Методы атак и защиты в агентных системах на основе больших языковых моделей

В.А. Евграфов, Б.М. Нутфуллин, Д.Е. Намиот

***Аннотация*—Агентные системы на основе больших языковых моделей (LLM) представляют собой новый класс автономных программных систем, способных планировать и выполнять многоэтапные задачи с использованием внешних инструментов, долгосрочной памяти и межагентного взаимодействия. Переход от чат-ботов к автономным агентам порождает принципиально новые поверхности атак, не охваченные классическими моделями угроз. Настоящая обзорная статья систематизирует актуальное состояние исследований в области безопасности LLM-агентов. Предложена расширенная таксономия атак, разделённая на семь классов: (1) инъекции подсказок; (2) атаки на память агента; (3) атаки через инструменты и протоколы интеграции; (4) атаки в мультиагентных конфигурациях; (5) мультимодальные атаки; (6) атаки на цепочки инструментов и поставки; (7) темпоральные атаки. Систематизированы методы защиты по уровням вмешательства: текстовый (фильтрация и обнаружение инъекций), модельный (анализ внутренних представлений и активаций), инструментальный (контроль привилегий и политики вызовов), протокольный (расширения безопасности MCP), межсетевой (агентные файрволы) и системный (формальная верификация политик и криптографические подходы). Существующие защиты от indirect prompt injection остаются уязвимыми к адаптивным атакам, что ставит вопрос о необходимости оценки защит против адаптивного противника как стандартной практики. Анализ реальных инцидентов безопасности – от захвата платформ оркестрации до эксфильтрации данных через межагентное доверие – подтверждает практическую значимость теоретических моделей угроз. Особую остроту приобретают атаки с межсессионной персистентностью, при которых компрометация сохраняется между сессиями агента.**

***Ключевые слова*—большие языковые модели, LLM-агенты, инъекция подсказок, безопасность ИИ, атаки на память, Model Context Protocol, мультиагентные системы, цепочки инструментов, мультимодальные атаки**

I. Введение

В производственных агентных системах – от платформ оркестрации до инструментов автоматизации разработки – регулярно обнаруживаются критические уязвимости [1], [2]. Наделение LLM автономностью – способностью вызывать инструменты, изменять файловую систему, совершать транзакции – порождает угрозы, не имеющие аналогов в традиционных программных системах [3], [4], [5], [6].

Владимир Андреевич Евграфов, МГУ им. М.В. Ломоносова, факультет ВМК, кафедра информационной безопасности, Москва, Россия (e-mail: evgrafov.vladimir@gmail.com).

Булат Маратович Нутфуллин, МГУ им. М.В. Ломоносова, факультет ВМК, кафедра информационной безопасности, Москва, Россия (e-mail: bulat15g@gmail.com).

Дмитрий Евгеньевич Намиот, МГУ им. М.В. Ломоносова, факультет ВМК, кафедра информационной безопасности, Москва, Россия; доктор технических наук (e-mail: dnamiot@gmail.com).

Существующие обзоры сосредоточены преимущественно на инъекции подсказок [3], [7]. Атаки на память, протокольные уязвимости, мультимодальные угрозы и цепочки инструментов остаются без внимания. Бенчмарки не охватывают контекстно-зависимые задачи [3], защиты не оцениваются против адаптивного противника [8], а систематический анализ реальных инцидентов отсутствует.

Настоящая работа систематизирует угрозы и защиты LLM-агентов, вводит расширенную таксономию атак и анализирует реальные CVE.

II. Архитектура агентных систем на основе LLM

A. Компоненты агентной системы

Современная LLM-агентная система включает следующие ключевые компоненты [3], [4], [9].

LLM-ядро (Reasoning Core) – центральный модуль планирования и генерации действий. Уязвимость ядра к внедрённым инструкциям является первопричиной большинства рассматриваемых атак.

Инструменты (Tool Use) – внешние функциональные модули (поиск, исполнение кода, работа с файлами, вызов API), определяемые схемой, на основе которой ядро генерирует вызовы [10], [11].

Память – краткосрочная (контекстное окно), долгосрочная (персистентное хранилище) и RAG (извлечение из внешнего корпуса) [12], [13], [14]. Каждый уровень создаёт собственный вектор атаки.

Протоколы интеграции – прежде всего Model Context Protocol (MCP), де-факто стандарт взаимодействия агентов с внешними сервисами [15], [16].

Мультиагентные конфигурации – оркестратор координирует субагентов [17]. Межагентное доверие и каскадное распространение контекста порождают дополнительные поверхности атак.

III. Таксономия угроз

Таксономия организована по принципу расширяющейся поверхности атаки – от манипуляции входными данными через компрометацию памяти и инструментов к мультиагентным, мультимодальным, цепочечным и темпоральным угрозам.

A. Атаки на входные данные: инъекция подсказок

Инъекция подсказок (Prompt Injection, PI) остаётся наиболее изученным классом атак на LLM-агентов. Суть проблемы – фундаментальная неспособность языковой модели надёжно разграничить *инструкции* и *данные* внутри единого текстового потока [3].

