Two-Level Detection Method of DDoS Attack Mimicking CDN Caches

Kazuya Taniguchi* and Noriaki Kamiyama†
* Graduate School of Information Science and Engineering, Ritsumeikan University, Osaka 567-8570, Japan
†College of Information Science and Engineering, Ritsumeikan University, Osaka 567-8570, Japan
Email: is0512he@ed.ritsumei.ac.jp, kamiaki@fc.ritsumei.ac.jp

***Abstract*—Distributed Denial of Service (DDoS) attacks have become increasingly frequent, overwhelming target servers by flooding them with packets from bots. Though firewalls can block illegitimate traffic, attacks become harder to detect when bots spoof the IP addresses of trusted content delivery network (CDN) cache servers. To mitigate this, we propose a two-stage detection method that uses a dynamic based on packet arrival intervals and DNS log analysis. By incorporating Z-scores, our method adapts to changing traffic patterns and significantly reduces the load on the origin server (OS). Performance evaluations show that our approach enhances DDoS detection accuracy while lowering processing costs, with optimal thresholds designed for varying environments.**

I. Introduction

content delivery networks (CDN) consist of geographically distributed cache servers (CS) that cache and efficiently deliver various types of internet content, such as HTML pages, images, and video. The global CDN market is projected to grow from $19.96 billion in 2024 to $42.46 billion by 2029 [1]. Attacks targeting CDN operations not only threaten the functionality of CDNs but can also lead to significant reputational damage. As such, safeguarding CDNs from security threats is critical. In addition to protecting against content theft and data loss, CDNs must ensure content availability by effectively mitigating security attacks [2].

One of the most prevalent threats in recent years has been Distributed Denial of Service (DDoS) attacks, where bots, widely distributed across a network, flood a target host with large volumes of packets, causing the server to malfunction [3]. While DDoS attacks impose a heavy load on the target server, the impact can be mitigated if multiple servers are involved, as the load is distributed across them [4]. However, attacks on networks utilizing CDN infrastructure can still be effective. For instance, the June 4, 2020, attack on Akamai—a major CDN—was a DNS amplification attack, where the attacker exploited vulnerable DNS servers to generate large amounts of legitimate traffic, peaking at 1.44 Tbps [5].

Moreover, attackers can launch DDoS attacks against an Origin Server (OS) by sending attack packets directly to its IP address. In a CDN setup, the OS only receives requests when the requested content is not available in the selected CS. Therefore, an OS can defend against DDoS attacks by using a firewall to reject any delivery requests not originating from a CS [6] [7]. However, if a bot spoofs the CS's IP address as the source, the firewall will be unable to detect the attack.

An effective approach to protect CDNs involves exploring and analyzing network access logs to detect abnormal behavior and attack patterns [8]. While DNS name resolution logs are typically generated during normal queries from cache servers (CS), direct requests from bots bypass DNS name resolution, leaving no corresponding logs. Therefore, by inspecting DNS logs, it is possible to distinguish between legitimate content delivery requests from CS and DDoS packets from bots. However, checking DNS logs for every delivery request incurs high processing costs.

To address this, we propose a two-stage detection method that sets a threshold for packet arrival intervals, examining DNS logs only when requests from the same IP address arrive at intervals shorter than the threshold. Since attackers may dynamically alter their packet rate to evade filtering, the proposed method employs the Z-score technique [9], which dynamically adjusts the detection threshold. Additionally, we focus on the detection accuracy of DDoS attacks and the method's ability to reduce the load on the OS. The effectiveness of the proposed approach is demonstrated through computer simulations.

Section II provides an overview of related research, while section III details the DDoS attack method. Section IV presents the proposed scheme, followed by a performance evaluation in section V. Finally, section VI offers a summary of the paper.

II. Related Research

A. Possible direct attack on the OS

The OS IP addresses of services that do not use a CDN, such as mail, FTP, and SSH, are public. Therefore, an attacker can collect source addresses from DNS records of these services (e.g., MX records that refer to mail services). Content owners also use hidden subdomains for some services, such as SSH (e.g., ssh.owner.com). Using a dictionary attack, an attacker can guess the hidden subdomains and execute queries to collect the source IP addresses [6]. In addition to these, it has been pointed out that there are various other ways

for attackers to obtain or infer the IP address of an OS [10].

