Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 08, August-2023

Smart Reverse Proxy with Remote Management

Swayam Atul Mehta
Computer Science and Engineering
Vellore Institute of Technology
Vellore, India

Krushn Pathak
Computer Science and Engineering
Vellore Institute of Technology
Vellore, India

Anusha Garg
Information Technology
Vellore Institute of Technology
Vellore, India

***Abstract*—In this paper we plan to develop a cloud-based firewall at level 7, specifically designed as a reverse proxy. A reverse proxy functions as a server situated behind a private network's firewall, directing client requests to the appropriate backend server. This initiative aims to construct an advanced reverse proxy that prioritises observability, incorporating a random load distribution system. Additionally, a Telegram bot will be implemented to streamline remote proxy administration tasks for the administrator. The results of experiments indicate that this design provides the benefits of universality, extensibility, and efficiency. These results are generalizable across the most advanced model-serving frameworks.**

Keywords—remote proxy management, forward proxy, reverse proxy, Telegram Bot, load-balancing

I. INTRODUCTION

A forward proxy, also known as an outgoing proxy, is situated between a client (user's device) and the internet. When a client requests a resource from the internet, the request is first sent to the forward proxy server, which then forwards the request to the target server on behalf of the client. The response from the target server is then sent back to the proxy, which, in turn, delivers it to the client. Forward proxies are typically used to provide anonymity and control over outbound traffic from clients.

A reverse proxy, also known as an inbound proxy, is positioned between the internet and a server (or a group of servers) responsible for serving content. When clients request resources, the request is directed to the reverse proxy, which then forwards the request to one of the backend servers. The response from the backend server is sent back to the reverse proxy, which, in turn, delivers it to the client. Reverse proxies are commonly used to enhance security, improve performance, and distribute incoming traffic across multiple servers.

Admins can streamline security administration by employing a reverse proxy and load balancing. Centralised security controls, SSL termination, web application firewalls, and access control can be implemented at the proxy layer. This approach simplifies authentication, isolates traffic, and enhances monitoring. By spreading the security load across load-balanced backend instances, admins can ensure consistent protection, logging, and patch management. This strategy simplifies security management while offering scalability and DDoS mitigation.

The goal of this project is to create a clever reverse proxy that is observable. It employs a uniformly distributed random load balancing mechanism. We also intend to make remote proxy management very simple for the administrator through a Telegram bot.

II. LITERATURE SURVEY

The literature survey encompasses ten research papers that contribute to the understanding of reverse proxy servers, load balancing, and security measures. Takenaka, Kato, and Okamoto[1] propose an adaptive load balancing content address hashing routing for reverse proxy servers. Lin, Liu, and Lien[2] present a detection method against web flooding attacks using reverse proxy. Tao and Chen[3] introduce an extensible universal reverse proxy architecture. Kato and Okamoto[4] present a load balancing routing algorithm for reverse proxy servers. Karimi et al.[5] propose a fuzzy logic-based adaptive load balancing algorithm for reverse proxy servers. Long and Li[6] focus on designing secure sessions based on reverse proxies. Wang, Douglis, and Rabinovich[7] discuss forwarding requests among reverse proxies. Chhabra [8] studies recent research trends of proxy servers. Agarwal and Sirsikar[9] present an efficient technique for finding SQL injection using reverse proxy servers. Arnaldy and Hati[10] analyse the performance of reverse proxy and web application firewall with Telegram Bot as an attack notification on web servers. These papers collectively contribute valuable insights into various aspects of reverse proxy technologies, load balancing approaches, and security mechanisms, aiding in the advancement of computer networks and web services.

III. OVERVIEW OF PROPOSED SYSTEM

A. *Proposed Methodology*

In order to prevent any one server from being overworked, load balancing involves dividing network traffic across several servers. Our reverse proxy uses a random load-balancing strategy, in which the proxy chooses at random which server instance a specific user will visit at a specific time. Despite being random, it guarantees that each server instance is chosen equally. In our situation, with three instances, each instance would be chosen 33% of the time, or around 333 times, out of a total of 1,000 requests. In the event that the reverse proxy server is experiencing high traffic, it notifies the administrator.

