Cloud shared hosting DDoS resistance and potential ways of protection

Alexander Chizhov[1,†], Andriy Fesenko[1,2,*,†], Vadym Ziuziun[3,†] and Dinara Basshykyzy[4,†]

[1] National Aviation University, Liubomyra Huzara Ave. 1, Kyiv, 03058, Ukraine
[2] State Scientific and Research Institute of Cybersecurity Technologies and Information Protection, Maksym Zalizniak Str., 3/6, Kyiv, 03142, Ukraine
[3] Taras Shevchenko National University of Kyiv, Volodymyrska Str., 64/13, Kyiv, 01601, Ukraine
[4] Yessenov University, Microdistrict, 32, Aktau, 130000, Kazakhstan

Abstract

In this study, we examine a method for optimizing the load on computing resources for websites with low traffic. The main idea of the method is to host websites not on distributed servers (several hundred websites on one independent server), which is the industrial standard for shared hosting, but rather to consolidate servers into a cloud cluster to provide hosting as a cloud service. This approach aims to significantly increase the reliability of the service and optimize the use of computing resources. With this architecture, computing resources can be dynamically allocated and reallocated based on load (number of requests per second). It is evident that DDoS attacks such as HTTP Flood and Brute Force can have an extremely negative impact on the efficiency of the entire cluster and, consequently, the entire hosting company. Unlike traditional shared hosting, where a DDoS attack on a single client website disables only one server (affecting only the websites hosted on that server, a small fraction of all websites managed by the company), the proposed architecture necessitates additional measures to minimize the damage from such attacks. In this work, we propose the use of artificial neural networks and machine learning for HTTP/HTTPS requests filtering. Based on the decisions made by the neural network, IP addresses can be marked as potentially compromised. When there is a significant increase in traffic to the cluster, requests from such IP addresses can be blocked using CAPTCHA challenges, and if the negative impact accumulates, all requests from these IP addresses can be temporarily completely blocked.

Keywords

cloud, hosting, DDoS, http flood, brute force, protection

1. Introduction

The optimization of load for low-traffic websites is a pressing issue for several reasons. The number of low-traffic websites is in the millions and is growing rapidly every day. As of February 2024, there are approximately 1.1 billion websites worldwide, with about 200 million of them being active [1, 2]. Around 252,000 websites are created daily [3]. According to global statistics, only 71% of businesses had a website in 2023 [2]. More than 43% of small businesses plan to invest in their website in the near future. The hosting services market shows an annual growth rate of 17.35% [1].

Hosting companies face the challenge of minimizing costs for maintaining these online resources while providing affordable hosting rental prices. Clients of such services include small and medium-sized businesses, individuals, non-profit, and public organizations, making the cost of hosting a significant concern. Therefore, the only viable solution in the market is shared hosting, where the cost of owning and maintaining a server is effectively shared among many consumers. On the other hand, website visitors increasingly demand higher quality service. Website availability 24/7 and fast

CH&CMiGIN'24: Third International Conference on Cyber Hygiene & Conflict Management in Global Information Networks, January 24–27, 2024, Kyiv, Ukraine
* Corresponding author.
† These authors contributed equally.
alex.definch@gmail.com (A. Chizhov); aafesenko88@gmail.com (A. Fesenko); vadym.ziuziun@knu.ua (V. Ziuziun); dinara.bashykyzy@yu.edu.kz (D. Basshykyzy)
0000-0002-3992-8522 (A. Chizhov); 0000-0001-5154-5324 (A. Fesenko); 0000-0001-6566-8798 (V. Ziuziun); 0000-0001-8601-3870 (D. Basshykyzy)

loading are key indicators. Statistics show that 47% of web resource visitors do not want to wait longer than 2 seconds for a web page to load [4–6]. Thus, it is evident that there is a problem where, on one hand, more and more resources are required to maintain millions of websites, and on the other hand, the demands for service quality are getting higher. The issue of optimizing resource usage and improving service quality is the subject of this article.

2. Classic shared hosting architecture

The most common budget solution in the hosting services market is shared hosting [7, 8]. The architecture of shared hosting involves hosting several hundred low-traffic websites on a single server [9, 10]. Consequently, all websites hosted on the server compete for the same server resources. This architecture is illustrated in Figure 1.

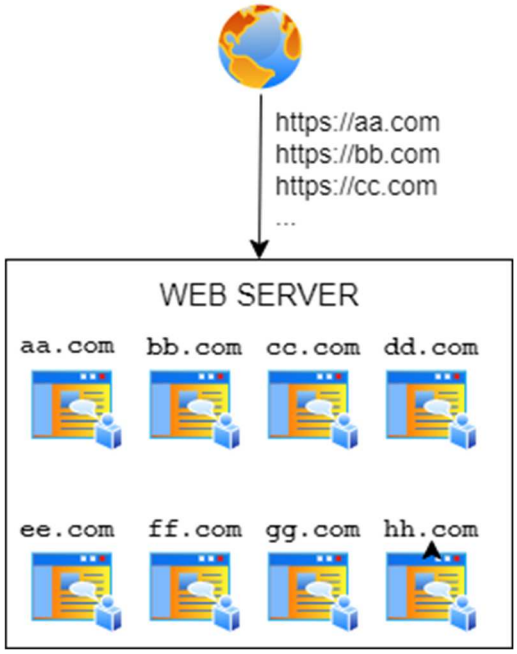

Figure 1: Classic shared hosting architecture.

