International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V69I1P107

Volume 69 Issue 1, 34-37, January 2021

Original Article

AI-Driven (Smart) Throttling in API Management using Stream Processor and trained ML Models

Bharathan Kasthuri Rengan

Principal Architect, Virtusa Corporation, 65 W Newell Ave, Rutherford NJ, USA

Received Date: 30 November 2020
Revised Date: 15 January 2021
Accepted Date: 17 January 2021

Abstract - *APIs have become the new diamond [part of New Normal as well] of enterprise transformation initiatives and strategy empowering customers, employees, partners, and other stakeholders to access their applications, business, data of their systems.*

We see new waves of cyber-attacks from hackers against these enterprise assets and initiatives disrupting the industry from time to time. It is imperative to build robust API security (static, dynamic, AI-driven policies) for large enterprises to serve their customer and stakeholders better to address this. Gartner has predicted by 2022, the most frequent attack vendor for enterprise will be in the space of API.

Enterprises have a solid response to this scenario by having a comprehensive API management solution. APIs are must-have capabilities to govern, control, and access the API ecosystem. However, while building the API strategy, they need also to provide a comprehensive solution around the most sophisticated vector attacks on APIs, thereby implementing static, dynamic, and AI-driven security (around throttling and rate-limiting).

This paper aims to address API management solutions that enterprises should incorporate to address integrity, security, scalability, and API ecosystem availability.

Keywords *- API Throttling, API Management, API Governance, API Gateway*

I. INTRODUCTION

The idea behind APIs has existed since the beginning of computing; however, in the last 10 years, they have grown significantly in number and sophistication. They are increasingly scalable, monetized, and ubiquitous, with more than 12,000 listed on ProgrammableWeb, which manages a global API directory.

Future-looking scenarios involving smartphones, tablets, social outlets, wearables, embedded sensors, and connected devices will have inherent internal and external dependencies in underlying data and services. APIs can add features, reach, and context to new products and services or become products and services themselves.

A. The Need for API Throttling

Throttling is a key functionality (one of the key pillars) of API Management. API back ends can't serve unlimited requests. Throttling plays an important role in monetizing an API. Ensuring that your business APIs can be exposed to the public with a reputation and ensure that each user gets a fair share. Shaping the traffic as your business changes. To make an API, application, or resource available to a consumer at different service levels, usually for monetization purposes.

B. Central Policy Server: Throttling Manager

Throttling Manager provides both design and run-time support to the author and enables throttle policies in API Gateway. The following illustration provides the interaction of API Manager components.

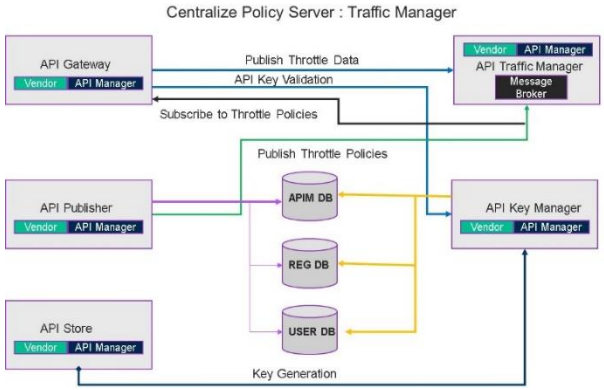

Fig.1 Central Policy Server: Throttling Manager

The Traffic Manager processes data of each API request and makes throttling decisions based on the applicability of available throttle policies. Throttling decisions are made by the runtime and published to the JMS topic.
The gateways subscribed to this JMS topic get instantly notified of the decisions.

C. Levels of Throttling

Throttling policies can be defined at API Publisher /Developer Portal and enforced in API Gateway (GW). Following are the levels of throttling.

a) Subscription Level Throttling (API Publisher)

- Rate Limiting (Burst Control)
- Control the usage of APIs within smaller time durations
- Protect the backend from sudden request bursts

Define rate limit; for example, one can define 1000 requests per day or 10 requests per second in single throttle policy. This way, enforcing a rate limit that protects the back end can throttle burst requests and controls.

b) Subscription Level Throttling (API Subscriber)

- After selecting the subscription level, throttling tiers are set, and the API is published. The API consumers can log in to the API store at subscription time and select which tier they are interested in when subscribing to the API.

c) Application Level Throttling (Application Developer)

d) Advanced Throttling (API Publisher)

- Applicable in API-Level as well as Resource Level throttling when publishing API.
  i. 10 K per min, 20 K per min, 50 K per min, Unlimited
- Advanced throttling can be defined in Publisher Portal
- Resource level throttling tiers are set to HTTP verbs of API resources.
- Advanced Policy could be based on IP address range, JWT header (specific claim value), Header, and Query parameters.

