Автономная многослойная платформа анализа цифрового следа с ансамблем малых языковых моделей, контролируемой эксплуатацией уязвимостей и сквозной анонимизацией

Тельбух Вячеслав Владимирович
кандидат технических наук, преподаватель, ВКА им. А.Ф. Можайского,

Данилов Виталий Валерьевич
кандидат технических наук, преподаватель, ВКА им. А.Ф. Можайского,

Романов Павел Андреевич
кандидат технических наук, преподаватель ВКА им. А.Ф. Можайского

Современные задачи анализа цифрового следа требуют одновременного выполнения пассивного сбора данных из открытых источников, семантического анализа полученной информации, автоматической идентификации уязвимостей и, в отдельных случаях, контролируемой верификации эксплуатируемости обнаруженных уязвимостей. При этом ужесточение регуляторных требований и необходимость работы на объектах критической информационной инфраструктуры (КИИ) исключают использование облачных сервисов и крупных языковых моделей, требующих GPU-ускорителей. В статье представлена архитектура и экспериментальная оценка полностью локальной платформы анализа цифрового следа, функционирующей на оборудовании потребительского класса (Intel Core i7, 64 ГБ ОЗУ, без GPU). Платформа объединяет 232 модуля сбора данных, ансамбль из 16 специализированных малых языковых моделей (Small Language Models, SLM, ≤9B параметров) с адаптивной маршрутизацией [2], контекстно-зависимый RAG-индекс, каталог из 26 инструментов безопасности с универсальным Python-fallback, многоагентную оркестрацию на базе LangGraph [3], а также подсистемы контролируемой эксплуатации и сквозной анонимизации оператора. Экспериментально установлено, что предложенный ансамбль SLM потребляет на 32,3 % меньше пиковой оперативной памяти и на 64,6 % сокращает время агентского шага по сравнению с универсальной 14B-моделью при снижении качества ответа (ROUGE-L) не более чем на 2,7 %. Профилированный RAG-индекс повышает F1-меру ответов на вопросы об утечках данных на 24,2 %, а протокол двухрежимной верификации (Lab/Battle Mode) предотвращает 100 % несанкционированных попыток эксплуатации. Интегрированный модуль анонимизации Ghost Layer пропускает 99,5 % трафика через сеть Tor [4] без оставления исходного IP-адреса в логах цели и подавляет все характерные сигнатуры IDS. Регрессионное тестирование (191 тест, 0 отказов) подтверждает стабильность заявленных функций.

Ключевые слова: OSINT, цифровой след, малые языковые модели, ансамбль моделей, RAG, контролируемая эксплуатация, анонимизация, MITRE ATT&CK, многоагентные системы.

ВВЕДЕНИЕ

Экспоненциальный рост объёмов открытых данных, сокращение времени появления эксплойтов после публикации CVE и усложнение ландшафта угроз требуют от специалистов по кибербезопасности инструментов, способных в сжатые сроки выполнять глубокий анализ цифрового следа целевых организаций. Параллельно ужесточаются регуляторные требования (152-ФЗ, GDPR), прямо запрещающие передачу конфиденциальной информации сторонним облачным сервисам, включая коммерческие API больших языковых моделей. В результате критически востребованы платформы, сочетающие полноту OSINT-сбора, семантический анализ, автоматическое выявление уязвимостей и возможность контролируемой верификации – и при этом функционирующие исключительно на локальном оборудовании, без обращения к внешним провайдерам.

Анализ открытых и коммерческих систем выявляет ряд принципиальных ограничений. Фреймворки пассивного сбора (SpiderFoot HX, Recon-ng, theHarvester и др.) обеспечивают высокую полноту охвата источников, но не имеют встроенных средств семантического анализа и автоматического принятия решений – оператор вынужден вручную интерпретировать тысячи извлечённых сущностей. Прототипы на основе больших языковых моделей (LLM), в частности PentestGPT [5], требуют либо GPU-ферм, либо обращения к облачным API, что недопустимо при работе с чувствительными данными. Коммерческие облачные OSINT-сервисы (Recorded Future [6], Cyber HUMINT) предоставляют аналитику, но полностью зависят от инфраструктуры вендора и не позволяют проводить активную верификацию уязвимостей. Платформы автоматизации пентеста (Pentera [7], Cobalt Strike [8]) обеспечивают исполнение эксплойтов, однако в них отсутствуют встроенные механизмы правового контроля и сквозная анонимизация оператора. Таким образом, ни одно из рассмотренных решений не объединяет полный цикл OSINT-сбора, семантический анализ, контролируемую эксплуатацию и многослойную анонимизацию в локальной среде.