3. Adaptation of the Erlang B model for a distributed computing resource

In a telephone line system with a limited number of channels, a call will be rejected if all lines are occupied. In shared hosting, the server resources (CPU time and RAM) serve as the equivalent of these channels [11, 12]. Server resources can be quantified by the number of requests the server can handle simultaneously. Handling HTTP requests [13] on the web server side typically doesn't involve significant computational tasks [14]. Instead, it mainly impacts RAM, where the script language interpreter loads and web application files are assembled [15, 16]. For modern web application frameworks, the recommended memory limit per request is generally around 256 MB of RAM [17, 18]. Therefore, 256 MB of RAM can be considered analogous to one telephone line. Suppose a modern hosting server typically has 64GB of RAM. Allocating up to 4 GB for the operating system and other overheads [19–21], the remaining memory provides for approximately 240 parallel processes. To ensure a 20% safety margin under real-world conditions, this value is adjusted to about 200 parallel requests.

Example Calculation:

- Total RAM: 64 GB = 65,536 MB.
- RAM reserved for OS and overhead: 4 GB = 4,096 MB.

- Usable RAM: 65,536 MB - 4,096 MB = 61,440 MB.
- RAM per request: 256 MB.
- Number of parallel requests: 61,440 MB / 256 MB = 240 (rounded down to 200 for safety margin).

The Erlang B model estimates the probability of call blocking. This value can be adapted for distributed computing architectures as the probability that a request will be rejected or will take significantly longer to execute due to server overload and insufficient resources, such as processor or memory.

The telephone line load in Erlangs [11] can be adapted for use in the context of distributed computing resources as the number of requests per second per website multiplied by the number of websites (analogous to the number of incoming calls per unit time), as well as the average processing time per request (average call duration).

$$E = \lambda \times H.$$

In the context of shared hosting:

- λ - represents the average number of requests per second,
- H - represents the average processing time per request in seconds.

The average number of requests per second (λ) in shared hosting depends on the number of websites hosted on the server.

$$\lambda = rps \times w,$$

where:

- *rps* - represents the average number of requests per second for a lightly loaded website,
- *w* - denotes the number of websites on a single server.

Global statistics indicate that the majority of lightly loaded websites receive no more than 50,000 visits per month, with each visit averaging 4-6 pages [22], resulting in approximately 0.114 requests per second (rps). Analytics from the 2023 Statista Survey [23] show that every second of delay in website loading decreases visitor satisfaction by 16%, increasing the likelihood of visitors leaving the site. Therefore, optimal response time is considered within 2 seconds.

The Erlang B formula:

$$B(E,N) = \frac{\frac{E^N}{N!}}{\sum_{k=0}^{N} \frac{E^k}{k!}}$$

In the context of shared hosting:

- *E* - represents the load in Erlangs, calculated as the average number of requests per second to one lightly loaded website, multiplied by the number of websites, and further multiplied by the average time spent on one request,
- *N* - denotes the maximum number of concurrent requests that the server can handle.

The calculations of the blocking probability according to the Erlang B model are provided in Table 1.

The graph depicting the dependency of the probability of request blocking on the number of websites on a single server is shown in Figure 2.

Therefore, we can use the adapted Erlang B model effectively to forecast the load on a server hosting websites with sufficient accuracy, supported by practical experience from hosting companies. For instance, servers with configurations like 16 cores and 64 GB of RAM typically host 700-900 lightly loaded websites, ensuring high uptime.

Table 1
Dependence of the Probability of Request Blocking on the Number of Websites Hosted on the Server

w	*rps*	λ	*H*	*E*	*N*	*B*
100	0.114	11.4	2	22.8	200	0.01%
200	0.114	22.8	2	45.6	200	0.01%
300	0.114	34.2	2	68.4	200	0.01%
400	0.114	45.6	2	91.2	200	0.01%
500	0.114	57.0	2	114.0	200	0.01%
600	0.114	68.4	2	136.8	200	0.01%
700	0.114	79.8	2	159.6	200	0.01%
800	0.114	91.2	2	182.4	200	1.00%
900	0.114	102.6	2	205.2	200	7.00%
1000	0.114	114.0	2	228.0	200	15.00%