B. Protecting DDoS with CDN and CBSP

CDN and Cloud-based Security Providers (CBSPs) have a common capability to intercept requests to web servers and either serve cached content or forward requests to web servers for dynamic responses [10]. CDN inspect requests and They use intelligent caching techniques and are well suited to provide cloud-based security. Because traffic is already redirected through the CDN, it is easy to chain scrubbing centers and WAFs within the infrastructure. Geographically distributed CDN are ideal for handling distributed attacks and absorbing large volumes of malicious traffic using anycast. the overlapping capabilities of CDN and CBSPs are blurring the lines as CDN providers and CBSPs merge. Thus, it applies to both CDN and CBSPs with security extensions.

III. DDoS ATTACK METHOD BY TRICKING CDN AND DETECTION METHOD BY DNS LOG

The primary patterns of DDoS attacks include the following: Volume-based attacks, where massive amounts of traffic are sent simultaneously to a target host, overwhelming its capacity. Application-layer attacks, which involve precisely targeted assaults on specific applications or services, leading to the exhaustion of server resources. Reflection attacks, in which an attacker sends numerous query packets from bots to public servers (such as DNS servers) while spoofing the IP address of the target host as the origin. This causes the public servers to respond by flooding the target host with a large volume of response packets. These attacks are often combined with IP spoofing, where the attacker disguises their own IP address to anonymize the attack and make it harder to trace. Continuous monitoring, traffic filtering, and robust security measures are critical for defending against these types of attacks.

A. DDoS attacks tricking CDN cache server

DDoS attacks that exploit the IP addresses of CDN cache servers represent a serious security risk to networks using CDNs. The specific attack we focus on in this paper—the IP address of the CDN cache server spoofing—is particularly dangerous. This type of attack is difficult for firewalls to detect because spoofed traffic is often perceived as legitimate, with firewalls generally trusting the source IP addresses of CS traffic. As a result, as shown in Figure 1, DDoS attack packets can bypass the CDN's CS and reach the OS directly, mingling with legitimate traffic and enabling unauthorized access to the OS.

Additionally, spoofing attacks can be used to launch volumetric attacks directly on the OS. When an attacker impersonates a trusted CS IP address, the traffic is accepted as coming from a legitimate source, allowing a flood of requests to overwhelm the OS. This leads to server overload, making it difficult for the OS to process legitimate traffic. Firewalls typically fail to detect this type of attack, and the resulting impact on the OS can be severe, reducing its availability and performance.

In conclusion, IP address spoofing attacks pose a significant risk as they exploit vulnerabilities in the security infrastructure and are particularly challenging for firewalls to detect.

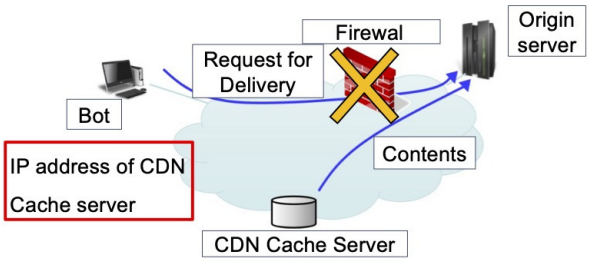

Fig. 1. DDoS attacks against the OS using spoofing of originating addresses

B. DNS Name Resolution Method

It is effective to use the logs left over from DNS name resolution to detect such attacks as described in the previous section: In the case of using a CDN, as shown in Figure 2, when a user requests delivery, in addition to the name resolution procedure with the authoritative DNS server of the content provider (CP), a name resolution procedure with the DNS server of the CDN provider occurs. In addition to the name resolution procedure with the authoritative DNS server of the CP, a name resolution procedure with the DNS server of the CDN provider occurs when a user requests delivery. The name resolution procedures of CDN are shown below.

1) LDNS (Local DNS) server request, CP's authoritative DNS server replies CNAME to LDNS server.
2) LDNS server requests CNAME name resolution to the CDN operator's authoritative DNS server
3) CDN operator's authoritative DNS server selects CS and replies its IP address to LDNS server
4) LDNS server replies to the user with the IP address of the selected CS, and the user accesses the specified CS
5) If the content is not cached in the CS, the CS retrieves the content from the OS, caches it, and delivers it to the user.

