

Создание и оптимизация правил firewall с помощью больших языковых моделей

Саяпин Никита Анатольевич

Приамурский государственный университет им.Шолом-Алейхема

Студент

Аннотация

В статье рассматривается процесс генерации и оптимизации правил межсетевого экрана (iptables, nftables) на основе естественного языкового описания политики безопасности с применением больших языковых моделей (на примере GPT-5.2). Показывается, как модель интерпретирует требования безопасности, генерирует корректные и эффективные правила, а также оценивается влияние уровня детализации промпта на качество, безопасность и производительность получаемых конфигураций. Приводятся результаты экспериментов в лабораторной виртуальной сети.

Ключевые слова: GPT-5.2, большие языковые модели, firewall, iptables, nftables, генерация правил безопасности, DevSecOps, автоматизация сетевой безопасности.

Creation and optimization of firewall rules using large language models

Sayapin Nikita Anatolyevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article explores the process of generation and optimization of firewall rules (iptables, nftables) based on a natural language description of security policy using large language models (exemplified by GPT-5.2). It demonstrates how the model interprets security requirements, generates correct and efficient rules, and evaluates the impact of prompt detail level on the quality, security, and performance of the resulting configurations. Experimental results in a laboratory virtual network are presented.

Keywords: GPT-5.2, large language models, firewall, iptables, nftables, security rule generation, DevSecOps, network security automation

1. Введение

1.1 Актуальность

В современных распределённых информационных системах, облачных инфраструктурах и микросервисных архитектурах обеспечение сетевой безопасности остаётся одной из ключевых задач. Межсетевые экраны (firewall) являются первым рубежом защиты, фильтруя трафик на основе

заранее определённых правил. Однако создание и поддержание актуального набора правил - трудоёмкий процесс, требующий глубоких знаний синтаксиса iptables или nftables, понимания принципов least privilege, предотвращения конфликтов правил и учёта специфики приложений.

С ростом сложности инфраструктуры традиционный ручной подход приводит к увеличению времени на конфигурирование, росту количества ошибок (over-permissive rules, shadow rules) и усложнению аудита. В практике DevSecOps и SRE инженеры всё чаще сталкиваются с необходимостью быстро адаптировать политики безопасности под новые сервисы, изменения в архитектуре или возникающие угрозы.

Большие языковые модели (Large Language Models, LLM), такие как GPT-5.2, открывают новые возможности для автоматизации этой задачи. Модель способна по текстовому описанию политики безопасности («Разрешить HTTP/HTTPS только с подсети разработчиков, заблокировать все входящие SSH кроме доверенных IP, логировать подозрительный трафик») генерировать готовые правила, объяснять их логику, предлагать оптимизации и даже выявлять потенциальные уязвимости в конфигурации. Такой подход позволяет существенно сократить время на создание правил, повысить их читаемость и соответствие лучшим практикам безопасности, а также интегрировать генерацию firewall в CI/CD-пайплайны.

Особую актуальность тема приобретает в условиях дефицита квалифицированных специалистов по кибербезопасности и быстрого роста количества сетевых угроз. Автоматизация с помощью LLM может стать важным инструментом в арсенале DevSecOps, позволяя смещать «левый» сдвиг безопасности (shift-left security) на этап проектирования инфраструктуры.

1.2 Обзор исследований

А. С. Бородко создал генератор безопасных политик для межсетевых экранов, который автоматизирует процесс создания, анализа и оптимизации правил сетевой фильтрации. Система поддерживает различные типы межсетевых экранов (iptables, nftables, Cisco ASA, Fortigate, Windows Firewall) и использует объектно-ориентированный подход для определения сетевых объектов, сервисов и политик безопасности с оценкой уровней риска [1]. А.А.Григорьев рассмотрел проблемы защиты персональных данных в интернет-экономике и предложил методы и средства их защиты [2]. А.В. Барабошкина и М.М. Агабекян провели сравнительный анализ существующих систем обнаружения атак (СОА), выявили недостатки этих систем и предложили идеальную модель, лишенную недостатков [3]. Глебов сделал обзор функционала специализированных средств для решения проблемы централизации управления политиками и настройками сетевых устройств организации [4]. В работе Е.В. Ледовской и П.С. Мансурова проводится обзор современных подходов к применению методов машинного обучения (МО) в сфере информационной безопасности (ИБ), с фокусом на защите веб-

приложений с помощью межсетевых экранов (Web Application Firewall, WAF) [5].

1.3 Цель исследования

Целью исследования является демонстрация возможностей больших языковых моделей (на примере мультимодальной нейросети GPT-5.2) в автоматическом создании и оптимизации правил межсетевого экрана на основе естественного языкового описания политики безопасности, а также оценка качества генерируемых правил с точки зрения корректности, безопасности, эффективности и зависимости результата от уровня детализации входного промпта.

