Сеидов М. С-А.
Студент
3 курс, факультет «Комплексной безопасности топливно-энергетического комплекса»
РГУ нефти и газа (НИУ) имени И.М. Губкина
Россия, г. Москва

Ясевич Б. О.
Студент
3 курс, факультет «Комплексной безопасности топливно-энергетического комплекса»
РГУ нефти и газа (НИУ) имени И.М. Губкина
Россия, г. Москва

АВТОРИЗАЦИЯ И ВАЛИДАЦИЯ ВХОДНЫХ ДАННЫХ В D-BUS API NETWORKMANAGER: ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ

ВВЕДЕНИЕ

Современные Linux-системы применяются не только как рабочие станции, но и как основа управляемой корпоративной инфраструктуры, где сетевые параметры, правила доступа, службы аутентификации и централизованное администрирование становятся значимыми элементами безопасности. Для отечественных дистрибутивов Linux эта задача особенно актуальна в условиях импортозамещения и перехода на российские программные платформы. В работах, посвященных ОС Альт и другим отечественным операционным системам, D-Bus рассматривается как один из основных каналов межпроцессного взаимодействия, через который пользовательские приложения обращаются к привилегированным компонентам системы [1–3]. Значимость защиты системного сетевого управления подтверждается и отраслевыми исследованиями, где кибербезопасность связывается с устойчивостью технологических процессов и снижением риска несанкционированного воздействия на информационные системы [15].

Одним из ключевых сервисов сетевого управления в Linux является NetworkManager, значительная часть функциональности которого доступна через D-Bus API [4]. Поэтому безопасность NetworkManager определяется не только корректностью сетевой конфигурации, но и строгостью авторизации D-Bus вызовов, защитой секретов профилей и устойчивостью обработки некорректных входных данных.

В связи с этим исследование авторизации и валидации входных данных в D-Bus интерфейсах NetworkManager актуально для администрирования Linux-систем и прикладной информационной безопасности. Научная значимость работы состоит в определении границ ответственности между D-Bus как транспортом и моделью вызова методов, Polkit как механизмом авторизации и NetworkManager как сервисом семантической проверки сетевых профилей.

Целью работы является исследование механизмов авторизации и валидации входных данных в D-Bus API NetworkManager в среде операционной системы Альт. ля достижения поставленной цели были сформулированы следующие задачи:

1. Проанализировать архитектуру D-Bus API NetworkManager, включая объектную модель службы, структуру профилей соединений и методы работы с параметрами сетевой конфигурации.
2. Рассмотреть механизм Polkit-авторизации при обращении к D-Bus интерфейсам NetworkManager и определить роль субъекта доступа, пользовательского сеанса и политики безопасности при выполнении привилегированных операций.
3. Разработать методику экспериментальной проверки, предусматривающую получение матрицы разрешений, проверку доступа к открытым параметрам и секретам профилей, а также формирование набора корректных и некорректных D-Bus вызовов.
4. Провести экспериментальное исследование поведения NetworkManager при чтении профилей, получении секретов, создании

соединений и изменении существующих профилей через методы GetSettings, GetSecrets, AddConnection, Update и Update2.

5. Обобщить результаты экспериментов и определить, какие уровни защиты участвуют в обеспечении безопасности D-Bus API NetworkManager, включая типизацию D-Bus сообщений, Polkit-авторизацию и прикладную валидацию параметров внутри NetworkManager.

Методологическую основу работы составляют анализ научно-технической литературы и официальной документации, сравнительный анализ, инструментальное тестирование и метод «черного ящика». Анализ источников использовался для раскрытия архитектуры D-Bus, NetworkManager и Polkit, а также для выбора проверяемых D-Bus методов. Сравнительный анализ применялся при сопоставлении разрешений пользователя root и обычного пользователя, а также при оценке различий между методами GetSettings и GetSecrets. Метод «черного ящика» использовался при передаче в NetworkManager корректных и некорректных структур профилей без обращения к внутренней реализации службы. Инструментальное тестирование выполнялось средствами gdbus, busctl, nmcli и pkaction, применявшимися для вызова D-Bus методов, получения матрицы разрешений, анализа Polkit-действий и фиксации результатов эксперимента.