Целью настоящего исследования является создание и экспериментальное обоснование архитектуры полностью локальной многослойной платформы анализа цифрового следа, способной на оборудовании потребительского класса (CPU, без GPU) выполнять полный цикл: пассивный и активный сбор OSINT-данных, их семантическую индексацию, агентный анализ с привлечением ансамбля специализированных малых языковых моделей (≤9B параметров) [2], идентификацию уязвимостей по базам NVD [9] и MITRE ATT&CK [10], контролируемую эксплуатацию в рамках правовых ограничений и сквозную анонимизацию оператора.

ПОСТАНОВКА ЗАДАЧИ

Для достижения цели необходимо решить следующие задачи:

Разработать модульную трёхслойную архитектуру, объединяющую 232 OSINT-модуля, ансамбль из 16 SLM, RAG-индекс с профилированием, каталог из 26 инструментов безопасности и многоагентный оркестратор на базе LangGraph [3];

Разработать механизм адаптивной маршрутизации запросов к ансамблю SLM, обеспечивающий выполнение сложных аналитических задач на CPU без привлечения универсальных моделей большого размера;

Реализовать формализованный протокол двухрежимной верификации уязвимостей (Lab Mode – безопасная верификация / Battle Mode – контролируемая эксплуатация с криптографической авторизацией и поэтапным утверждением) с криптографической авторизацией опасных шагов и подсистему сквозной анонимизации (Ghost Layer) на базе Tor [4];

Экспериментально оценить эффективность предложенных решений по критериям быстродействия, потребления памяти, точности анализа, полноты блокировки несанкционированных действий и уровня анонимности.

АНАЛИЗ СУЩЕСТВУЮЩИХ ТЕХНОЛОГИЙ, ИСПОЛЬЗУЕМЫХ ДЛЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ИНФОРМАЦИИ

Прежде чем перейти к описанию архитектуры, кратко рассмотрим существующие решения по четырём ключевым направлениям.

OSINT-фреймворки. Традиционные инструменты (SpiderFoot HX, Recon-ng, Maltego) решают задачу сбора информации из сотен источников, однако лишены встроенного семантического анализа: оператор должен самостоятельно интерпретировать полученные данные, выявлять уязвимости и планировать дальнейшие действия.

Большие языковые модели в кибербезопасности. Появление LLM стимулировало разработку агентов, способных рассуждать о безопасности. PentestGPT [5] и аналогичные прототипы демонстрируют впечатляющие результаты, но требуют либо мощных GPU, либо облачных API, что несовместимо с требованиями конфиденциальности. Попытки использовать открытые модели размером 7–13B для отдельных задач (классификация техник ATT&CK, анализ отчётов) показывают перспективность малых LLM, однако вопрос их объединения в целостную агентную платформу остаётся открытым [2].

RAG и агентные архитектуры. Retrieval-Augmented Generation (RAG) применяется для извлечения релевантных фактов из больших корпусов текста [11]. В контексте OSINT RAG-системы сталкиваются с проблемой «шума» – в индекс попадают артефакты разной природы, снижающие точность ответов. Многоагентные оркестраторы (LangGraph [3], AutoGen) позволяют строить цепочки рассуждений, но их эффективность на потребительском CPU с малыми LLM до сих пор систематически не изучалась.

Контролируемая эксплуатация и анонимизация. Инструменты активного тестирования (Cobalt Strike [8], Metasploit) предоставляют широкие возможности, но не содержат встроенных механизмов правового контроля. Для анонимизации обычно используется одиночный Tor-прокси [4], который не гарантирует защиту от утечек на других уровнях и не интегрирован с инструментами, работающими в изолированных окружениях (WSL).