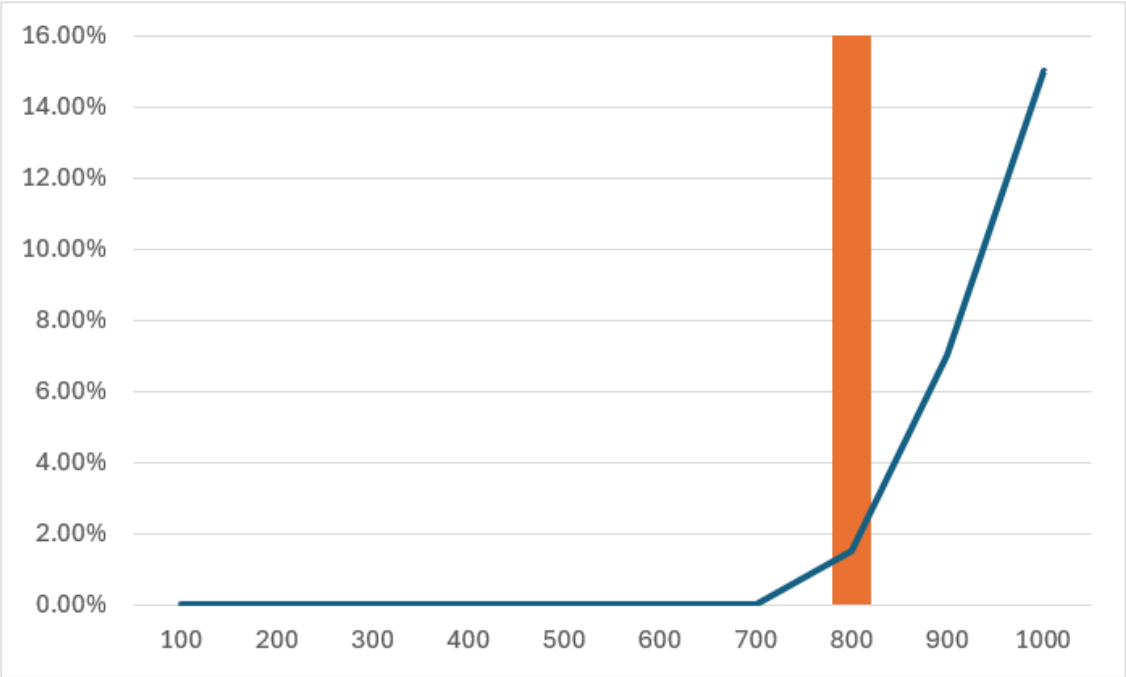

Figure 2: Dependency of the probability of request blocking on the number of websites hosted on the server.

4. Cyber attacks on a computing resource

In today's interconnected world, internet resources are globally accessible, allowing any user to access any public website. However, this openness also exposes these websites to various types of cyber attacks targeting different levels of network resources, from the physical network to application layers. Considering cloud architecture as a replacement for shared hosting suggests a shift towards application-level management. This approach prioritizes cybersecurity and resilience against attacks at the application level, while the lower network layers remain largely unchanged in terms of vulnerability.

The most common application-level attacks include:

- Cross-Site Scripting (XSS). Attackers inject malicious scripts into web pages viewed by other users.
- SQL Injection (SQLi). Attackers inject SQL queries into input fields of web forms, exploiting vulnerabilities to manipulate the application's database.

- Cross-Site Request Forgery (CSRF). The goal of the attacker is to compel the victim to unwittingly send a malicious web request to a website accessible to the victim.
- Session Fixation. The attack is aimed at session hijacking of the victim and gaining access to secure data.
- HTTP Flood. Sending many GET and POST requests to a website with the intention of consuming all server resources.
- Brute Force. An attack aimed at obtaining client authentication credentials by systematically attempting to guess passwords using a dictionary.

Most of the discussed attacks exploit vulnerabilities within the application itself, often stemming from programming errors, with the exception of HTTP Flood and Brute-Force attacks. These two types are variations of DoS/DDoS attacks. A DDoS (Distributed Denial of Service) attack is a cyberattack where the malicious actor aims to overwhelm the computational resources of a server that hosts a website. The goal is to flood the server with a large volume of requests, depleting its resources and preventing it from handling requests from legitimate users. Unlike a classic DoS attack, a DDoS attack utilizes multiple resources to send requests to the server [24, 25].

HTTP Flood is a type of attack where the attacker sends numerous GET and POST requests, effectively opening pages of a website from multiple IP addresses simultaneously. This leads to resource exhaustion on the server, making it unable to process requests from genuine visitors and thus disrupting the website's operation.