Объектом исследования являются D-Bus интерфейсы NetworkManager, используемые для управления сетевыми соединениями, параметрами профилей, состоянием сетевых устройств и доступом к чувствительным данным сетевой конфигурации в операционной системе Альт. Предметом исследования являются механизмы авторизации D-Bus вызовов и валидации входных данных, реализуемые при обращении к методам NetworkManager.

Научная новизна исследования заключается в комплексной экспериментальной оценке D-Bus интерфейсов NetworkManager на уровнях типизации D-Bus вызовов, Polkit-авторизации и прикладной валидации параметров. В отличие от работ, посвященных общему применению D-Bus в ОС Альт, управлению графическими окружениями и сетевой подсистеме

отечественных операционных систем [1–3], в работе анализируется поведение конкретных методов NetworkManager при чтении настроек, получении секретов, создании и изменении профилей. Дополнительно сопоставлены результаты для привилегированного и непривилегированного пользователя и классифицированы ошибки обработки некорректных структур D-Bus API.

Практическая значимость работы состоит в применимости результатов для администраторов Linux-систем, специалистов по информационной безопасности, аудиторов и разработчиков политик доступа. Предложенная методика позволяет проверить доступность действий NetworkManager, выявить операции с Polkit-аутентификацией, оценить защиту секретов профилей и реакцию службы на некорректные параметры. Результаты могут использоваться при настройке Polkit, проверке разграничения административных полномочий и воспроизведении аналогичных тестов в других версиях NetworkManager.

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ БЕЗОПАСНОСТИ D-BUS API NETWORKMANAGER

Анализ известных уязвимостей показывает, что безопасность D-Bus и Polkit зависит не только от базовой архитектуры, но и от корректности прикладных проверок в системных службах. Уязвимость CVE-2021-3560 позволяла обойти проверку учетных данных Polkit для D-Bus запросов и повысить привилегии локального пользователя [11]. Для D-Bus также фиксировались уязвимости отказа в обслуживании, связанные с обработкой некорректных сигнатур типов и сообщений с файловыми дескрипторами [12, 13]. Уязвимость CVE-2023-4104 демонстрирует, что ошибочная Polkit-проверка и отсутствие аутентификации для D-Bus методов могут позволить локальному пользователю изменять параметры VPN-службы [14].

D-Bus представляет собой механизм межпроцессного взаимодействия Linux-систем, предназначенный для обмена сообщениями между пользовательскими приложениями и системными службами. Его объектная модель предполагает обращение клиента не к процессу напрямую, а к имени

службы на шине, объектному пути, интерфейсу и методу. Это формализует взаимодействие компонентов, но требует строгого контроля доступа, поскольку через системную шину могут выполняться привилегированные операции управления устройствами, сетевыми параметрами и системной конфигурацией [5]. Спецификация D-Bus определяет модель шины сообщений, объектные пути, интерфейсы и типизацию аргументов, однако прикладная авторизация конкретного действия остается задачей службы, обрабатывающей запрос [6].

В Linux используются сеансовая шина D-Bus, обслуживающая процессы конкретного пользовательского сеанса, и системная, предназначенная для взаимодействия с фоновыми службами общесистемного значения. NetworkManager работает как системная служба и регистрирует на системной шине имя org.freedesktop.NetworkManager, через которое доступны объекты состояния сетевой подсистемы, устройств, активных соединений, профилей настроек и вспомогательных механизмов управления сетью.

Профиль соединения NetworkManager представляет собой структурированное описание сетевой конфигурации. Он включает общую секцию connection с идентификатором, UUID, типом соединения и параметрами автоподключения, а также технологические секции: 802-3-ethernet для Ethernet, vlan для VLAN, ipv4 и ipv6 для IP-адресации, 802-1x для сетевой аутентификации. В D-Bus API эти настройки передаются как вложенная структура a{sa{sv}}, где верхний уровень задает секции, а внутренний словарь содержит свойства секции.

