнал строится как оценка неожиданности фактически наблюдаемого действия относительно истории сессии. Чем ниже вероятность текущего шага по оценке модели, тем выше подозрительность продолжения сессии.

$$s(a_t) = -\log P(a_t \mid a_{t-1}, \ldots, a_{t-k}) \qquad (4)$$

Здесь a_t обозначает фактически наблюдаемое действие на шаге t, a_{t-1}, ..., a_{t-k} - k предыдущих действий, $P(a_t \mid a_{t-1}, \ldots, a_{t-k})$ - условную вероятность следующего действия по оценке модели, а $s(a_t)$ - оценку неожиданности текущего шага.

Такой способ оценки удобен тем, что модель обучается только на нормальном поведении, а аномалия задаётся как отклонение от него, а не как заранее перечисленный класс атаки [7], [18].

IV. Экспериментальная оценка поведенческого контура

A. Методика эксперимента

Обучение поведенческой модели выполнялось только на нормальных последовательностях, реконструированных из обучающей части CSIC 2010. Проверка проводилась на отдельных нормальных последовательностях и на модифицированных сценариях атак, сформированных на основе тех же маршрутов. Такая схема важна потому, что модель учится описывать допустимое развитие сессии, а не запоминать ограниченный набор заранее размеченных атак.

Для итоговой оценки выбраны два сценария злоупотребления установленной сессией. Первый соответствует автоматизированной атаке ботами: скрипт повторяет короткую подпоследовательность действий быстрее и регулярнее, чем это делает человек. Второй сценарий моделирует обход бизнес-логики, когда злоумышленник использует легитимные конечные точки, но пытается перейти к более чувствительному действию раньше, чем это допускает наблюдаемая история.

Эксперименты проводились на сетке параметров: размер скрытого состояния принимал значения 16, 32, 64 и 128, а длина истории изменялась от 3 до 20.

В качестве основной метрики использована ROC-AUC, отражающая способность модели ранжировать аномальные примеры выше нормальных без жёсткой фиксации порога. Для рассматриваемой постановки это особенно важно, поскольку в реальном контуре допуска пороги могут различаться в зависимости от чувствительности операции.

Отдельным элементом протокола является ранняя остановка по валидационной функции потерь. Она используется для ограничения переобучения на обучающей части нормальных последовательностей и для выбора состояния модели, которое затем оценивается на отложенной выборке. В такой постановке валидационная часть и ранняя остановка уменьшают влияние избыточного дообучения на сравнение конфигураций по длине истории и размеру скрытого состояния.

B. Результаты и их интерпретация

Результаты показали различную сложность двух исследуемых сценариев. Для автоматизированной атаки ботами уже при короткой истории качество находится примерно на уровне 0.70, а при увеличении длины истории до 10 достигает значений 0.93-0.94 (рис. 1). Дальнейшее увеличение истории не приводит к столь же заметному выигрышу, что согласуется с природой автоматизированного поведения: повторяемость и регулярность достаточно быстро начинают отличаться от нормальной человеческой траектории.

Для сценария обхода бизнес-логики отклонение от нормы выражено менее явно. При короткой истории значения ROC-AUC находятся около 0.55, а в лучших конфигурациях достигают значений 0.77-0.79 (рис. 2). Это означает, что модели требуется более длинный контекст, чтобы распознать нарушение последовательности, при котором отдельные шаги могут оставаться правдоподобными сами по себе.

Влияние размера скрытого состояния оказалось заметно слабее влияния длины истории. Переход от 16 к 32 и 64 скрытым признакам дал умеренный выигрыш, особенно для обхода бизнес-логики, однако дальнейшее увеличение до 128 не меняло картину радикально. Следовательно, в данной постановке полезнее вкладываться в корректное восстановление истории и формирование последовательностей, чем в избыточное наращивание внутренней ёмкости модели.

Таким образом, разные классы отклонений требуют различной политики реакции. Для автоматизированной атаки ботами поведенческий контур может использоваться как ранний триггер для усиленной проверки или ограничения доступа. Для обхода бизнес-логики более естественной выглядит ступенчатая схема: сначала понижение доверия, затем усиленная аутентификация и только после накопления сигнала – отказ в операции.

Тем самым экспериментальные результаты поддерживают саму идею непрерывного доверия к сессии. Поведенческий анализ не заменяет классические механизмы защиты токена, но позволяет пересчитывать доверие по мере развития сессии и использовать динамическую оценку в контуре принятия решения о доступе.