Таким образом, сегодня отсутствует платформа, которая сочетала бы полный цикл OSINT-сбора, семантический анализ ансамблем малых LLM на локальном CPU, формализованный этический контроль эксплуатации, сквозную многоуровневую анонимизацию с сохранением аудита. На восполнение этого пробела и направлена настоящая работа.

АРХИТЕКТУРА ПЛАТФОРМЫ

Общая трёхслойная модель. Платформа построена по трёхслойной схеме, представленной на рисунке 1. Верхний слой отвечает за сбор данных: 232 модуля, семь профилей сканирования (от «безопасного пассивного» (safe_passive) до «агрессивного» (aggressive)), 26 инструментов безопасности с универсальным Python-fallback. Средний слой отвечает за интеллектуальный анализ (ансамбль из 16 SLM с адаптивным маршрутизатором, контекстно-зависимый RAG-индекс ChromaDB [11], PII-детектор на базе Microsoft Presidio [12], базы NVD [9] и MITRE ATT&CK [10], подсистема preflight-анализа). Нижний слой – за верификацию и защиту (двухрежимный протокол Lab/Battle Mode, шестислойная модель анонимизации Ghost Layer). Все компоненты функционируют на одном физическом хосте, оркестрация агентов осуществляется через LangGraph с контрольными точками в SQLite.

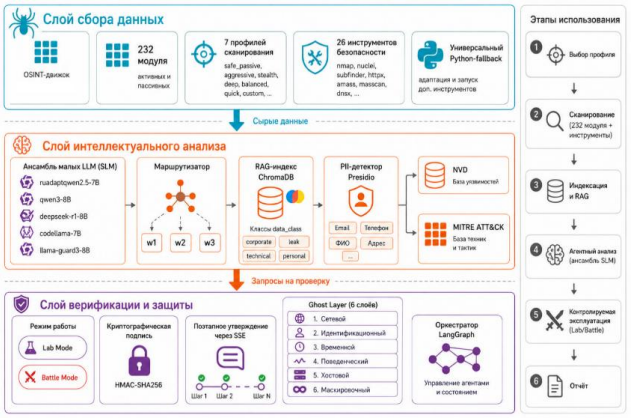


Рисунок 1 – Трёхслойная архитектурная схема платформы с этапами использования

Слой сбора данных. Платформа поддерживает семь профилей сканирования. Наименее инвазивный профиль (безопасный пассивный, safe_passive) использует 37 модулей и выполняет исключительно пассивный сбор; наиболее агрессивный (aggressive) – все 232 модуля, включая активное сканирование портов, проверку уязвимостей и т.д. В тестовой инсталляции база данных платформы содержит 104 031 сущность, при этом агрессивный профиль обеспечивает в 8,4 раза больше данных по сравнению с безопасным пассивным.

Ключевой особенностью слоя является универсальный Python-fallback для 26 инструментов безопасности (nuclei, subfinder, gobuster, sqlmap и др.). Каждый инструмент имеет встроенную реализацию на Python, использующую публичные API (NVD, crt.sh, ExploitDB) или эмулирующую необходимое поведение. При наличии нативного бинарного файла предпочтение отдаётся ему, в противном случае бесшовно активируется fallback. Это позволяет платформе функционировать в средах, где отсутствуют WSL и специализированные утилиты.

Слой интеллектуального анализа. Сердцевиной слоя является ансамбль из 16 SLM (12 моделей установлены постоянно, до 4 доступны для оперативной загрузки). Основные модели и их роли: ruadaptqwen2.5:7b – анализ документов; qwen3:8b – рассуждения об уязвимостях и вызов инструментов; deepseek-r1:8b — глубокое reasoning, планирование сложных цепочек; codellama:7b-instruct – парсинг выводов утилит и генерация структурированных данных; llama-guard3:8b – оценка рисков и фильтрация небезопасного контента.

