DOI: 10.15514/ISPRAS-2026-38(3)-4

# Подход к применению мандатного контроля целостности в ОС Astra Linux для управления доступом к пользовательским данным

*П.Н. Девянин, ORCID: 0000-0003-2561-794X <pdevyanin@astralinux.ru>*
*В.В. Горбатов, ORCID: 0009-0000-3891-9192 <vgorbatov@astralinux.ru>*
*А.А. Трубников, ORCID: 0009-0009-1932-7242 <atrubnikov@astralinux.ru>*

*ООО «РусБИТех-Астра»,*
*127254, г. Москва, Огородный проезд, д. 16/1с5.*

**Аннотация.** Мандатный контроль целостности (МКЦ) в операционной системе (ОС) Astra Linux как в ряде других ОС, например, семейства Microsoft Windows или IBM AIX, традиционно применяется для защиты системного высокоцелостного (доверенного) программного обеспечения от несанкционированной модификации или захвата управления со стороны пользовательского низкоцелостного (недоверенного) нарушителя. Для управления доступом к пользовательским данным, как правило, используется штатное для ОС семейства Linux дискреционное управление доступом. Оно не задает четких правил управления доступом, что часто приводит к ошибкам при администрировании и затрудняет научное обоснование безопасности. Более гибкое и развитое ролевое управление доступом потенциально могло бы решить эту задачу, но оно еще полнофункционально не реализовано в ОС Astra Linux. Изначально используемое в рассматриваемой ОС мандатное управление доступом (МРД) позволяет задавать учетным записям пользователей, субъектам (процессам) и сущностям (файлам, каталогам) неиерархические категории конфиденциальности, отражающие либо содержание данных, либо их принадлежность к структурным подразделениям организации, где используется ОС Astra Linux. Однако правила МРД накладывают жесткие ограничения на управление доступом, избыточные для случаев, когда не требуется предотвращение утечки конфиденциальных данных. В статье предлагается для защиты пользовательских данных рассмотреть возможность применения МКЦ, который также использует неиерархические категории. Если термин «целостность» дополнить «доверием», то уровни целостности (доверия) МКЦ хорошо соотносятся с должностными иерархиями организаций или многодоменными сетевыми структурами, поскольку правила МКЦ по умолчанию запрещают доступ на запись «снизу-вверх» (от менее целостного к более целостному), а при необходимости с применением специального флага сущностей SSI позволяют запретить и доступ на чтение «снизу-вверх». В качестве примера практической апробации предлагаемого подхода в статье рассматривается технология использования МКЦ для управления доступом к пользовательским данным в сетевых файловых системах Samba и NFS. В целом этот подход не требует существенной доработки МКЦ в ОС Astra Linux, а для научного обоснования его безопасности уже разработана мандатная сущностно-ролевая ДП-модель управления доступом и информационными потоками в ОС семейства Linux (МРОСЛ ДП-модель).

**Ключевые слова:** операционная система; мандатный контроль целостности; МРОСЛ ДП-модель; операционная система Astra Linux.

**Для цитирования:** Девянин П.Н., Горбатов В.В., Трубников А.А. Подход к применению мандатного контроля целостности в ОС Astra Linux для управления доступом к пользовательским данным. Труды ИСП РАН, том 38, вып. 3, часть 1, 2026, стр. 71–86. DOI: 10.15514/ISPRAS–2026–38(3)–4.





# Implementing Mandatory Integrity Control in Astra Linux OS for Access Control of User Data

*P.N. Devyanin, ORCID: 0000-0003-2561-794X <pdevyanin@astralinux.ru>*
*V.V. Gorbatov, ORCID: 0009-0000-3891-9192 <vgorbatov@astralinux.ru>*
*A.A. Trubnikov, ORCID: 0009-0009-1932-7242 <atrubnikov@astralinux.ru>*

*RusBITech-Astra*
*16/1b5, proyezd Ogorodny, Moscow, 127254, Russia.*

**Abstract.** Mandatory Integrity Control (MIC) in the Astra Linux OS, as in many other OS such as the Microsoft Windows family or IBM AIX, is traditionally applied to protect high-integrity (trusted) system software from unauthorized modification or takeover of control by low-integrity (untrusted) user adversaries. For access control to user data, the Discretionary Access Control (DAC) standard for Linux family OS is typically used. However, it does not establish clear rules for access control, which often leads to administrative errors and complicates the scientific justification of security. A more flexible and advanced Role-Based Access Control (RBAC) could potentially solve this problem, but it has not yet been fully implemented in the Astra Linux OS. The Multilevel security (MLS) originally employed in this OS under consideration enables the assignment of non-hierarchical confidentiality categories to user accounts, subjects (processes), and objects (files, directories), reflecting either the content of the data or their affiliation with structural units of the organization where the Astra Linux OS is used. However, MLS rules impose stringent restrictions on access control that are excessive in scenarios where preventing the leakage of confidential data is not a priority. In this context, the paper proposes considering the application of MIC for protecting user data, as it also utilizes non-hierarchical categories. By associating integrity levels with trust, MIC aligns with organizational hierarchies or multi-domain networks. It enforces the no-write-up rule, preventing subjects from modifying higher-integrity objects, and optionally, via the SSI flag, the no-read-up rule, restricting reads from higher-integrity objects. As an example of practical validation of the proposed approach, the paper examines technologies for employing MIC to control access to user data in the Samba and NFS network file systems. Overall, this approach does not require significant modifications to MIC in the Astra Linux OS, and for the scientific justification of its security, the MROSL DP-model has already been developed.

**Keywords:** operating system; mandatory integrity control; MROSL DP-model; Astra Linux.

**For citation:** Devyanin P.N. Gorbatov V.V., Trubnikov A.A. Implementing Mandatory Integrity Control in Astra Linux OS for Access Control of User Data. Trudy ISP RAN/Proc. ISP RAS, vol. 38, issue 3, part 1, 2026. pp. 71-86 (in Russian). DOI: 10.15514/ISPRAS-2026-38(3)-4.

## *1. Введение*

В современных операционных системах (ОС) механизм управления доступом, как правило, решает спектр задач по обеспечению безопасности самих ОС и обрабатываемых ими данных. Во-первых, это защита системного программного обеспечения (ПО) ОС от несанкционированного изменения реализуемых им функций, включая захват управления над ним со стороны нарушителя через, например, внедрение закладок или заражение вирусами. То есть можно говорить, что в этом случае механизм управления доступом направлен на обеспечение целостности программной среды защищенной ОС. Во-вторых, это управление (разграничение) доступом к сущностям (объектам доступа, например, файлам или каталогам) субъектов (процессов) от имени в большинстве случаев непривилегированных учетных записей пользователей, использующих для работы с данными прикладное ПО. По мере развития информационных технологий эта задача только усложняется, так как при таком управлении доступом приходится учитывать специфику разнородного прикладного ПО в сетевых, доменных, облачных инфраструктурах, в средах контейнеризации или виртуализации.

