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

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

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

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

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

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

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

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

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

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

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

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

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

Введение

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

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

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

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

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

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

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

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

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

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

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

2. Анализ недекларируемых возможностей ОС Astra Linux

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

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

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

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

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

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

Проверка отсутствия НДВ включает в себя:

– анализ исходных кодов ОС Astra Linux;

– документирование всех программных интерфейсов;

– подтверждение отсутствия перечисленной выше функциональности.

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

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

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

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

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

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

3. Особенности динамического тестирования безопасности веб-приложений

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

Таблица 1

Сравнение динамических и статических подходов

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

Окончание табл.1

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

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

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

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

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

4. Методы оценки защищенности ИТ-инфраструктуры

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

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

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

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

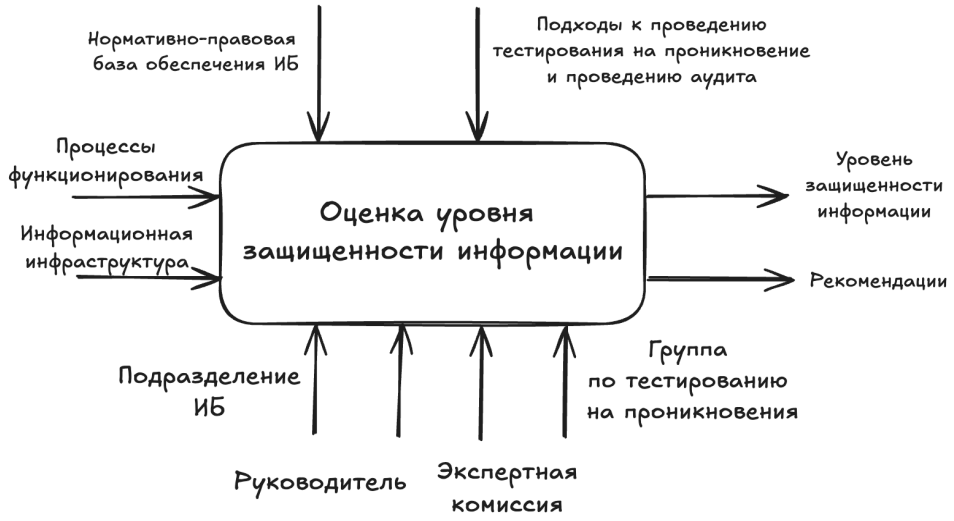


Рис. 1. Структурная схеме процесса оценки уровня защищенности информации

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

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

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

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

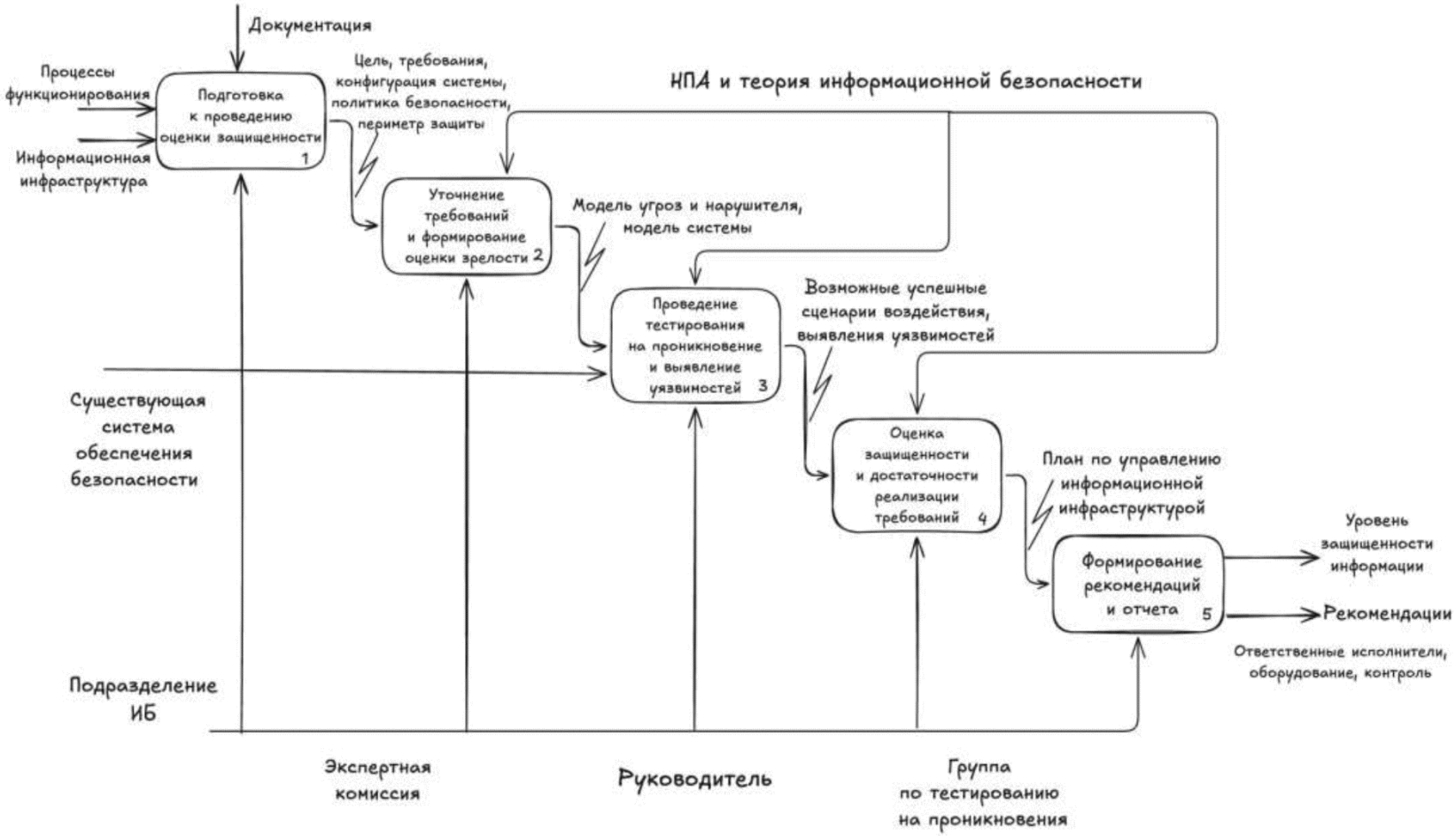


Рис. 2. Структурная схема процесса оценки уровня защищенности информации

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

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

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

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

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

Заключение

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

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

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

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

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

Литература

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

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

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

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

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

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

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

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

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

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

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

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

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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