УДК 004.056

Баженов Михаил Сергеевич магистрант
2 курс, СПбГУТ им. проф. М. А. Бонч-Бруевича
Россия, г. Санкт-Петербург
Алексеева Ксения Евгеньевна магистрант
2 курс, СПбГУТ им. проф. М. А. Бонч-Бруевича
Россия, г. Санкт-Петербург

РАЗРАБОТКА МЕТОДИКИ ПЕРВИЧНОГО АУДИТА ЗАЩИЩЕННОСТИ ASTRA LINUX БЕЗ ИЗМЕНЕНИЯ СОСТОЯНИЯ ПРОВЕРЯЕМОЙ СИСТЕМЫ

Аннотация: *Рассматривается задача быстрой проверки фактической настройки узлов операционной системы Astra Linux. Предложена методика первичного аудита, при которой программный агент работает в режиме чтения и не вносит изменений в проверяемый узел. Сформулировано рабочее определение уязвимого конфигурационного состояния, перечислены группы собираемых диагностических признаков и описан их перенос в структурированный документ JSON. Каталог проверок разбит по областям контроля, для отдельных проверок показан способ обоснования критичности. Отмечено, что подход не заменяет сертификационную оценку, но дает воспроизводимый и понятный администратору срез конфигурационного риска, в котором каждое замечание сопровождается источником логики и рекомендацией по устранению.*

Ключевые слова: *Astra Linux, защита информации, первичный аудит, конфигурация, режим чтения, диагностические признаки, каталог проверок*

Annotation: *The problem of quickly checking the actual configuration of Astra Linux hosts is considered. A primary audit methodology is proposed in which the*

software agent operates in read mode and makes no changes to the audited host. A working definition of a vulnerable configuration state is given, the groups of collected diagnostic features are listed, and their transfer into a structured JSON document is described. The set of checks is divided by control areas, and a way to justify the criticality of individual checks is shown. The approach does not replace a certification assessment but provides a reproducible and administrator-friendly slice of configuration risk, in which every finding is accompanied by a logic source and a remediation recommendation.

Key words: *Astra Linux, information protection, primary audit, configuration, read mode, diagnostic features, check catalog*

Введение

Расширение применения российских операционных систем в защищенных сегментах инфраструктуры повышает значимость регулярного контроля их фактической настройки. Astra Linux востребована там, где предъявляются повышенные требования к доверенной программной базе, регламентному администрированию и регистрации событий [1]. Практика показывает, что наличие сертифицированной платформы не означает безопасной настройки отдельного узла: ослабленные параметры удаленного входа, забытые служебные учетные записи, открытые наружу порты, давно не устанавливавшиеся обновления и пробелы в журналировании заметно повышают итоговый риск.

Дополнительная трудность связана с тем, что в повседневной работе администратора опасность далеко не всегда сводится к зарегистрированной записи об известной уязвимости. Значимыми оказываются и сами эксплуатационные настройки, отклоняющиеся от ожидаемой роли узла. Отсюда вытекает потребность в инструменте, который быстро и прозрачно фиксирует такие отклонения, оставаясь при этом безопасным для работающей системы.

Постановка задачи

Цель работы заключается в построении методики первичной проверки настройки Astra Linux, отвечающей нескольким условиям. Во-первых, безопасность применения: сбор данных не должен затрагивать конфигурацию, перезапускать сервисы или иным образом воздействовать на узел. Во-вторых, прозрачность вывода: для каждого зафиксированного отклонения должно быть ясно проверяемое условие, причина его опасности и предлагаемое действие. В-третьих, повторяемость: при неизменных входных данных результат проверки не должен меняться. В-четвертых, учет особенностей платформы: штатные сервисы и официальные репозитории Astra Linux не должны помечаться как нарушение автоматически.

Под уязвимым конфигурационным состоянием здесь понимается зафиксированный набор настроек и эксплуатационных характеристик, которые увеличивают шанс успешной атаки либо затрудняют сопровождение. Такая трактовка охватывает больше, чем перечень известных уязвимостей, но при этом ограничена практической связью с риском. Методические документы по оценке угроз предписывают учитывать не только сам факт уязвимости, но и объект воздействия, последствия и реализуемость угрозы [2], поэтому проверка строится как оценка риска, а не как формальный ответ о наличии признака.

Состав признаков и порядок обработки

Исходные сведения узла представлены документом JSON, который собирает агент в режиме чтения. В него входит паспорт узла, а также тематические блоки. Среди них параметры службы удаленного доступа, локальные учетные записи и права повышения полномочий, владельцы и режимы критичных файлов, перечень слушающих портов, активные и автозапускаемые сервисы, сведения о пакетах и репозиториях и журнальные показатели.

Параметры удаленного входа выделены отдельно, поскольку именно этот канал чаще всего становится точкой проникновения на серверный узел [3].

Считывание выполняется через безопасное чтение системных файлов и набор диагностических команд. Если какая-то команда недоступна, агент не останавливается, а помечает пропуск предупреждением, которое позже влияет на оценку полноты результата. Порядок обработки последователен. Сначала агент формирует документ узла, затем выбирается профиль роли, далее применяется каталог проверок, после чего вычисляется риск и готовится отчет с пояснениями.

Перечень собираемых характеристик сознательно сужен до того, что можно получить без воздействия на систему. Агент не подбирает пароли и не обращается к сетевым службам извне, а опирается только на уже доступную информацию о настройке. Для первичной проверки этого достаточно, потому что задача состоит в быстром выявлении типовых административных ошибок, а не в демонстрации эксплуатации.