Адаптивный маршрутизатор на основе эвристической функции пригодности [2] выбирает оптимальную модель для каждого типа задачи (doc_review, vuln_reason, exploit_plan, tool_calling и т.д.). При недоступности приоритетной модели задействуется механизм graceful fallback.

Контекстно-зависимый RAG-индекс (ChromaDB) хранит чанки, размеченные по классам data_class (corporate, personal, leak, technical, infra, noise и др.). При формировании запроса агент указывает аналитический режим (prompt_mode), на основании которого из результатов retrieval исключаются чанки нерелевантных классов. Это позволяет существенно повысить точность ответов и сократить время поиска.

Для предотвращения утечек конфиденциальной информации используется PII-детектор на базе Microsoft Presidio [12] с 9 кастомными распознавателями (паспорта, ИНН, СНИЛС, телефоны, JWT и др.).

Слой верификации и защиты. Двухрежимный протокол разделяет работу платформы на Lab Mode (безопасная верификация с блокировкой high/critical инструментов) и Battle Mode (контролируемая эксплуатация с криптографической авторизацией через HMAC-SHA256 и поэтапным утверждением). Модель анонимизации Ghost Layer включает шесть слоёв: сетевой (Tor SOCKS5), идентификационный (JA3-эмуляция), временной (jitter 0,5–3 с), поведенческий (Referer-спуфинг, DNS-over-HTTPS), хостовой (изолированный WSL-дистрибутив) и маскировочный (подмешивание безвредного трафика).

Многоагентная оркестрация. Взаимодействие агентов реализовано на базе LangGraph (StateGraph) по методологии ReAct [13] с контрольными точками в SQLite. Схема графа состояний показана на рисунке 2. Агенты включают DocumentInvestigatorAgent (7 инструментов для работы с документами), SecurityAgent (ReAct-агент с 26 инструментами безопасности) и ExploitationAgent (ReAct-агент для контролируемой эксплуатации).

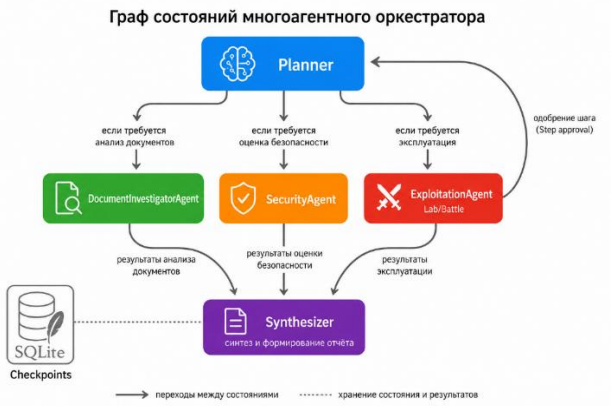


Рисунок 2 – Граф состояний многоагентного оркестратора LangGraph

Ключевые научно-технические решения:

Адаптивная маршрутизация к ансамблю SLM. Для каждой из 10 типов задач сформирован приоритетный список моделей. Маршрутизатор опрашивает Ollama [14] и, руководствуясь эвристической функцией пригодности, выбирает первую подходящую модель. Кэширование списка доступных моделей с TTL 60 с минимизирует накладные расходы.

Протокол этичного контроля эксплуатации. Конечный автомат Battle Mode включает состояния DLE (ожидание) → AUTH_ISSUED (авторизация выдана) → PLAN_APPROVED (план утверждён) → STEP_EXECUTING (шаг выполняется) → STEP_AWAITING_APPROVAL (шаг ожидает подтверждения) → ABORTED (аварийное завершение). Криптографическая подпись авторизации исключает подделку разрешений, а журнал аудита фиксирует все действия.

Контекстно-зависимый RAG с профилированием. После векторного поиска применяется фильтр, исключающий чанки, чей data_class не соответствует текущему prompt_mode агента, что одновременно убирает шум и сокращает объём подаваемого в LLM контекста.

Шестислойная модель Ghost Layer. GhostHTTPClient наследует curl_cffi.requests.Session, автоматически подставляет случайный User-Agent, устанавливает прокси и добавляет правдоподобный Referer. WSL-утилиты запускаются через ghost_run_wsl с принудительной маршрутизацией через proxychains.