Для решения задач каждой из этих двух групп сложилась многолетняя практика применения конкретных видов политик управления доступом, основные из которых определены в

ГОСТ Р 59453.1-2021 [1]. Поскольку дискреционное управление доступом, основанное на непосредственном и независимом друг от друга назначении прав доступа субъектам к сущностям, много лет являлось относительно простым и по сути единственным реализованным штатно в большинстве ОС, в том числе семейства Linux, то оно за неимением альтернативы использовалось для решения задач обеих групп. Но оно не задает четких правил управления доступом, что часто приводит к ошибкам при администрировании ОС. Также для него доказана в общем случае алгоритмическая неразрешимость задачи проверки безопасности [2-3], что затрудняет его использование как единственного механизма управления доступом в сертифицированных ОС, соответствующих высоким классам защиты согласно нормативным документам ФСТЭК России [4].

Более гибкое и развитое ролевое управление доступом (РУД) потенциально могло бы решить задачи обеих групп, но оно полнофункционально (когда назначение субъектам прав доступа к сущностям осуществляется только через роли, используются обычные, административные, запрещающие роли, их иерархии, механизм ограничений, это согласовано с другими политиками управления доступом путем, например, назначения ролям уровней конфиденциальности или целостности) еще не реализовано в ОС Astra Linux [5-7], а также в большинстве других ОС общего назначения, так как это является достаточно сложным практически. При этом известны примеры реализации РУД в ОС, включая ОС IBM AIX [8] и пакет безопасности Security Enhanced Linux (SELinux) [9]. Однако оба этих решения непосредственно направлены на выполнение задач первой группы (защита и администрирование системного ПО), а используемую в их основе базовую модель RBAC (Role-Based Access Control) [10-11, 3] сложно назвать адекватной условиям функционирования современных ОС. Кроме того, в рамках этой модели не приводится доказательство условий безопасности РУД.

Напротив мандатное управление доступом (МРД), реализуемое в том числе в подсистеме безопасности PARSEC ОС Astra Linux, изначально ориентировано на защиту пользовательских данных с использованием назначения сущностям уровней конфиденциальности, субъектам – уровней доступа, а на основе сравнения этих уровней принимаются решения о предоставлении доступов с конечной целью предотвратить создание информационных потоков (скрытых каналов) с высокого уровня конфиденциальности на низкий («сверху-вниз») [12, 3]. При этом для задания уровней конфиденциальности (доступа) МРД в общем случае используется решетка многоуровневой безопасности MLS (Multi Level Security, MLS-решетка), являющаяся прямым (декартовым) произведением линейной решетки уровней конфиденциальности и решетки подмножеств множества неиерархических категорий конфиденциальности, отражающих либо содержание данных, либо их принадлежность к структурным подразделениям организации, использующей МРД. Для МРД в рамках соответствующих формальных моделей, как правило, формулируются и доказываются условия его безопасности.

Вместе с тем правила МРД накладывают жесткие ограничения на управление доступом, часто избыточные для случаев, когда не требуется предотвращение утечки конфиденциальных данных. Например, МРД запрещено копирование данных между сущностями, обладающими равными линейными уровнями конфиденциальности, но разными неиерархическими категориями. То есть если каждому подразделению некоторой организации соответствует собственная неиерархическая категория, то передача данных между этими подразделениями (чтение и запись) будет запрещена МРД.

В этом контексте в случае, когда от реализованного в ОС механизма управления доступом не требуется защита от утечки конфиденциальных данных (информационных потоков «сверху-вниз»), а необходимо управление (разграничение) доступом на основе должностных иерархий организаций или многодоменных сетевых структур, может оказаться востребованным мандатный контроль целостности (МКЦ) [13, 3]. Он аналогично МРД для задания уровней целостности субъектов и сущностей позволяет использовать решетку уровней целостности, являющуюся прямым (декартовым) произведением линейной решетки уровней целостности и





решетки подмножеств множества неиерархических категорий целостности. Эти уровни целостности сравниваются при принятии решения о предоставлении доступов с целью предотвратить получение субъектом с меньшим уровнем целостности управления другим субъектом с большим уровнем целостности или доступа на запись к сущности с большим уровнем целостности.

Хотя в модели Биба (первой формальной модели МКЦ) не утверждалось, что МКЦ направлен только на обеспечение целостности системного ПО, дальнейшее практическое применение МКЦ (например, в ОС IBM AIX в режиме Trusted AIX [8], в ОС семейства Microsoft Windows [14] или ОС KasperskyOS [15]), как правило, сводилось к решению задач именно этой группы. То есть первоочередной задачей МКЦ являлась защита привилегированных высокоцелостных компонент (субъектов или сущностей) системного ПО ОС от их несанкционированного изменения или захвата управления над ними со стороны непривилегированных низкоцелостных субъектов (нарушителей). Именно для этого МКЦ был реализован в подсистеме безопасности PARSEC ОС Astra Linux, став в ней основным механизмом защиты, согласованным с дискреционным и мандатным управлением доступом, а также механизмом замкнутой программной среды (ЗПС) [6, 7]. При этом научной базой механизмов управления доступом в ОС Astra Linux является соответствующая критериям ГОСТ Р 59453.1-2021 [1] мандатная сущностно-ролевая ДП-модель управления доступом и информационными потоками в ОС семейства Linux (МРОСЛ ДП-модель) [3], в рамках которой, в том числе, осуществляется доказательство условий безопасности МКЦ.

Однако по мере накопления опыта внедрения МКЦ в ОС Astra Linux были начаты исследования по его более широкому применению, включая управление доступом к пользовательским данным. Так, например, в [16] был предложен подход по адаптации к МКЦ технологии контейнерной виртуализации, когда потенциально «опасное» ПО запускается в изолированных на отрицательных линейных уровнях целостности (в сессиях непривилегированного пользователя, работающего на нулевом уровне целостности) контейнерах-«песочницах» (например, docker). Этот подход позволяет обеспечить защиту от эксплуатации многих типовых уязвимостей прикладного ПО, в том числе защиту от вирусов (например, «шифровальщиков»). Кроме ОС Astra Linux в рамках проведенных авторами исследований были учтены результаты разработки экспериментальных ОС HiStar [17] и DStar [18], в которых для систем децентрализованного контроля информационных потоков предлагалось объединить уровни конфиденциальности и целостности сущностей, разрешив их понижение субъектами, получившими доступ к этим сущностям в процессе их передачи между компонентами распределенной системы. Хотя непосредственно применить такой подход для управления доступом к пользовательским данным в ОС Astra Linux не представляется возможным, т. к. в ней не допускается неконтролируемое понижение уровней конфиденциальности и целостности сущностей, а также для этого подхода отсутствует доказательство условий безопасности МРД и МКЦ в рамках формальной модели.