The combination of load balancing and monitoring mechanisms offers an efficient and user-friendly solution for managing microservice traffic distribution and ensuring system stability even in high-demand scenarios.

B. *Algorithm and Explanation*

1) Import Required Modules

Import necessary modules like ‘express’, ‘morgan’, ‘http-proxy-middleware’, ‘express-ipfilter’, ‘node-telegram-bot-api’, ‘request’, etc.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 08, August-2023

2) Create Express Server

Create an instance of the Express server using 'express()'.

3) Configure Server Constants

Set up constants like PORT, HOST, API_SERVICE_URL, token, and others.

4) Set Up Logging

Use 'morgan' middleware for logging HTTP requests in a specific format. Create a Telegram bot instance using the provided token and enable polling.

5) IP Filtering

Define an array containing the allowed IP addresses. Use 'express-ipfilter' middleware to restrict access to the server to only the specified IP addresses.

6) Define Endpoints

Create several endpoint handlers using 'app.get()'.

- /info: Respond with a message including user-agent from the request headers.
- /ms1, /ms2, /ms3: Respond with simple HTML messages for microservices clones.
- /site: Handle the site endpoint, checking for high traffic and sending notifications using the Telegram bot. If not in downtime, proxy the request to one of the URLs.

7) Handle Authorization

Use middleware to check for an authorization header in incoming requests. If an authorization header is present, allow the request to proceed, otherwise send a "403 Forbidden" response.

8) Telegram Bot Handling

Set up event listeners for incoming messages from the Telegram bot. Respond to messages containing specific keywords like "Metrics", "Block IP", "Kill Switch", and "Toggle Live Logs". Perform corresponding actions based on the message content. Initialise the Telegram bot when the /start command is received. Provide a keyboard for interacting with the bot.

9) Proxy Endpoints

Use the 'createProxyMiddleware' to set up a proxy for requests to the '/json_placeholder' path. These requests are redirected to the API_SERVICE_URL with path rewriting.

10) Start the Server

Use the app.listen method to start the Express server on the specified PORT and HOST.

C. Proposed Architecture

A front-end tier and a backend tier make up the suggested structure. The front-end tier displays which instance of the microservice the user has accessed (this has been done for understanding and explanation purposes), while the backend tier is in charge of the reverse proxy functionality. Standard HTML and CSS are used on the front end while node.js is used to design the proxy.

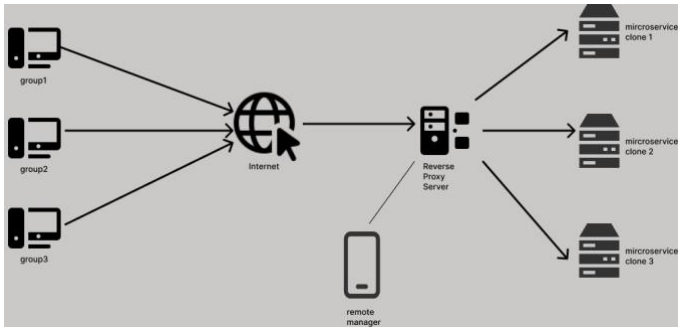

Fig. 1. System Architecture

Additionally, we have a Telegram bot that gives the administrator a condensed perspective of the network. Sending the bot "/start" grants access to the bot. The bot replies by sending a greeting. The bot offers the following capabilities:

- Live logging: Through this feature, the administrator may view the endpoint and IP address of the client, as well as the time that a request was made to the proxy.
- Kill Switch: This toggle switch, when activated, kills the proxy, preventing all incoming requests to the microservices it is hiding. To resume regular operation, toggle it on once more.
- Metrics: This feature enables the administrator to check the overall performance of the proxy and displays the total number of requests made to it since the last time this parameter was checked. It is reset to 0 after checking request count.

