УДК 004

Тумгоев М.Р.

студент физико-математического факультета ИнгГУ

ФГБОУ ВО «Ингушский государственный университет» г. Магас

Пиакартов М.Х.

научный руководитель, кандидат педагогических наук, старший преподаватель кафедры «ИСиТ»

ФГБОУ ВО «Ингушский государственный университет» г. Магас

ИССЛЕДОВАНИЕ МЕТОДОВ ЗАЩИТЫ API ОТ DDOS-АТАК

Аннотация: В статье рассматриваются методы защиты API от распределённых атак типа «отказ в обслуживании» (DDoS). Анализируются основные виды атак, их влияние на работу сервисов и бизнес-процессы. Описаны ключевые подходы к защите: использование сетей доставки контента (CDN), ловушек для злоумышленников (honeypot), ограничение частоты запросов (rate limiting), Web Application Firewall с парсингом API-трафика и API Gateway со встроенными очередями. Особое внимание уделяется необходимости многоуровневой эшелонированной защиты. Статья предназначена для разработчиков, архитекторов и администраторов, интересующихся вопросами безопасности API.

Ключевые слова: API, DDoS-атака, защита API, CDN, сеть доставки контента, honeypot, ловушка для злоумышленников

Tumgoev M.R.

Student of the Physics and Mathematics Department at IngSU

Ingush State University, Magas

Piakartov M.K.

Research supervisor, Candidate of Pedagogical Sciences, Senior Lecturer at the Department «ISaT»

Ingush State University, Magas

METHODS FOR RESEARCHING API PROTECTION AGAINST DDOS ATTACKS

Abstract: This article examines methods for protecting APIs from distributed denial-of-service (DDoS) attacks. It analyzes the main types of attacks and their impact on services and business processes. Key approaches to protection are described, including the use of content delivery networks (CDNs), honeypots, rate limiting, Web Application Firewalls with API traffic

parsing, and API Gateways with built-in queues. Particular attention is paid to the need for multi-layered, in-depth defense. This article is intended for developers, architects, and administrators interested in API security.

Keywords: API, DDoS attack, API protection, CDN, content delivery network, honeypot.

API являются основой современных приложений, обеспечивая работу всего, от мобильных приложений до устройств IoT. Однако их критически важная роль также делает их главной мишенью для распределенных атак типа «отказ в обслуживании» (DDoS) , которые могут нарушить работу сервисов, нанести ущерб бизнесу и негативно повлиять на пользовательский опыт. Защита API от DDoS-атак необходима для поддержания бесперебойной работы, обеспечения целостности данных и сохранения доверия клиентов.

Что такое DDoS-атаки и как они влияют на API?

Распределенная атака типа «отказ в обслуживании» (DDoS) - это злонамеренная попытка перегрузить сервер или API огромным количеством запросов, сделав их недоступными для законных пользователей. В таких атаках часто используются ботнеты — сети скомпрометированных устройств - для генерации огромного трафика из множества источников, что затрудняет блокировку атаки только по IP-адресу.

Почему именно API становятся объектом пристального внимания?

API являются высокоприоритетными целями, поскольку служат шлюзами к критически важным системам и данным. Они обеспечивают работу таких важных сервисов, как мобильные приложения, платежные системы и интеграции, поэтому простои могут привести к серьезным сбоям. Злоумышленники используют API для следующих целей:

- Вызывает сбои в работе сервиса.
- Ресурсы выхлопной системы.
- Похищение данных или использование уязвимостей.
- Простои: DDoS-атаки могут сделать API недоступными, что приводит к потере дохода и снижению доверия клиентов.
- Истощение ресурсов: API, подвергающиеся атакам, могут потреблять чрезмерное количество вычислительных ресурсов и пропускной способности, что повлияет на работу других сервисов.

- Ущерб репутации: Длительные сбои или низкая производительность могут нанести ущерб репутации вашего бренда.

Перейдём непосредственно к рассмотрению методов защиты

Существуют способы фильтрации запросов. Например, CDN скрывает исходный код вашего приложения, одновременно предоставляя данные на уровне приложения с использованием кэшированного контента. Она работает как средство защиты вышестоящего уровня безопасности, фильтруя запросы к вашим приложениям и помогая пользователям получать данные с низкой задержкой благодаря кэшированию. Вы можете использовать сторонние инструменты, предлагающие решения CDN, такие как AWS CloudFront . Тем не менее, полезно иметь минимальный план реагирования до обращения к вашему интернет-провайдеру. Размещение ваших пользовательских сервисов, получающих доступ к веб-контенту, например, видео и музыке, в защищенном кэшированном хранилище также может помочь.