Таким образом, в настоящей статье развиваются подходы по управлению доступом к пользовательским данным в ОС Astra Linux в первую очередь за счет гибкого использования неиерархических категорий уровней целостности МКЦ. При этом если термин целостность дополнить доверием, то уровни целостности (доверия) МКЦ хорошо соотносятся с должностными иерархиями организаций или многодоменными сетевыми структурами («лесами» доменов), поскольку правила МКЦ по умолчанию запрещают доступ на запись «снизу-вверх» (от менее целостного, недоверенного к более целостному, доверенному), а при необходимости с применением специального флага сущностей SSI (который потенциально может назначаться на экземпляры ОС и даже отдельные домены) позволяют запретить и доступ на чтение «снизу-вверх». В качестве примера практической апробации предлагаемого подхода в статье рассматривается технология использования МКЦ для управление доступом к пользовательским данным в востребованных сетевых файловых системах Samba [19] и NFS

[20]. Также важно отметить, что все перечисленное не требует существенной доработки МКЦ в ОС Astra Linux и базируется на применении МРОСЛ ДП-модели.

В связи с изложенным статья организована следующим образом. В следующем разделе анализируются технологии и возникающие при этом проблемы применения МРД для управления доступом к пользовательским данным на примере ОС Astra Linux. В разделе 3 описывается подход к решению такой задачи с использованием МКЦ. В разделе 4 излагается технология реализации МКЦ в сетевых файловых системах Samba и NFS. Заключение завершает статью, в нем подводятся итоги проведенного исследования технологий применения механизма МКЦ ОС Astra Linux для управление доступом к пользовательским данным, а также рассматриваются дальнейшие направления развития этих технологий.

## 2. Анализ технологий применения МРД для управления доступом к пользовательским данным

МРД для защиты от утечки конфиденциальных данных начали реализовывать в ОС, начиная с созданной в 1976 г. ОС Multics [12, 3], в которой этот механизм управления доступом базировался на классической формальной модели Белла-ЛаПадулы. На основе этой же модели с 1999 г. МРД функционирует в пакете безопасности SELinux [9]. С момента появления на рынке в 2008 г. ОС Astra Linux МРД стало ее штатным механизмом (в режиме защиты «Смоленск»), в основе которого с 2012 г. применяется МРОСЛ ДП-модель.

Поскольку для обеспечения гибкости МРД не достаточно только принадлежащих линейной решетке сравнимых между собой уровней доступа или конфиденциальности (например, «несекретно», «для служебного пользования», «секретно», «совершенно секретно») учетных записей пользователей, субъектов (процессов) и сущностей (файлов, каталогов, сокетов и др.), как уже было отмечено, в нем часто используется решетка подмножеств множества неиерархических категорий конфиденциальности, отражающих либо содержание данных (например, «экономические», «политические», «военные» и т.п.), либо их принадлежность к структурным подразделениям организации, использующей МРД (например, «отдел № 1», «бухгалтерия», «отдел кадров», «научно-исследовательский отдел» и т.п.). В совокупности прямое (декартово) произведение линейной решетки и решетки подмножеств множества неиерархических категорий конфиденциальности образуют MLS-решетку. Для задания уровней конфиденциальности первого вида, как правило, используются положительные целые числа, второго вида – маска бит.

На основе сравнения уровней конфиденциальности (доступа) из MLS-решетки механизмом МРД принимаются решения о предоставлении доступов с конечной целью предотвратить создание информационных потоков (скрытых каналов) «сверху-вниз». Например, при предоставлении субъекту доступа на чтение к сущности проверяется, что линейный уровень доступа субъекта не ниже линейного уровня конфиденциальности сущности, а биты неиерархических категорий доступа субъекта включают биты неиерархических категорий конфиденциальности сущности. Следует также отметить, что некоторые уровни конфиденциальности (доступа) могут быть несравнимы друг с другом (например, когда биты неиерархических категорий конфиденциальности одного уровня не включают все биты неиерархических категорий конфиденциальности другого уровня, и наоборот).

В таком виде MLS-решетка (целое число от 0 до 255 и маска из 64 бит неиерархических категорий) реализуется в ОС Astra Linux. При этом для предотвращения утечки конфиденциальных данных, в том числе через создание скрытых каналов по памяти или по времени «сверху-вниз» [21] в этой ОС внедрен спектр технологий, некоторые из которых создают неизбежные трудности для работы пользователей с их данными.

Так домашние каталоги учетных записей пользователей подвергаются виртуализации в зависимости от параметров МРД его сеанса [7]. В каталоге /home/.pdp создаются подкаталоги вида /home/.pdp/%username%, в каждом из которых создаются подкаталоги вида:





/home/.pdp/%username%/l<level>:i<int>:c<category>:t0x0

где

- <level> – линейный уровень доступа сеанса работы пользователя;
- <int> – неиерархические категории уровня целостности сеанса работы пользователя (будут рассмотрены далее);
- <category> – битовая маски неиерархических категорий уровня доступа сеанса работы пользователя;
- t0x0 – зарезервирован.

При старте сеанса работы пользователя все обращения к домашнему каталогу его учётной записи (то есть имена файлов и каталогов, начинающиеся с /home/%username%) преобразуются в обращения к тому подкаталогу каталога /home/.pdp/%username%, который соответствует текущему уровню доступа процессов сеанса. В результате процессы учётной записи пользователя «видят» в своём домашнем каталоге только файлы и подкаталоги с соответствующим уровнем конфиденциальности. При этом для доступа к файлам и подкаталогам, созданным в сеансах работы с уровнем доступа не выше текущего уровня доступа процесса, нужно указать их полное имя вида:

/home/.pdp/%username%/l<level>:i<int>:c<category>:t0x0.

Хотя такая виртуализация домашних каталогов позволяет существенно сократить ресурсы на адаптацию системного и прикладного ПО, предназначенного для ОС семейства Linux, не поддерживающих МРД, она очевидно создает трудности при работе в ОС Astra Linux пользователя, осуществляющего вход в нее с различными наборами линейных и неиерархических категорий уровней доступа. Создаваемые им файлы располагаются в разных каталогах, а копирование данных разрешается только из файлов и каталогов с меньшими уровнями конфиденциальности в файлы и каталоги с большими уровнями конфиденциальности.

Рассмотрим пример, в котором для наглядности не будем использовать линейные уровни конфиденциальности, а для неиерархических категорий возьмем маску только из 3 бит. Пусть в некоторой организации функционируют 3 отдела, каждому из которых присвоена индивидуальная неиерархическая категория конфиденциальности (001, 010, 100, соответственно). Также в организации есть руководитель, очевидно, с набором всех неиерархических категорий – 111, а также общий ресурс для обмена данными без категорий – 000. Согласно правилам МРД в такой структуре будут разрешены или запрещены права доступа на чтение (r) или на запись (w) в соответствии со схемой на рис. 1. В результате сотрудники отделов не будут иметь никаких прав доступа к ресурсам и данным других отделов, а также к ресурсам и данным руководителя, но будут иметь возможность получать данные из общего ресурса. В свою очередь руководитель будет «видеть» данные и ресурсы всех подчиненных отделов. Однако руководитель не сможет непосредственно осуществлять запись и передавать данные (руководить) в ресурсы отделов, для этого ему необходимо входить в отдельные сессии с неиерархической категорией доступа соответствующего отдела, не «видя» своих собственных ресурсов и данных, а также ресурсов и данных других отделов. Аналогично отделы не смогут непосредственно обмениваться между собой данными через общий ресурс, т. к. не смогут осуществлять запись в него.