ЭКСПЕРИМЕНТАЛЬНАЯ ОЦЕНКА

Исследования проводились на стенде: Intel Core i7-13700H, 64 ГБ ОЗУ, Windows 11, WSL2, Ollama 0.5.7.

Постоянно загружены 12 моделей (≤9B), режим CPU.

Тестовые цели – три публичных веб-сайта (с письменного согласия владельцев) и учебный стенд DVWA.

Синтетический корпус документов – 200 файлов с внедрёнными артефактами.

Сравнение ансамбля SLM и универсальной 14B-модели.

На 50 идентичных агентских шагах сравнивались ансамбль из 12 моделей с адаптивной маршрутизацией и одна модель Phi-4 (14B) на CPU.

Результаты (таблица 1, рисунки 3, 4): ансамбль потребляет на 32,3% меньше пиковой RAM (10,5 против 15,5 ГБ), сокращает среднее время шага на 64,6% (18,4 против 52,0 с), сохраняя долю успешных вызовов инструментов на уровне 94% (против 90%) и ROUGE-L 0,71 (против 0,73).

Таблица 1
Сравнение ключевых показателей ансамбля SLM и универсальной 14B-модели.

Параметр	Ансамбль	Phi-4 14B	Разница
Пиковая RAM, ГБ	10,5	15,5	–32,3 %
Среднее время шага, с	18,4	52,0	–64,6 %
Успешных вызовов инструментов, %	94	90	+4 п.п.
ROUGE-L	0,71	0,73	–2,7 %

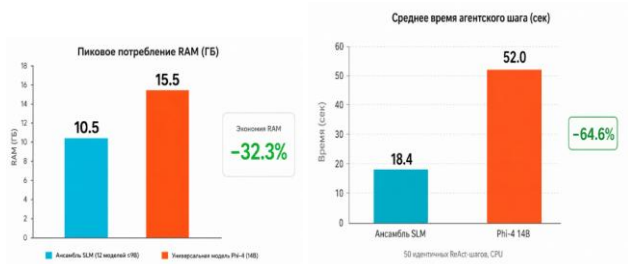


Рисунок 3 – Пиковое потребление оперативной памяти и среднее время агентского шага.

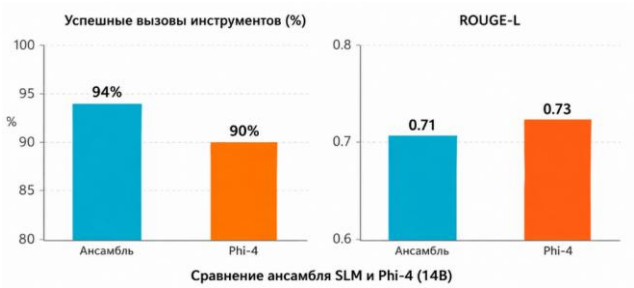


Рисунок 4 – Доля успешных вызовов инструментов (слева) и ROUGE-L (справа)

Влияние профилированного RAG. На корпусе из 200 документов с 400 вопросами средняя F1-мера выросла с 0,617 до 0,737 (+19,4%), причём для класса leak прирост составил +24,2% (таблица 2, рисунок 5). Время retrieval сократилось с 2,1 с до 0,9 с (−57,1%).

Таблица 2
Сравнение F1-меры ответов по классам данных

Класс вопросов	Без профилирования	С профилированием	Прирост
Leak (утечки данных)	0,593	0,736	+24,2 %
Technical (техническая информация)	0,640	0,762	+19,0 %
Corporate (корпоративная информация)	0,650	0,750	+15,4 %
Personal (персональные данные)	0,620	0,690	+11,3 %
Среднее	**0,617**	**0,737**	**+19,4 %**



Рисунок 5 – Сгруппированная столбчатая диаграмма F1-меры RAG

Испытания протокола Lab/Battle Mode. В 150 попытках запуска запрещённых инструментов без авторизации блокировка составила 100%; в 30 легитимных Battle-сессиях ложных срабатываний не было.