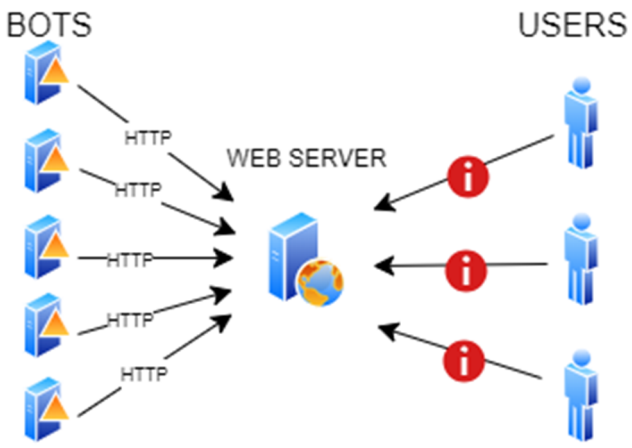

Figure 3: Principle of organizing a distributed HTTP Flood attack.

On the other hand, a Brute-Force attack targets hidden resources of a website, attempting to guess passwords by systematically trying different combinations from a dictionary. The complexity in identifying a DDoS attack lies in the fact that the malicious actor uses various IP addresses to send malicious requests. These requests appear indistinguishable from those of legitimate users, essentially comprising ordinary requests to open web pages or submit forms.

What are the risks of DDoS attacks for businesses [26]:

- Direct financial losses due to potential customers being unable to place orders or purchase goods on the website.
- Operational losses as more resources are consumed during the attack, requiring intervention by specialists to prevent both the attack itself and its consequences.
- Damage to brand reputation. As customer service declines, so does satisfaction with the brand overall.
- Legal consequences if the company has committed to providing a certain level of service and cannot do so due to a DDoS attack.

Considering the fact that year after year, DDoS attacks are becoming more massive and powerful, protection against such threats is increasingly critical. In November 2021, Microsoft faced the largest attack in history, involving 10,000 devices from around the world and peaking at 340 million requests per second [27]. Figure 4 depicts statistics on the scale of HTTP attacks over recent years according to Cloudflare data [28].

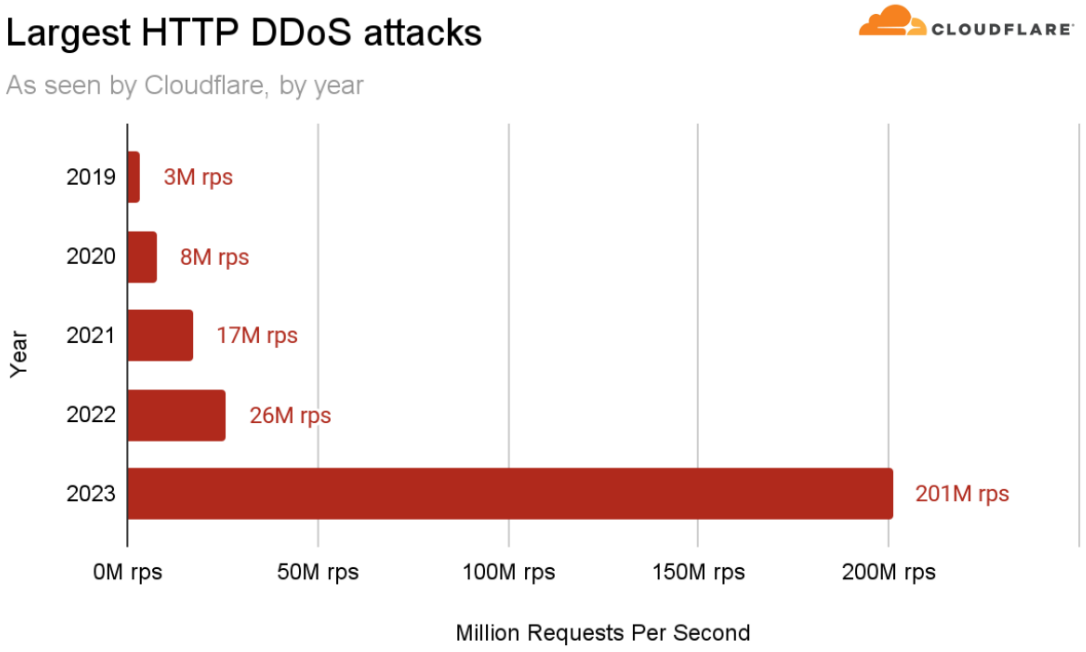

Figure 4: Cloudflare statistics about HTTP DDoS attacks.

Over the past 5 years, the scale of attacks has increased by 67 times [28]. According to the Cisco Global Report [29], the number of DDoS attacks has been growing at an average rate of 14% per year over the last 5 years. The total number of attacks has doubled to 15.4 million per year over this period [29, 30].