Авторизация D-Bus вызовов NetworkManager не сводится к факту подключения к системной шине. Доступ к шине позволяет отправить сообщение, но не дает автоматического права изменять системный профиль, отключать сеть, получать секреты или перезагружать конфигурацию. Для разграничения таких действий применяется Polkit, который проверяет право субъекта на выполнение операции и при необходимости инициирует административную аутентификацию. Итоговое решение зависит от

пользователя, активного сеанса, политики безопасности и типа запрашиваемого действия [7].

Общая схема обработки D-Bus вызова NetworkManager представлена на рисунке 1. Клиентский процесс передает сообщение на системную шину, где проверяются адресация объекта, интерфейс, имя метода и соответствие аргументов сигнатуре. Затем NetworkManager определяет чувствительность операции и при необходимости обращается к Polkit для проверки прав субъекта. После успешной авторизации выполняется прикладная обработка запроса, включая проверку структуры профиля, допустимости свойств, типов значений и семантических ограничений параметров.

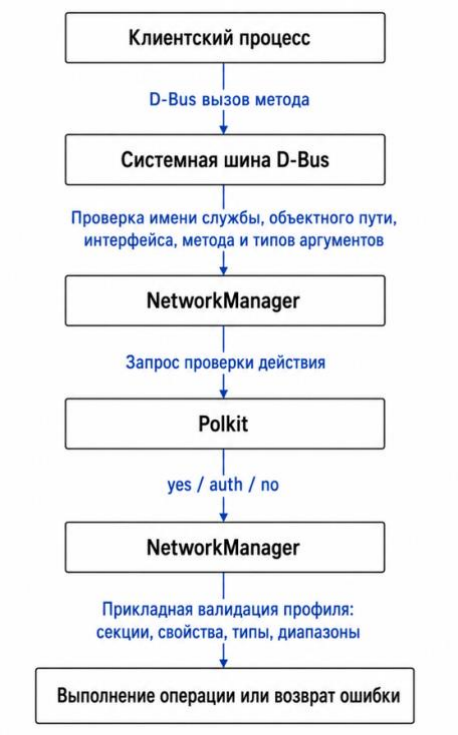

Рисунок 1 – Схема взаимодействия клиентского процесса, системной шины D-Bus, Polkit и NetworkManager

Для формализации авторизации модель доступа представлена как M = {S, A, P}, где S – множество субъектов, A – множество действий NetworkManager, P – множество политик проверки. В эксперименте S = {root, user}, A соответствует зарегистрированным Polkit-действиям NetworkManager, а P = {yes, auth, no}, где yes означает разрешение без дополнительной

аутентификации, auth — необходимость подтверждения через Polkit, no — запрет операции.

В NetworkManager контролируемые действия представлены Polkit-идентификаторами вида org.freedesktop.NetworkManager.*, охватывающими управление сетью, беспроводными интерфейсами, системными и пользовательскими профилями, именем узла, глобальными DNS-параметрами и перезагрузкой конфигурации. Результат проверки может принимать значения yes, auth или no, поэтому при анализе авторизации необходимо различать доступность D-Bus объекта, возможность вызова метода, решение Polkit и итоговое выполнение операции внутри NetworkManager.

В NetworkManager обычные параметры профиля и секреты обрабатываются раздельно: метод GetSettings возвращает несекретную структуру соединения, а чувствительные значения запрашиваются через GetSecrets. К секретам относятся пароли беспроводных сетей, параметры VPN, учетные данные 802.1X и другие защищаемые данные. Согласно документации NetworkManager, GetSecrets возвращает секреты указанной секции из постоянного хранилища или Secret Agent сеанса запрашивающего пользователя [8].

Валидация входных данных в D-Bus интерфейсах NetworkManager выполняется на трех уровнях. Первый связан с D-Bus типизацией, при которой сообщение должно соответствовать сигнатуре метода и ожидаемым типам аргументов. Второй уровень относится к структуре профиля, в котором NetworkManager проверяет допустимость секций, имен свойств и типов значений. Третий уровень является семантическим, поскольку значение с корректным типом может быть отклонено из-за недопустимого диапазона, неподдерживаемого режима или несоответствия типу соединения, например при проверке vlan.id или допустимых значений ipv4.method.