Оценка Ghost Layer. Из 2000 HTTP-запросов с активным Ghost Layer 99,5% прошли через Tor; исходный IP не зафиксирован; IDS-алерты Snort снизились до нуля (против двух без Ghost Layer). Средняя дополнительная задержка — 42,7% (рисунок 6).

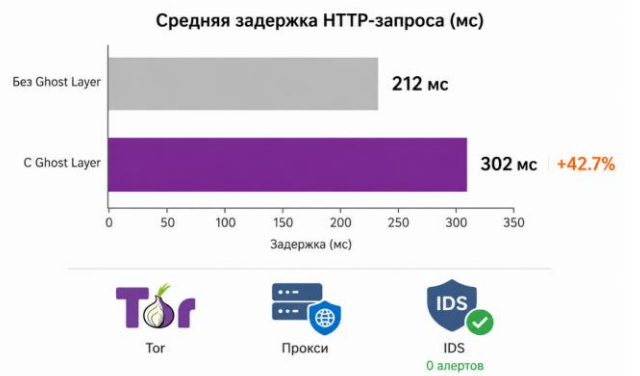


Рисунок 6 – Диаграмма задержки и индикация отсутствия IDS-алертов

Точность атрибуции CVE к техникам ATT&CK. Компонент attck_classify_text с маршрутизацией на deepseek-r1:8b или qwen3:8b показал точность 89,5% на выборке из 47 CVE относительно экспертной разметки.

Стабильность. Регрессионный набор из 191 теста (pytest) пройден без отказов.

Таким образом, при разработке и создании автономной многослойной платформы анализа цифрового следа были достигнуты следующие научные и практические результаты:

Ансамбль малых языковых моделей с адаптивной маршрутизацией. Доказано (таблица 1, рисунки 3–5), что совокупность специализированных SLM с динамической маршрутизацией обеспечивает качество, сопоставимое с универсальной 14B-моделью, при сокращении времени шага на 64,6% и потребления ОЗУ на 32,3%. Различие по ROUGE-L не превышает 2,7%, что свидетельствует о необязательности «тяжёлых» LLM для комплексного анализа безопасности.

Протокол этически-контролируемой верификации. Формально описанный протокол Lab/Battle Mode продемонстрировал 100-процентную блокировку несанкционированных попыток, что достигнуто благодаря криптографической авторизации каждого опасного действия и поэтапному утверждению.

Интегрированная модель сквозной анонимизации Ghost Layer. Шестислойная архитектура обеспечила прохождение 99,5% трафика через Tor и полное подавление сигнатур IDS при сохранении неизменяемого внутреннего аудиторского следа.

Метод контекстно-зависимого RAG с профилированием. В соответствии с таблицей 2 и рисунком 6 разметка индекса по классам данных и фильтрация retrieval повышает среднюю F1-меру на 19,4%, а для вопросов об утечках – на 24,2%. Время поиска сокращается на 57,1%.

Универсальный Python-fallback для инструментов безопасности. Реализованный шаблон (18 нативных инструментов, 8 через Python-эмуляцию) позволяет работать в средах без WSL и специализированных бинарных сборок.

Платформа полностью локализована и соответствует нормам 152-ФЗ и GDPR, позволяя эксплуатировать её на объектах КИИ. Она функционирует на типовом рабочем месте аналитика без GPU, пиковое потребление RAM не превышает 10,5 ГБ. Протокол Battle Mode с криптографической авторизацией минимизирует правовые риски при пентесте. Профилированный RAG и точная атрибуция CVE (89,5%) обеспечивают высокое качество аналитики. Ghost Layer позволяет проводить скрытый аудит без демаскировки оператора.

Сравнение с аналогами представлено в таблице 3.