Не меньшие сложности могут возникнуть при использовании МРД при управлении доступом к ресурсам доменных инфраструктур. Так как важным для безопасности домена его компонентам (контроллерам домена, файловым серверам и т.п.) не могут назначаться высокие уровни конфиденциальности, поскольку взаимодействие с ними может быть необходимо всем компонентам домена.

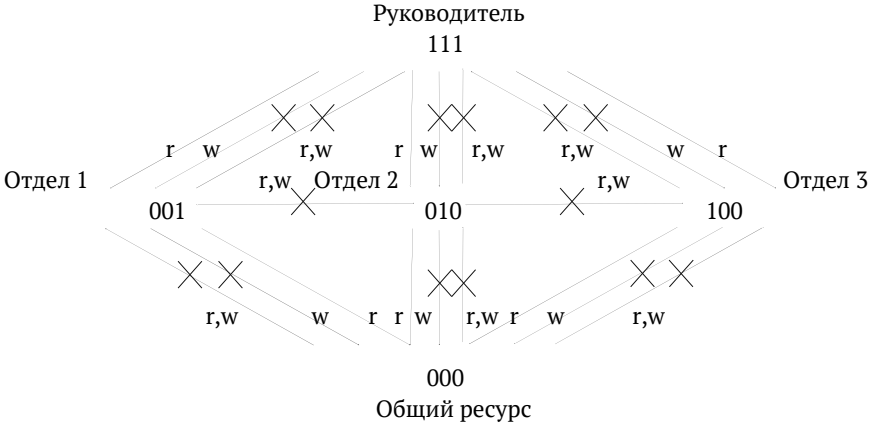


*Рис. 1. Схема разрешенных или запрещенных прав доступа в зависимости от неиерархических категорий конфиденциальности.*
*Fig. 1. A scheme of permitted or denied access rights depending on non-hierarchical confidentiality categories.*

Таким образом, в случае, когда нет необходимости в обеспечении защиты от утечки конфиденциальных данных «сверху-вниз» применение для управления доступом МРД, в том числе в ОС Astra Linux, создает существенные технические трудности особенно в организациях с большим числом подразделений, сложными должностными иерархиями и многодоменными сетевыми структурами. Все это побудило авторов провести исследования возможности использования для этих целей МКЦ как другого штатного механизма защиты ОС Astra Linux.

## 3. Подход к использованию МКЦ для управления доступом к пользовательским данным в ОС Astra Linux

Реализованный на основе МРОСЛ ДП-модели механизм МКЦ (как в ряде ОС, перечисленных в разделе 1) первоначально был использован в ОС Astra Linux для решения задач защиты системного ПО. Для этого в ней, начиная с ее релиза 2021 г., в режимах защиты «Воронеж» и «Смоленск» была применена решетка уровней целостности, являющаяся прямым (декартовым) произведением решетки подмножеств множества неиерархических категорий целостности (задаваемых маской из 4 байт или 32 бит) и линейных уровней целостности (задаваемых 1 байтом знаковых чисел от -128 до 127). Представление таких уровней целостности имеет следующий вид: <ilev>:<ilinear> (например, 0x00000002:-128), где ilev – неиерархические категории, ilinear – линейный уровень.

Эти уровни целостности сравниваются при принятии решений о предоставлении доступов в первую очередь на запись с целью предотвратить получение субъектом с меньшим уровнем целостности управления другим субъектом с большим уровнем целостности. Например, при предоставлении субъекту доступа на запись к сущности проверяется, что биты неиерархических категорий целостности субъекта включают биты неиерархических категорий целостности сущности, а линейный уровень целостности субъекта не ниже линейного уровня целостности сущности. Также как уровни конфиденциальности некоторые уровни целостности могут быть несравнимы друг с другом. Кроме того, в отличие от уровней конфиденциальности, в которых традиционно на практике большую роль играет линейный уровень, в уровнях целостности ОС Astra Linux более важны неиерархические категории целостности. С их помощью задается принадлежность к важным системным компонентам ОС, например, самый младший бит указывает, что это сетевые службы, следующий за ним – средства виртуализации.





При этом пока из 4 байт неиерархических категорий по умолчанию в ОС Astra Linux используется только младший с установленным максимальным уровней целостности для всей ОС – 0x0000003F:0 (десятичное 63). Линейный уровень целостности задействуется еще редко, например, для изоляции на отрицательных линейных уровнях с помощью средств контейнерной виртуализации потенциально «опасного» ПО такого, как браузеры или офисные приложения [16].

Достаточная выразительность уровней целостности МКЦ, наличие в ОС Astra Linux «свободных» 3 байт неиерархических уровней целостности, в ряде случаев более «мягкие», чем у МРД, правила управления доступом, накопление практик применения МКЦ для защиты системного ПО и настройки ОС в целом, позволили авторам рассмотреть возможность его использования для управления доступом к пользовательским данным.

Чтобы описать предлагаемый подход проанализируем пример, аналогичный приведенному в предыдущем разделе. Пусть в некоторой организации функционируют 3 отдела, каждому из которых присвоена индивидуальная неиерархическая категория целостности (001, 010, 100, соответственно), есть руководитель с неиерархическими категориями 111, и общий ресурс с 000. Согласно правилам МКЦ в такой структуре будут разрешены или запрещены права доступа на чтение (r) или на запись (w) в соответствии со схемой на рис. 2. Где все будут иметь права доступа на чтение к ресурсам и данным друг друга. Однако отделы не смогут осуществлять запись в ресурсы друг друга, но смогут обмениваться данными через общий ресурс, а руководитель сможет непосредственно осуществлять запись и передавать данные (руководить) в ресурсы отделов.