В практическом исследовании необходимо отделять ошибки авторизации от ошибок валидации. Если непривилегированный пользователь получает отказ доступа, это не подтверждает проверку переданного профиля,

поскольку запрос мог быть отклонен до семантической обработки. Следовательно валидация проверялась от имени root, авторизация анализировалась отдельно для root и обычного пользователя для выявления различий между разрешенными действиями, операциями с требованием аутентификации и запретами.

Для экспериментального анализа использовался метод GetPermissions, отражающий оценку прав текущего субъекта NetworkManager для конкретного сеанса. Его результат задается значениями yes, auth и no, где yes означает выполнение без дополнительной аутентификации, auth – необходимость подтверждения через Polkit, а no – запрет операции. Для управления профилями NetworkManager использует несколько D-Bus объектов. Объект /org/freedesktop/NetworkManager отражает состояние сетевой подсистемы, а объект /org/freedesktop/NetworkManager/Settings предназначен для работы со списком сохраненных соединений и создания новых профилей. Каждый профиль представлен отдельным путем вида /org/freedesktop/NetworkManager/Settings/N, на котором вызываются методы чтения, изменения, сохранения и удаления параметров соединения.

Методы AddConnection, Update и Update2 использовались для проверки входных данных, поскольку они работают с полной структурой настроек соединения. AddConnection позволяет оценить реакцию NetworkManager на некорректные параметры при создании профиля, Update – проверить отклонение ошибочной конфигурации уже существующего профиля, а Update2 – дополнительно исследовать обработку флагов и служебного словаря аргументов D-Bus метода [9]. Защита секретов в NetworkManager основана на разделении открытых параметров профиля и чувствительных значений. К открытым параметрам относятся имя, тип, UUID, настройки IPv4/IPv6, автоподключение и параметры интерфейса, тогда как пароли Wi-Fi, учетные данные 802.1X и ключи VPN должны обрабатываться отдельно.

ЭКСПЕРИМЕНТАЛЬНОЕ ИССЛЕДОВАНИЕ D-BUS ИНТЕРФЕЙСОВ NETWORKMANAGER

Экспериментальная часть направлена на проверку поведения D-Bus интерфейсов NetworkManager в ОС Альт при чтении профилей, получении секретов, создании соединений и изменении существующих параметров. Цель экспериментов состояла в оценке разграничения доступа к чувствительным операциям и обработки корректных и некорректных данных, передаваемых через системную шину D-Bus. Для воспроизводимости полные материалы исследования, включая команды, выводы gdbus, nmcli, busctl, pkaction и результаты тестирования, размещены в открытом репозитории GitHub [10].

Все проверки носили функциональный характер и выполнялись однократно, поскольку оценивались не временные или статистические показатели, а детерминированная реакция NetworkManager на конкретный D-Bus вызов. Эксперимент включал четыре направления:

– проверку авторизации root и обычного пользователя;

– сравнение доступа к открытым параметрам и секретам через GetSettings и GetSecrets;

– валидацию входных данных при создании профилей через AddConnection;

– проверку изменения существующих профилей методами Update и Update2 для оценки отклонения некорректной конфигурации без частичного применения.

Экспериментальное исследование проводилось в операционной системе ALT Рабочая Станция 11.1, относящейся к ветке p11. В качестве ядра использовалась версия Linux 6.12.41-6.12-alt1 для архитектуры x86_64. Используемое в рамках исследование ПО представлено в таблице 1.

Таблица 1 – Используемое ПО

ПО	**Версия**
NetworkManager	1.52.1-alt1
D-Bus	1.16.0-alt2
Polkit	124-alt2
glib2	2.84.2-alt1

libgio	2.84.2-alt1

Доступность NetworkManager через D-Bus была подтверждена проверкой системной шины (рисунок 2). На шине было зарегистрировано имя org.freedesktop.NetworkManager, связанное с процессом NetworkManager, выполняющимся от имени пользователя root.

Рисунок 2 - Проверка доступности NetworkManager через D-Bus

После фиксации стенда с помощью pkaction был получен перечень Polkit-действий NetworkManager, по которым служба запрашивает разрешение на выполнение привилегированных операций. В тестовой среде зарегистрировано 17 действий, охватывающих управление сетью и беспроводными интерфейсами, изменение системных и пользовательских профилей, имени узла, глобальных DNS-параметров и перезагрузку конфигурации. Результат представлен на рисунке 3.