Therefore, when a CS makes a normal query, a name resolution log is kept on the CP's DNS server, but a request from a BOT directly using the OS IP address does not use DNS name resolution, so no name resolution log is kept. Therefore, by examining the DNS logs, it is possible to distinguish between normal delivery requests from CS and DDoS packets from bot. However, since many delivery requests arrive at the OS, checking the DNS logs for all delivery requests may increase the processing load on the OS.

C. Challenges of detection method in name resolution

As mentioned in the previous section, checking DNS logs is an important means of detecting IP address

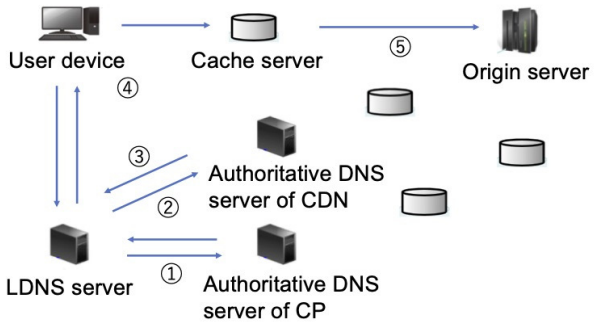

Fig. 2. DNS name resolution

spoofing attacks. However, the load on the OS increases if the DNS log check process is performed for all the large number of delivery requests that arrive at the OS. Excessive DNS log check processing degrades OS performance and delays responses to legitimate traffic, thus defeating the purpose of DDoS attacks against the OS. Therefore, DNS log checking should be limited to only those requests that need it. In this paper, we focus on the difference in request generation patterns between DDoS attacks and normal content delivery requests and propose a method to detect request packets that are highly likely to be DDoS attack packets based on the arrival interval of the previous request for the same content, and to perform DNS log checking only for such packets.

IV. PROPOSED METHOD

The proposed method consists of a two-stage detection method with a threshold between request arrivals to reduce the load of DNS name resolution log checks for the detection of the IP address of the CDN cache server spoofing attacks, and a dynamic threshold setting method using the Z-score method to cope with dynamic environment changes. Details of each technique are described below.

A. Two-stage detection method

While name resolution logs are kept on the DNS servers of content providers during normal queries from CS, requests from the bot directly using IP addresses do not use DNS name resolution, and therefore do not keep name resolution logs. Therefore, by examining the DNS logs, it is possible to determine whether the request is a normal delivery request from a CS or a DDoS packet from a BOT. However, since many delivery requests arrive at the OS, checking the DNS logs for all delivery requests may increase the processing load on the OS. For this reason, a threshold T is set for the query interval, the request arrival interval is measured for each content, and the length of the request arrival interval relative to T is used to narrow down the requests for which it is necessary to check whether a DNS server query has been made. Specifically, requests with an interval longer than T are considered to be normal requests from CS, while requests with an interval shorter than T are checked against the DNS log for the presence or absence of queries, considering the possibility of DDoS attacks from bot. This is due to the difference in request generation patterns: normal delivery requests are generated in the event of cache misses, whereas DDoS packets are generated continuously at short time intervals. If a query is received, it is assumed to be a normal request from the CS, and if no query is received, it is assumed to be a DDoS attack and access is dismissed. This method reduces the load of searching DNS logs and detects DDoS attacks efficiently.

In the two-step detection method for DDoS attacks, if the threshold T is set too large, the number of query checks at the DNS server increases and the OS load increases. On the other hand, if the threshold T is set too small, DDoS attacks may not be detected and the performance of protection against DDoS attacks will deteriorate. To solve this problem, it is necessary to set an optimal threshold T. The threshold T should be maximized within the allowable upper limit of the DNS inspection rate. The upper limit of DNS server throughput is the DNS inspection rate per content for all content. Here, we assume the total DNS inspection rate. To find the optimal threshold T, we need the total DNS inspection rate. Let M be the number of inspected contents and $r_m(T)$ be the DNS inspection rate of m contents for the threshold T, then the total DNS inspection rate R_n for T

$$R_n(T) = \sum_{m=1}^{M} r_m(T) \tag{1}$$

is obtained by $R_n(T) = U_n$. Using this total DNS inspection rate, set T to the maximum value of T such that $R_n(T) = U_n$ when U_n is the upper limit of the DNS inspection rate. In this way, the optimal threshold value T can be obtained.