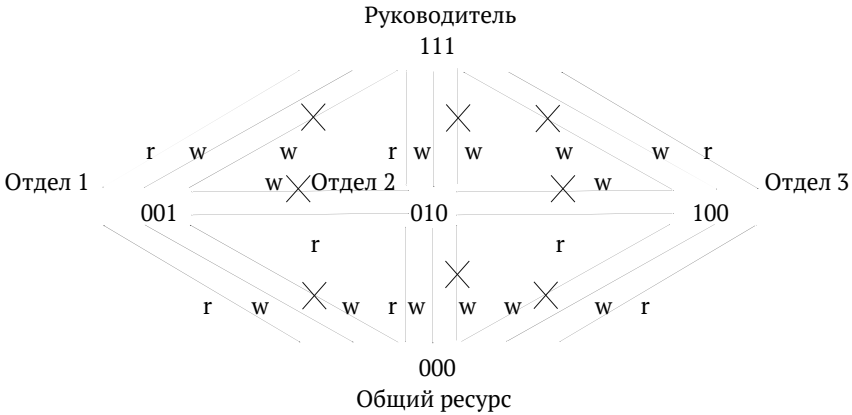


*Рис. 2. Схема разрешенных или запрещенных прав доступа в зависимости от неиерархических категорий целостности.*
*Fig. 2. A scheme of permitted or denied access rights depending on non-hierarchical integrity categories.*

При этом возможность без ограничений читать данные друг друга может оказаться неприемлемой в ряде случаев, например, когда руководителю необходимо сделать «закрытыми» часть его ресурсов, или когда необходимо сделать ресурсы какого-либо отдела «невидимыми» для других отделов, но не для руководителя организации. В таких случаях может быть использован реализованный в ОС Astra Linux назначаемый на сущности флаг SSI, который разрешает доступ к сущности на чтение субъекту только с текущим уровнем целостности большим или равным уровню целостности этой сущности. В настоящее время такой флаг может быть задан для файлов и каталогов. В перспективе рассматривается возможность его назначения на отдельные экземпляры ОС и даже домены.

В рассматриваемом примере флаг SSI может быть задан для ресурсов руководителя (или на какой-то отдельный его сетевой каталог) и на все ресурсы отдела 2, тогда права доступа на чтение изменятся согласно схеме на рис. 3.

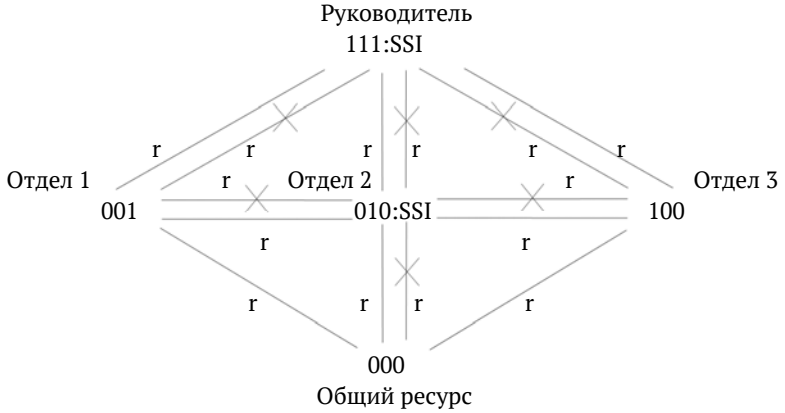


*Рис. 3. Схема разрешенных или запрещенных прав доступа на чтение в зависимости от неиерархических категорий целостности.*
*Fig. 3. A scheme of permitted or denied read access rights depending on non-hierarchical integrity categories.*

В результате такое применение неиерархических категорий уровней целостности и флага SSI МКЦ позволяет реализовать, с одной стороны, менее жесткие чем у МРД ограничения при управлении доступом, с другой стороны, он предоставляет возможность создавать информационные потоки (осуществлять управление) от субъектов и сущностей с большим уровнем целостности к субъектам и сущностям с меньшим уровнем целостности, при необходимости изолируя некоторые из них от менее целостных компонент системы. Это обеспечивает большее соответствие МКЦ востребованным практикам управления доступом к пользовательским данным с учетом должностных иерархий, структур подразделений организаций или многодоменных сетевых структур их информационных систем, построенных на основе ОС Astra Linux. При этом если совместно с термином уровень целостности использовать термины уровень доверия, важности или критичности, то предлагаемый подход становится еще яснее. Например, становятся очевидными правила применения МКЦ вида: уровень целостности (доверия или важности) руководителя должен быть выше уровней целостности (доверия или важности) всех подчиненных ему сотрудников и подразделений, или контроллер домена должен иметь уровень целостности (критичности) не ниже, чем уровни целостности входящих в домен серверов и рабочих станций.

Для использования рассматриваемого подхода уже сейчас можно задействовать в уровнях целостности ОС Astra Linux три «свободных» старших байта неиерахических категорий, в таком случае, например:

- битовые маски старших байт 1 и 2 будут соответствовать структурным подразделения или должностным иерархиям организации (как в приведенном в разделе примере);
- битовая маска байта 3 использоваться для задания уровней целостности компонент домена («лесов» домена) с учетом их критичности;
- байт 4 как в текущей реализации для задания уровней целостности компонент экземпляра ОС;
- байт линейного уровня целостности для изоляции с помощью «песочниц» потенциально «опасного» ПО (браузеры, офисные приложения и др.).





К примеру уровень целостности процессов руководителя всей организации (используются все неиерархические категории байт 1 и 2), функционирующего на файловом сервере домена (используется только младший бит байта 3) с правами администратора ОС (аналогично текущий реализации ОС значение байта 4 соответствует десятичному 63) не в «песочнице» (линейный уровень целостности равен 0), будет 0xFFFF013F:0. То есть МКЦ в ОС Astra Linux может эффективно использоваться и для защиты системного ПО, и для управления доступом к пользовательским данным.

Кроме того, в изложенном подходе с помощью МКЦ не предлагается полностью заменить МРД. Когда требуется защита от утечки конфиденциальных данных их можно применять совместно. При этом вместе с востребованными практикой линейными уровням конфиденциальности (доступа), задавая нулевыми неиерархические категории уровней конфиденциальности (доступа), для гибкости управления доступом к пользовательским данным и защиты системного ПО использовать неиерархические категории уровней целостности. Все перечисленное также не требует существенной доработки текущей реализации МКЦ и МРД в ОС Astra Linux.

## *4. Технология реализации МКЦ в сетевых файловых системах Samba и NFS*

### 4.1. Реализация МКЦ в сетевой инфраструктуре

Кроме предложений по управлению доступом к пользовательским данным в отдельном экземпляре ОС рассмотрим технологию применения механизма МКЦ для аналогичных целей в сетевых файловых системах Samba и NFS. В настоящее время в актуальных версиях ОС Astra Linux механизм МКЦ для этих файловых систем функционирует преимущественно на стороне клиентских рабочих станций. Вследствие этого серверные компоненты данных сетевых файловых систем, по умолчанию работающие на максимальном уровне целостности, не могут противостоять скомпрометированным клиентским рабочим станциям, с которых нарушитель может инициировать сетевые запросы на изменение высокоцелостных пользовательских данных, хранящихся на файловых серверах.

