SDN-based Detection and Mitigation System for DNS Amplification Attacks

Kaan Özdinçer
Department of Computer Engineering
Gebze Technical University
Kocaeli, Turkey
Email: kaanozdincer@gtu.edu.tr

Hacı Ali Mantar
Department of Computer Engineering
Gebze Technical University
Kocaeli, Turkey
Email: hamantar@gtu.edu.tr

***Abstract*—Abstract—DNS amplification is a type of reflection-based DDoS attacks, and they are very hazardous for the reliability of victims within the network. To prevent or mitigate such attacks, a significant amount of work is being done both on conventional networks and on SDN-based networks. This study aimed to detect and reduce the effects of DNS amplification attacks in SDN-based with the developed system. This system aims to monitor the variations in the amplification factor and TTL header to initiate mitigation and sustain the victim's life. It also ensures that legitimate packets are not suspected in the process. In doing so, it is aimed to generate alarms and mitigation by using the central management feature of SDN, by writing the metrics into a time series database immediately. Experimental results show that this system can be used SDN-based networks and prevent an attack in reactively. It has also been observed that it can be used not only for DNS amplification attacks but also for other UDP-based amplification/reflection attacks.**

***Index Terms*—DNS Amplification, SDN, DDoS, TTL, Amplification Factor**

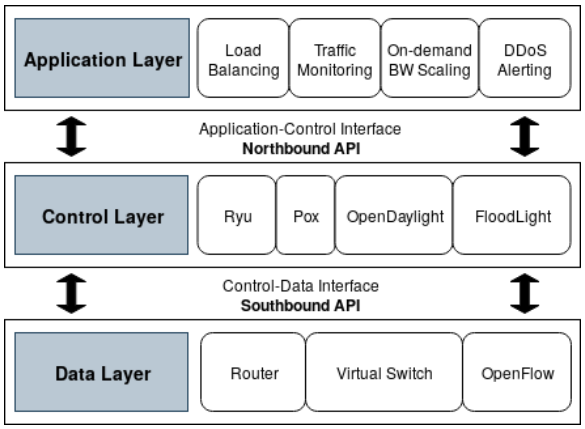

Fig. 1. The three-layer architecture of SDN.

I. Introduction

When we talk about SDN and security, two main issues come to mind. The first is the security of SDN's infrastructure and the protocol used by SDN-based network. The second is SDN-based security applications. In this study, a sub-branch of the second subject will be studied.

As discussed in the study of Scott-Hayward et al. [1], a considerable amount of work on SDN-based security applications. Also, studies have been carried out on many different security weaknesses. Solutions are created against unauthorized access, protecting controller from intruders, distributed control model, and intelligent systems that understand and take banned accesses accordingly. Solutions for malicious applications on the network are: addressing the on-demand resources, detecting and blocking by looking at the behavior of applications, identifying new paths to address possible network problems have been produced. Steps generally taken against network attacks are: understanding the attack from network patterns and making the necessary network settings, scaling the controller, verifying the source address, real-time network monitoring, policy conflicts detection for misconfiguration. Security of the controller's channel that communicates controller debugger against system-level SDN problems are: collecting data, analyzing traffic and update of rules, DoS/DDoS protection, middlebox security and infrastructure security problems, agile network access control and global traffic monitoring for ad-hoc networks like IoT networks.

In this study, the related network information will be aggregated (Figure 1) and mitigation stage will be tried to prevent DNS-amplification attacks which is a type of DRDoS attacks.

A survey conducted by IHS Markit [2] with 23 service providers showed that 2 out of 3 of these service providers would use SDN by the end of 2018 and this would increase to 87% by the end of 2019. Top two reasons for investing and deploying SDN are simplification and automation of service provisioning and controlling all parts of network elements easily.

