Nevel

БЕЗОПАСНОСТЬ

Сетевая защита и свой AI firewall

Сегментация, исходящие соединения, встраивание корпоративного средства защиты, TLS, лимиты и отсутствие обхода.

Облако / В контуре

https://docs.nevel.sh/security/network/

Nevel позволяет организовать контролируемый сетевой путь между пользователем, Platform, Guardrail и моделью. Для ИБ существенны две разные задачи: защитить канал передачи и проверить содержание обращения. VPN или TLS решает транспортную задачу, но сам по себе не анализирует prompt injection, персональные данные или допустимость вызова инструмента.

Корпоративный AI firewall встраивается в разрешённую цепочку обработки как отдельный контроль. Его адрес, аутентификация, точки расшифрования и поведение при ошибке входят в схему внедрения. Указание совместимого API позволяет подключить модельный шлюз, но не является обещанием проверенной интеграции со всеми продуктами ИБ.

Какие маршруты разрешает on-prem

On-prem определяет место Platform. Из установленной у заказчика платформы запрос по политике может идти локальной модели, российскому провайдеру, например Яндексу или Сберу, self-hosted модели Nevel в РФ либо глобальному провайдеру. Каждый внешний endpoint и промежуточный сервис включается в список разрешённых получателей. Отсутствие исходящих модельных соединений требуется для отдельного изолированного профиля.

Сетевая граница инфраструктуры и государственная граница различаются. Соединение из on-prem к российской площадке Nevel или российского провайдера является выходом из инфраструктуры заказчика; фактическая география обработки оценивается отдельно. Матрица и схема — в вариантах установки.

Сегментация и исходящие соединения

Пользовательский вход, административный доступ, модельный трафик, хранилища и доставка аудита разделяются по назначению. Базы, хранилище соответствий и источники секретов не публикуются как пользовательский API. Доступ к ним получают только необходимые сервисы и уполномоченные эксплуатационные роли.

Поток	Инициатор и получатель	Содержание	Правило
Пользовательский	Браузер → Platform	Сессия, сообщения, файлы	TLS, авторизация, ограничения загрузки
Прикладной API	Корпоративное приложение → Gateway	Ключ, запрос и параметры модели	Отдельный ключ, лимиты и обязательный защищённый маршрут
Внутренний	Platform → Guardrail и хранилища	Контекст и служебные данные	Сервисные полномочия, закрытый доступ, защита канала
Модельный	Router → разрешённый обработчик	Подготовленный запрос	Разрешённый адрес и цепочка получателей
Интеграционный	Исполнитель → корпоративный источник	Параметры инструмента и результат	Ограниченные операции и разрешённая сеть

Поток	Инициатор и получатель	Содержание	Правило
Аудитный	Audit → SIEM	Минимальные метаданные	Защищённый транспорт и контролируемый получатель
Административный	Уполномоченный инженер → управление	Настройки и эксплуатационные команды	Отдельный доступ, ограничение времени и журнал

Исходящие соединения ограничиваются перечнем необходимых получателей. Разрешение «любой адрес по HTTPS» не доказывает отсутствие утечки: HTTPS защищает отправку в том числе запрещённому получателю. Правило оценивает назначение соединения и контекст операции, включая переадресации и обращения фоновых обработчиков.