IV. Results and Discussion

The proposed resolution demonstrates encouraging outcomes in enhancing the efficiency of network infrastructure.The exploration of advanced machine learning algorithms for predictive analysis, enhancement of security protocols using encryption techniques, and the enhancement of the system's flexibility to accommodate different network structures introduce fresh avenues for research. By integrating state-of-the-art technology and continuously advancing the concept of an intelligent reverse proxy, there is the potential to further enhance network efficiency and drive the progress of resilient, efficient, and secure digital environments. The system can further be developed upon and iterated to make use of different algorithms and technologies to improve the efficacy of the proposed system.

V. Conclusion and Future Works

In summary, the article "Smart Reverse Proxy with Remote Management" provides a thorough investigation of the creation and application of an intelligent reverse proxy system that makes use of remote administration capabilities. The suggested solution shows promising results in increasing network infrastructure efficiency by successfully resolving the issues of load balancing, security, and performance optimization. The investigation of cutting-edge machine learning algorithms for predictive analysis, the improvement of security measures through encryption methods, and the expansion of the system's adaptability to various network topologies open new research directions. The incorporation of cutting-edge technology and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 08, August-2023

ongoing development of the idea of a smart reverse proxy hold the promise of further optimising network performance and advancing the development of strong, effective, and secure digital ecosystems.

REFERENCES

[1] T. Takenaka, S. Kato and H. Okamoto, "Adaptive load balancing content address hashing routing for reverse proxy servers," 2004 IEEE International Conference on Communications (IEEE Cat. No.04CH37577), 2004, pp. 1522-1526 Vol.3, doi: 10.1109/ICC.2004.1312765.

[2] C. -H. Lin, J. -C. Liu and C. -C. Lien, "Detection Method Based on Reverse Proxy against Web Flooding Attacks," 2008 Eighth International Conference on Intelligent Systems Design and Applications, 2008, pp. 281-284, doi: 10.1109/ISDA.2008.72.

[3] Y. Tao and G. Chen, "An Extensible Universal Reverse Proxy Architecture," 2016 International Conference on Network and Information Systems for Computers (ICNISC), 2016, pp. 8-11, doi: 10.1109/ICNISC.2016.012.

[4] Satosi Kato, Hidetosi Okamoto, Toyofumu Takenaka "Load Balancing Routing Algorithm for Reverse Proxy Servers", IEICE Transactions on Communications, September 2015

[5] Abbas Karimi, Faraneh Zarafshan, Adznan b. Jantan, A.R. Ramli, M. Iqbal b.Saripan, "A Fuzzy Logic Based Adaptive Load Balancing Algorithm for Reverse Proxy Servers",(IJCSIS) International Journal of Computer Science and Information Security, Vol. 6, No. 1, 2009

[6] Long, Wen-Guang, and Jian-Ping Li. "Designing secure session based on reverse proxy." 2012 International Conference on Wavelet Active Media Technology and Information Processing (ICWAMTIP). IEEE, 2012.

[7] Wang, Limin, Fred Douglis, and Michael Rabinovich. "Forwarding requests among reverse proxies." International Web Caching and Content Delivery Workshop, ser. IWCW'00. No. 5. 2000.

[8] Chhabra, Yogita. "A Study of Recent Research Trends of Proxy Server." International Journal of Advanced Technology in Engineering and Science 3.01 (2015): 159-164.

[9] Agarwal, Raj, and Sumedha Sirsikar. "An efficient technique for finding sql injection using reverse proxy server." International Research Journal of Engineering and Technology (IRJET) 6.09 (2019): 1564-1569.

[10] Arnaldy, Defiana, and Tio Setia Hati. "Performance Analysis of Reverse Proxy and Web Application Firewall with Telegram Bot as Attack Notification On Web Server." 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE). IEEE, 2020