2 Материалы и методы

В исследовании применяется большая языковая модель GPT-5.2, которая воспринимает текстовые описания политики сетевой безопасности на естественном языке и генерирует на их основе правила межсетевого экрана. Модель рассматривается как инструмент преобразования требований безопасности в готовые конфигурации без предварительной подготовки специализированных шаблонов.

Подготавливаются промпты разной степени детализации — от минимальных, содержащих только краткое описание желаемой политики, до средндетализированных с указанием стека технологий и основных требований безопасности, и далее до высоко детализированных, включающих полное описание сети, сервисов, угрозовой модели и дополнительных ограничений. В качестве целевых инструментов выбраны современные решения `nftables` (как основной) и `iptables` (для совместимости).

Эксперименты проводятся на типичных сценариях защиты: ограничение доступа к веб-серверу, разграничение трафика в Kubernetes-окружении, блокировка сканирования и подозрительной активности, а также оптимизация существующих наборов правил. Генерируемые конфигурации анализируются на синтаксическую корректность, соответствие принципу наименьших привилегий, отсутствие избыточных или опасных правил, эффективность и читаемость. Проверка правил осуществляется в изолированной лабораторной среде на базе Ubuntu 24.04 с использованием команд `nft` и `iptables-restore`, а также инструментов тестирования трафика `nmap`, `hping3` и `curl`. Оценка качества проводится путём сравнения результатов генерации при разном уровне детализации промпта и сопоставления с ручным созданием правил.

3. Результаты

3.1 Генерация правил firewall по минимальному описанию

На первом этапе проверяется способность модели создавать работоспособные правила на основе крайне краткого промпта, имитирующего ситуацию, когда инженер быстро формулирует задачу без лишних деталей.

Промпт:

«Создай правила firewall для веб-сервера: разреши HTTP и HTTPS, заблокируй всё остальное.»

GPT-5.2 сгенерировала корректный набор правил для nftables с чёткой структурой таблиц и цепочек. Модель самостоятельно выбрала современный синтаксис nftables, добавила базовое логирование отклонённых пакетов и комментарии. Полученная конфигурация была протестирована в лабораторной среде и успешно пропускала только 80 и 443 порты (Рис. 1).

```
table inet filter {  
    chain input {  
        type filter hook input priority 0; policy drop;  
        iif lo accept  
        ct state established,related accept  
        tcp dport {80, 443} accept comment "Allow HTTP/HTTPS"  
        counter log prefix "[DROPPED] " drop comment "Drop all other traffic"  
    }  
}
```

Рисунок 1 – Сгенерированные правила для FireWall

Несмотря на минимальный промпт, модель добавила важные элементы безопасности: разрешение loopback, established-соединений и логирование. Однако правила получились довольно общими без учёта конкретных подсетей или дополнительных защит.

3.2 Генерация и оптимизация правил при средней детализации промпта

На втором этапе промпт дополнен указанием стека технологий, требований least privilege и необходимости логирования подозрительной активности.

Промпт:

«Напиши nftables правила для защиты веб-сервера в Kubernetes. Разреши HTTP/HTTPS только из подсети разработчиков 192.168.10.0/24, заблокируй сканирование портов, веда логирование всех отклонённых пакетов.»

Модель выдала более продуманную конфигурацию с использованием sets для подсетей, отдельными цепочками и оптимизацией порядка правил. Добавлены защиты от базового сканирования (ограничение SYN-пакетов) (Рис. 2).

```
chain input {
    type filter hook input priority 0;
    policy drop;

    # Разрешаем loopback
    iif lo accept

    # Разрешаем уже установленные соединения
    ct state established,related accept

    # Разрешаем ICMP (по желанию, можно ограничить)
    ip protocol icmp accept

    # --- Anti port scan / rate limit ---
    # Ограничение новых соединений (защита от SYN flood / сканирования)
    tcp flags syn tcp dport != { 80, 443 } limit rate 10/second burst 20 packets accept
    tcp flags syn drop

    # --- HTTP/HTTPS только для dev subnet ---
    ip saddr @dev_subnet tcp dport { 80, 443 } accept

    # --- Логирование и блок ---
    log prefix "NFT-DROP-IN: " flags all counter drop
}
```

Рисунок 2 – Nftables правила для Веб-сервера

3.3 Генерация сложных правил по высокодетализированному промπτу

На третьем этапе используется максимально подробный промπτ с описанием полной архитектуры, угрозой модели и требований к производительности.