C. Анализ различий между двумя сценариями

Автоматизированная атака ботами меняет ритм и геометрию последовательности. Повторы становятся короче, переходы – однообразнее, а сам маршрут теряет естественные колебания, присущие человеку. Поэтому даже умеренный контекст позволяет модели быстро заметить отклонение. Для такого сценария оказывается достаточно сравнительно простого поведенческого контроля, поскольку аномалия грубее и заметнее уже на ранних стадиях развития сессии.

Обход бизнес-логики устроен иначе. Здесь отдельные шаги могут оставаться правдоподобными сами по себе, а нарушение проявляется в их относительном положении. Из-за этого модель должна помнить больше предыдущих действий и выявлять не редкий токен, а неестественный переход между допустимыми на вид операциями. Именно поэтому рост качества здесь медленнее и сильнее зависит от увеличения длины истории. Этот результат подтверждает, что для тонких нарушений бизнес-процесса решающее значение имеет накопленный контекст маршрута.

D. Практическая интерпретация результатов для механизма допуска

Результаты эксперимента показывают, что для предложенного механизма определяющим фактором является объём контекста, доступного модели в момент оценки. В обоих сценариях качество возрастает при увеличении длины истории, однако глубина контекста, необходимая для устойчивого отделения аномалии от нормы, различается: для автоматизированного поведения достаточно более короткой

последовательности, тогда как для обхода бизнес-логики требуется более длинная история. При этом влияние размера скрытого состояния оказывается заметно слабее.

Таким образом, работа механизма допуска должна опираться на накопленную историю сессии как на обязательную часть состояния, поскольку оценка изолированного запроса не даёт сопоставимой надёжности. Следовательно, при внедрении предложенной схемы критическим рабочим параметром становится длина анализируемой последовательности, а доверие к сессии должно пересчитываться по мере накопления новых действий. Тем самым экспериментальные результаты обосновывают использование механизма динамического доверия как сессионного контура допуска, в котором решение определяется не только валидностью токена, но и накопленным контекстом взаимодействия.

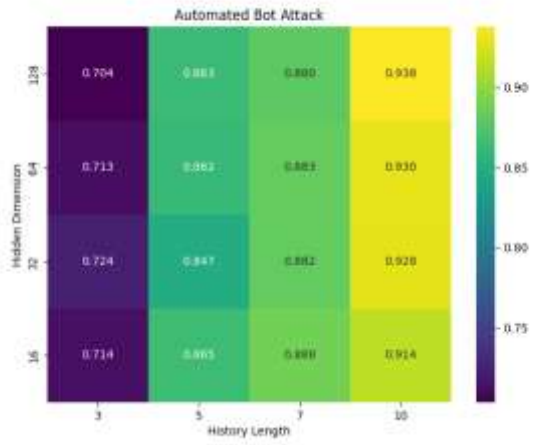

Рисунок 1. Зависимость ROC-AUC от длины истории и размера скрытого состояния для автоматизированной атаки ботами

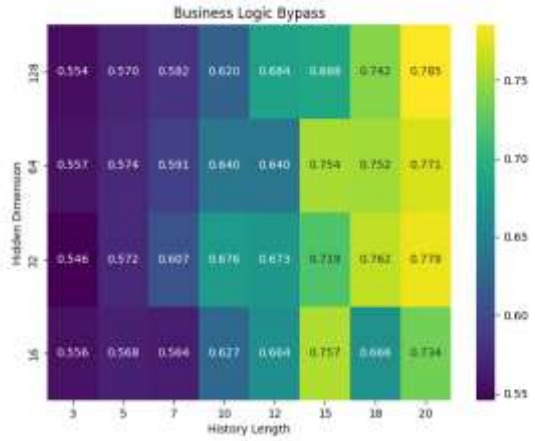

Рисунок 2. Зависимость ROC-AUC от длины истории и размера скрытого состояния для обхода бизнес-логики

V. Заключение

В работе рассмотрена задача защиты веб-сессии после аутентификации, когда одного факта валидности токена уже недостаточно для обеспечения безопасного доступа. Показано, что для практической веб-безопасности необходимо различать два уровня защиты: криптографический, отвечающий за корректность и свежесть токена, и поведенческий, отвечающий за то, как развивается активная сессия внутри сервиса.