Этот подход фильтрует трафик до того, как он достигнет вашей сети, что упрощает управление серверами. Но для защиты в случае обнаружения и компрометации вашей среды всё равно необходимы дополнительные меры. Именно здесь может помочь ловушка для злоумышленников (honeypot).

Метод Honeypot (ловушка для злоумышленников)

Honeypot, или ловушка для злоумышленников, — это один из самых необычных и в то же время эффективных методов защиты цифровых систем. Суть его проста: администратор намеренно создаёт ложную цель — сервер, API или целый сегмент сети, который выглядит как настоящий и содержит нарочно оставленные уязвимости. Когда злоумышленник начинает атаку, он попадает не на реальную систему, а именно в эту ловушку.

Главная сила honeypot в том, что он работает как обманный манёвр. Атакующий тратит время и ресурсы на взлом ложной цели, а защитники в это время получают уникальную возможность наблюдать за его действиями в реальном времени, фиксировать используемые инструменты и методы, а также изучать новые типы атак без риска для реальной инфраструктуры. По сути, honeypot превращает злоумышленника из охотника в добычу.

Honeypot редко используется как самостоятельное решение. Обычно он встраивается в многоуровневую систему защиты: наряду с файрволами,

системами обнаружения вторжений и антивирусами. Его задача — не столько предотвратить атаку, сколько вовремя её обнаружить, изучить и выиграть время для реагирования. В этом смысле honeypot — это не броня, а скорее сигнализация с функцией отвлечения.

Тем не менее, для организаций, работающих с особо ценными данными — финансовыми учреждениями, государственными структурами, крупными технологическими компаниями, — honeypot становится практически необходимым инструментом. Он позволяет быть на шаг впереди злоумышленников и превращает пассивную защиту в активную разведку.

Ограничение сетевых ресурсов

Вы можете настроить сетевой контроллер таким образом, чтобы он обрабатывал максимальный трафик за сессию. Ограничение скорости может осуществляться как программным, так и аппаратным способом. Первый управляет количеством одновременных вызовов, а второй - конфигурацией коммутатора и маршрутизатора. Ограничение скорости сетевых ресурсов гарантирует работоспособность вашего приложения, несмотря на то, что некоторые пользователи могут испытывать повышенную задержку из-за атак на ваши сервисы.

Как ловушка для вредоносных программ и CDN могут улучшить вашу защиту

Как уже упоминалось, CDN будет предоставлять контент на уровне вашего приложения, охватывая лишь одну часть вашего плана безопасности. Вы можете извлечь выгоду из использования ловушки для вредоносных программ в качестве первой поверхности атаки, и она должна находиться в контролируемой среде, где размещено ваше приложение. Ваш план безопасности должен использовать сочетание сервисов, ориентированных на различные области применения приложений, а принцип руководства по безопасности усиливает безопасность взаимосвязанных частей. Таким образом, сочетание вашей CDN и ловушки для вредоносных программ может помочь вашей команде реализовать существующий план реагирования, смягчая замедление и недоступность ваших сервисов. Это даст вам достаточно времени, чтобы более безопасно подтвердить снижение эффективности, не создавая новых угроз.

Особого внимания заслуживает метод Web Application Firewall с парсингом API-трафика который является критически важным для современной защиты. В отличие от традиционных WAF, настроенных на веб-сайты, API-ориентированный WAF анализирует структуру запросов в форматах JSON, XML или GraphQL, проверяя их на соответствие ожидаемой схеме. Это необходимо потому, что многие DDoS-атаки на прикладном уровне (L7) маскируются под легитимный трафик: злоумышленник отправляет корректно сформированные, но логически «тяжёлые» запросы — например, с чрезмерно глубокой вложенностью объектов или рекурсивными структурами, — которые заставляют сервер выполнять огромный объём вычислений при небольшом размере самого запроса. Обычный rate limiting здесь бессилен, а WAF способен обнаружить аномалии, блокировать подозрительные запросы и даже перенаправлять их в honeypot. Однако у метода есть ограничения: парсинг каждого запроса увеличивает задержку, а поддержание актуальной схемы API требует постоянного внимания со стороны разработчиков. Тем не менее, в сочетании с CDN, honeypot и ограничением частоты запросов WAF с парсингом API-трафика закрывает критическую уязвимость на прикладном уровне, без чего защита API от современных DDoS-атак остаётся неполной.