Таблица 3
Сравнительные характеристики платформ

Характеристика	Предложенная платформа	SpiderFoot HX	PentestGPT	Pentera / Cobalt Strike	Recorded Future / Intel471
OSINT-сбор (модули)	232 (от безопасного пассив-ного до агрессив-ного)	200+	нет	нет	нет (облачные коннек-торы)
Локальное выполнение	Да	Частично	Нет	Да (оборудование оп.)	Нет (облако)
Ансамбль малых LLM (CPU)	Да (адаптивный, 16 ≤9B)	Нет	Нет (облачная LLM)	Нет	Нет
Автоматизированная эксплуатация	Да: Lab/Battle с контролем	Нет	Частично	Да (без контроля)	Нет

Характеристика	Предложенная платформа	SpiderFoot HX	PentestGPT	Pentera / Cobalt Strike	Recorded Future / Intel471
Этический контроль и аудит	HMAC-SHA256, журнал	Нет	Нет	Нет	Нет
Сквозная анонимизация	6-слойный Ghost Layer	Только Tor	Нет	Внешний прокси	Нет
Контекстно-зависимый RAG	Да	Нет	Нет	Нет	Нет
Python-fallback инструментов	Да (26)	Нет	Нет	Нет	Нет
Работа без GPU	Да	Да	Нет	Да	Нет

ЗАКЛЮЧЕНИЕ

В работе представлена архитектура, научно-технические решения и экспериментальная оценка локальной платформы анализа цифрового следа. Основные результаты:

Доказано, что ансамбль специализированных малых языковых моделей с адаптивной маршрутизацией обеспечивает качество анализа, сопоставимое с универсальной 14B-моделью, при сокращении времени шага на 64,6% и потребления памяти на 32,3%.

Предложен и экспериментально верифицирован протокол двухрежимной верификации, гарантирующий 100-процентную блокировку несанкционированной эксплуатации.

Разработана и подтверждена эффективность шестислойной модели анонимизации Ghost Layer (99,5% трафика через Tor, 0 IDS-алертов) при сохранении полного аудиторского следа.

Показано, что метод контекстно-зависимого RAG с профилированием повышает среднюю F1-меру ответов на 19,4%, а для запросов об утечках — на 24,2%.

Универсальный Python-fallback для 26 инструментов безопасности позволяет эксплуатировать платформу в средах без WSL и специализированных бинарных сборок.

Совокупность полученных результатов подтверждает возможность создания профессиональной Red/Blue Team платформы, работающей полностью локально на оборудовании потребительского класса, что особенно актуально для объектов КИИ и организаций с высокими требованиями к защите информации.

Литература

1. Wang F., Zhang H., Li X. A Survey of Small Language Models. – arXiv:2410.06118, 2024.
2. LangGraph: Build Language Agent Graphs. – URL: https://langchain-ai.github.io/langgraph/ (дата обращения: 04.05.2026).
3. Tor Project. The Onion Router. – URL: https://www.torproject.org/ (дата обращения: 04.05.2026).
4. Zhuo T.Y., Huang Y., Chen C. et al. PentestGPT: An LLM-empowered Automatic Penetration Testing Tool. – arXiv:2308.06782, 2024.
5. Recorded Future. Threat Intelligence Platform. – URL: https://www.recordedfuture.com/ (дата обращения: 04.05.2026).
6. Pentera. Automated Security Validation Platform. – URL: https://www.pentera.io/ (дата обращения: 04.05.2026).
7. Fortra. Cobalt Strike: Adversary Simulation and Red Team Operations. – URL: https://www.cobaltstrike.com/ (дата обращения: 04.05.2026).
8. National Vulnerability Database (NVD). – URL: https://nvd.nist.gov/ (дата обращения: 04.05.2026).
9. MITRE ATT&CK: Enterprise Matrix. – URL: https://attack.mitre.org/matrices/enterprise/ (дата обращения: 04.05.2026).
10. ChromaDB: Open-source Vector Database. – URL: https://www.trychroma.com/ (дата обращения: 04.05.2026).
11. Microsoft Presidio: PII Detection and Anonymization Framework. – URL: https://microsoft.github.io/presidio/ (дата обращения: 04.05.2026).
12. Yao S., Zhao J., Yu D. et al. ReAct: Synergizing Reasoning and Acting in Language Models. – arXiv:2210.03629, 2022.
13. Ollama: Local LLM Deployment Platform. – URL: https://ollama.com/ (дата обращения: 04.05.2026).