Косвенная инъекция подсказок (IPI) значительно опаснее прямой: вредоносные инструкции поступают через внешние данные, извлекаемые агентом, – веб-страницы, документы, результаты API-вызовов [3], [7].

IPI через RAG представляет особую опасность для агентов с внешними корпусами документов. Предложенный метод разделяет вредоносный контент на *триггерный* и *атакующий* фрагменты в разных частях корпуса, достигая близкой к 100% частоты извлечения на 11 бенчмарках [12].

Интересно, что обнаружен и скрытый канал через URL-превью (Silent Egress): механизм автоматической генерации превью побуждает агента выполнить исходящий запрос, эксфильтрируя контекст выполнения [18].

На вершине эскалации находятся адаптивные атаки. Фреймворк AdapTools ориентирован на системы с интеграцией через MCP и динамически подстраивает полезную нагрузку под конкретную конфигурацию инструментов агента, обеспечивая 2,13-кратное улучшение ASR [19].

B. Атаки на память агента

MINJA (Memory INJection Attack) демонстрирует, что долгосрочная память агента может быть отравлена через обычные пользовательские запросы без привилегированного доступа. Экспериментальная оценка показала более 95% успешности инъекции [14].

MemoryGraft нацеливается на RAG-хранилище: злоумышленник имплантирует вредоносные шаблоны процедур, замаскированные под записи об успешном решении задач [13]. Агент воспринимает отравленные данные как собственный проверенный опыт.

Наивысшую степень персистентности обеспечивает концепция **Zombie Agents** [20]. Злоумышленник имплантирует полезную нагрузку, которая *переживает смену сессий* и превращает агента в перманентную марионетку.

Ключевое отличие – *межсессионная персистентность* полезной нагрузки.

C. Атаки через инструменты и протоколы

На протокольном уровне выявлены три фундаментальные уязвимости MCP: (1) отсутствие аттестации возможностей серверов; (2) двунаправленная выборка без аутентификации источника; (3) неявное транзитивное доверие в мультисерверных конфигурациях [15].

Проблема несбалансированного инструментального агентства [10] порождает дилемму между избыточными разрешениями и чрезмерными ограничениями. ToolHijacker [21] внедряет вредоносное описание инструмента в библиотеку агента. Атака Log-To-Leak [22] маскирует эксфильтрацию данных через MCP под рутинное логирование.

Для CUA-агентов, взаимодействующих с графическим интерфейсом, идентифицированы семь самостоятельных классов уязвимостей [5].

D. Атаки в мультиагентных системах

Атака OMNI-LEAK демонстрирует утечку на уровне оркестрации: оркестратор агрегирует чувствительный контекст от множества источников и передаёт его субагентам, расширяя поверхность утечки [17].

Agent-in-the-Middle (AiTM) – атака, при которой LLM-управляемый адверсарный агент перехватывает и манипулирует межагентными сообщениями [23]. В отличие от классического MITM, перехватчик сам является языковой моделью с адаптивной модификацией сообщений.

Показана возможность произвольного выполнения кода в мультиагентных системах с успешностью 58–90% на GPT-4o и до 100% в отдельных конфигурациях [24].

MAD-Spear [25] эксплуатирует склонность LLM к конформности: компрометация даже 1 из 6 агентов существенно влияет на результат мультиагентных дебатов [25].

Причём 82,4% моделей оказались уязвимы через канал межагентного доверия: LLM отклоняют прямые вредоносные команды, но выполняют идентичные нагрузки от «доверенных» агентов [26]. В мультиагентных конфигурациях атака Log-To-Leak приобретает дополнительную скрытность, маскируясь под легитимную операцию обслуживания [22].

E. Мультимодальные атаки

CrossInject [27] исследует кросс-модальную инъекцию: согласованные адверсарные сигналы встраиваются одновременно в визуальную и текстовую модальности, обходя защиты отдельных каналов. Атака достигает +30,1% ASR по сравнению с SOTA [27].

Бенчмарк **ARE / VisualWebArena-Adv** предоставляет 200 целевых адверсарных задач для стандартизированной оценки устойчивости мультимодальных агентов [28].

Исследование **Mind the GAP** [29] демонстрирует фундаментальный разрыв: безопасность на уровне текста не переносится на безопасность вызовов инструментов. Тестирование 6 фронтирных моделей показало, что модели, корректно отклоняющие вредоносные текстовые запросы, выполняют идентичные по сути вредоносные вызовы инструментов [29].

F. Атаки на цепочки инструментов и поставки

Фреймворк **STAC** [30] формализует многоходовую атаку: *каждый отдельный вызов инструмента проходит проверку безопасности*, но их совокупность приводит к компрометации. ASR превышает 90% для большинства агентов [30].

Обманная регистрация инструментов в экосистемах MCP/A2A (**Tool Squatting**), аналогичная typosquatting в пакетных менеджерах, позволяет подменить легитимный инструмент вредоносным [31]. Фреймворк **SkillJect** [32] предлагает автоматизированный метод инъекции через агентные навыки, достигающий среднего ASR 95,1% [32].

