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
†College of Information Science and Engineering, Ritsumeikan University, Shiga 525-8577, Japan
Email: tanikazu1221@icloud.com, kamiaki@fc.ritsumei.ac.jp

***Abstract*—In recent years, DDoS (Distributed Denial of Service) attacks have been occurring frequently, in which bot, widely distributed throughout the network, send large numbers of packets to the target host, causing the target server to malfunction. Since DDoS attacks place an enormous load on the target server, the attack is not feasible if there are multiple servers, as the load is distributed among them. If an attacker knows the IP address of the origin server (OS), the attacker can establish a DDoS by sending attack packets from the bot to the OS IP address. Such an attack can be prevented by using a firewall that allows the OS to reject packets other than those requested by the CS. If a bot sends packets to the OS falsely claiming the CS IP address as its originating address, it cannot be detected by firewall. By examining DNS logs, it is possible to determine whether the packet is a normal delivery request from the CS or a DDoS packet from a bot. However, if DNS logs are examined for all delivery requests, the processing cost increases. Therefore, the authors focused on the fact that many DDoS attack packets are generated in short time intervals and proposed a two-stage detection method for DDoS attacks that trick CDN and a method for designing optimal thresholds, in which a threshold is set for arrival time intervals and DNS logs are checked only when requests arrive from the same IP address at intervals below the threshold. However, in general, the occurrence pattern of delivery requests changes dynamically, and this method sets a fixed threshold value, making it difficult to cope with dynamic environments. In this paper, we propose a two-stage detection method using Z-scores that can cope with dynamic environments. In the performance evaluation, we evaluate the detection accuracy of DDoS attacks and prove that the proposed method reduces the load on the OS. We also evaluate the required training time of the proposed method by evaluating the detection accuracy against the training time.**

I. Introduction

CDN (content delivery networks) consist of geographically distributed cache server (CS) that cache and efficiently deliver Internet content such as HTML pages, images, and video. The market value of CDN is expected to reach $42.46 billion by 2029 from 19.96 billion in 2024[1]. Attacks on CDN operations have the potential to compromise CDN functionality and generate negative publicity. Therefore, it is very important to protect CDN from security attacks. In addition, CDN must not only protect against content theft and loss, but also ensure content availability by mitigating security attacks.

On the other hand, DDoS (Distributed Denial of Service) attacks have been occurring frequently in recent years, in which a large number of packets are sent to the target host by bot that are widely distributed on the network, causing the target server to malfunction. Since DDoS attacks place an enormous load on the target server, if there are multiple servers, the load will be distributed and the attack may not be as impactful[4]. However, attacks targeting a network using a CDN have the potential to succeed. In fact, the June 4, 2020 attack targeting Akamai, a CDN, was a DNS amplification attack in which the attacker used a vulnerable DNS server to send a large amount of legitimate traffic. The peak attack traffic was reported to be 1.44 Tbps[5].

In addition, an attacker can launch a DDoS attack targeting an OS by sending attack packets directly to the OS (Origin Server) using the Origin Server's IP address as the destination. However, with CDN, requests are delivered to the OS only when the requested content does not exist in the selected CS in response to a user's delivery request. Therefore, if a BOT sends a packet directly to the target OS, the OS can prevent a DDoS attack by rejecting delivery requests from non-CS through its firewall[2][6]. However, if a BOT sends a packet to the OS falsely claiming the CS's IP address as its originating address, the firewall cannot detect it.

While name resolution logs are kept on the DNS servers of content providers during normal queries from CS, direct requests from bot do not use DNS name resolution, so no name resolution logs are kept. 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, if DNS logs are examined for all delivery requests, the processing cost increases. To address this issue, we propose a two-stage detection method that sets a threshold for the arrival time interval and checks the DNS logs only when a request arrives from the same IP address at an interval less than the threshold. An attacker may dynamically change the packet rate to avoid filtering. Therefore, the proposed method uses the Z-score method [7], which can dynamically change the detection threshold. Additionally, we focus on the detection accuracy of DDoS attacks and the effectiveness of the proposed method in reducing the OS load. We clarify the effectiveness of the proposed method through computer simulations.

We describe the DDoS attack method that tricks CDN cache servers and the detection method using DNS logs in section , II, and related research in section III. Section, IV describes the details of the proposed method, section, V presents the performance evaluation results, and section, VI summarizes the whole paper.

II. DDoS ATTACK METHOD BY TRICKING CDN CACHE SERVER AND DETECTION METHOD BY DNS LOG

The main DDoS attacks include the following attack patterns. In a volume attack, a large amount of traffic is sent to the target host simultaneously, causing the target host to become overloaded. Application-layer attacks, in which a tightly planned attack is launched against a specific application or service, resulting in server resource exhaustion. In a reflection attack, an attacker sends a large number of query packets from a BOT to a public server, such as a DNS server, falsely claiming the IP address of the target host as the originating address, causing a large number of public servers to send a large number of response packets to the target host. This is sometimes combined with IP spoofing, in which the attacker spoofs his own IP address to anonymize the attack and make it difficult to trace. Continuous monitoring, traffic filtering, and security measures are essential to counter these attacks.