DNS-amplification DDoS attacks are a way to targeted server or network that sends heavy traffic to make bottlenecks. They aim to exhaust the resources of the target (server or network). Small queries sent to DNS servers provide very large answers and the source IP address is targeted by the victim's IP address (Figure 2). Since UDP is connectionless, with a small request, a big answer can be obtained. Besides, because of the real IP address is hidden, it becomes a reflection attack. This type of attack can be resolved by preventing DNS servers that are used as reflectors from responding to the ANY

978-1-7281-3789-6/19/$31.00 ©2019 IEEE

query. However, servers need to respond to this query or have this feature turned on by mistake. In these cases, network administrators need to take action.

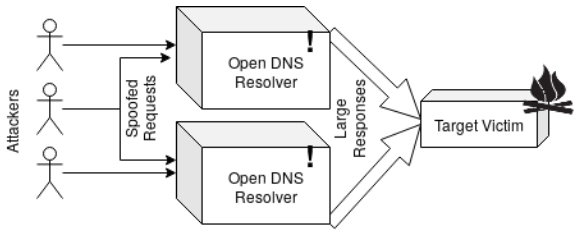

Fig. 2. Dns Amplification Attack.

The ratio of the total size of the requests to the DNS server used as the reflector to the total size of the responses is called the amplification factor. The amplification factor is one of the most important metrics for the measurement of the attack. In addition, verification of the spoofed IP address is another important circumstance for understanding the attack. Also, with similar protocol features, the same type of attack can be done using NTP [3] etc. (Table I). The product of this study is expected to be able to detect and prevent NTP amplification attacks.

TABLE I
AMP FACTORS AND PORTS PER NETWORK SERVICE PROTOCOLS.

Protocol	Port	Amp Factor
SNMP	161	6.3
NTP	123	556.9
DNS ns	53	28
DNS or	53	54
NetBios	137	3.8
SSDP	1900	30.8

II. RELATED WORKS

Several approaches have been presented in the literature regarding the DDoS detection and mitigation of DNS amplification attacks. The methods that stand out are the time-based solutions, statistical entropy methods, ML algorithms based on distribution patterns, DNS query history based matching methods etc.

Dharma et al. [4] offer a time-based solution for DDoS detection and mitigation on SDN controller. The method aimed to observe high traffic, clustering the patterns and threshold-based mitigation.

K. Kalkan et al. [5] used statistical entropy methods for the detection and prevention of DDoS attacks in SDN. This method reduces system load compared to ML algorithms. Furthermore, due to the statistical method, the proposed procedures can detect and mitigate unknown attack types.

In that context, their previous work SDNScore [6] is a packet-based statistical model that relies on capable switches. It investigates packets and calculates scores with their attribute values.

R. Wang et al. [7] also used a statistical model to generate profiles based on statistical features collected during a legitimate period and drop attack packets by comparing the packets with these profiles.

L. Li et al. [8] proposed an attack detection algorithm by calculating distribution patterns in network packet headers. It calculates cumulative traffic entropy instead of packet classification and detects illegitimate traffic. The method implements anomaly detection instead of using predefined thresholds. If the anomaly continues for a while, the traffic is marked as abnormal.

Another entropy-based approach [9] used a distributed entropy calculation. Source IP address entropy and Chi-square method is used. But in this case, memory and computational overheads are also converged at a single point.

H. Wang. et al.[10] [11] developed Hop Count Filter (HPC) by finding hop counts from the TTL value in the IP header and matching it to the source IP address. Then designed a defensive system against spoofed IP addresses using HPC.

Chih-Chieh Chen et al. [12] offer a detection based on machine learning for DRDoS attacks. The system categorizes DNS/NTP packets in SDN with an SVM classifier and blocks what they think is amplification.

Several predefined threshold approach systems were proposed. Comparison of the data collected by Sflow with the predefined thresholds [13]. D. Huistra et al. [14] presented a method based on the the calculation of DNS packet sizes and threshold-based detection.

J. Zeng et al. [15] proposed a technique named reinforcing anti-DDoS action in realtime (RADAR), a real-time defense application developed on commercial off-the-shelf (COTS) unmodified SDN switches to detect and defend against a wide range of flooding-based DDoS attacks within adaptive correlation analysis.