However, in a network environment where the packet rate is constantly changing, a single fixed threshold cannot be used as a criterion for the first step. In the next section, we describe a dynamic thresholding method based on the Z-score algorithm that enables outlier detection by dynamically adjusting the threshold value for dynamic environments.

B. Z-score method

The proposed method uses the Z-score algorithm as the threshold setting method. The Z-score method is an algorithm for detecting outliers in data. The Z-score algorithm is described below.

$$S_i = \begin{cases} 1 & E_c - \mu_{i-1} > \eta\delta_{i-1} \\ -1 & E_c - \mu_{i-1} < \eta\delta_{i-1} \\ 0 & otherwise \end{cases} \tag{2}$$

$$E_i = \begin{cases} E_c, S_i = 0 \\ \alpha \times E_c + (1-\alpha) \times E_{i-1}, otherwise \end{cases} \tag{3}$$

$$\mu_i = mean(E_{i-L+1}, E_{i-L+2}, \cdots, E_i) \quad (4)$$

$$\delta_i = std(E_{i-L+1}, E_{i-L+2}, \cdots, E_i) \quad (5)$$

When S_i is 1 or -1, an alarm is generated in time slot i. The standard deviation value η is the sensitivity of the signal detection. The influence α is the strength of the signal influence on the signal correction during detection. Outliers are detected using the mean and standard deviation calculated from the estimated values for the past L periods, and the measured values detected as outliers are updated to the weighted sum of the previous measured values. The Z-score method enables outlier detection that reflects past data and detects differences between normal content delivery requests and DDoS attack request generation patterns.

C. Degree of reduction in processing load by introducing the Z-score method

In this paper, the Z-score method is used to detect highly likely DDoS attack packets among arriving delivery requests. Therefore, the degree to which the processing load is reduced from the arrival of DDoS packets to the rejection of DDoS packets is an important evaluation item for the effectiveness of the proposed method. Comparing the processing load with and without the proposed method is roughly equivalent to comparing the processing load for calculating whether or not a packet is an outlier in the Z-score method and the processing load for searching for the corresponding request in a log containing a large number of DNS queries. By comparing the processing load of these two methods using the order notation, we show the validity of the proposed method in reducing the processing load.

In the Z-score, the mean and standard deviation updates in (3)-(5) need to be done for each time slot, whereas (2) needs to be done for each requested packet arrival, so the computational complexity of (2) is dominant, but (2) is only a simple comparison and the time complexity of the Z-score algorithm is $\mathcal{O}(1)$. In addition, although there are various possible DNS log retrieval methods, assuming a simple linear search, the worst-case time complexity of the DNS log retrieval process is $\mathcal{O}(n)$ for the number of entries n. Therefore, we can confirm that the processing load can be reduced by using Z-scores.

D. Detection Method

When a content request arrives at the OS, the interval between the arrival of the previous request for that content is recorded. the Z-score method detects when the difference between the measured value and the average value is large, but compared to normal content requests, requests tend to arrive at shorter intervals in DDoS attacks, and if the arrival interval is used as the Z score, DDoS cannot be detected. Therefore, the inverse of the arrival interval is used as the input E_c of the Z-score method.

The Z-score is then used to check whether an arriving request is an outlier or not, and if an alarm occurs, the DNS log is checked to determine the possibility of an attack and a final attack decision is made. During an ongoing attack, the mean and standard deviation used for Z-score detection are distorted when the mean and standard deviation of the past L are updated using attack data, unlike when the arrival interval of requests is only normal requests. Therefore, after an attack is detected, the mean and standard deviation are not updated using the Z-score method. Then, when the DNS log check results show that the request has been successfully received P times in a row, the attack is considered over and the mean and standard deviation updates are resumed. Using this method, it is possible to check for outliers using only the data of normal requests before the attack occurred.

V. PERFORMANCE EVALUATION

We evaluate the effectiveness of the proposed method by computer simulation. The cache replacement method is an LRU method, and the number of contents is set to $N = 100$ and the cache capacity is set to $C = 10$. The content with the xth popularity is denoted as RANKx. The Z-score parameter was set to $L = 10$, $\eta = 4.0$, and $\alpha = 0.5$. The simulation was run for 10,000 seconds, and DDoS attacks were generated for 3,000 seconds starting 5,000 seconds after the start of the simulation. The attacker was allowed to decide on the termination of the attack after the detection of the attack. The P used to determine the end of attacks after attack detection was set to $P = 5$. We confirm the effectiveness of the proposed method by evaluating the appropriateness of the learning time, the contents with different popularity, and the number of DDoS packet arrivals.