Предложенный механизм динамического доверия объединяет эти два уровня в единую систему. Формальная модель включает требования непереносимости токена, устойчивости к повторному использованию и непрерывности доверия. Архитектурно механизм представлен в виде трёх взаимодействующих сервисов: сервиса аутентификации, сервиса оценки риска и ресурсного сервиса, в котором применяется итоговая политика допуска.

Практическая часть подтвердила, что последовательностный поведенческий анализ может использоваться как дополнительный контур контроля доступа.

Полученные результаты позволяют рассматривать динамическое доверие к веб-сессии как рабочий механизм для архитектуры ZeroTrust. Он не заменяет существующие протокольные средства защиты, но дополняет их и позволяет переходить от статической бинарной схемы допуска к непрерывной переоценке поведения внутри активной сессии.

В качестве дальнейшего развития подхода можно выделить несколько направлений. Одно из них связано с расширением признакового описания сессии: в модель могут быть включены временные интервалы между действиями, признаки клиентской среды, параметры устройства и компактные характеристики самого запроса. Другое направление относится к экспериментальной базе и предполагает добавление новых сценариев злоупотребления сессией, а также использование более реалистичных журналов веб-приложений.

Благодарности

Вопросы использования Искусственного интеллекта в кибербезопасности являются одним из основных научных направлений кафедры ИБ факультета ВМК МГУ имени М.В. Ломоносова и рассматривались во множестве магистерских диссертаций и научных работ [20, 21, 22]. Также, традиционно, отмечаем работы В.П. Куприяновского и его соавторов, положивших начало цифровой тематике в журнале INJOIT [23, 24].

Библиография

[1] NIST. Zero Trust Architecture. NIST Special Publication 800-207. Gaithersburg: National Institute of Standards and Technology, 2020

[2] OWASP Foundation. Session Management Cheat Sheet [Электронный ресурс]. URL: https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (дата обращения: 30.03.2026)

[3] Barth A. HTTP State Management Mechanism. RFC 6265. IETF, 2011

[4] Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). RFC 7519. IETF, 2015

[5] Fett D., Campbell B., Bradley J., Lodderstedt T., Jones M., Waite D. OAuth 2.0 Demonstrating Proof of Possession (DPoP). RFC 9449. Internet Engineering Task Force (IETF), 2023

[6] Lodderstedt T., Campbell B., Bradley J., Sakimura N., Parecki A., Waite D. OAuth 2.0 Security Best Current Practice. RFC 9700. IETF, 2025

[7] Du M., Li F., Zheng G., Srikumar V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning // Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017. URL: https://doi.org/10.1145/3133956.3134015
[8] Ferrag M., Maglaras L., Moschoyiannis S., Janicke H. Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study // Journal of Information Security and Applications. 2020. Vol. 50
[9] Calzavara S., Jonker H., Krumnow B., Rabitti A. Measuring Web Session Security at Scale // Computers & Security. 2021. Vol. 111. Article 102472
[10] Sivakorn S., Polakis I., Keromytis A. The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information // IEEE Symposium on Security and Privacy. 2018
[11] Rahman R. U., Tomar D. S. New biostatistics features for detecting web bot activity on web applications // Computers & Security. 2020. Vol. 97. Article 102001. DOI: 10.1016/j.cose.2020.102001
[12] Iliou C., Kostoulas T., Tsikrika T., Katos V., Vrochidis S., Kompatsiaris I. Detection of Advanced Web Bots by Combining Web Logs with Mouse Behavioural Biometrics // Digital Threats: Research and Practice. 2021. Vol. 2. No. 3. DOI: 10.1145/3447815
[13] Deepa G., Thilagam P. S., Praseed A., Pais A. R. DetLogic: A black-box approach for detecting logic vulnerabilities in web applications // Journal of Network and Computer Applications. 2018. Vol. 109. P. 89-109. DOI: 10.1016/j.jnca.2018.01.008
[14] Metin B., Wynn M., Tunali A., Kepir Y. Business Logic Vulnerabilities in the Digital Era: A Detection Framework Using Artificial Intelligence // Information. 2025. Vol. 16. No. 7. Article 585. DOI: 10.3390/info16070585
[15] Kwon D., Kim H., Kim J., Suh S. C., Kim I., Kim K. J. A Survey of Deep Learning-Based Network Anomaly Detection // Cluster Computing. 2019. Vol. 22. Suppl. 1. P. S949-S961. DOI: 10.1007/s10586-017-1117-8
[16] Alaoui R. L., Nfaoui E. H. Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review // Future Internet. 2022. Vol. 14. No. 4. Article 118. DOI: 10.3390/fi14040118
[17] Cho S., Cha S. SAD: Web Session Anomaly Detection Based on Parameter Estimation // Computers & Security. 2004. Vol. 23. No. 4. P. 312-319. DOI: 10.1016/j.cose.2004.01.006
[18] Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. Vol. 9. No. 8
[19] Sculley P. CSIC 2010 HTTP Dataset in CSV Format for WEKA Analysis [Электронный ресурс]. URL: https://petescully.co.uk/research/csic-2010-http-dataset-in-csv-format-for-weka-analysis/ (дата обращения: 30.03.2026)
[20] Юдова, Е. А., and О. Р. Лапонина. "Анализ возможностей использования технологий машинного обучения для выявления атак на веб-приложения." International Journal of Open Information Technologies 10.1 (2022): 61-68.
[21] Зубриенко, Г. А., and О. Р. Лапонина. "Методы оптимизации выборки данных для определения аномального трафика." International Journal of Open Information Technologies 4.10 (2016): 1-8.
[22] Намиот, Д. Е. Схемы атак на модели машинного обучения / Д. Е. Намиот // International Journal of Open Information Technologies. – 2023. – Т. 11, № 5. – С. 68-86. – EDN YVRDOB.
[23] Интернет цифровой железной дороги / В. П. Куприяновский, Г. В. Суконников, С. А. Синягов [и др.] // International Journal of Open Information Technologies. – 2016. – Т. 4, № 12. – С. 53-68. – EDN XETADZ.
[24] О работах по цифровой экономике / В. П. Куприяновский, Д. Е. Намиот, С. А. Синягов, А. П. Добрынин // Современные информационные технологии и ИТ-образование. – 2016. – Т. 12, № 1. – С. 243-249. – EDN XEQRFJ.