Ещё одним важнейшим методом защиты, является использование API Gateway со встроенными механизмами лимитирования и очередей. API Gateway выступает единой точкой входа для всего API-трафика, что делает его идеальным местом для применения политик ограничения частоты запросов (rate limiting) на разных уровнях: можно задать жёсткие лимиты для анонимных пользователей, более щадящие - для авторизованных клиентов, и отдельные - для конкретных эндпоинтов (например, более строгие для тяжёлых операций вроде поиска или экспорта данных). Однако по-настоящему надёжную защиту обеспечивает дополнение лимитов механизмом очередей. Когда поток запросов превышает допустимую нагрузку, вместо того чтобы отклонять лишние запросы с ошибкой, API Gateway помещает их в очередь с фиксированной глубиной и таймаутами, позволяя бэкенду обрабатывать запросы в комфортном для него темпе. Это сглаживает пиковые нагрузки, защищает бэкенд-сервисы от внезапных всплесков трафика и даёт время для автоматического масштабирования. При этом клиент не получает отказа мгновенно, а ожидает обработки в пределах заданного времени. Такой подход

особенно эффективен в сочетании с паттернами Circuit Breaker и Retry с экспоненциальной задержкой - когда перегруженный сервис «отключается» на время, давая себе восстановиться, а клиенты повторяют запросы с увеличивающимися интервалами. К недостаткам метода можно отнести необходимость настройки параметров очередей под конкретную бизнес-логику (слишком маленькая очередь будет отбрасывать запросы, слишком большая — создавать недопустимые задержки), а также дополнительную нагрузку на сам API Gateway. Тем не менее, современные решения вроде Kong, NGINX, AWS API Gateway или Envoy предоставляют все необходимые инструменты, и их грамотное использование превращает API Gateway из потенциального узкого места в мощный щит против DDoS-атак на прикладном уровне.

В данном исследовании мы подробно рассмотрели, как защитить API от DDoS-атак. Мы увидели, что эти атаки могут серьезно нарушить работу сайтов и приложений, лишив пользователей доступа к нужным сервисам. Чтобы этого избежать, нужен комплексный подход.

Мы изучили разные методы защиты, от простых, вроде ограничения количества запросов от одного пользователя, до более сложных, использующих *машинное обучение* для выявления подозрительной активности. Каждый из этих методов имеет свои плюсы и минусы. Например, ограничение скорости помогает справиться с простыми атаками, но может помешать обычным пользователям, если настроено слишком строго.

Важно понимать, что ни один метод сам по себе не является панацеей. Только сочетание различных технических решений, таких как специализированные сетевые экраны (WAF) и сервисы доставки контента (CDN), вместе с четкими организационными правилами – например, своевременное обновление систем и обучение персонала – сможет обеспечить надежную защиту. Это значит, что нужно не только установить программы, но и правильно их настроить и следить за их работой.

Также мы упомянули, что эта область постоянно развивается. Появляются новые виды атак, и, соответственно, разрабатываются новые методы противодействия. Поэтому важно не останавливаться на достигнутом, а продолжать изучать новые технологии и совершенствовать существующие стратегии защиты. Это поможет держать наши онлайн-сервисы в безопасности.

Список литературы

1. **Cloudflare**. What is a DDoS attack? [Электронный ресурс]. URL: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/ (дата обращения: 06.04.2026).
2. **AWS**. Amazon CloudFront Documentation. [Электронный ресурс]. URL: https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/Introduction.html (дата обращения: 06.04.2026).
3. **Kaspersky**. Что такое honeypot и как он работает? [Электронный ресурс]. URL: https://usa.kaspersky.com/resource-center/threats/what-is-a-honeypot (дата обращения: 06.04.2026).
4. **NIST**. Glossary: Vulnerability. [Электронный ресурс]. URL: https://csrc.nist.gov/glossary/term/vulnerability (дата обращения: 06.04.2026).
5. **Wikipedia**. Rate limiting. [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Rate_limiting (дата обращения: 06.04.2026).
6. **Infoblox**. Layer 7 of the OSI Model: Application Layer. [Электронный ресурс]. URL: https://www.infoblox.com/glossary/layer-7-of-the-osi-model-application-layer/ (дата обращения: 06.04.2026).
7. **Cloud Security Alliance**. Security Guidance for Critical Areas of Focus in Cloud Computing. [Электронный ресурс]. URL: https://cloudsecurityalliance.org/research/guidance/ (дата обращения: 06.04.2026).
8. **Hosain, M., Shuvo, S. A., Ogbe, M., et al.** Web Technologies Security in the AI Era: A Survey of CDN-Enhanced Defenses. *2025 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)*, 2025, pp. 180-186.