На вершине эскалации – концепция **Viral Agent Loop** [33]: самораспространяющийся генеративный червь, функционирующий без программных уязвимостей на уровне кода. Вредоносная нагрузка передаётся между агентами через штатные каналы коммуникации [33].

G. Темпоральные атаки (TOCTOU)

Уязвимости Time-of-Check to Time-of-Use (TOCTOU) эксплуатируют гонку состояний между проверкой безопасности вызова и его выполнением. Предложенные

защиты сокращают окно атаки на 95%, однако полное устранение требует архитектурных изменений [34].

Сравнительный анализ атак представлен в Таблице I.

IV. Методы защиты

Защиты систематизированы по стадии вмешательства [3]: текстовый, модельный, уровень выполнения, протокольный и архитектурный.

A. Защита на уровне текста и контекста

PromptArmor [35] использует стандартную LLM в качестве охранного барьера для обнаружения инъектированных подсказок. На бенчмарке AgentDojo FPR и FNR составляют менее 1%, а ASR после применения защиты падает до менее 1% [35].

В работе [8] показано, что протестированные авторами IPI-защиты остаются уязвимыми к адаптивным атакам (подробнее см. раздел VI). Этот результат касается конкретно IPI-ориентированных защит и не затрагивает защиты инструментального [11], протокольного [16] или архитектурного уровня [36], [37]. Тем не менее он ставит вопрос об оценке *любых* защит против адаптивного противника как стандартной практики.

B. Защита на уровне модели

ICON [38] реализует парадигму «зондирование – смягчение»: IPI-атаки оставляют характерные сигнатуры чрезмерной фокусировки в скрытом пространстве модели, что позволяет обнаруживать их без преждевременного завершения легитимных процессов [38].

AgentSentry [39] моделирует IPI как *темпоральный причинный захват* управления. Система локализует точки захвата через контрфактические переисполнения и обеспечивает безопасное продолжение через причинно-ориентированную очистку контекста [39].

ARGUS [40] применяет управление активациями для нейтрализации мультимодальных инъекций – через изображения, видео и аудио – непосредственно во внутренних представлениях модели.

C. Управление привилегиями инструментов

Progent [11] – первый фреймворк контроля привилегий для LLM-агентов. Архитектура включает предметно-ориентированный язык (DSL) для задания политик и детерминированную среду исполнения с доказуемыми гарантиями безопасности [11].

AgenTRIM [10] применяет принцип минимальных привилегий на каждом шаге выполнения, существенно снижая успешность атак на AgentDojo [10].

AgentArmor [41] трансформирует трассы выполнения в графы программных зависимостей (PDG), снижая ASR до 3% при 1% функциональных накладных расходов [41].

MindGuard [42] обнаруживает отравленные вызовы инструментов с точностью 94–99% при времени обработки менее 1 секунды [42].

D. Протокольный уровень защиты

SMCP [16] расширяет MCP средствами безопасности, устраняя три уязвимости из [15]: аттестацию серверов, аутентификацию источника и управление транзитивным доверием.

VIGIL [43] реализует протокол «верификация перед фиксацией», снижая успешность атак на 22% и удваивая полезность агента под атакой [43].

E. Межсетевые экраны для агентных систем

MS Firewalled Agentic Networks [36] обеспечивают трёхуровневый контроль на границах между агентами, инструментами и внешними источниками. Утечка данных снижается с 70% до менее 2%, атаки на изменение состояния – с 45% до 0% [36].

ControlValve [44] устраняет уязвимости LlamaFirewall [6], применяя принципы целостности потока управления из классической компьютерной безопасности.

GAF [45] предлагает четырёхуровневую модель безопасности с задержкой менее 10 мс и пропускной способностью более 20 000 запросов/с [45].

F. Формальная верификация и политики привилегий

Следует отметить, что полная формальная верификация LLM-агентных систем на практике невозможна из-за недетерминированности языковых моделей. Тем не менее ряд работ предлагают частичные формальные гарантии.

PCAS [37] применяет Datalog-производный язык для задания политик безопасности (улучшение соответствия с 48% до 93%) [37]. **Authenticated Workflows** [46] вводят криптографический уровень доверия с языком политик MAPL [46]. **AgentBound** [47] реализует модель контроля доступа для MCP-серверов с автогенерацией политик из исходного кода (80,9% точности).

G. Защита памяти и среды выполнения

Composite Trust Scoring [14] присваивает каждой записи памяти оценку достоверности, градуированно снижая вес потенциально скомпрометированных записей при принятии решений.

H. Фундаментальное ограничение: трилемма безопасности

В работе [8] показано, что рассмотренные авторами защиты от indirect prompt injection остаются уязвимыми к адаптивным атакам: для всех протестированных ими IPI-защит достигнут ASR $>$50%. Этот результат относится к IPI-ориентированным защитам; инструментальные [11], протокольные [16] и архитектурные [36], [37] защиты требуют отдельной проверки. Оценка любых защит против адаптивного противника должна стать стандартной практикой.