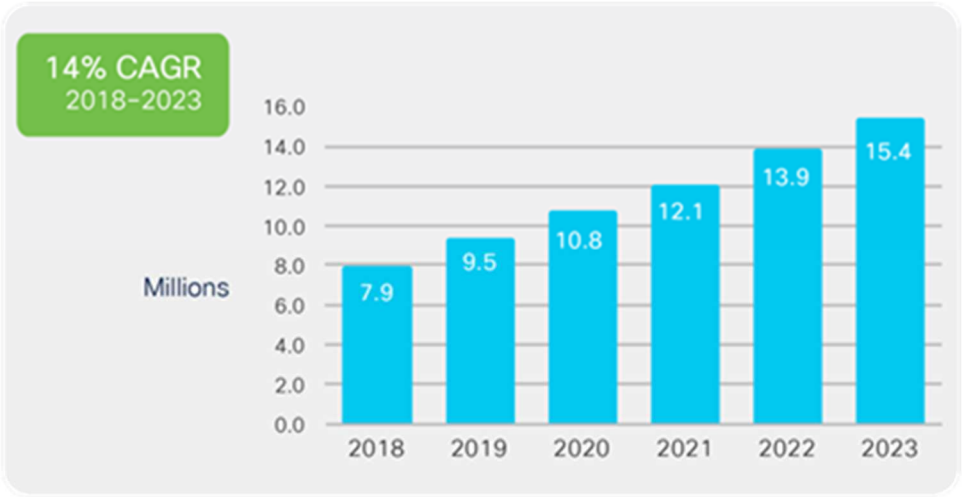

Figure 5: Number of DDoS attacks. Cisco Global Report.

The annual losses from DDoS attacks alone amount to tens of millions of dollars. The downtime for IT companies globally costs between $300,000 to $1 million per hour [30, 31].

5. Modeling increased workload and comparing architectures

Considering classic shared hosting with one distributed server hosting 800 websites, we can model an increase in workload on one of the sites to assess the threat level of an HTTP Flood attack. Table 2 illustrates the relationship between the number of requests to the target site (Wrps) and the probability of service denial to legitimate visitors of sites hosted on the same server.

Table 2
Dependency of the Number of Requests to the Target Site and the Probability of Service Denial

Wrps	λ	*H*	*E*	*N*	*B*
0.114	91.2	2	182.4	200	1.5%
1	92.1	2	184.2	200	1.6%
10	101.1	2	202.2	200	6.5%
20	111.1	2	222.2	200	13.0%
50	141.1	2	282.2	200	30.0%
100	191.1	2	382.2	200	53.0%
200	291.1	2	582.2	200	65.0%
300	391.1	2	782.2	200	74.0%
500	591.1	2	1182.2	200	82.0%

When a site on a server is loaded with just 50 requests per second, the server denies service with a 30% probability. With a minor DDoS attack of 500 requests per second, this figure rises to 82%. It can be said that under such conditions, the server almost ceases to serve clients. The cost of organizing a DDoS attack on an unprotected site, with low traffic, is around $200-300 for a load of 1000 requests per second [32, 33]. Such an attack guarantees to fully load the server, causing all sites on the server to cease service.

Hosting companies manage thousands of websites. Using shared hosting architecture, they own several dozen servers, each hosting several hundred websites. The proposed clustering architecture involves dynamic clustering, where the cluster uses exactly as many servers as needed to serve all sites at any given moment. Thus, during off-peak hours and weekends, when visitor activity is lower, the cluster will have significantly fewer servers than in the case of shared hosting. However, if there is an increase in load on any site, the cluster can add servers to maintain service quality across the company's balance. In the case of clustered architecture, all websites share the resources of the entire cluster. Figure 6 illustrates the clustered architecture of cloud shared hosting.

An adapted Erlang B model for a cluster would look like:

$$\lambda = rps \times wC,$$

where:

- rps — average number of requests per second for a lightly loaded website,
- wC — number of websites served by the cluster.

$$N = n \times S,$$

where:

- n — number of requests processed by one server,
- S — number of servers in the cluster.

If we scale the task to a cluster of 20 servers serving 16,000 websites, an attack at 1000 requests per second on one site would have negligible impact on other resources. With an attack at 2000 requests per second on one site, the probability of service denial would be around 5%. An attack at 5000 requests per second on one or several sites (e.g., 5 sites at 1000 requests per second each) would result in a 60% chance of service denial, significantly affecting customer service quality but not leading to a complete failure of websites in the cluster.