Исходя из этого, для реализации МКЦ в сетевых файловых системах Samba и NFS предлагается использовать третью доверенную сторону – контроллер домена, на который возлагаются функции централизованного назначения и хранения уровней целостности для каждой учетной записи пользователя, рабочей станции и файлового сервера. Такой механизм МКЦ не должен позволять при управлении доступом к пользовательским данным на файловом сервере использовать потенциальную возможность локального повышения нарушителем уровня целостности взломанной им рабочей станции клиента выше централизованно заданного на контролере домена значения. Кроме того, при реализации технологии требуется, чтобы подсистема управления доступом к пользовательским данным на стороне файлового сервера функционировала в соответствии с МРОСЛ ДП-моделью и использовала централизованно назначенные уровни целостности учетной записи пользователя (от имени которой процесс на рабочей станции клиента осуществляет запрос на доступ к файловому серверу), рабочей станции клиента и самого сервера, а также хранящиеся локально на файловом сервере уровни целостности сущностей (объектов доступа, например файлов или каталогов), состоящие из набора неиерархических категорий уровня целостности и специальных флагов МКЦ. При этом средствами подсистемы безопасности PARSEC на клиентской рабочей станции также может осуществляться управление доступом субъектов (процессов) к сущностям на файловом сервере через механизм монтирования на рабочей станции его сетевой файловой системы. Однако в случае реализации описываемой технологии при компрометации рабочей станции клиента или учетной записи пользователя исключается возможность распространения атаки нарушителя на сущности сетевых файловых систем, имеющие уровни целостности выше или несравнимые

централизованно заданных на контроллере домена для рабочей станции клиента или учетной записи пользователя значений.

Для централизованного управления учетными записями пользователей, рабочими станциями и серверами предлагается использовать доменную инфраструктуру на основе программного комплекса FreeIPA, доступного в репозиториях ОС Astra Linux. Данная инфраструктура включает в себя контроллер домена, файловые серверы Samba и NFS, а также клиентские рабочие станции, входящие в домен. Каждый из перечисленных компонент должен функционировать под управлением ОС Astra Linux в режимах защиты «Воронеж» или «Смоленск». Чтобы исключить возможность подмены заданных на контроллере домена уровней целостности клиентских рабочих станций и учетных записей пользователей, для их передачи на файловые серверы предлагается использовать протокол аутентификации Kerberos, а именно интегрированную во FreeIPA и поддерживаемую в Samba и NFS реализацию MIT Kerberos [19].

## 4.2. Реализация технологии на контроллере домена

В актуальной версии FreeIPA из репозитория ОС Astra Linux реализована функциональность, позволяющая администратору домена назначать уровни целостности учетным записям пользователей посредством изменения для них значений атрибута x-ald-user-mic-level, хранящегося на контроллере домена в LDAP-каталоге (в базе данных службы каталогов 389 Directory Server). Однако для учетных записей рабочих станций и серверов аналогичный атрибут отсутствует. Исходя из этого, предлагается расширить структуру данных на контроллере домена путем добавления нового атрибута x-ald-host-mic-level для учетных записей рабочих станций и файловых серверов, значение которого будет назначаться администратором домена в процессе их регистрации на контроллере домена.

Для передачи значений атрибутов x-ald-user-mic-level и x-ald-host-mic-level на файловые серверы без возможности их подмены клиентом предлагается включать их в сервисные билеты Kerberos (Ticket Granting Service – TGS), выдаваемые контроллером домена, поскольку их структура содержит имена учетной записи пользователя и сетевой службы и тем самым формирует необходимый контекст для применения ограничений МКЦ. При этом основная сложность реализации данного подхода заключается в том, что по умолчанию запросы на получение билетов (AS-REQ и TGS-REQ) не содержат имени учетной записи клиентской рабочей станции. Для этого авторами предлагается использовать технологию FAST (Flexible Authentication Secure Tunneling), которая обеспечивает передачу в запросах дополнительного билета, выдаваемого для учетной записи рабочей станции клиента. Это позволит реализовать извлечение ее имени на стороне центра распределения ключей MIT Kerberos (Key Distribution Center – KDC) в процессе обработки запроса на выдачу билета и добавление этого имени непосредственно в билет в качестве индикатора аутентификации. Для реализации всего перечисленного в MIT Kerberos есть несколько средств, наиболее оптимальным из которых является внесение изменений в исходный код KDC, а именно в функции обработки AS-REQ и FAST. В результате обеспечивается упрощение архитектурного решения предлагаемой технологии и оптимальная производительность по сравнению с разработкой отдельного плагина, использование которого может привести к накладным расходам времени выполнения на вызовы интерфейсных функций и потенциальные переключения контекста.

Поиск значений уровней целостности учетной записи пользователя и рабочей станции клиента в базе данных службы каталогов, а также добавление их в билеты, предлагается реализовать в FreeIPA путем внесения изменений в исходный код плагина IPA-KDB, отвечающего за взаимодействие с базой данных службы каталогов и выдачу атрибутных сертификатов привилегий (Privilege Attribute Certificate – PAC). В рамках этих изменений осуществляется поиск и преобразование значений уровней целостности в идентификаторы безопасности (Security Identifier – SID) и их добавление в структуру PAC в билетах.





Для уровней целостности принят формат SID вида: S-1-16-0-<HOST_ILEV> – для учетной записи рабочей станции, S-1-16-1-<USER_ILEV> – для учетной записи пользователя. Указанные SID маркируются как обязательные, что гарантирует их принудительное применение, и добавляются в PAC без нарушения формата билета, сохраняя совместимость с другими сетевыми службами. Следует отметить, что является достаточной реализация вышеописанных изменений в исходном коде KDC и плагина IPA-KDB для первичных билетов (Ticket Granting Ticket – TGT), поскольку FreeIPA штатными средствами реализует копирование SID из первичных билетов в сервисные. В результате TGS становится носителем централизованно назначенных уровней целостности, передаваемых серверным компонентам Samba и NFS, что исключает влияние на МКЦ в сетевой инфраструктуре его возможных локальных изменений на отдельных рабочих станциях.

Схема используемых компонент домена представлена на рис. 4, где в качестве примера для контроллера домена задано значение уровня целостности 0x000003FF (десятичное 1023), для файлового сервера 0x000001FF (десятичное 511), для рабочей станции клиента 0x0000003F (десятичное 63).

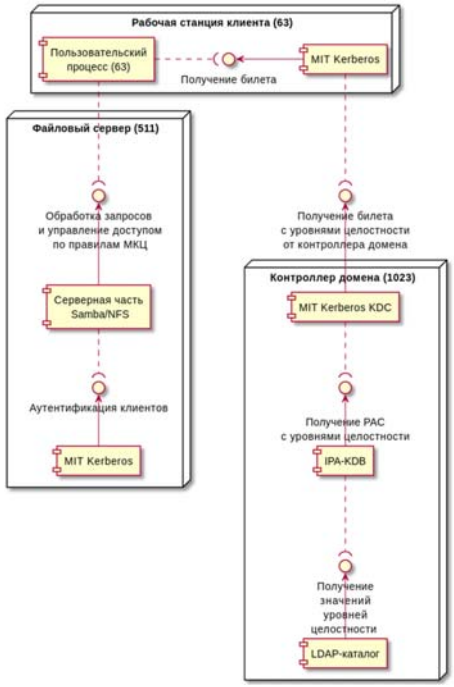


*Рис. 4. Схема компонент домена, используемых в реализации МКЦ.*
*Fig. 4. Scheme of domain components used in the MIC implementation.*

## 4.3. Реализация технологии на файловом сервере Samba

Для реализации МКЦ на стороне файлового сервера Samba [19] возможна разработка модуля для подсистемы Samba VFS (Virtual File System), осуществляющего проверки сетевых запросов на доступ к сущностям. Стоит отметить, что данный подход широко применяется разработчиками при создании для Samba дополнительных механизмов безопасности. Так, например, в виде VFS-модулей подсистемы Samba часто реализованы различные антивирусные

сканеры, системы аудита и мониторинга файловых операций, а также механизмы управления доступом на основе расширенных атрибутов.