На более фундаментальном уровне сформулирована трилемма безопасности агентных систем: ни один подход не обеспечивает одновременно высокую защищённость, высокую полезность и низкую задержку [3]. Причина – рассинхронизация динамических доверительных состояний и статических границ авторизации [4].

Сравнительный анализ методов защиты представлен в Таблице II.

Таблица I
Сравнительный анализ атак на LLM-агентные системы

Атака	Вектор	Персист.	Цель	ASR	Ист.
IPI	Внешние данные	Нет	Захват инструментов	Варьируется	[3], [7]
IPI через RAG	Векторное хранилище	Нет	Гарантир. извлечение	≈100%	[12]
AdapTools	MCP / инструменты	Нет	Обход защит	2,13× baseline	[19]
MINJA	Пользоват. запросы	Да	Долгосроч. память	>95% инъекция	[14]
MemoryGraft	RAG-хранилище	Да	Процедурная память	—	[13]
Zombie Agents	Безобидные сессии	Да	Марионетка	—	[20]
MCP-атаки	MCP-протокол	Нет	Расшир. привилегий	—	[15]
OMNI-LEAK	Оркестратор	Нет	Утечка данных	—	[17]
CrossInject	Визуал. + текст	Нет	Мультимод. агенты	+30,1% к SOTA	[27]
STAC	Последов. вызовов	Нет	Композ. результат	>90%	[30]
AiTM	Межагент. коммун.	Нет	Перехват / манипул.	—	[23]
MAD-Spear	Мультиагент. дебат	Нет	Манипул. результат	Эскалация	[25]
Межагент. доверие	Запросы агентов	Нет	Обход выравнив.	82,4% мод.	[26]
ToolHijacker	Библ. инструм.	Нет	Перенапр. вызовов	—	[21]
Tool Squatting	Реестры инстр.	Да	Подмена инстр.	—	[31]
SkillJect	Скрипты / SKILL.md	Нет	Авто PI	95,1%	[32]
Viral Agent Loop	Самораспростран.	Да	Генерат. червь	—	[33]
Log-To-Leak	MCP-логирование	Нет	Утечка через MCP	—	[22]

V. Бенчмарки, методология оценки и реальные инциденты

A. Обзор бенчмарков

Основные бенчмарки для оценки безопасности LLM-агентов:

- **AGENTPI** [3] – контекстно-зависимые задачи для оценки PI с учётом состояния агента.
- **SIREN** [43] – 959 случаев инъекции в поток инструментов.
- **AgentDojo** [48] – 97 задач, 629 случаев; GPT-4o достигает 69% полезности при 53,1% ASR.
- **ASB** [49] – 10 сценариев, свыше 400 инструментов, 27 методов атаки (макс. ASR 84,30%).
- **b3** [50] – 194 331 атака; способности к рассуждению повышают безопасность, размер модели *не коррелирует* с ней.
- **AgentLAB** [51] – первый бенчмарк адаптивных атак с длинным горизонтом (5 семейств, 28 сред, 644 случая); даже GPT-5.1 показывает ASR ≈70% [8].
- **MCPTox** [52] – отравление инструментов на 45 реальных MCP-серверах (353 инструмента, 1312 случаев).
- **SafeSearch** [53] – макс. ASR 90,5% на поисковых агентах.

Интересно, что практически ни один бенчмарк, кроме AgentLAB, не тестирует адаптивного противника [8].

B. Реальные инциденты и CVE

Теоретические атаки из раздела III находят подтверждение в зафиксированных уязвимостях производственных систем.

CVE-2025-53773 (GitHub Copilot RCE, CVSS 7.8) [1] – непрямая инъекция через невидимые Unicode-символы в комментариях кода, обеспечивающая удалённое выполнение кода. Обнаружен потенциал червеобразного распространения [33].

CVE-2025-68664 (LangChain, CVSS 9.3) [54] эксплуатирует уязвимости сериализации: доверие агента к результатам десериализации позволяет внедрить произвольный код.

CVE-2026-27966 (Langflow CSV Agent RCE) [55] демонстрирует опасность чрезмерных привилегий: жёстко заданный параметр `allow_dangerous_code=True` превращает агент в вектор полного компрометирования системы.

CVE-2026-21858 (n8n, CVSS 10.0) [2] – неаутентифицированная утечка файлов и захват платформы оркестрации. Компрометация платформы автоматически компрометирует всех развёрнутых агентов.

ServiceNow Now Assist [56] – низкопривилегированный агент обманом заставляет высокопривилегированного агента экспортировать конфиденциальные файлы. Это прямая валидация атак на межагентное доверие [26] и проблемы Trust-Authorization Mismatch [4].