Каталог проверок и обоснование критичности

Проверки сгруппированы по смысловым областям, что упрощает чтение отчета: видно не только общее значение риска, но и направление, в котором накопились замечания. Сводка областей приведена в таблице 1.

Таблица 1.

Области контроля каталога первичного аудита

Область	Что охватывает	Типичные проверяемые характеристики
Удаленный доступ	настройки службы входа	вход суперпользователя, парольный вход, пустые пароли
Учетные записи	пользователи и полномочия	второй идентификатор 0, состав sudo, лишние интерактивные пользователи
Файлы	критичные системные файлы	владелец и режим shadow, passwd, sudoers

Сеть и сервисы	поверхность и состав ПО	незащищенные службы, порты и сервисы вне профиля
Пакеты и журналы	сопровождение и наблюдаемость	задержка обновлений, неуспешные входы, сбойные модули

Каждая запись каталога снабжена идентификатором, уровнем критичности, весом, источником логики, доказательством и рекомендацией. Подход к назначению уровня поясним на примерах. Разрешенный вход суперпользователя по защищенному каналу повышает цену компрометации пароля и относится к высокому уровню. Появление второй учетной записи с нулевым идентификатором равнозначно второму администратору и трактуется как критическое. Интерактивный пользователь, не внесенный в список разрешенных, расценивается как замечание среднего уровня, поскольку увеличивает число точек входа без немедленной угрозы.

Принципиальной чертой является адаптация к платформе. Штатные сервисы регистрации и диагностики, а также официальный расширенный репозиторий не считаются нарушением сами по себе; отклонением признается несоблюдение утвержденного профиля роли либо локальной политики сопровождения [1]. Для этого вводятся профили ролей с допустимыми портами, сервисами, пользователями и источниками пакетов, что согласуется с практиками управления конфигурацией признанных наборов мер [4; 5].

Результаты и обсуждение

Методика обеспечивает три практически важных свойства. Повторяемость достигается тем, что один и тот же документ узла при одном и том же каталоге дает одинаковый вывод. Прозрачность обеспечивается тем, что каждое замечание привязано к проверке, ее источнику, доказательству и рекомендации. Предметность достигается ролевыми профилями, отделяющими штатные

компоненты от настоящих отклонений и снижающими долю ложных срабатываний.

Назначение подхода ограничено первичной оценкой. Он не претендует на роль сертификационной проверки или промышленного сканера, а дает быстрый и объяснимый снимок конфигурационного риска, который удобно повторять и обсуждать. Слабым местом остается зависимость от полноты собранных данных и от точности профилей ролей, что снимается их сопровождением и калибровкой на накопленных результатах.

Заключение

Предложена методика первичного аудита Astra Linux, работающая в режиме чтения и не изменяющая проверяемый узел. Определено рабочее понятие уязвимого конфигурационного состояния, описан состав собираемых характеристик, предложена разбивка проверок по областям контроля и показан способ обоснования их критичности с учетом особенностей платформы. Подход пригоден для регулярного внутреннего контроля и учебных стендов; направления развития связаны с пополнением каталога и уточнением профилей специализированных узлов.

Использованные источники:

1. Операционная система Astra Linux Special Edition. Эксплуатационная документация // Справочный центр Astra Linux. URL: https://wiki.astralinux.ru/ (дата обращения: 21.06.2026).

2. Методика оценки угроз безопасности информации (утв. ФСТЭК России 05.02.2021) [Электронный ресурс]. URL: https://www.consultant.ru/document/cons_doc_LAW_378330/ (дата обращения: 21.06.2026).

3. Linux man-pages: sshd_config(5) [Электронный ресурс]. URL: https://man7.org/linux/man-pages/man5/sshd_config.5.html (дата обращения: 21.06.2026).

4. NIST SP 800-53 Rev. 5. Security and Privacy Controls for Information Systems and Organizations [Электронный ресурс]. URL: https://csrc.nist.gov/pubs/sp/800/53/r5/upd1/final (дата обращения: 21.06.2026).

5. CIS Critical Security Controls Version 8.1 [Электронный ресурс]. URL: https://www.cisecurity.org/controls/v8-1 (дата обращения: 21.06.2026).

6. Common Vulnerability Scoring System v3.1: Specification Document / FIRST [Электронный ресурс]. URL: https://www.first.org/cvss/v3.1/specification-document (дата обращения: 21.06.2026).

7. ГОСТ Р 56545-2015. Защита информации. Уязвимости информационных систем. Правила описания уязвимостей. М.: Стандартинформ, 2015. 16 с.

8. Linux man-pages: shadow(5) [Электронный ресурс]. URL: https://man7.org/linux/man-pages/man5/shadow.5.html (дата обращения: 21.06.2026).

9. Девянин П. Н. Результаты переработки уровней ролевого управления доступом и мандатного контроля целостности формальной модели управления доступом операционной системы Astra Linux // Труды ИСП РАН. 2023. Т. 35. Вып. 5. С. 7-22.

10. Банк данных угроз безопасности информации ФСТЭК России [Электронный ресурс]. URL: https://bdu.fstec.ru/ (дата обращения: 21.06.2026).

11. CISOfy. Lynis Documentation [Электронный ресурс]. URL: https://cisofy.com/documentation/lynis/ (дата обращения: 21.06.2026).