A. Time variation of detection rate

In this section, we evaluate the time variation of the detection rate of DDoS packets using the Z-score of the proposed method. The detection rate is defined as the percentage of DDoS packets arriving at the OS that are detected by the Z-score method, since a DDoS packet can be reliably detected by a DNS check if it is detected by the Z-score method. In order to investigate the impact regardless of popularity in the evaluation, the detection accuracy against elapsed time for the case of attacks against RANK1, RANK50, and RANK100 is shown in Figure3. We set the average interval between DDoS packet occurrences to four patterns: 0.5, 1, 5, and 10 seconds. By observing the change in the detection rate with respect to the time of DDoS packet generation, we consider the point at which the detection rate is almost constant and does not change to be the point at which learning is complete. This suggests that the system has learned the normal occurrence pattern appropriately.

Although DDoS attacks on low-popular content are expected to require more time to learn the Z-score after the simulation starts, even for RANK100, the least popular content evaluated, the detection rate leveled off after 5,000 seconds and remained almost constant. The detection rate is almost constant. As shown above, it

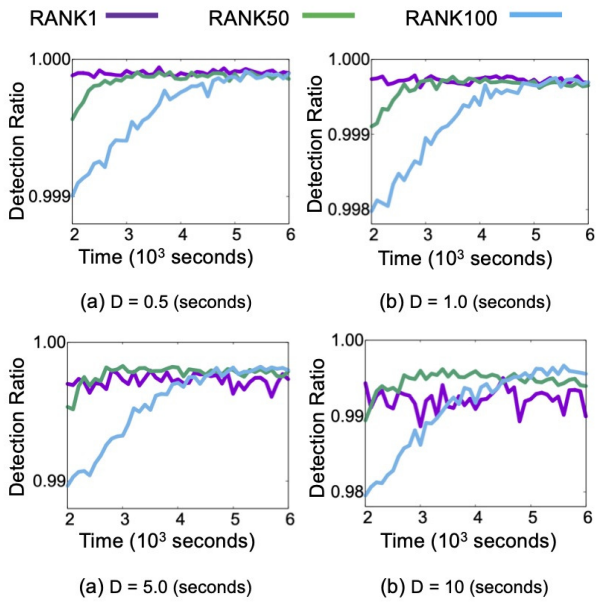

Fig. 3. DDoS packet detection rate over time for each of the three contents

takes 5,000 seconds before an attack can be detected, but after that time, the proposed method can detect attacks almost accurately. In the case of highly popular content, the detection rate of RANK1 remains around 99% even when DDoS packets are generated with an average generation interval of $D = 10$ seconds, although the algorithm of the proposed method is expected to have more difficulty in detecting DDoS packets as the interval between the generation increases.

B. Evaluation for contents with different popularity

In the proposed method, it is expected that low-popularity content with a low arrival rate of requests from CS will be easier to detect DDoS packets because the difference in the rate of request generation from DDoS packets will be more pronounced. On the other hand, for highly popular content, it may be difficult to detect DDoS packets, depending on the set value of the interval between the occurrence of DDoS packets. In such cases, not only is detection difficult, but there are also cases where a normal delivery request is regarded as highly dangerous. However, in such cases, there is no danger of mistakenly rejecting a normal packet because it can be determined by the presence or absence of a query when the DNS name resolution log is checked, and false positives for normal requests from CS can be completely avoided. However, as the number of false positives increases, the number of DNS log detections increases and the processing load on the OS increases. Therefore, it is still better to keep false positives small. In this section, we evaluate the detection rate of DDoS packets and the false positive rate of normal packets for the case where each content is a DDoS target.