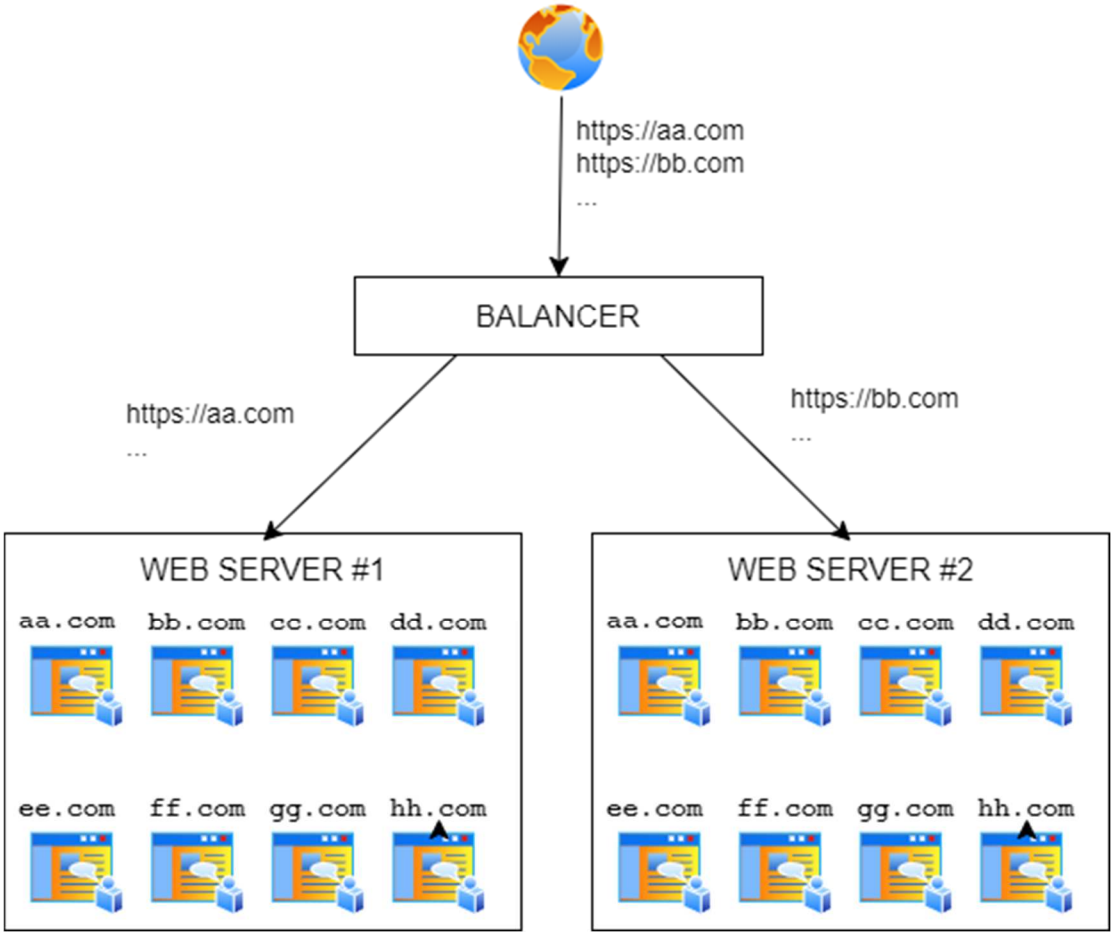

Figure 6: Number of DDoS attacks. Cisco Global Report.

Thus, we have demonstrated that clustered architecture is a more reliable and resilient solution for hosting lightly loaded websites. However, architectural changes do not completely eliminate the threat. The idea of clustering in shared hosting aims to save resources by serving the same number of websites with fewer servers. On the other hand, a more massive HTTP flood attack could result in a complete loss of the entire cluster, which would be catastrophic for the hosting company financially and reputationally.

6. Solution for protecting the cluster from distributed HTTP Flood and Brute-Force attacks

Filtering malicious traffic from legitimate traffic becomes challenging because malicious traffic masquerades as legitimate. Essentially, this includes GET or POST requests to website URIs, making it physically impossible to distinguish one request from another. Attackers emulate request headers similar to those of real browsers, use multiple IP addresses, and send a small number of requests from each address per unit of time to mimic real user behavior. A progressive solution in this scenario involves the use of artificial neural networks and machine learning. With a sufficient dataset of requests across the entire cluster, it is possible to create a profile of typical visitor behavior for each site and identify anomalies with high accuracy. In the task of filtering malicious traffic, the goal should not aim for high precision in identifying malicious requests. Achieving a detection probability of 70-80% appears quite feasible and would provide significant resilience to cluster attacks, making the attack economically unfeasible.

To achieve a significant result in training the neural network, it will be essential to utilize as much information as possible about both the specific request and the IP address from which the request originated. The neural network's objective should not be to determine the maliciousness of a specific

request but rather to output the probability that the IP address is infected and should be placed in quarantine. Subsequent requests from IP addresses in quarantine would be served a CAPTCHA puzzle [34] page instead of the actual website content. If the CAPTCHA puzzle is successfully solved, the IP address is removed from quarantine. If an IP address enters quarantine multiple times within a limited time period, it is placed on a blacklist. Requests from IP addresses on the blacklist are blocked and rejected at the load balancer level. After a specified period in the blacklist, the IP address moves back to quarantine, where it can be removed upon successful CAPTCHA completion or after the quarantine period expires. On Figure 7, a logical diagram of the protection module operation is presented.