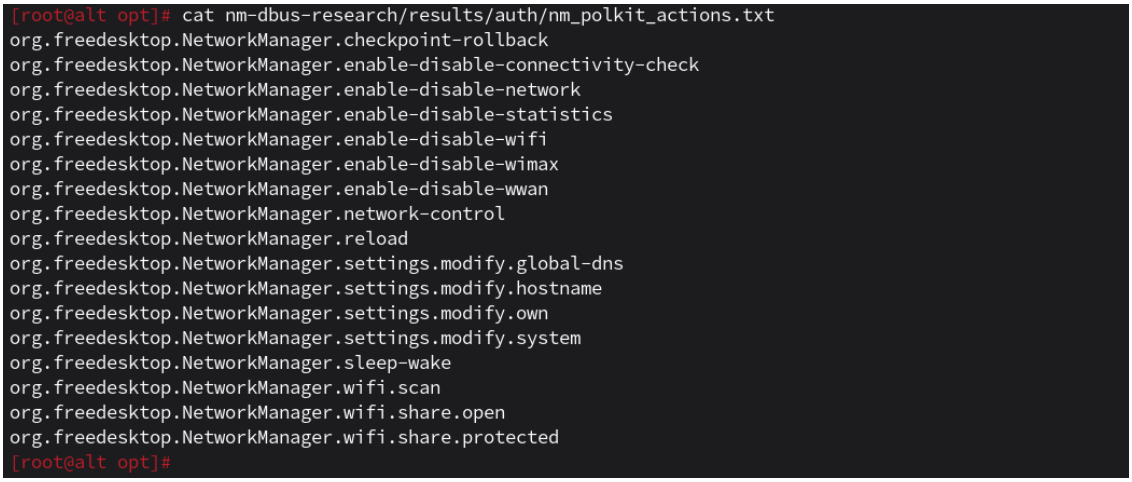

Рисунок 3 – Перечень Polkit-действий NetworkManager, зарегистрированных в тестовой среде

Для проверки различий в авторизации использовались два субъекта доступа: привилегированный пользователь root и обычный пользователь user с идентификатором UID 1000. Результаты проверки разрешений NetworkManager представлены в таблице 2.

Таблица 2 – Матрица разрешений NetworkManager для пользователей root и user

Действие NetworkManager	**root**	**user**

checkpoint-rollback	yes	auth
enable-disable-connectivity-check	yes	yes
enable-disable-network	yes	yes
enable-disable-statistics	yes	yes
enable-disable-wifi	yes	yes
enable-disable-wimax	yes	yes
enable-disable-wwan	yes	yes
network-control	yes	yes
reload	yes	auth
settings.modify.global-dns	yes	auth
settings.modify.hostname	yes	auth
settings.modify.own	yes	yes
settings.modify.system	yes	auth
sleep-wake	yes	yes
wifi.scan	yes	yes
wifi.share.open	yes	yes
wifi.share.protected	yes	yes

Сравнение результатов показывает, что NetworkManager применяет контекстно-зависимую авторизацию. Доступ к D-Bus службе не означает права на выполнение любой операции. Обычный пользователь может выполнять часть действий без повышения привилегий, однако изменение общесистемных параметров требует подтверждения через Polkit.

Для проверки доступа к секретам был создан тестовый профиль nm-dbus-secret-test с отключенным автоподключением, не предназначенный для фактического соединения с сетью. В профиль были добавлены параметры 802.1X-аутентификации, включая идентификатор пользователя и тестовый пароль.

Рисунок 4 – Создание тестового профиля NetworkManager с параметрами 802.1X-аутентификации

Метод GetSettings вернул структуру тестового профиля с секциями connection, 802-3-ethernet, 802-1x, ipv4, ipv6 и proxy, однако в секции 802-1x были отображены только открытые параметры аутентификации без значения пароля. При вызове GetSecrets от имени root для секции 802-1x сохраненный пароль был возвращен. Результат сравнения GetSettings и GetSecrets представлен на рисунке 5.