As in the previous section, we set the average interval between DDoS packets to 0.5, 1, 5, and 10 seconds. The

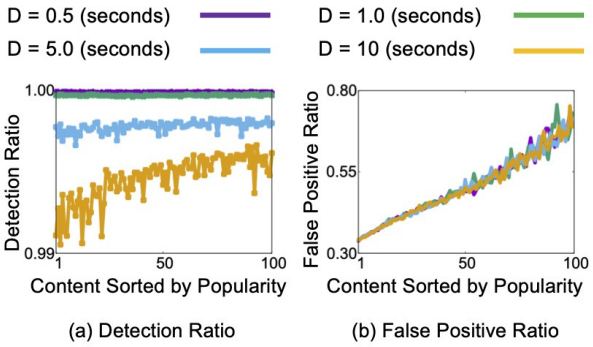

Fig. 4. Detection accuracy when each content is the target of an attack

detection rate of DDoS attacks on each content is plotted in Figure fig:figure2 on the left, and the probability that a normal request (from a CS) is wrongly detected by the Z-score method (False positive ratio) is plotted for each in order of popularity of the non-attack content.

Since the proposed method focuses on the arrival interval, it is presumed that highly popular content is more difficult to detect. However, from Figure4 (a), the detection rate is higher than 0.99 regardless of the popularity. For attacks with a frequency of about $D = 10$ seconds, the detection accuracy is equivalent for all contents. On the other hand, however, the false positives are large, ranging from 0.3 to 0.8. Even if a false positive is detected by the Z-score, the proposed method checks the DNS log as a second step to identify the attack packets to be finally filtered, thus preventing the rejection of normal request packets from the CS. However, the increase in OS processing load due to false positives is an issue.

C. Evaluation of DDoS packets against arrival rate

In this section, we evaluate the detection capability of the proposed method when the arrival rate (inverse of the arrival interval) of DDoS packets is varied. In order to investigate the detection capability for highly dangerous and low-popular content, we evaluate the proposed method for the case of attacks on RANK1, RANK2, RANK3, RANK5, RANK10, and RANK20. In addition to the detection rate of DDoS packets and the false positive rate of normal packets, we evaluate the OS processing load, DDoS attack strength, and attack rate. The DDoS attack strength is the number of times the OS delivers content per unit time, including both normal content delivery and content delivery for missed DDoS packets. The attack rate is the ratio of DDoS packets to attack strength.

In Figure 5, we present (a) detection rate, (b) false positive rate, (c) OS processing load, (d) DDoS attack intensity, and (e) attack rate, plotted against the arrival rate of DDoS packets to the OS for each of the six content types. As anticipated, the detection rate improves as the arrival rate of DDoS packets increases, as shown in (a), where it approaches nearly 100%. The false positive rate, shown in (b), remains largely independent of the DDoS

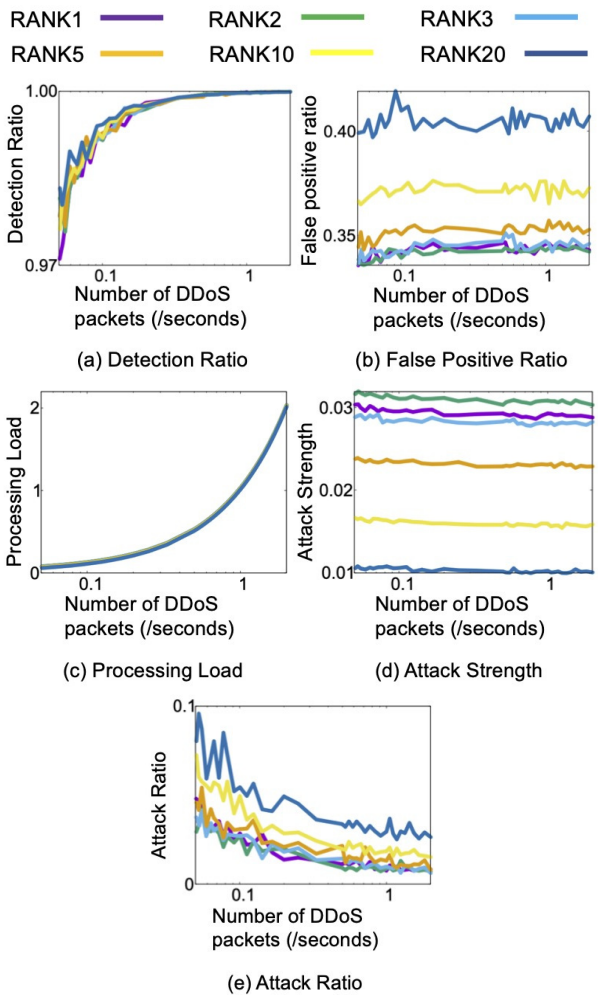

Fig. 5. Impact of DDoS packet arrival rates on various evaluation measures

packet arrival rate but fluctuates based on the popularity of the targeted content. This variability arises because a certain number of short-interval requests are included in the randomly generated traffic. However, as mentioned previously, legitimate content delivery requests are not mistakenly blocked.

