RitAPI

DDOS & Abuse Shield
Mitigation Playbook

BY

IT Security is
Our Main Concern

Sydeco

Introduction

The landscape of Distributed Denial of Service (DDoS) attacks has evolved significantly. Attackers are no longer limited to simple volumetric attacks that saturate network bandwidth. Modern threats now frequently target the application layer (Layer 7), using sophisticated, low-volume requests that mimic legitimate user traffic to exhaust server resources like CPU, memory, and database connections. These attacks are harder to detect and can bring critical services down with a fraction of the traffic.

RitAPI provides an API-aware defense mechanism specifically designed for this modern threat environment. Unlike traditional firewalls that are blind to API context, RitAPI understands your API's structure, endpoints, and expected traffic patterns. This allows it to distinguish malicious abuse from legitimate user activity and apply surgical mitigation strategies without impacting genuine users.

Types of Attacks Covered

RitAPI's shield is engineered to protect against a wide spectrum of API-specific attacks.

1. Volumetric Floods

- **Description**: The classic DDoS attack, aimed at overwhelming a network's bandwidth or a server's connection capacity with a massive volume of traffic. Examples include SYN floods, UDP floods, and ICMP floods.
- **Impact**: Prevents legitimate users from accessing any services by completely saturating the "pipe" to your infrastructure.

2. Bot Scraping

- **Description**: An application-layer attack where automated bots make a high frequency of requests to API endpoints to harvest data, reverse-engineer pricing, or steal intellectual property. This traffic can appear legitimate on a per-request basis but is identifiable by its high rate and repetitive nature.
- **Impact**: Degrades performance for real users, increases infrastructure costs, and can lead to the loss of proprietary data.

3. Low-and-Slow Abuse

- **Description**: A subtle and dangerous form of attack where a malicious actor sends a small number of seemingly valid requests over a long period. This technique is designed to fly under the radar of traditional rate-limiting tools. Examples include credential stuffing attacks on login endpoints, slow enumeration of user IDs, or resource-intensive API calls that slowly exhaust server resources.
- **Impact**: Can lead to account takeovers, data breaches, and gradual service degradation that is difficult to diagnose.

Mitigation Strategies

RitAPI employs a multi-layered, configurable defense model to precisely counter these threats.

1. Rate Limiting (Per Endpoint / IP / Token)

- **Function**: Enforces a maximum number of requests allowed within a specific time window. This is the first line of defense against brute-force and high-frequency bot attacks.
- **Granularity**: RitAPI allows you to define unique limits based on:

➤ **Source IP**: Limit requests from a single client.
➤ **API Token/Key**: Enforce usage quotas for specific users or applications.
➤ **Endpoint**: Apply stricter limits to sensitive or resource-intensive endpoints (e.g., `/auth/login`, `/search`).

2. Burst Windows

- **Function**: Allows legitimate traffic to temporarily exceed the defined rate limit for short periods. This "burst" capacity accommodates natural traffic spikes (e.g., a marketing campaign launch) without penalizing users, while still blocking sustained, high-volume attacks.
- **Use Case**: If your rate limit is 100 requests/second, you can configure a burst limit of 200 to handle a sudden, short-lived influx of traffic.

3. Queueing Policies

- **Function**: Instead of immediately rejecting requests that exceed the burst limit, RitAPI can place them in a temporary queue. This smooths out traffic peaks and improves user experience during minor load spikes. If the queue continues to grow, it indicates a persistent attack, and subsequent requests can then be dropped.
- **Benefit**: Prevents service disruption during brief, intense periods of activity and provides a buffer before more aggressive mitigation actions are taken.

Example Config Snippet

Below is a simple YAML configuration snippet demonstrating how to apply a basic DDoS protection profile to an API gateway or a specific endpoint using RitAPI.

```
# RitAPI Protection Profile for a public-facing endpoint

ddos_protection:
    # The sustained request rate allowed from a single source
(IP or token).
    # Any traffic consistently above this will be throttled.
    max_requests_per_second: 100

    # The number of requests allowed in a short, temporary
burst.
    # This allows for legitimate traffic spikes without
immediate blocking.
    burst_limit: 200

    # The action to take when the burst_limit is exceeded.
    # Options: 'rate_limit' (HTTP 429), 'queue', 'drop'.
    action: rate_limit
```

Emergency Response Checklist

When an active attack is suspected, follow these steps to quickly mitigate the threat and restore service.

Step 1: Identify Endpoint Under Attack

Use the RitAPI dashboard to view real-time traffic metrics. Look for endpoints with abnormally high request counts, elevated error rates (e.g., 429 Too Many Requests), or increased latency. Isolate the attack to a specific API path (e.g., `/api/v1/products`).

Step 2: Apply Emergency Profile

Activate a pre-configured "Emergency" profile for the identified endpoint. This profile should have much stricter rate limits, a lower burst window, and potentially a more aggressive action like `drop` instead of `rate_limit`. This will immediately cut off the majority of malicious traffic.

Step 3: Monitor Queue Depth

If you are using a queuing policy, watch the queue depth monitor in the RitAPI dashboard. A constantly full or rapidly growing queue indicates the attack volume exceeds your processing capacity even with the emergency profile. You may need to further tighten the limits or block the offending IP ranges.

Step 4: Export Outage Report

Once the attack has subsided, use RitAPI's reporting tools to generate an outage report. This report should include key metrics such as attack duration, peak requests per second, source IPs/regions, the endpoint targeted, and the mitigation actions taken. Use this data for post-mortem analysis and to refine your protection profiles for the future.