II. IMPLEMENTATION

Traffic Manager has the responsibility of making throttling decisions - The Traffic Manager Runtime in Traffic Manager processes events from API gateways. Policies deployed in traffic manager are executed on each event - An event that triggers a condition in a policy will be notified to gateways through a JMS topic- Each gateway maintains a throttle data map to check whether a request is within the allowed quota. Gateways update the throttle data map from the JMS Topic, which is notified by the Traffic Manager.

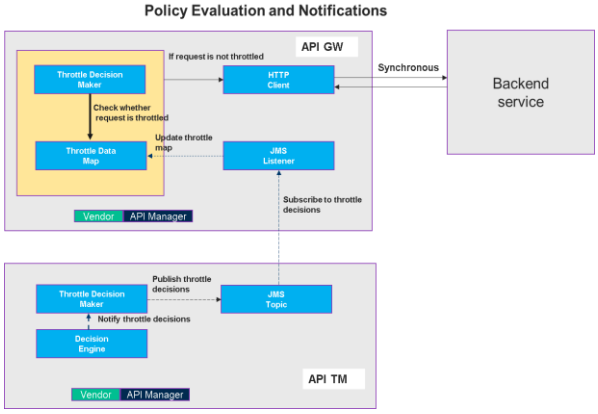

Fig. 2 Policy Evaluation and Notification

A. Need for Smart AI-driven solution in API Management

The Next Gen API gateway and Throttling capabilities are emerging as vast API's are exposed to the external world for an enterprise. The key is each API has a different access pattern.

A legitimate user access pattern in a specific API could be malicious for another API as context, and usage is completely different. The threat opportunity is different for each API and resources.

For example, a user doing a multiple search API could be considered perfectly normal, whereas multiple requests from the same user within a time-period access pattern could ring "alarm bell" for systems as a stolen credit card could be used by a hacker. Therefore, each API access pattern has to be examined carefully to determine the correct response

B. AI security layer as Solution

To fill the cracks left by traditional policy-based API protections, modern security teams need AI-based API security by applying AI models to inspect and report all API activities continuously.

The traditional approach would detect anomalies and risks; however, it can take months to discover them. As contrast, AI-driven Solution using prebuilt models as an additional AI security layer to detect some of these attacks in near real-time.

Importantly, AI engines can run outside of API gateways and communicate the decision to them.

API gateway does not have to expend resources to process these requests. The addition of AI security typically does not impact runtime performance.

C. Integrate Policy-based and AI-Driven API Security

Defining a security enforcement point and decision point is critical to implement AI-driven API security. These 2 endpoints are independent of each other to achieve high computation power to reduce the latency and not affect their efficiency.

The API gateway intercepts API requests and applies various policies. Linked to it is the security enforcement point, which describes each request's attributes (API call) to the decision point, requests a security decision, and then enforces that decision in the gateway.

The decision point, powered by AI, continuously learns each API access pattern's behavior, detects anomalous behaviors, and flags different attributes of the request.

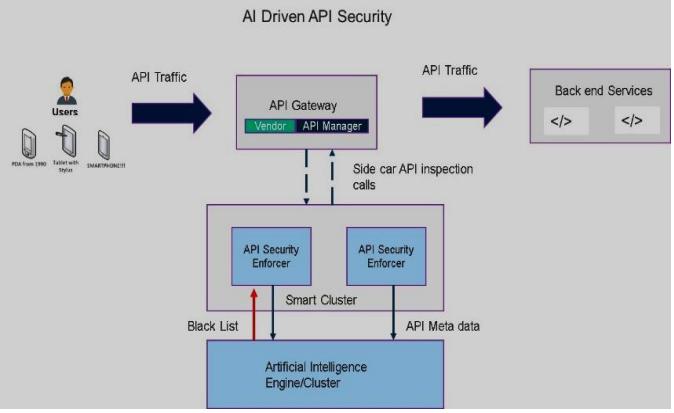

Fig. 3 Illustration of AI Engine in API Management – Throttling, Rate Limiting

D. AI Models in Action – API security

Step A: Use Cases: For API throttling, the following data is collected from network analysis