The OS processing load, illustrated in (c), rises with the DDoS packet arrival rate, but since the number of false positives for legitimate packets remains stable, the results primarily reflect the attack frequency over time. The detection accuracy for DDoS packets remains high despite the increasing arrival rate. In (d), rejecting DDoS packets influences the cache miss intervals for legitimate content delivery requests, with the delivery frequency increasing as content popularity grows. Interestingly, when comparing RANK1 and RANK2, we observe that while RANK1 receives more requests, its higher cache hit rate leads to shorter average cache miss intervals.

Finally, the results in (e) confirm that the proportion of DDoS packets within the total delivery requests is low, and the rate of missed DDoS detections is minimal, significantly lowering the risk of large-scale volumetric attacks.

VI. SUMMARY

When an attacker identifies the IP address of a CDN cache server, they may launch a direct DDoS attack against the origin server by spoofing the IP address. Such attacks are challenging for firewalls to detect. However, analyzing DNS name resolution logs can help distinguish between legitimate content delivery and DDoS attacks. In this paper, we focus on the differences in packet arrival intervals between normal traffic and DDoS traffic and propose a detection method using the Z-score to dynamically narrow down requests that require DNS log analysis. This approach reduces processing costs due to the lower computational complexity of the Z-score method compared to DNS log retrieval. Numerical evaluation shows that the proposed method is highly effective at detecting DDoS packets. In the future, we plan to develop a technique that focuses on the types of source IP addresses to better isolate potentially compromised CDN cache servers, reducing the number of servers to which the proposed method needs to be applied and further lowering detection costs. We also intend to explore ways to address cache hits on local DNS servers that bypass authoritative DNS logs.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 23K21665, 23K21664, and 23K28078.

REFERENCES

[1] Mordor Intelligence, Content Delivery Network Market Size & Share Analysis - Growth Trends & Forecasts (2024 - 2029), https://www.mordorintelligence.com/industry-reports/content-delivery-market
[2] R. Guo, W. Li, B. Liu, S. Hao, J. Zhang, H. Duan, K. Shen, J. Chen, and Y. Liu, CDN Judo: Breaking the CDN DoS Protection with Itself, Network and Distributed Systems Security (NDSS) Symposium 2020
[3] O. Sanchez, M. Repetto, A. Carrega, R. bolla, Evaluating ML-based DDoS Detection with Grid Search Hyperparameter Optimization, 7th International Conference on Network Softwarization (NetSoft) 2021
[4] GMO.INTERNET GROUP, https://www.gmo.jp/security/cybersecurity/vulnerability-assessment/blog/ddos-attack/
[5] D. Wagner, D. Kopp, M. Wichtlhuber, C. Dietzel, O. Hohlfeld, G. Smaragdakis, A. Feldmann, United We Stand: Collaborative Detection and Mitigation of Amplification DDoS Attacks at Scale, ACM CCS 2021
[6] M. Ghaznavi, E. Jalalpour, A. Salahuddin, R. Boutaba, D. Migault, and S. Preda, Content Delivery Network Security: A Survey, IEEE Communications Survey & Tutorials, Vol. 23, No. 4, Fourth Quarter 2021
[7] Protecting Websites from Attack with Secure Delivery Networks, Comp. Mag, 2015
[8] L. Yang, A. Moubayed, A. Shami, P. Heidari, A.Boukhtouta, A. Lalabi, R. Brunner, S. Preda, D. Migault, Multi-Perspective Content Delivery Networks Security Framework Using Optimized Unsupervised Anomaly Detection, IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT 2021
[9] C. Hou, H. Han, Z. Liu, and M. Su, A Wind Direction Forecasting Method Based on Z Score Normalization and Long Short Term Memory, ICGEA 2019
[10] T. Vissers, et al., Maneuvering Around Clouds: Bypassing Cloud-based Security Providers, ACM CCS 2015