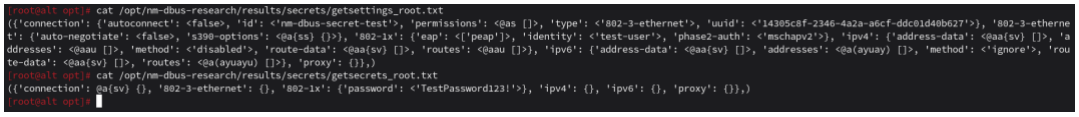

Рисунок 5 – Сравнение вывода методов GetSettings и GetSecrets для тестового профиля с параметрами 802.1X

После этого GetSecrets был вызван от имени обычного пользователя. NetworkManager не вернул секрет автоматически, а инициировал административную аутентификацию через Polkit. Следовательно, доступность профиля через D-Bus не означает доступа ко всем его данным, а защита секретов обеспечивается совместной логикой NetworkManager и Polkit.

На третьем этапе валидация входных данных проверялась через метод AddConnection от имени root. Корректный ethernet-профиль был создан успешно, а некорректные значения отклонялись с диагностикой по конкретному свойству и типу ошибки. Результаты представлены в таблице 3.

Таблица 3 - Результаты проверки валидации входных данных при создании профиля

Тестовый случай	**Передаваемое значение**	**Результат**	**Уровень проверки**
Корректный ethernet-профиль	Валидные connection, 802-3-ethernet, ipv4, ipv6	Профиль создан	Контрольный успешный вызов

Некорректный UUID	connection.uuid = not-a-valid-uuid	Ошибка InvalidProperty	Семантическая проверка идентификатора
Недопустимый метод IPv4	ipv4.method = wrong-method	Ошибка InvalidProperty	Проверка допустимого набора значений
Неизвестное свойство	connection.unknown-property	Ошибка неизвестного свойства	Проверка структуры профиля
Неверный D-Bus тип	connection.autoconnect = 'false' вместо логического значения	Ошибка несоответствия типов b и s	Проверка типа значения
Недопустимый VLAN ID	vlan.id = 4095	Ошибка диапазона 0–4094	Семантическая проверка диапазона

На четвертом этапе проверялась валидация изменения существующего профиля методами Update и Update2. Корректный вызов Update был применен успешно, а передача недопустимого ipv4.method вызвала ошибку InvalidProperty. Последующее чтение подтвердило сохранение прежних корректных параметров. Результаты представлены в таблице 4.

Таблица 4 - Результаты проверки изменения существующего профиля

Метод	**Тестовый случай**	**Результат**	**Интерпретация**
Update	Корректное изменение имени профиля	Операция выполнена успешно	Валидная структура профиля принимается
Update	ipv4.method = wrong-method	Ошибка InvalidProperty	Некорректное изменение отклоняется
Update	Проверка профиля после ошибки	Профиль сохранил прежние корректные параметры	Частичное применение ошибочной конфигурации не выполняется
Update2	Корректный вызов с flags = 0 и пустым args	Операция выполнена успешно	Метод принимает корректные служебные параметры
Update2	Строковое значение вместо числового flags	Ошибка разбора параметра типа u	Ошибка выявляется на уровне сигнатуры аргумента

Update2	Неизвестный ключ unknown-key в args	Ошибка Unsupported argument	Выполняется проверка дополнительных аргументов метода

ЗАКЛЮЧЕНИЕ

Эксперименты показали, что авторизация действий зависит от субъекта доступа и политики Polkit, секреты профилей не раскрываются через обычное чтение настроек, а данные при создании и изменении профилей проверяются по структуре, типам, допустимым свойствам и семантическим диапазонам. Безопасность взаимодействия с NetworkManager через D-Bus обеспечивается сочетанием типизированного D-Bus вызова, Polkit-авторизации и прикладной валидации внутри NetworkManager. Ограничениями исследования являются использование одной версии ОС Альт и одного набора пакетов NetworkManager, D-Bus и Polkit, а также ограниченный набор методов и типов профилей, включающий Ethernet, VLAN и параметры 802.1X без анализа VPN, Wi-Fi и пользовательских Secret Agent. Дальнейшие исследования целесообразно направить на сравнение разных Linux-дистрибутивов и версий NetworkManager, анализ Polkit-правил, расширение набора типов соединений и автоматизацию отрицательных тестов D-Bus API.