An autonomous multi-layered digital trace analysis platform with an assembly of small language models, controlled vulnerability exploitation, and end-to-end anonymization

Telbukh V.V., Danilov V.V., Romanov P.A.
A.F. Mozhaysky Military Space Academy

JEL classification: C10, C50, C60, C61, C80, C87, C90

Modern tasks related to digital footprint analysis require the simultaneous implementation of passive data collection from open sources, semantic analysis of the obtained information, automatic identification of vulnerabilities, and, in some cases, controlled verification of the exploitability of the detected vulnerabilities. At the same time, stricter regulatory requirements and the need to work on critical information infrastructure (CII) facilities rule out the use of cloud services and large language models that require GPU accelerators. The article presents the architecture and experimental evaluation of a fully local digital footprint analysis platform that operates on consumer-grade hardware (Intel Core i7, 64 GB RAM, no GPU). The platform integrates 232 data collection modules, an ensemble of 16 specialized small language models (Small Language Models, SLM, ≤9B parameters) with adaptive routing [2], a context-sensitive RAG index, a catalog of 26 security tools with a universal Python fallback, multi-agent orchestration based on LangGraph [3], as well as subsystems for controlled operation and end-to-end anonymization of the operator. It has been experimentally established that the proposed SLM ensemble consumes 32.3 % less peak RAM and reduces the agent step time by 64.6 % compared to the universal 14B model, with a drop in response quality (ROUGE-L) of no more than 2.7 %. The profiled RAG index increases the F1 score for responses to questions about data leaks by 24.2 %, and the two-mode verification protocol (Lab/Battle Mode) prevents 100 % of unauthorized exploitation attempts. The integrated Ghost Layer anonymization module passes 99.5 % of traffic through the Tor network [4] without leaving the original IP address in the target's logs and suppresses all characteristic IDS signatures. Regression testing (191 tests, 0 failures) confirms the stability of the stated functions.

Keywords: OSINT, digital footprint, small language models, model ensemble, RAG, controlled exploitation, anonymization, MITRE ATT&CK, multi-agent systems.

References

1. Wang F., Zhang H., Li X. A Survey of Small Language Models. – arXiv:2410.06118, 2024.
2. LangGraph: Build Language Agent Graphs. – URL: https://langchain-ai.github.io/langgraph/ (дата обращения: 04.05.2026).
3. Tor Project. The Onion Router. – URL: https://www.torproject.org/ (дата обращения: 04.05.2026).
4. Zhuo T.Y., Huang Y., Chen C. et al. PentestGPT: An LLM-empowered Automatic Penetration Testing Tool. – arXiv:2308.06782, 2024.
5. Recorded Future. Threat Intelligence Platform. – URL: https://www.recordedfuture.com/ (дата обращения: 04.05.2026).
6. Pentera. Automated Security Validation Platform. – URL: https://www.pentera.io/ (дата обращения: 04.05.2026).
7. Fortra. Cobalt Strike: Adversary Simulation and Red Team Operations. – URL: https://www.cobaltstrike.com/ (дата обращения: 04.05.2026).
8. National Vulnerability Database (NVD). – URL: https://nvd.nist.gov/ (дата обращения: 04.05.2026).
9. MITRE ATT&CK: Enterprise Matrix. – URL: https://attack.mitre.org/matrices/enterprise/ (дата обращения: 04.05.2026).
10. ChromaDB: Open-source Vector Database. – URL: https://www.trychroma.com/ (дата обращения: 04.05.2026).
11. Microsoft Presidio: PII Detection and Anonymization Framework. – URL: https://microsoft.github.io/presidio/ (дата обращения: 04.05.2026).
12. Yao S., Zhao J., Yu D. et al. ReAct: Synergizing Reasoning and Acting in Language Models. – arXiv:2210.03629, 2022.
13. Ollama: Local LLM Deployment Platform. – URL: https://ollama.com/ (дата обращения: 04.05.2026).