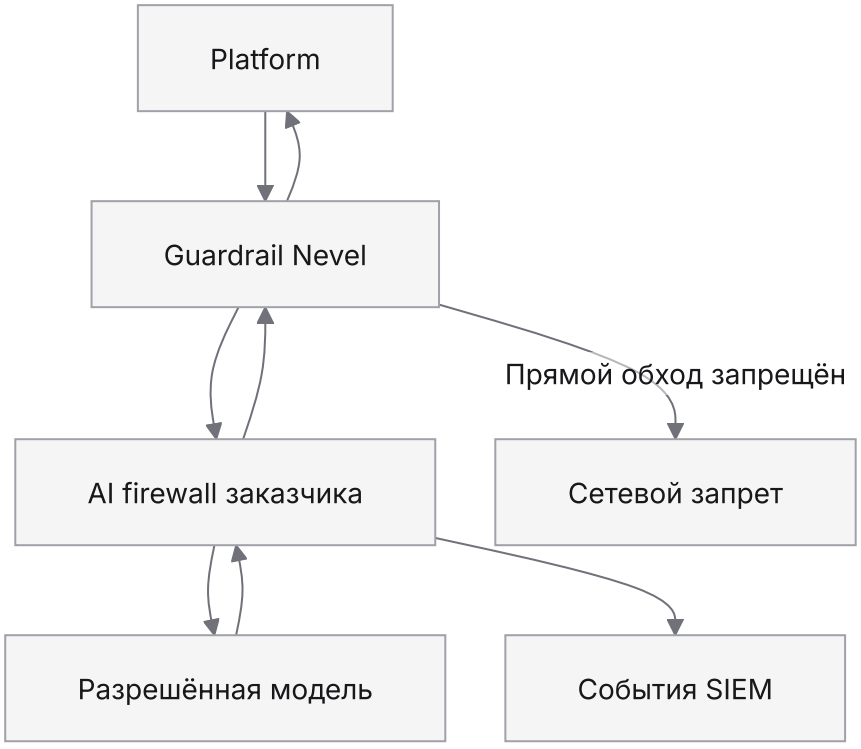
Как подключается собственный AI firewall

Средство заказчика может стоять перед модельным endpoint как reverse proxy или совместимый шлюз. Platform направляет модельный трафик на разрешённый адрес средства, оно выполняет проверки и передаёт запрос далее. Сетевая политика препятствует прямому обходу этого адреса. Ответ возвращается через ту же контрольную цепочку.

Для российских SIEM, DLP, сетевых шлюзов и IdP см. корпоративную среду: там разделены документированные интерфейсы продуктов, точки до/после маскирования и условия проверки интеграции. ICAP-сервис DLP, модельный reverse proxy и приёмник событий SIEM имеют разные контракты; один адрес нельзя подставлять вместо другого. Пошаговая доставка событий описана в интеграции с SIEM.

СХЕМА 1

Корпоративное средство в модельном маршруте



Место установки определяет видимое содержание. После маскирования средство анализирует подготовленный текст и не видит заменённые исходные значения. До маскирования оно видит оригинал и становится обработчиком соответствующих чувствительных данных. Если требуется контроль в обе стороны до восстановления, средство видит ответ с масками; восстановленный ответ доступен уже на внутреннем участке.

Точка завершения TLS также влияет на видимость. Средство, которое только пересылает зашифрованные байты без расшифрования, не анализирует текст модельного запроса. При TLS inspection настраивается доверенная цепочка сертификатов; отключение проверки сертификатов в клиентах не считается корректной интеграцией.

Что проверяется на уровне протокола

Совместимость проверяется на реальных методах и форматах, которые использует приложение. В минимальный набор входят обычный ответ, потоковая передача, ошибки, большие входы, отмена, вызовы инструментов и передача поддерживаемых вложений. Поля, неизвестные средству, не должны незаметно удалять ограничения или менять смысл запроса.

Условие	Ожидаемое поведение интеграции
Обычный текстовый запрос	Сохраняются модель, сообщения, параметры и идентификатор связи
Потоковый ответ	Средство корректно обрабатывает фрагменты, завершение и разрыв соединения
Вызов инструмента	Структура вызова сохраняется, полномочия действия проверяет сервер
Ограничение размера	Возвращается понятный отказ до неконтролируемой обработки
Превышение лимита	Клиент получает сигнал ограничения без бесконечной немедленной повторной отправки
Тайм-аут проверки	Обязательная проверка не обходится прямым вызовом модели
Ошибка аутентификации	Секрет не раскрывается в ответе, новая попытка не использует чужой ключ
Запрет модели или категории	Ошибка не превращается в автоматический выбор запрещённого получателя

Идентификатор запроса проходит цепочку или связывается с идентификаторами отдельных компонентов. Это позволяет сопоставить решение Guardrail, AI firewall и модели. При диагностике не требуется передавать исходные значения масок в заголовке для удобства корреляции.