Список использованных источников

1. Прокофьева, А. С. Применение d-bus для управления GNOME в ОС Альт. Вопросы безопасности / А. С. Прокофьева, М. С. Прокофьева // Теория и практика современной науки. – 2026. – № 1(127). – С. 62-72. – EDN BQPFVV.

2. Шевченко, Д. А. Применение D-Bus для управления KDE в ОС Альт / Д. А. Шевченко, И. С. Боготов // Вестник науки. – 2025. – Т. 5, № 6-2(87). – С. 324-334. – EDN VYCEYU.

3. Потапов, А. А. Работа сетевой подсистемы в отечественных ос / А. А. Потапов, Е. В. Чулисов // Мировая наука. – 2025. – № 1(94). – С. 96-103. – DOI 10.5281/zenodo.14889548. – EDN WFMPOZ.

4. NetworkManager D-Bus API Specification [Электронный ресурс] // NetworkManager Documentation. URL: https://networkmanager.dev/docs/api/latest/spec.html (дата обращения 22.05.2026).

5. Макеев, В. В. Работа модуля PolicyKit с использованием системы межпроцессного взаимодействия D-BUS / В. В. Макеев // Цифровая трансформация образования и транспортной отрасли: анализ опыта и перспективы развития : Сборник статей Международного научно-практического IT-форума, Санкт-Петербург, 23–24 ноября 2023 года. – Санкт-Петербург: Петербургский государственный университет путей сообщения Императора Александра I, 2024. – С. 131-136. – EDN MHTHKY.

6. D-Bus Specification [Электронный ресурс] // freedesktop.org. URL: https://dbus.freedesktop.org/doc/dbus-specification.html (дата обращения 23.05.2026).

7. polkit(8): Authorization Framework [Электронный ресурс] // Linux man page. URL: https://linux.die.net/man/8/polkit (дата обращения 24.05.2026).

8. NetworkManager Settings.Connection D-Bus API [Электронный ресурс] // freedesktop.org. URL: https://people.freedesktop.org/~lkundrak/nm-dbus-api/gdbus-org.freedesktop.NetworkManager.Settings.Connection.html (дата обращения 23.05.2026).

9. NMRemoteConnection: libnm Reference Manual [Электронный ресурс] // NetworkManager Documentation. URL: https://networkmanager.dev/docs/libnm/latest/NMRemoteConnection.html (дата обращения 26.05.2026).

10. https://github.com/mirsa313/-D-Bus-NetworkManager-.git

11. CVE-2021-3560 Detail [Электронный ресурс] // National Vulnerability Database. National Institute of Standards and Technology. URL: https://nvd.nist.gov/vuln/detail/CVE-2021-3560 (дата обращения: 01.06.2026).

12. CVE-2022-42010 Detail [Электронный ресурс] // National Vulnerability Database. National Institute of Standards and Technology. URL: https://nvd.nist.gov/vuln/detail/CVE-2022-42010 (дата обращения: 01.06.2026).

13. CVE-2022-42012 Detail [Электронный ресурс] // National Vulnerability Database. National Institute of Standards and Technology. URL: https://nvd.nist.gov/vuln/detail/CVE-2022-42012 (дата обращения: 01.06.2026).

14. CVE-2023-4104 Detail [Электронный ресурс] // National Vulnerability Database. National Institute of Standards and Technology. URL: https://nvd.nist.gov/vuln/detail/CVE-2023-4104 (дата обращения: 01.06.2026).

15. Греков, В. С. Перспективы кибербезопасности в нефтегазовой отрасли / В. С. Греков, А. Г. Уймин // Губкинский университет в решении вопросов нефтегазовой отрасли России : Тезисы докладов VI Региональной научно-технической конференции, посвященной 100-летию М.М. Ивановой, Москва, 19–21 сентября 2022 года. – Москва: Российский государственный университет нефти и газа (национальный исследовательский университет) имени И.М. Губкина, 2022. – С. 1108-1109.