Developing a dynamic trust mechanism to protect web sessions in the ZeroTrust architecture

E.O. Lycheva, O.R. Laponina

Abstract - **This paper proposes a dynamic trust mechanism for web session protection, combining cryptographic and behavioral loops. The cryptographic loop ensures token validity, freshness, and non-portability, while the behavioral loop evaluates the current trajectory's compliance with a typical user scenario. The formal model of the mechanism includes requirements for token non-portability, reuse resistance, and trust continuity.**

The practical part is implemented as a prototype consisting of an authentication service, a risk assessment service, and a resource service. The behavioral loop utilizes an LSTM model trained on normal action sequences reconstructed from the CSIC 2010 dataset.

Experimental validation is performed on two scenarios of session abuse: an automated bot attack and business logic bypass. The primary metric used is ROC-AUC, which reflects the model's ability to rank abnormal examples above normal ones without a fixed threshold. The experimental results show that the decisive factor for the proposed mechanism is the amount of context available to the model at the time of evaluation. In both scenarios, quality increases with increasing history length, but the depth of context required to reliably distinguish anomalies from norms differs: a shorter sequence is sufficient for automated behavior, while a longer history is required for bypassing business logic. Moreover, the impact of the hidden state size is significantly weaker.

Therefore, the operation of the admission mechanism must rely on the accumulated session history as a mandatory part of the state, since evaluating an isolated request does not provide comparable reliability. Consequently, when implementing the proposed scheme, the length of the analyzed sequence becomes a critical operational parameter, and session trust must be recalculated as new actions accumulate.

Keywords - **Zero Trust, LSTM, web session anomalies, web session, dynamic trust, behavioral analysis, access control.**

REFERENCES

[1]NIST. Zero Trust Architecture. NIST Special Publication 800-207. Gaithersburg: National Institute of Standards and Technology, 2020

[2]OWASP Foundation. Session Management Cheat Sheet [Elektronnyj resurs]. URL: https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html (data obrashcheniya: 30.03.2026)

[3]Barth A. HTTP State Management Mechanism. RFC 6265. IETF, 2011

[4]Jones M., Bradley J., Sakimura N. JSON Web Token (JWT). RFC 7519. IETF, 2015

[5]Fett D., Campbell B., Bradley J., Lodderstedt T., Jones M., Waite D. OAuth 2.0 Demonstrating Proof of Possession (DPoP). RFC 9449. Internet Engineering Task Force (IETF), 2023

[6]Lodderstedt T., Campbell B., Bradley J., Sakimura N., Parecki A., Waite D. OAuth 2.0 Security Best Current Practice. RFC 9700. IETF, 2025