Таблица II
Сравнительный анализ методов защиты LLM-агентов

Метод	**Подход**	**Ключевые результаты**	**Ист.**
PromptArmor	Текст.: LLM-guardrail для обнаружения PI	FPR и FNR <1% на AgentDojo	[35]
ICON	Модельн.: обнаружение over-focusing	Сохраняет непрерывность задачи	[38]
AgentSentry	Модельн.: темпор. причинная диагностика	Причинная очистка контекста	[39]
ARGUS	Модельн.: управление активациями	Защита от вид./аудио PI	[40]
Progent	Инструм.: DSL-политики привилегий	Доказуемые гарантии	[11]
AgenTRIM	Инструм.: least-privilege per-step	Снижает атаки на AgentDojo	[10]
AgentArmor	Инструм.: PDG-анализ трасс	ASR до 3%, 1% overhead	[41]
MindGuard	Инструм.: DDG для обнаружения отравления	94–99% precision, <1 с	[42]
SMCP	Протокольн.: расширения безопасности MCP	Аттестация, аутентификация	[16]
VIGIL	Протокольн.: verify-before-commit	>22% снижение ASR, 2× утилита	[43]
MS Firewalled Networks	Межсет. экран: трёхуровневый	Утечка 70%→<2%, атаки 45%→0%	[36]
ControlValve	Межсет. экран: CFI + least privilege	Исправляет уязвим. LlamaFirewall [6]	[44]
GAF	Межсет. экран: 4-уровневая модель	<10 мс, >20K запр./с	[45]
PCAS	Формальн.: Datalog-политики, монитор	48%→93% compliance, 0 нарушений	[37]
Auth. Workflows	Формальн.: криптографические доказательства	4 границы защиты	[46]
AgentBound	Формальн.: Android-стиль MCP-разрешения	80,9% точность автогенерации	[47]
Composite Trust	Память: скоринг доверия записей	Снижает влияние MINJA	[14]

VI. Обсуждение

Агентная парадигма порождает качественно новые классы угроз: межсессионную персистентность [20], [13], [33], атаки на протокольную экосистему [15], эксплуатацию межагентного доверия [17], [23]. Эти угрозы не сводятся к инъекциям подсказок. Как отмечено в разделе IV, в работе [8] показано, что IPI-защиты уязвимы к адаптивным атакам. Реальные инциденты подтверждают практическую значимость теоретических моделей.

Наиболее перспективны формально верифицируемые защиты, стандартизированные бенчмарки с адаптивным противником и решение проблемы Trust-Authorization Mismatch [4] на архитектурном уровне. Собственно говоря, от накопления эмпирических атак и контрмер необходим переход к системному проектированию безопасных агентных архитектур. Трилемма безопасности – одновременное обеспечение защищённости, полезности и низкой задержки [3] – остаётся нерешённой фундаментальной проблемой.

VII. Заключение

В работе предложена расширенная таксономия атак на LLM-агентные системы из семи классов, систематизированы методы защиты по уровням вмешательства и проанализированы реальные CVE. Показано, что переход от языковых моделей к автономным агентам качественно расширяет поверхность атаки, а существующие защиты не выдерживают проверки адаптивным противником. Необходим переход к системному проектированию безопасных агентных архитектур с формально верифицируемыми гарантиями.

Список литературы

[1] Red Embrace The. GitHub Copilot RCE via prompt injection (CVE-2025-53773). — URL: https://embracethered.com/blog/posts/2025/github-copilot-remote-code-execution-via-prompt-injection/. — 2025.
[2] CSO Online. Critical RCE flaw allows full takeover of n8n AI workflow platform (CVE-2026-21858). — URL: https://www.csoonline.com/article/4113980/. — 2026.
[3] Wang X. et al. The landscape of prompt injection threats in LLM agents. — arXiv:2602.10453. — 2026.
[4] Shi Y. et al. SoK: Trust-authorization mismatch in LLM agent interactions. — arXiv:2512.06914. — 2025.
[5] Jones E. et al. A systematization of security vulnerabilities in computer use agents. — arXiv:2507.05445. — 2025.
[6] Meta AI. LlamaFirewall: Open source guardrail system for secure AI agents. — arXiv:2505.03574. — 2025.
[7] Ji Z. et al. Taxonomy, evaluation and exploitation of IPI-centric LLM agent defense frameworks. — arXiv:2511.15203. — 2025.
[8] Zhan Q. et al. Adaptive attacks break defenses against indirect prompt injection // NAACL 2025 Findings. — 2025.
[9] Jiang H. et al. SoK: Agentic skills — beyond tool use in LLM agents. — arXiv:2602.20867. — 2026.
[10] Betser N. et al. AgenTRIM: Tool risk mitigation for agentic AI. — arXiv:2601.12449. — 2026.