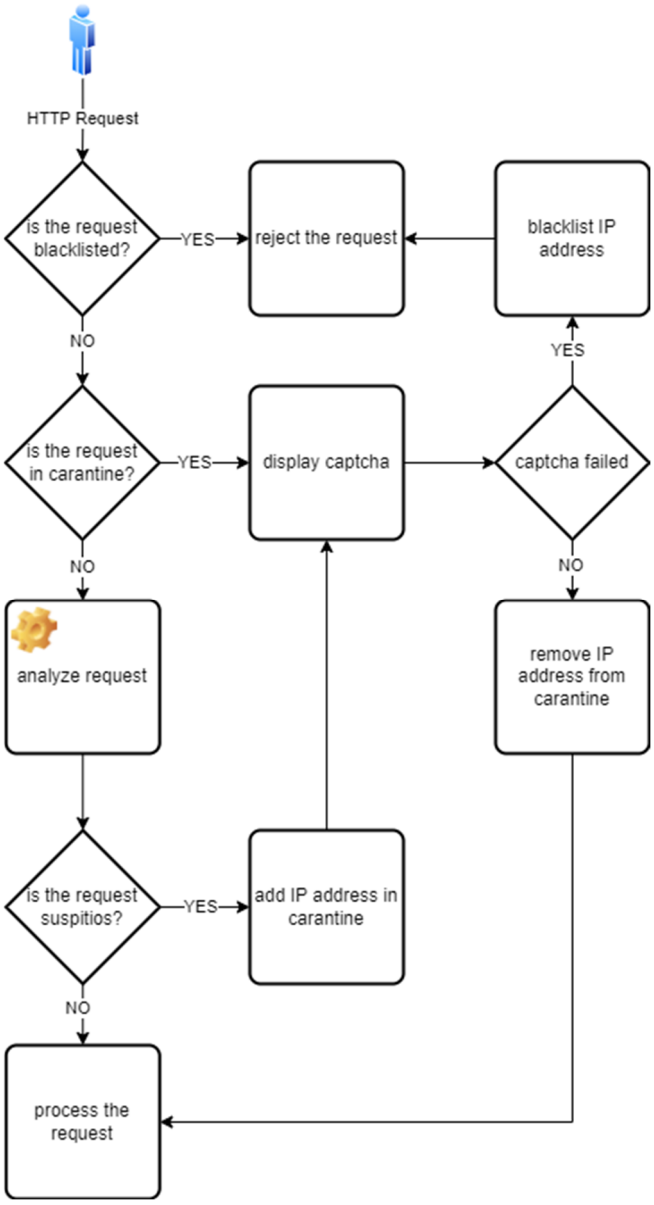

Figure 7: A logical diagram of the protection module.

7. Conclusions

The study examined the proposed clustered shared hosting architecture in terms of resilience to DDoS attacks, considered among the most common and likely threats. It is evident that the clustered system is significantly more resilient to DDoS attacks compared to traditional shared hosting. However, the architecture is not without its drawbacks, notably its monolithic nature, which can lead to overload under sufficiently high traffic volumes, affecting all sites uniformly. In contrast,

traditional shared hosting is more susceptible to server overload, but only a small fraction of sites are affected. A clear solution lies in implementing protection against DDoS attacks such as HTTP Flood and Brute-Force. Given the complexity of filtering malicious traffic from legitimate traffic, leveraging artificial neural networks and machine learning with multiple parameters based not only on packet properties but also historical data about IP addresses appears promising. The neural network's objective would be to determine the probability that an IP address is involved in a DDoS attack rather than belonging to a legitimate website user.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] M. Alda, Web hosting – worldwide, 2022. URL: https://www.statista.com/outlook/tmo/it-services/it-outsourcing/web-hosting/worldwide.

[2] K. Haan, Top Website Statistics For 2023, 2023. URL: https://www.forbes.com/advisor/business/software/website-statistics/.

[3] NJ. How Many Websites Are There in the World, 2023. URL: https://siteefy.com/how-many-websites-are-there/#How-Many-Active-Websites-Are-There.

[4] Page load time vs. response time – what is the difference?, 2023. URL: https://www.pingdom.com/blog/page-load-time-vs-response-time-what-is-the-difference/.

[5] Rasmussen E. How fast should your website be in 2023?, 2023. URL: https://www.enterspeed.com/blog/how-fast-should-your-website-be.

[6] A. Andy, The pros and cons of cheap shared web hosting, 2022. URL: https://medium.com/@andy_2959/the-pros-and-cons-of-cheap-shared-web-hosting-7efcf1eeaef4.

[7] A. Wandola, web hosting market share + 11 fast facts, 2024. URL: https://www.hostingadvice.com/how-to/web-hosting-services-market-share/.

[8] Shewale R. 70 web hosting statistics for 2024, 2024. URL: https://www.demandsage.com/web-hosting-statistics/.

[9] R. S. Sinha How many websites can you run on a dedicated server?, 2023. URL: https://www.host.co.in/blog/how-many-websites-can-you-run-on-a-dedicated-server/.