[7]Du M., Li F., Zheng G., Srikumar V. DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning // Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. 2017. URL: https://doi.org/10.1145/3133956.3134015

[8]Ferrag M., Maglaras L., Moschoyiannis S., Janicke H. Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study // Journal of Information Security and Applications. 2020. Vol. 50

[9]Calzavara S., Jonker H., Krumnow B., Rabitti A. Measuring Web Session Security at Scale // Computers & Security. 2021. Vol. 111. Article 102472

[10] Sivakorn S., Polakis I., Keromytis A. The Cracked Cookie Jar: HTTP Cookie Hijacking and the Exposure of Private Information // IEEE Symposium on Security and Privacy. 2018

[11] Rahman R. U., Tomar D. S. New biostatistics features for detecting web bot activity on web applications // Computers & Security. 2020. Vol. 97. Article 102001. DOI: 10.1016/j.cose.2020.102001

[12] Iliou C., Kostoulas T., Tsikrika T., Katos V., Vrochidis S., Kompatsiaris I. Detection of Advanced Web Bots by Combining Web Logs with Mouse Behavioural Biometrics // Digital Threats: Research and Practice. 2021. Vol. 2. No. 3. DOI: 10.1145/3447815

[13] Deepa G., Thilagam P. S., Praseed A., Pais A. R. DetLogic: A black-box approach for detecting logic vulnerabilities in web applications // Journal of Network and Computer Applications. 2018. Vol. 109. P. 89-109. DOI: 10.1016/j.jnca.2018.01.008

[14] Metin B., Wynn M., Tunali A., Kepir Y. Business Logic Vulnerabilities in the Digital Era: A Detection Framework Using Artificial Intelligence // Information. 2025. Vol. 16. No. 7. Article 585. DOI: 10.3390/info16070585

[15] Kwon D., Kim H., Kim J., Suh S. C., Kim I., Kim K. J. A Survey of Deep Learning-Based Network Anomaly Detection // Cluster Computing. 2019. Vol. 22. Suppl. 1. P. S949-S961. DOI: 10.1007/s10586-017-1117-8

[16] Alaoui R. L., Nfaoui E. H. Deep Learning for Vulnerability and Attack Detection on Web Applications: A Systematic Literature Review // Future Internet. 2022. Vol. 14. No. 4. Article 118. DOI: 10.3390/fi14040118

[17] Cho S., Cha S. SAD: Web Session Anomaly Detection Based on Parameter Estimation // Computers & Security. 2004. Vol. 23. No. 4. P. 312-319. DOI: 10.1016/j.cose.2004.01.006

[18] Hochreiter S., Schmidhuber J. Long Short-Term Memory // Neural Computation. 1997. Vol. 9. No. 8

[19] Sculley P. CSIC 2010 HTTP Dataset in CSV Format for WEKA Analysis [Elektronnyj resurs]. URL: https://petescully.co.uk/research/csic-2010-http-dataset-in-csv-format-for-weka-analysis/ (data obrashcheniya: 30.03.2026)

[20] YUdova, E. A., and O. R. Laponina. "Analiz vozmozhnostej ispol'zovaniya tekhnologij mashinnogo obucheniya dlya vyyavleniya atak na veb-prilozheniya." International Journal of Open Information Technologies 10.1 (2022): 61-68.

[21] Zubrienko, G. A., and O. R. Laponina. "Metody optimizacii vyborki dannyh dlya opredeleniya anomal'nogo trafika." International Journal of Open Information Technologies 4.10 (2016): 1-8.

[22] Namiot, D. E. Skhemy atak na modeli mashinnogo obucheniya / D. E. Namiot // International Journal of Open Information Technologies. – 2023. – T. 11, № 5. – S. 68-86. – EDN YVRDOB.

[23] Internet cifrovoj zheleznoj dorogi / V. P. Kupriyanovskij, G. V. Sukonnikov, S. A. Sinyagov [i dr.] // International Journal of Open Information Technologies. – 2016. – T. 4, № 12. – S. 53-68. – EDN XETADZ.

[24] O rabotah po cifrovoj ekonomike / V. P. Kupriyanovskij, D. E. Namiot, S. A. Sinyagov, A. P. Dobrynin // Sovremennye informacionnye tekhnologii i IT-obrazovanie. – 2016. – T. 12, № 1. – S. 243-249. – EDN XEQRFJ.