The work presented by [16] et al. provides a fair solution for DNS amplifications with keeping a one-to-one match for DNS queries, responses and understands spoofed packets. Whenever a suspicious packet arrives, the counter is incremented by one and the attack alert works when the specified threshold arrives. The problem in network is very large data must be stored.

VAVE [17] Virtual Source Address Validation Edge, expands NOX controller for investigating flow entries, filtering and address validation for DDOS prevention. It validates incoming packets and makes decisions.

Bohatei [18] monitors virtual routers, detects DDoS attacks if traffic flows the victim and exceeds a threshold.

This survey [19] demonstrates the categorization of DRDoS about amplification preventing, source ip spoofing detecting and filtering, location of defense (Figure 3).

In this study, a lightweight and reactive system was designed by using the size of the request, response packets and variations of the TTL value in the IP header, predefined thresholds and looking at only statistical changes without keeping historical data.

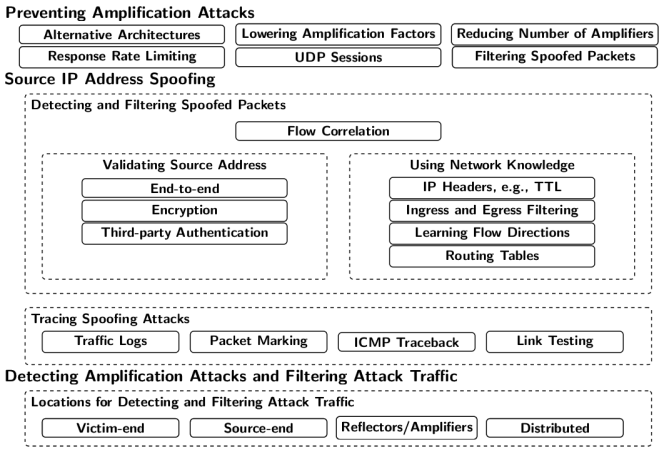

Fig. 3. Categorisation of DRDoS Literature.

III. The Proposed System

In this section, we describe our detection and mitigation mechanisms in detail. In this study, a system has been developed against DNS DRDoS attacks on SDN. We assume that one or more DNS servers and victim computers exist in SDN-based network. By using the DNS server as a reflector and amplifier, a huge UDP traffic will flow on the victim computer. Detection and mitigation of the attack will be examined separately, and a two-phase phase method will be applied during detection and aim to be mitigated. It also has a preliminary process pointing to the location of DNS servers.

A. Amplification Factor

The ratio between the size of the request to the DNS servers and the size of the response is called the Amplification Factor(AF) [19] [13]. It is known that the answer to the Recursive DNS query is greater than the answer to a query such as A or CNAME [20]. Two equations are generally used to show the AF, as following.

$$AF_1 = size(response) \div size(request)$$

$$AF_2 = sumofsize(response) \div sumofsize(request)$$

AF_1 gives the ratio of size of the response to that of the corresponding request. AF_2 measures the cumulative size of responses divided by that of a flow of the corresponding request. The increase in this metric is characteristic of a possible attack [21]. However, where the answer to this query is small or the responses of the TXT, NS, MX records are huge, the AF may not be sufficient to understand the attack [22]. In literature two defined the amplification factor exists. Bandwith amplification factor (BAF) is the bandwidth multiplier referring to the number of UDP payload that an amplifier response, compared to the number of UDP payload of the request. Packet amplification factor (PAF) is a packet multiplier referring to the number of IP packets, to which an amplifier responds. that an amplifier responds to. In this study, we use DNS ANY queries as a bandwidth amplification factor.

B. TTL & Hop Count

TTL is used to prevent packets from being routed forever when the destination host can not be located in a fixed number of hops. The TTL is a valuable header for detecting spoofed packets. When routes change, they do not cause an important change in the number of hops. There are two important pieces of information for TTL header in the IP header. The first is TTL initial value, the second is a decrease in this number for each hop it passes. This initial number varies according to the type of host [23] which the IP packet is created. Due to changes in the default number of the client on which the IP packet was created, Hop Count should be calculated. Because of the operating systems of the clients used by attackers who have hidden the source, the IP address of the victim computer may be different. As stated in the study [24], the HC is calculated as in (Alg. 1). With HC, the number of distinct values within a given window range is calculated. This is Hop Count Variation (HCV).