[11] Shi L. et al. Progent: Programmable privilege control for LLM agents. — arXiv:2504.11703. — 2025.
[12] Chang Y. et al. Overcoming the retrieval barrier: Indirect prompt injection in the wild. — arXiv:2601.07072. — 2026.
[13] Srivastava A. et al. MemoryGraft: Persistent compromise via poisoned experience retrieval. — arXiv:2512.16962. — 2025.
[14] Sunil R. et al. Memory poisoning attack and defense on memory based LLM-agents. — arXiv:2601.05504. — 2026.
[15] Maloyan A., Namiot D. Breaking the protocol: Security analysis of the model context protocol. — arXiv:2601.17549. — 2026.
[16] Hou Y. et al. SMCP: Secure model context protocol. — arXiv:2602.01129. — 2026.
[17] Naik A. et al. OMNI-LEAK: Orchestrator multi-agent network induced data leakage. — arXiv:2602.13477. — 2026.
[18] Lan M. et al. Silent egress: Implicit prompt injection makes LLM agents leak. — arXiv:2602.22450. — 2026.
[19] Wang L. et al. AdapTools: Adaptive tool-based indirect prompt injection attacks. — arXiv:2602.20720. — 2026.
[20] Yang K. et al. Zombie agents: Persistent control via self-reinforcing injections. — arXiv:2602.15654. — 2026.
[21] Shi J. et al. ToolHijacker: Prompt injection attack to tool selection. — arXiv:2504.19793. — 2025.
[22] others. Log-to-leak: Prompt injection via MCP. — OpenReview:UVgbFuXPaO. — 2025.
[23] He J. et al. Agent-in-the-middle: Red-teaming multi-agent systems via communication attacks. — arXiv:2502.14847. — 2025.
[24] Triedman H., Jha R., Shmatikov V. Multi-agent systems execute arbitrary malicious code. — arXiv:2503.12188. — 2025.
[25] Cui Y., Du H. MAD-Spear: Conformity-driven prompt injection on multi-agent debate. — arXiv:2507.13038. — 2025.
[26] Lupinacci L. et al. The dark side of LLMs: Agent-based attacks for complete computer takeover. — arXiv:2507.06850. — 2025.
[27] Cheng X. et al. CrossInject: Cross-modal prompt injection // ACM Multimedia 2025. — 2025.
[28] Wu C. H. et al. Dissecting adversarial robustness of multimodal LM agents // ICLR 2025. — 2025.
[29] Cartagena A., Teixeira A. Mind the GAP: Text safety does not transfer to tool-call safety. — arXiv:2602.16943. — 2026.
[30] Amazon Science. STAC: When innocent tools form dangerous chains to jailbreak LLM agents. — arXiv:2509.25624. — 2025.
[31] Narajala V. S. et al. Tool squatting: Zero trust registry-based approach. — arXiv:2504.19951. — 2025.
[32] Jiaxiaojun et al. SkillJect: Automated skill-based prompt injection for coding agents. — arXiv:2602.14211. — 2026.
[33] others. Agentic AI as a cybersecurity attack surface: Runtime supply chains. — arXiv:2602.19555. — 2026.
[34] Lilienthal D., Hong S. Mind the gap: TOCTOU vulnerabilities in LLM-enabled agents. — arXiv:2508.17155. — 2025.
[35] Shi T. et al. PromptArmor: Simple yet effective prompt injection defense. — arXiv:2507.15219. — 2025.
[36] Abdelnabi S. et al. Firewalls to secure dynamic LLM agentic networks. — arXiv:2502.01822. — 2025.
[37] Palumbo N. et al. PCAS: Policy compiler for secure agentic systems. — arXiv:2602.16708. — 2026.
[38] Wang R. et al. ICON: Indirect prompt injection defense via inference-time correction. — arXiv:2602.20708. — 2026.
[39] Zhang T. et al. AgentSentry: Mitigating indirect prompt injection via temporal causal diagnostics. — arXiv:2602.22724. — 2026.
[40] others. ARGUS: Defending against multimodal IPI via activation steering. — arXiv:2512.05745. — 2025.
[41] Wang Y. et al. AgentArmor: Program analysis on agent runtime trace. — arXiv:2508.01249. — 2025.
[42] others. MindGuard: Decision inspection against metadata poisoning. — arXiv:2508.20412. — 2025.
[43] Lin J. et al. VIGIL: Defending LLM agents against tool stream injection. — arXiv:2601.05755. — 2026.
[44] Jha R. et al. Breaking and fixing defenses against control-flow hijacking in MAS. — arXiv:2510.17276. — 2025.
[45] NeuralTrust. Generative application firewall (GAF). — arXiv:2601.15824. — 2026.
[46] Rajagopalan M., Rao V. Authenticated workflows: A systems approach. — arXiv:2602.10465. — 2026.
[47] others. AgentBound: Access control for MCP servers. — arXiv:2510.21236. — 2025.
[48] Debenedetti E. et al. AgentDojo: Dynamic environment for prompt injection evaluation. — arXiv:2406.13352. — 2024.
[49] Zhang H. et al. Agent security bench (ASB) // ICLR 2025. — 2025.
[50] Bazinska J. et al. Breaking agent backbones (b3 benchmark) // ICLR 2026. — 2026.
[51] others. AgentLAB: Long-horizon attack benchmark. — arXiv:2602.16901. — 2026.
[52] Wang Y., Gao et al. MCPTox: Benchmark for tool poisoning on real-world MCP servers. — arXiv:2508.14925. — 2025.
[53] Dong J. et al. SafeSearch: Red-teaming LLM search agents. — arXiv:2509.23694. — 2025.
[54] Cyata. LangChain “LangGrinch” (CVE-2025-68664). — URL: https://cyata.ai/blog/langgrinch-langchain-core-cve-2025-68664/. — 2025.
[55] CVE Reports. Langflow CSV Agent RCE (CVE-2026-27966). — URL: https://cvereports.com/reports/CVE-2026-27966. — 2026.
[56] Sombra Inc. ServiceNow Now Assist privilege escalation via second-order prompt injection. — URL: https://sombrainc.com/blog/llm-security-risks-2026. — 2026.