В архитектуре модуля ключевая роль отводится извлечению уровней целостности учетной записи пользователя и рабочей станции клиента, которые передаются в структуре PAC сервисного билета Kerberos, где они представлены в формате SID. Во многом выбор данного формата для представления уровней целостности обусловлен его поддержкой не только в FreeIPA, но и в Samba. В частности, анализ PAC выполняется автоматически средствами Samba при аутентификации клиента с последующим размещением всех SID в структуре контекста безопасности текущей сессии, доступной из интерфейсных функций VFS-модуля. Таким образом, в исходном коде модуля достаточно реализовать только поиск нужных SID в данной структуре без необходимости дополнительного разбора билета.

Предлагается при управлении доступом учитывать минимальный (наибольшую нижнюю границу в решетке подмножеств множества неиерархических категорий целостности) из централизованно назначенных уровней целостности учетной записи пользователя и рабочей станции клиента, далее называемый уровнем целостности сессии, вычисляемый как результат побитовой конъюнкции указанных значений.

Для реализации управления доступом разработанный авторами VFS-модуль перехватывает каждый сетевой запрос от клиента и обеспечивает соответствующие проверки МКЦ. В частности, перехватываются операции открытия, создания, чтения, записи, переименования, удаления сущностей, а также операции работы с атрибутами и символическими ссылками в соответствии с набором функций, предоставляемых VFS-интерфейсом Samba.

При управлении доступом особое внимание уделяется наличию у сущностей специальных флагов МКЦ. В случае, если у сущности есть флаг SSI, и уровень целостности сессии ниже или несравним с уровнем целостности сущности, реализуется полный запрет на любые операции с ней, независимо от наличия других флагов. Это обеспечивает изоляцию критически важных данных от доступа (на чтение и запись) со стороны менее доверенных (с меньшим или несравнимым уровнем целостности) компонент домена. В случае запроса на создание новой сущности модуль проверяет возможность операции, учитывая правила МКЦ: если уровень целостности родительского каталога сущности превышает или несравним с уровнем целостности сессии и отсутствует флаг IRELAX, то доступ запрещается. Кроме того, при успешном создании сущности VFS-модуль также реализует механизм наследования уровня целостности от родительского каталога при наличии на нем специальных флагов IRELAX или IINH путем вычисления минимального из уровней целостности сессии и родительского каталога и последующей установки этого уровня новой сущности.

### 4.4. Реализация технологии на файловом сервере NFS

Серверная часть сетевой файловой системы NFS [20] в ОС Astra Linux включает модуль ядра nfsd, который принимает и обрабатывает RPC-запросы (Remote Procedure Call) от клиентов, выполняя операции над локальной файловой системой, а также набор утилит nfs-utils, обеспечивающих взаимодействие пользовательского пространства с ядром и аутентификацию клиентов. В отличие от Samba, поддерживающей VFS-модули, NFS не предоставляет аналогичного механизма расширения, что требует внесения изменений в исходный код ее реализации для добавления новых функциональных возможностей.

Извлечение уровней целостности из структуры PAC, входящей в состав билета Kerberos, предлагается реализовать в исходном коде службы rpc.svcgssd из набора утилит nfs-utils. Эта служба отвечает за серверную сторону аутентификации с использованием универсального интерфейса программирования для служб безопасности GSSAPI (Generic Security Services Application Programming Interface), абстрагирующего механизмы аутентификации, включая различные реализации протокола Kerberos. Получение PAC из контекста GSSAPI осуществляется путем вызова функции gss_get_name_attribute с передачей имени атрибута





"urn:mspac:". При этом необходимо реализовать функции разбора структуры PAC и извлечения соответствующих SID с уровнями целостности учетной записи рабочей станции и учетной записи пользователя, поскольку в NFS в отличие от Samba такие функции штатно отсутствуют. По этим значениям вычисляется уровень целостности сессии как результат побитовой конъюнкции, аналогично реализации в Samba, который передается в модуль ядра nfsd через вызов ioctl и сохраняется в контексте текущей сессии.

В модуле ядра nfsd реализация проверок по правилам МКЦ интегрируется в функции управления доступом, такие как nfsd_permission, а также в функции обработки операций в подсистеме VFS. Уровень целостности сессии сравнивается с уровнями целостности сущностей, извлекаемыми из расширенного атрибута "security.PDPL". Также как в Samba при создании новых сущностей вызывается функция установки наследуемых атрибутов на дочернюю сущность через vfs_setxattr.

## *5. Заключение*

В настоящей статье изложен подход к применению МКЦ, реализованному в ОС Astra Linux, для управления доступом к пользовательским данным с сохранением основной текущей функции данного механизма – обеспечение целостности системного ПО этой ОС. Данный подход имеет большую, чем МРД, гибкость управления доступом и лучше соответствует практикам задания разрешенных информационных потоков и управления с учетом иерархических структур подразделений или должностей организаций, доменных иерархий сетевых инфраструктур их информационных систем. При этом для использования подхода не требуется существенная доработка текущей реализации МКЦ и МРД в ОС Astra Linux, которые могут функционировать совместно, дополняя друг друга, а также не нужна доработка МРОСЛ ДП-модели, поскольку подход ей соответствует. Для примера рассмотрена технология реализации предлагаемого подхода для использования МКЦ в сетевых файловых системах Samba и NFS.

В дальнейшем планируются исследования и макетирование МКЦ в ОС Astra Linux с целью развития функционала подхода. Например, предполагается проанализировать возможность задание для некоторых сущностей (контейнеров, экземпляров ОС, компонент домена и др.) уровней целостности с маской бит неиерархических категорий произвольной длины, что может позволить применять МКЦ в средах контейнеризации, виртуализации или облачных средах с потенциально бесконечным числом компонент. Кроме того, предполагается дополнить возможности управления доступом к пользовательским данным согласованными с МКЦ механизмами МРД и РУД.

## Список литературы / References

[1]. ГОСТ Р 59453.1-2021 «Защита информации. Формальная модель управления доступом. Часть 1. Общие положения». М.: Стандартинформ. 16 с. / GOST R 59453.1-2021 «Information protection. Formal access control model. Part 1. General principles», 2021 (in Russian).

[2]. Harrison M., Ruzzo W., Ullman J. Protection in operating systems // Communication of ACM. 19(8), 1976. pp. 461-471.