A. DDoS attacks tricking CDN cache server

There are DDoS attacks in which attackers exploit the IP addresses of CDN CS. The CDN CS IP address spoofing attack we focus on in this paper poses a serious security risk to networks using CDN. First, this attack is difficult to detect by firewalls. First, this attack is difficult to detect by firewalls because spoofing is considered legitimate traffic, and firewalls usually trust the source IP address of traffic as a CS. Therefore, as shown in Figure 1, DDoS attack packets can reach the OS directly without going through the CDN's CS, mixing with legitimate traffic and causing unauthorized access to the OS.

In addition, spoofing attacks can also be used as a means of launching volume attacks directly against the OS. If an attacker impersonates a trusted IP address of a CS, the traffic is accepted as coming from a trusted source, and a large number of requests may be sent to the OS under attack. This overloads the OS, making it difficult to handle legitimate traffic. Firewalls typically fail to detect this attack, and the direct impact on the OS can be significant, resulting in reduced availability and performance. Thus, CDN CS IP address spoofing attacks pose a serious risk because they exploit weaknesses in the security infrastructure and are difficult for firewalls to detect.

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,

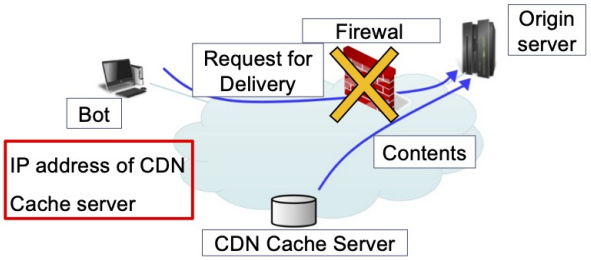

Fig. 1. DDoS attacks against operating systems using spoofing of originating addresses

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 content provider (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 a large number of 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 CDN CS IP address 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

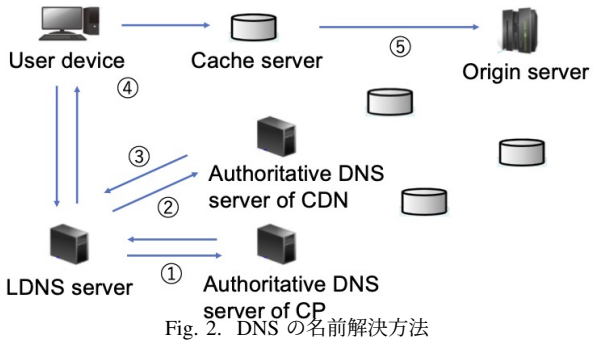

Fig. 2. DNS の名前解決方法

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.

III. 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[2]. 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[8].

B. Protecting DDoS with CDN and CBSP

CDN and CBSPs (Cloud-based Security Providers) have a common capability to intercept requests to web servers and either serve cached content or forward requests to web servers for dynamic responses. 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.

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 CDN CS IP address 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 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 a large number of 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.

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

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

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

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

When S_i is 1 or -1, an alarm is generated in time slot i. The threshold 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 make a decision 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

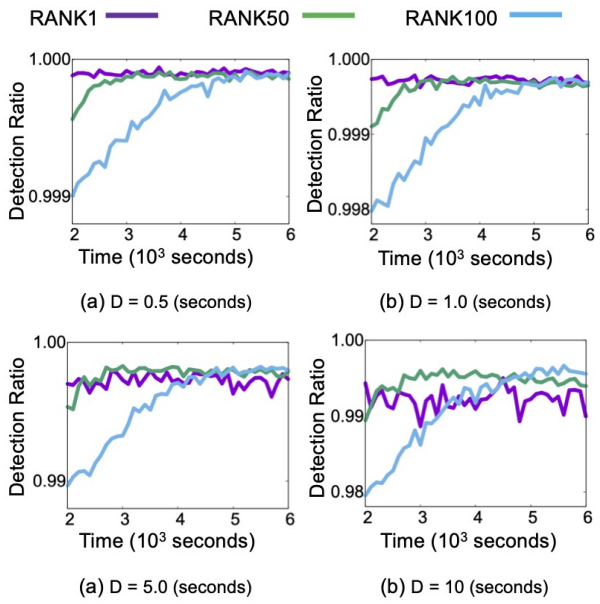

Fig. 3. DDoS packet detection rate over time for each of the three contents

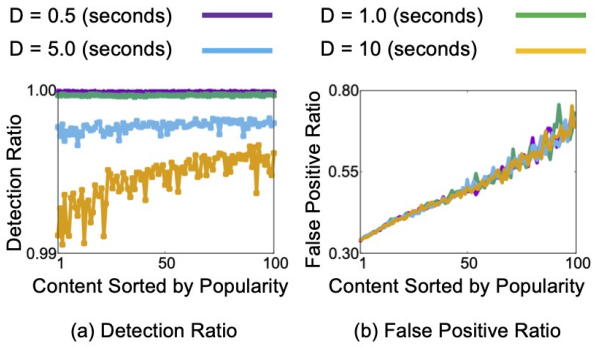

Fig. 4. Detection accuracy when each content is the target of an attack

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 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 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.