[10] V. Tkachuk, Y. Yechkalo, S. Semerikov, M. Kislova, Y. Hladyr, Using mobile ICT for online learning during COVID-19 lockdown, Communications in Computer and Information Science, 1308 (2021) 46–67. doi: 10.1007/978-3-030-77592-6_3.

[11] I. Angus, An introduction to Erlang B and Erlang C, Telemanagement 187.6 (2001): 8.

[12] E. Pinsky, A. Conway, W. Liu, Blocking formulae for the Engset model, IEEE transactions on communications 42.6 (1994) 2212–2214.

[13] T. Berners-Lee, R. Fielding, H. Frystyk, Hypertext transfer protocol--HTTP/1.0. No. rfc1945. 1996.

[14] D. Kunda, S. Chihana, M. Sinyinda, Web Server Performance of Apache and Nginx: A Systematic Literature Review, Computer Engineering and Intelligent Systems 8(2) (2017) 43–52.

[15] M. Laaziri, et al., A Comparative study of PHP frameworks performance, Procedia Manufacturing 32 (2019) 864–871.

[16] K. Lei, Y. Ma, Z. Tan, Performance comparison and evaluation of web development technologies in php, python, and node. js, in: Proceedings of 2014 IEEE 17th international conference on computational science and engineering, IEEE, Chengdu, China, 2014, pp. 661–668. doi: 10.1109/CSE.2014.142.

[17] D. Bonderud, Wordpress php memory limit: what it is, why it matters & how to increase it, 2022. URL: https://blog.hubspot.com/website/wordpress-php-memory-limit.

[18] R. S. Odarchenko, S. O. Gnatyuk, T. O. Zhmurko, O. P. Tkalich, Improved method of routing in UAV network, in: Proceedings of International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD), IEEE, Kyiv, Ukraine, 2015, pp. 294–297. doi: 10.1109/APUAVD.2015.7346624.
[19] Meeting minimum hardware requirements in Debian GNU/Linux installation guide, 2023. URL: https://www.debian.org/releases/bookworm/amd64/ch03s04.en.html.
[20] J. S. Al-Azzeh, M. Al Hadidi, R.S. Odarchenko, S. Gnatyuk, Z. Shevchuk, Z. Hu, Analysis of self-similar traffic models in computer networks, International Review on Modelling and Simulations 10(5) (2017) 328–336. doi: 10.15866/iremos.v10i5.12009.
[21] M. Zaliskyi, R. Odarchenko, S. Gnatyuk, Y. Petrova, A. Chaplits, Method of traffic monitoring for DDoS attacks detection in e-health systems and networks, CEUR Workshop Proceedings 2255 (2018) 193–204. URL: https://ceur-ws.org/Vol-2255/paper18.pdf.
[22] A. Fitzgerald, How Many Visitors Should Your Website Get, 2023. URL: https://blog.hubspot.com/blog/tabid/6307/bid/5092/how-many-visitors-should-your-site-get.aspx.
[23] Statista Research Department. How long are you willing to wait for a single webpage to load on your mobile phone before leaving the site?, 2022. URL: https://www.statista.com/statistics/276115/user-patience-with-website-loading-speeds-on-mobile-phones/.
[24] E. Alomari, et al., Botnet-based distributed denial of service (DDoS) attacks on web servers: classification and art, arXiv preprint arXiv:1208.0403 (2012).
[25] G. Mantas, et al., Application-layer denial of service attacks: taxonomy and survey, International Journal of Information and Computer Security 7.2-4 (2015) 216–239.
[26] M. Iavich, S. Gnatyuk, A. Fesenko, Cyber security European standards in business, Scientific and practical cyber security journal 3(2) (2019) 36–39.
[27] Azure DDoS Protection–2021 Q3 and Q4 DDoS attack trends, 2023. URL: https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/.
[28] O. Yoachimik, J. Pacheco, DDoS threat report for 2023 Q4, 2023. URL: https://blog.cloudflare.com/ddos-threat-report-2023-q4.
[29] Cisco annual internet report (2018–2023) white paper, 2023. URL: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.
[30] Five most famous ddos attacks and then some, 2023. URL: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/.
[31] The cost of IT downtime, 2023URL: https://www.the20.com/blog/the-cost-of-it-downtime/.
[32] A. El Tom, The true cost of a ddos attack: protect your business with proactive measures, 2023. URL: https://www.linkedin.com/pulse/true-cost-ddos-attack-protect-your-business-proactive-ali-el-tom/.
[33] Average price of selected malware and DDoS attack services for sale on the dark web as of March 2023. www.statista.com. URL: https://www.statista.com/statistics/1350155/selling-price-malware-ddos-attacks-dark-web/.
[34] W. G. Morein, A. Stavrou, D. L. Cook, A. D. Keromytis, V. Misra, D. Rubensteiny, Using graphic turing tests to counter automated DDoS attacks against web servers, in: Proceedings of the 10th ACM conference on Computer and communications security, Washington, DC, USA, 2003, pp. 1–12.