Historical – Syslog, routing, network packets analyze

and real-time – analyze the API payload (Header, body, query parameters, tokens)

Network analysis models (pre-built models) – traffic analysis

Post Authentication and Authorization, API behavior usage on usage patterns with following vector attacks

API takes over by legitimate internal user with valid credentials but based on usage patterns, and the threat is detected and alerted (overtime, real-time usage),

Transactions, skipping the gate with the valid cookie, token -> API usage pattern scanned. Deploy defensive or fake API's to trap the hackers go through the gold mines defensively.

E. Pattern Discovery

Following patterns are discovered as AI use cases

IaaS layer (through the proxy, Load Balancers – Big IP or Info blox), AWS, GCP, Cloud providers, or On-Prem

API usage patterns – internal, external, blue /green deployment are collected.

Following patterns are discovered as AI use cases

- IaaS layer (through the proxy, Load Balancers – Big IP or Info blox), AWS, GCP, Cloud providers, or On-Prem
- API usage patterns – internal, external, blue /green deployment are collected.
- Systems behind APIs, back end services, audit trail logs are collected.

F. Data Collection and Discovery

For the classification pattern on select related parameters on API take over, usage pattern, SVM (Support Vector Machines) is the best choice to extrapolate the regression and classification.

Defining the hyper-plane will be the key to address false-positive cases.

The key challenge is decisioning and communicating back to the API gateway, which is near real time to restrict API attacks.

G. Near Real-time feedback to API Gateway

These pre-trained models are injected into the AI engine and are ingested with sources (network analysis) – decision tree – for decision making and classification (SVM). Once the decision is made, after analysis and pattern match, the decision is propagated to the security introspection endpoint, cascaded to the API gateway as an asynchronous process. This is to make sure it is near real-time and doesn't degrade the performance of the API gateway (low latency requirements).

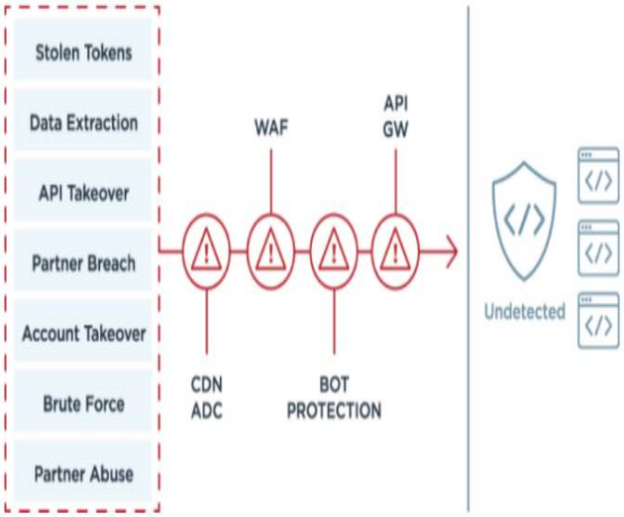

Fig. 4 AI-driven near time API security

III. CONCLUSION

One of the most cybersecurity threats is predicted around API security, thereby causing concerns for cybersecurity teams. TO address the new wave of intelligent attacks, it is imminent to build a nex-gen solution.

AI powers the API security where it learns from user behavior, address new threats. AI-driven API security is going to be the must-have capability for enterprises.

IV. REFERENCES

[1] From Batter to Cake: Bake your Own Security Model in API Management http://www.ijcttjournal.org/archives/ijctt-v68i10p103

[2] Ping Intelligence (AI pre-trained) models for API management https://www.pingidentity.com/en/software/pingintelligence.html

[3] Survey on the usage of Machine Learning Techniques for Malware Analysis https://www.researchgate.net/publication/320582721_Survey_on_the_Usage_of_Machine_Learning_Techniques_for_Malware_Analysis

[4] Surendiran,R., and Alagarsamy,K., Privacy Conserved Access Control Enforcement in MCC Network with Multilayer Encryption. SSRG International Journal of Engineering Trends and Technology (IJETT), 4(5) (2013) 2217-2224.