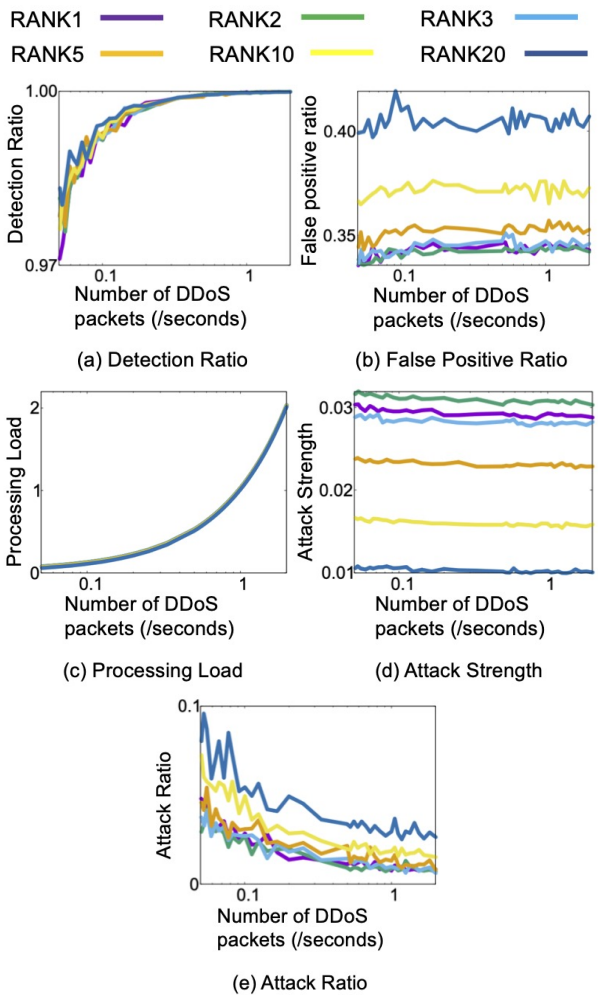

Fig. 5. Impact of DDoS packet arrival rates on various evaluation measures

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 plot (a) detection rate, (b) false positive rate, (c) OS processing load, (d) DDoS attack strength, and (e) attack rate against the arrival rate of DDoS packets to the OS when DDoS attacks are generated for each of the six contents. The detection rate is expected to be higher when the arrival rate of DDoS packets is higher, and the results in (a) reflect such an assumption. The detection rate increases as the detection rate increases, reaching almost 100%. The false positive rate is independent of the rate at which DDoS occurs, and varies depending on the popularity of the content under attack. This is because a certain number of short interval requests are included in the randomly generated requests. However, as mentioned in the previous section, normal content delivery requests are not mistakenly rejected. The processing load increases as the arrival rate of DDoS packets increases, but the number of false positives for normal packets remains constant and almost unchanged, so the results reflect the number of attacks that occur per unit time. Considering the arrival rate, the detection accuracy of DDoS packets is high. From (d), by rejecting DDoS packets, the results reflect the interval of cache misses of normal content delivery requests. In other words, the delivery frequency increases with the popularity of the content. However, a comparison of RANK1 and RANK2 shows that RANK1 has a higher number of requests but a higher cache hit rate, resulting in a reversal in the average interval of cache misses. The result in (e) confirms that the percentage of DDoS packets included in delivery requests is low; DDoS packet detection misses are small, greatly reducing the risk of volume attacks.

VI. SUMMARY

When the IP address of a CDN cache server is identified by an attacker, a direct attack on the origin server by DDoS packets that deceive the IP address is assumed. Such an attack is difficult to detect by firewalls. However, by checking DNS name resolution logs, it is possible to distinguish normal content delivery from DDoS attacks. In this paper, we focus on the difference in arrival interval between normal packets and DDoS packets, and propose a detection method that uses the Z-score method to dynamically narrow down the requests that need to check the DNS log. The proposed method can reduce the processing cost due to the difference in computational complexity between the Z-score method and DNS log retrieval. Numerical evaluation results show that the proposed method is highly accurate in detecting DDoS packets. In the future, we plan to implement a method that focuses on the type of source IP address to narrow down the list of dangerous CDN cache servers, thereby eliminating the need to apply the proposed method to a large number of CDN cache servers and reducing the cost of detection.

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] 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

[3] 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
[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] Protecting Websites from Attack with Secure Delivery Networks, Comp. Mag, 2015
[7] 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
[8] T. Vissers, et al., Maneuvering Around Clouds: Bypassing Cloud-based Security Providers, ACM CCS 2015