[3]. Девянин П.Н. Модели безопасности компьютерных систем. Управление доступом и информационными потоками. Учебное пособие для вузов. 3-е изд., перераб. и доп. М.: Горячая линия – Телеком, 2020. 352 с.: ил. / P.N. Devyanin. Security models of computer systems. Control for access and information flows. Hotline-Telecom, 2020, 352 p. (in Russian).

[4]. Выписка из Требований по безопасности информации, утвержденных приказом ФСТЭК России от 2 июня 2020 г. N 76. Доступно по ссылке: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/trebovaniya-po-bezopasnosti-informatsii-utverzhdeny-prikazom-fstek-rossii-ot-2-iyunya-2020-g-n-76, 26.11.2025 / Excerpts from Requirements for information security approved by FSTEK Russia order #76 of 2nd June 2020. Available at: https://fstec.ru/dokumenty/vse-dokumenty/spetsialnye-normativnye-dokumenty/trebovaniya-po-bezopasnosti-informatsii-utverzhdeny-prikazom-fstek-rossii-ot-2-iyunya-2020-g-n-76, accessed 26.11.2025 (in Russian).

[5]. Девянин П.Н. Результаты переработки уровней ролевого управления доступом и мандатного контроля целостности формальной модели управления доступом ОС Astra Linux // Труды ИСП РАН, том 35, вып. 5, 2023, С. 7-22 / 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. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 5, 2023, pp. 7-22 (in Russian).
[6]. Операционная система специального назначения Astra Linux Special Edition. Доступно по ссылке: https://astra.ru/software-services/os/, 26.11.2025. / Astra Linux Special Edition operating system. Available at: https://astra.ru/software-services/os/, accessed 26.11.2025.
[7]. Девянин П.Н., Тележников В.Ю., Третьяков С.В. Основы безопасности операционной системы Astra Linux Special Edition. Управление доступом. Учебное пособие. М., Горячая линия – Телеком, 2022, 148 с. / Devyanin P.N., Telezhnikov V.Y., Tret'yakov S.V. Astra Linux Special Edition security basics. Access control. Hotline-Telecom, 2022, 148 p. (in Russian).
[8]. AIX Version 7.3: Security / IBM Corporation. Available at: https://www.ibm.com/docs/en/ssw_aix_73/pdf/security_pdf.pdf, accessed 26.11.2025.
[9]. SELinux. Available at: https://ru.wikipedia.org/wiki/SELinux, accessed 26.11.2025.
[10]. Sandhu R. Rationale for the RBAC96 family of access control models // In Proceeding of the 1st ACM Workshop on Role-Based Access Control. ACM, 1997.
[11]. Sandhu R. Role-Based Access Control // Advanced in Computers. Academic Press, 1998. Vol. 46.
[12]. Bell D.E., LaPadula L.J. Secure Computer Systems: Unified Exposition and Multics Interpretation. Bedford, Mass.: MITRE Corp., 1976. MTR-2997 Rev. 1.
[13]. Biba K.J. Integrity Considerations for Secure Computer Systems. Bedford, Mass.: MITRE Corp., 1975. MTR-3153.
[14]. Conover M. Analysis of the Windows Vista security model / Technical Report, Symantec Corp., 2008, 18 p.
[15]. Буренков В.С., Кулагин Д.А. Модель мандатного контроля целостности в операционной системе KasperskyOS // Труды ИСП РАН, том 32, вып. 1, 2020, С. 27-56 / Burenkov V.S., Kulagin D.A. A Mandatory Integrity Control Model for the KasperskyOS Operating System. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 1, 2020. pp. 27-56 (in Russian).
[16]. Девянин П.Н., Старостин А.А., Панов Д.С., Усачев С.В. Проектирование и развитие механизма мандатного контроля целостности в операционной системе Astra Linux // Труды ИСП РАН, том 37, вып. 2, 2025, С. 61-78 / Devyanin P.N., Starostin A.A., Panov D.S., Usachev S.V. Design and Development of a Mandatory Integrity Control Mechanism in the Astra Linux Operating System. Trudy ISP RAN/Proc. ISP RAS, vol. 37, issue 2, 2025, pp. 61-78 (in Russian).
[17]. Zeldovich N., Boyd-Wickizer S., Kohler E., Mazières D. Making information flow explicit in HiStar // In Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI), 2006, pp. 263-278.
[18]. Zeldovich N., Boyd-Wickizer S., Mazières D. Securing distributed systems with information flow control // In Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI), 2008, pp. 293-308.
[19]. Samba + FreeIPA аутентификация пользователей Samba в Kerberos. Доступно по ссылке: https://wiki.astralinux.ru/pages/viewpage.action?pageId=44893460, 26.11.2025. / Samba user authentication in Kerberos Available at: https://wiki.astralinux.ru/pages/viewpage.action?pageId=44893460, accessed 26.11.2025 (in Russian).
[20]. Haynes T., Noveck D. Network File System (NFS) Version 4 Protocol. RFC 7530. IETF, 2015. Available at: https://datatracker.ietf.org/doc/html/rfc7530, accessed 26.11.2025.
[21]. ГОСТ Р 53113.1-2008 «Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Часть 1. Общие положения». М.: Стандартинформ. 12 с. / GOST R 53113.1-2008 «Information technology. Protection of information technologies and automated systems against security threats posed by use of covert channels. Part 1. General principles», 2008 (in Russian).

## Информация об авторах / Information about authors

Петр Николаевич ДЕВЯНИН – член-корреспондент Академии криптографии России, доктор технических наук, профессор, научный руководитель ООО «РусБИТех-Астра» («Группа Астра»). Область интересов: теория информационной безопасности, формальные модели безопасности компьютерных систем, разработка безопасного программного обеспечения, операционные системы семейства Linux.

Petr Nikolaevich DEVYANIN – Dr. Sci. (Tech.), Prof., corresponding member of Russian Academy of Cryptography, scientific director in RusBITech-Astra (Astra Linux). Field of Interest: information security theory, formal security models of computer systems, secure software development, operating systems of Linux family.

Вадим Владимирович ГОРБАТОВ – старший инженер ООО «РусБИТех-Астра» («Группа Астра»). Область интересов: формальные модели безопасности компьютерных систем, операционные системы семейства Linux, разработка безопасного программного обеспечения.

Vadim Vladimirovich GORBATOV – senior engineer in RusBITech-Astra (Astra Linux). Field of Interest: formal security models of computer systems, operating systems of Linux family, secure software development.

Арсений Александрович ТРУБНИКОВ – инженер ООО «РусБИТех-Астра» («Группа Астра»). Область интересов: формальные модели безопасности компьютерных систем, операционные системы семейства Linux, разработка безопасного программного обеспечения.

Arseniy Aleksandrovich TRUBNIKOV – engineer in RusBITech-Astra (Astra Linux). Field of Interest: formal security models of computer systems, operating systems of Linux family, secure software development.