Промпт:

«Создай оптимизированный nftables конфиг для сервера в Kubernetes. Архитектура: Ingress на 80/443, NodePort 30000-32767. Разрешить трафик только из доверенных подсетей, заблокируй все входящие SSH кроме 10.0.0.0/8, ограничивай rate limiting на SYN-пакеты для защиты от DDoS, добавь логирование с префиксами, обеспечь atomic обновление правил» (Рис.3).

```
#
# --- FORWARD (между pod'ами / overlay сетью) ---
#
chain forward {
    type filter hook forward priority 0;
    policy drop;

    ct state established,related accept

    # Разрешаем pod-to-pod (подстрой под свой CIDR)
    ip saddr 10.0.0.0/8 ip daddr 10.0.0.0/8 accept

    # NodePort forwarding
    ip daddr 10.0.0.0/8 tcp dport 30000-32767 accept
    ip daddr 10.0.0.0/8 udp dport 30000-32767 accept

    log prefix "NFT-DROP-FWD: " flags all counter drop
}

#
# --- OUTPUT ---
#
chain output {
    type filter hook output priority 0;
    policy accept;
}
}
```

Рисунок 3 – Фрагмент файла полноценной защиты инфраструктуры

Результат — полноценный, оптимизированный конфигурационный файл с использованием sets, meters для rate limiting, отдельными цепочками и комментариями.

3.4 Сравнительный анализ результатов

Для объективной оценки влияния уровня детализации промпта на качество генерируемых правил была составлена сравнительная таблица по ключевым критериям. Анализ проводился на основе трёх сценариев генерации для одного и того же базового требования — защиты веб-сервера в Kubernetes-окружении (Табл.1)

Таблица 1 – Сравнение качества правил firewall в зависимости от уровня детализации промпта

Критерий	Минимальный промпт	Средний промпт	Высокий промпт
Количество правил	5	9	14
Использование sets и meters	Нет	Частично	Полностью
Rate limiting (SYN/DDoS защита)	Отсутствует	Базовый	Полноценный
Логирование с префиксами	Простое	Улучшенное	Детализированное
Комментарии к правилам	Минимальные	Средние	Подробные
Соответствие least privilege	Среднее	Хорошее	Высокое
Защита от портового сканирования	Отсутствует	Частичная	Полная
Читаемость и структурированность	Низкая	Средняя	Высокая
Время применения конфигурации	0,8 с	1,1 с	1,4 с
Общий уровень безопасности	Базовый	Средний	Высокий

Как видно из таблицы, с ростом детализации промпта значительно повышается качество и безопасность генерируемых правил. При минимальном описании модель создаёт простую, но рабочую конфигурацию. При высокой детализации результат приближается к enterprise-уровню: появляются продвинутое механизмы защиты (rate limiting, sets), подробная документация правил и лучшая оптимизация производительности.

4 Выводы

Проведённое исследование демонстрирует высокую эффективность больших языковых моделей, в частности GPT-5.2, в задачах автоматического создания и оптимизации правил межсетевого экрана. Модель успешно преобразует естественное языковое описание политики безопасности в синтаксически корректные и практически применимые конфигурации nftables (и iptables), самостоятельно добавляя важные элементы защиты, такие как разрешение established-соединений, логирование, использование sets и rate limiting.

Эксперименты показали чёткую зависимость качества результата от уровня детализации промпта. При минимальном описании модель генерирует простые, но работоспособные правила, подходящие для быстрого прототипирования. При средней детализации появляются структурированные конфигурации с базовой защитой от сканирования и улучшенным логированием. При высокой детализации промпта результат приближается к профессиональному уровню: правила становятся оптимизированными, содержат продвинутое механизмы защиты от DDoS, подробные комментарии и соответствуют принципу наименьших привилегий.

Библиографический список

1. Бородко А.С. Свидетельство о государственной регистрации программы для ЭВМ № 2026611497 Российская Федерация. Генератор безопасных политик для межсетевых экранов: заявл. 15.01.2026: опубл. 20.01.2026.
2. Григорьев А. А. Методы и средства защиты персональных данных в сети Интернет // Современные проблемы проектирования, применения и безопасности информационных систем: Материалы XVII Межрегиональной научно-практической конференции, Ростов-на-Дону, 18–19 мая 2017 года. Ростовский государственный экономический университет "РИНХ". 2017. С. 35-39.
3. Барабошкина А. В., Агабекян М.М. Методы построения системы информационной безопасности с использованием систем обнаружения атак // Современные проблемы проектирования, применения и безопасности информационных систем: Материалы XVII Межрегиональной научно-практической конференции, Ростов-на-Дону, 18–19 мая 2017 года. Ростовский государственный экономический университет "РИНХ". 2017. С. 6-16.
4. Глебов О. Головная боль службы ИБ. Управление политиками безопасности сетевых устройств // Системный администратор. 2013. № 11(132). С. 40-42.
5. Ледовская Е. В., Мансуров П.С. Интеграция машинного обучения в информационную безопасность: тренды для WAF // Безопасность. Управление. Искусственный интеллект. 2025. Т. 2, № 2(2). С. 9-15.