Attack Methods and Defenses in LLM-Based Agentic Systems

V.A. Evgrafov, B.M. Nutfullin, D.E. Namiot

***Abstract*—LLM-based agentic systems represent a new class of autonomous software capable of planning and executing multi-step tasks using external tools, long-term memory, and inter-agent communication. The transition from chatbots to autonomous agents introduces fundamentally novel attack surfaces not captured by classical threat models. This survey systematizes the current state of LLM-agent security research. We propose an extended threat taxonomy divided into seven classes: (1) prompt injection attacks; (2) memory attacks; (3) tool and protocol attacks; (4) multi-agent attacks; (5) multi-modal attacks; (6) tool chain and supply chain attacks; (7) temporal attacks. Defense methods are systematized by intervention level: textual (injection filtering and detection), model-level (analysis of internal representations and activations), tool-level (privilege control and call policies), protocol-level (MCP security extensions), firewall (agentic firewalls), and systemic (formal policy verification and cryptographic approaches). Existing indirect prompt injection defenses remain vulnerable to adaptive attacks, raising the question of evaluating defenses against adaptive adversaries as a standard practice. Analysis of real-world security incidents – from orchestration platform takeovers to data exfiltration via inter-agent trust – validates the practical significance of theoretical threat models. Attacks with cross-session persistence, where compromise survives agent session boundaries, are particularly acute.**

***Keywords*—large language models, LLM agents, prompt injection, AI security, memory poisoning, Model Context Protocol, multi-agent systems, tool chain attacks, multi-modal attacks**

References

1. Embrace The Red, “GitHub Copilot RCE via Prompt Injection (CVE-2025-53773),” 2025.
2. CSO Online, “Critical RCE Flaw Allows Full Takeover of n8n AI Workflow Platform (CVE-2026-21858),” 2026.
3. Wang, X. and others, “The Landscape of Prompt Injection Threats in LLM Agents,” arXiv:2602.10453, 2026.
4. Shi, Y. and others, “SoK: Trust-Authorization Mismatch in LLM Agent Interactions,” arXiv:2512.06914, 2025.
5. Jones, E. and others, “A Systematization of Security Vulnerabilities in Computer Use Agents,” arXiv:2507.05445, 2025.
6. Meta AI, “LlamaFirewall: Open Source Guardrail System for Secure AI Agents,” arXiv:2505.03574, 2025.
7. Ji, Z. and others, “Taxonomy, Evaluation and Exploitation of IPI-Centric LLM Agent Defense Frameworks,” arXiv:2511.15203, 2025.
8. Zhan, Q. and others, “Adaptive Attacks Break Defenses Against Indirect Prompt Injection,” *NAACL 2025 Findings*, 2025. arXiv:2503.00061.
9. Jiang, H. and others, “SoK: Agentic Skills — Beyond Tool Use in LLM Agents,” arXiv:2602.20867, 2026.
10. Betser, N. and others, “AgenTRIM: Tool Risk Mitigation for Agentic AI,” arXiv:2601.12449, 2026.
11. Shi, L. and others, “Progent: Programmable Privilege Control for LLM Agents,” arXiv:2504.11703, 2025.
12. Chang, Y. and others, “Overcoming the Retrieval Barrier: Indirect Prompt Injection in the Wild,” arXiv:2601.07072, 2026.
13. Srivastava, A. and others, “MemoryGraft: Persistent Compromise via Poisoned Experience Retrieval,” arXiv:2512.16962, 2025.
14. Sunil, R. and others, “Memory Poisoning Attack and Defense on Memory Based LLM-Agents,” arXiv:2601.05504, 2026.
15. Maloyan, A. and Namiot, D., “Breaking the Protocol: Security Analysis of the Model Context Protocol,” arXiv:2601.17549, 2026.
16. Hou, Y. and others, “SMCP: Secure Model Context Protocol,” arXiv:2602.01129, 2026.
17. Naik, A. and others, “OMNI-LEAK: Orchestrator Multi-Agent Network Induced Data Leakage,” arXiv:2602.13477, 2026.
18. Lan, M. and others, “Silent Egress: Implicit Prompt Injection Makes LLM Agents Leak,” arXiv:2602.22450, 2026.
19. Wang, L. and others, “AdapTools: Adaptive Tool-based Indirect Prompt Injection Attacks,” arXiv:2602.20720, 2026.
20. Yang, K. and others, “Zombie Agents: Persistent Control via Self-Reinforcing Injections,” arXiv:2602.15654, 2026.
21. Shi, J. and others, “ToolHijacker: Prompt Injection Attack to Tool Selection,” arXiv:2504.19793, 2025.
22. “Log-To-Leak: Prompt Injection via MCP,” OpenReview:UVgbFuXPaO, 2025.
23. He, J. and others, “Agent-in-the-Middle: Red-Teaming Multi-Agent Systems via Communication Attacks,” arXiv:2502.14847, 2025.
24. Triedman, H. and Jha, R. and Shmatikov, V., “Multi-Agent Systems Execute Arbitrary Malicious Code,” arXiv:2503.12188, 2025.
25. Cui, Y. and Du, H., “MAD-Spear: Conformity-Driven Prompt Injection on Multi-Agent Debate,” arXiv:2507.13038, 2025.
26. Lupinacci, L. and others, “The Dark Side of LLMs: Agent-based Attacks for Complete Computer