Fail-closed и доступность

Для обязательного контрольного компонента применяется отказ без обхода: если проверка недоступна или не завершилась, непроверенное содержание не отправляется дальше. Возможен резервный экземпляр того же разрешённого средства или другой заранее разрешённый маршрут с эквивалентными ограничениями. Внешняя модель не становится допустимой только потому, что внутренняя перегружена.

Важна разница между отказом проверки и отрицательным решением. Тайм-аут означает отсутствие достаточного результата, а блокировка — явный запрет по правилу. Оба события должны быть различимы в аудите. Пользователь получает безопасное описание и идентификатор, а техническая команда — причину на соответствующем уровне доступа.

Если контур использует несколько endpoint, перечисляются все пути: обычный и защищённый API, интерфейс, фоновые задачи, файловые обработчики и инструменты. Закрытие только одного URL оставляет обход через другой маршрут. Защита обязательна на уровне фактической операции, а не только выбранной кнопки интерфейса.

Лимиты и защита доступности

Контролируются размер входа, число параллельных задач, частота запросов, лимит пользователя или ключа и стоимость обработки. Для файлов отдельно важен объём после распаковки и извлечения. Маленький архив может породить большой объём данных; короткий запрос — запустить многократные инструменты и модельные обращения.

Числовые лимиты задаются для конкретной поставки, оборудования и задач. Единое значение максимального файла или запросов в секунду не применяется ко всем установкам. Приёмка фиксирует применённые пределы и проверяет поведение на границе, чуть выше границы и при конкурентной нагрузке нескольких пользователей.

Сервис возвращает ограничение до истощения ресурсов, сохраняет управляемость и не перекладывает весь бюджет на соседнее пространство. Повторные попытки клиента имеют ограничение и задержку; бесконечный автоматический retry способен усилить перегрузку или повторить платную внешнюю обработку.

SSRF, ссылки и инструменты

Внешняя ссылка из документа или ответа модели не считается разрешённым адресом автоматически. Исполнитель инструмента ограничивает схемы, домены и доступ к внутренним служебным адресам. Проверяются конечный адрес после перенаправления и возможность обращения к инфраструктурным метаданным. Права на веб-поиск не дают права обращаться к произвольному внутреннему сервису.

Подключение к корпоративной базе или API использует назначенный адрес и ограниченные операции. Модель не меняет адрес сервера или учётные данные через содержимое запроса. В обычном on-prem инструменты могут обращаться к внутренним и внешним ресурсам по политике. В изолированном профиле рабочие инструменты используют только ресурсы внутри заданного контура; внешняя ссылка не должна приводить к скрытому выходу.

Облачные получатели и OpenRouter

Для API-маршрутов, использующих OpenRouter, этот сервис включается в цепочку получателей наряду с конечным модельным провайдером. Необходимо учитывать фактический запрос, доступные настройки хранения и географию каждого участника. Прямой вызов провайдера и вызов через дополнительный шлюз имеют разные границы доверия; утверждение «все обращения идут только напрямую» не применяется ко всей системе.

Использование собственного ключа провайдера меняет способ авторизации и учёта, но не устраняет сетевой путь и получателей. Требование обработки только в РФ и запрет выхода из инфраструктуры заказчика являются разными ограничениями: первое проверяет географию всей цепочки, второе исключает и российские внешние сервисы. Ключ пользователя не может обойти ни одно из этих правил.

Негативные проверки

Отключите обязательный AI firewall, DLP и разрешённую модель по очереди. Убедитесь, что исходный запрос не ушёл напрямую и что событие объясняет стадию отказа. Затем попробуйте альтернативный endpoint, изменение адреса, запрещённую переадресацию, недоверенный сертификат, запрос выше лимита и повтор после блокировки.

Для изолированного профиля запустите текст, скан, индексирование файла, инструмент и создание артефакта при запрещённом внешнем выходе. Сетевые журналы должны подтвердить внутренний маршрут каждой операции. Документируйте версию средства ИБ, правила, конфигурацию Nevel и фактические результаты — этого требует проверка интеграции, а не одного заявления о совместимом API.

Смежные темы: архитектура, шифрование, аудит, внешнее API.