Takeover," arXiv:2507.06850, 2025.
27. Cheng, X. and others, "CrossInject: Cross-Modal Prompt Injection," *ACM Multimedia 2025*, 2025. arXiv:2504.14348.
28. Wu, C. H. and others, "Dissecting Adversarial Robustness of Multimodal LM Agents," *ICLR 2025*, 2025. arXiv:2406.12814.
29. Cartagena, A. and Teixeira, A., "Mind the GAP: Text Safety Does Not Transfer to Tool-Call Safety," arXiv:2602.16943, 2026.
30. Amazon Science, "STAC: When Innocent Tools Form Dangerous Chains to Jailbreak LLM Agents," arXiv:2509.25624, 2025.
31. Narajala, V. S. and others, "Tool Squatting: Zero Trust Registry-Based Approach," arXiv:2504.19951, 2025.
32. Jiaxiaojun and others, "SkillJect: Automated Skill-Based Prompt Injection for Coding Agents," arXiv:2602.14211, 2026.
33. "Agentic AI as a Cybersecurity Attack Surface: Runtime Supply Chains," arXiv:2602.19555, 2026.
34. Lilienthal, D. and Hong, S., "Mind the Gap: TOCTOU Vulnerabilities in LLM-Enabled Agents," arXiv:2508.17155, 2025.
35. Shi, T. and others, "PromptArmor: Simple yet Effective Prompt Injection Defense," arXiv:2507.15219, 2025.
36. Abdelnabi, S. and others, "Firewalls to Secure Dynamic LLM Agentic Networks," arXiv:2502.01822, 2025.
37. Palumbo, N. and others, "PCAS: Policy Compiler for Secure Agentic Systems," arXiv:2602.16708, 2026.
38. Wang, R. and others, "ICON: Indirect Prompt Injection Defense via Inference-Time Correction," arXiv:2602.20708, 2026.
39. Zhang, T. and others, "AgentSentry: Mitigating Indirect Prompt Injection via Temporal Causal Diagnostics," arXiv:2602.22724, 2026.
40. "ARGUS: Defending Against Multimodal IPI via Activation Steering," arXiv:2512.05745, 2025.
41. Wang, Y. and others, "AgentArmor: Program Analysis on Agent Runtime Trace," arXiv:2508.01249, 2025.
42. "MindGuard: Decision Inspection Against Metadata Poisoning," arXiv:2508.20412, 2025.
43. Lin, J. and others, "VIGIL: Defending LLM Agents Against Tool Stream Injection," arXiv:2601.05755, 2026.
44. Jha, R. and others, "Breaking and Fixing Defenses Against Control-Flow Hijacking in MAS," arXiv:2510.17276, 2025.
45. NeuralTrust, "Generative Application Firewall (GAF)," arXiv:2601.15824, 2026.
46. Rajagopalan, M. and Rao, V., "Authenticated Workflows: A Systems Approach," arXiv:2602.10465, 2026.
47. "AgentBound: Access Control for MCP Servers," arXiv:2510.21236, 2025.
48. Debenedetti, E. and others, "AgentDojo: Dynamic Environment for Prompt Injection Evaluation," arXiv:2406.13352, 2024.
49. Zhang, H. and others, "Agent Security Bench (ASB)," *ICLR 2025*, 2025. arXiv:2410.02644.
50. Bazinska, J. and others, "Breaking Agent Backbones (b3 Benchmark)," *ICLR 2026*, 2026. arXiv:2510.22620.
51. "AgentLAB: Long-Horizon Attack Benchmark," arXiv:2602.16901, 2026.
52. Wang, Y. and Gao and others, "MCPTox: Benchmark for Tool Poisoning on Real-World MCP Servers," arXiv:2508.14925, 2025.
53. Dong, J. and others, "SafeSearch: Red-Teaming LLM Search Agents," arXiv:2509.23694, 2025.
54. Cyata, "LangChain 'LangGrinch' (CVE-2025-68664)," 2025.
55. CVE Reports, "Langflow CSV Agent RCE (CVE-2026-27966)," 2026.
56. Sombra Inc., "ServiceNow Now Assist Privilege Escalation via Second-Order Prompt Injection," 2026.