Algorithm 1 ttl2hc: - Compute HC value using TTL

Require: $ttl > 0$
1. **if** $ttl < 33$ **then**
2. $hc = 32 - ttl$
3. **else if** $(ttl) < 65$ **then**
4. $(hc) = 64 - ttl$
5. **else if** $96 < ttl < 129$ **then**
6. $hc = 128 - ttl$
7. **else if** $224 < ttl < 256$ **then**
8. $hc = 255 - ttl$
9. **else**
10. **return** $hc - 1$
11. **end if**
12. **return** hc

C. Attack Detection

Initially, to determine the location of the DNS servers in the network, the server is scanned periodically. For this purpose, only the packets coming to port 53 and from port 53 for this scan are decoded up to layer7 and it is decided whether it is a DNS server or not. Although port 53 is used by default for DNS servers, it can optionally be used by another application. Thus, non-DNS servers that communicate through port 53 will not be monitored. This phase must be done before proceeding to the two-phase detection stage. Because the metric calculation will be started immediately in the next phase. Actually, This attack type includes amplification, reflection, and IP spoofing behavior. Therefore, two-phase method has been proposed. And finally, we will call this system and it's implementation briefly YARASA(Yet Another Reflection/Amplification Attack in SDN Application) is represented in Figure 4. YARASA decodes packets up to layer 4 only. It does not look at packages until the application layer.

Phase 1: Amplification Factor Monitoring
In this section, the lengths of requests and outgoing responses of detected DNS servers are collected via controller and AF is calculated. Increasing AF value means an attack may be taking place. DNS amplification attack is usually used with ANY queries. However, the AF value can be increased by making other queries such as DNSKEY, TXT, NS, MX. For this reason, increasing AF does not mean that the attack has started immediately. To eliminate this problem, the detection stage has phase 2. There are 2 thresholds for the AF examined in this phase. Lower threshold and upper threshold. When the lower threshold is exceeded, the YARASA goes to phase 2 to avoid dropping the legitimate queries. When the upper threshold is exceeded, the second phase is skipped and the goes to the mitigation stage directly.

Phase 2: Hop Count Variation Monitoring
This section will examine the change of TTL values in the IP headers of client's DNS requests to the server with a high AF value. This high AF value is examined for the cause of the attack. If the HCV value is higher than the specified threshold value, then the mitigation stage is started. If not, nothing will be done and phase 2 will continue to wait for a trigger from phase 1. This means the AF value was high in phase 1, but this is not a DNS amplification attack. HCV calculation is not computing continuously, only when phase 2 is reached. This eliminates the need for excessive computational loads and a large data storage like IP-TTL mapping tables. Because they maintain an IP-TTL mapping table for each client IP address. However, YARASA only performs this calculation for the clients of the DNS servers which are passed to phase 2. If there is an attack with the AF value that remains between 2 thresholds and the HCV does not change, a possible attack may continue. However, it will not be large enough to cause damage. To cause damage, it is necessary to pass the 2nd threshold, in which case attack will be passed to the mitigation stage. In any case, the selection of the 1st threshold value affects the number of passes to phase 2. The choice of the value of the 2nd threshold affects the mitigation of underloaded attacks.

D. Attack Mitigation

If YARASA has triggered the Mitigation stage, then there is a DNS server used as a reflector. The victim is blocked from responding to the DNS server to reduce incoming traffic. In practice, even if the DNS server responds, these packets will be dropped on the first switch. In other words, FLOW_MOD is sent to the switches by the controller. This drops the amplified responses generated by the DNS server from the network. This is to represent closing the recursive query of the Open Resolver DNS server as recommended here [25]. Optionally, requests can also be blocked. However, in this case, legitimate requests by the Victim will be dropped. Although Source IP validation and rate-limiting are recommended as other mitigation techniques, they will not be applied in this study. Because the system that we want to focus on in this study is the two-phase detection system.

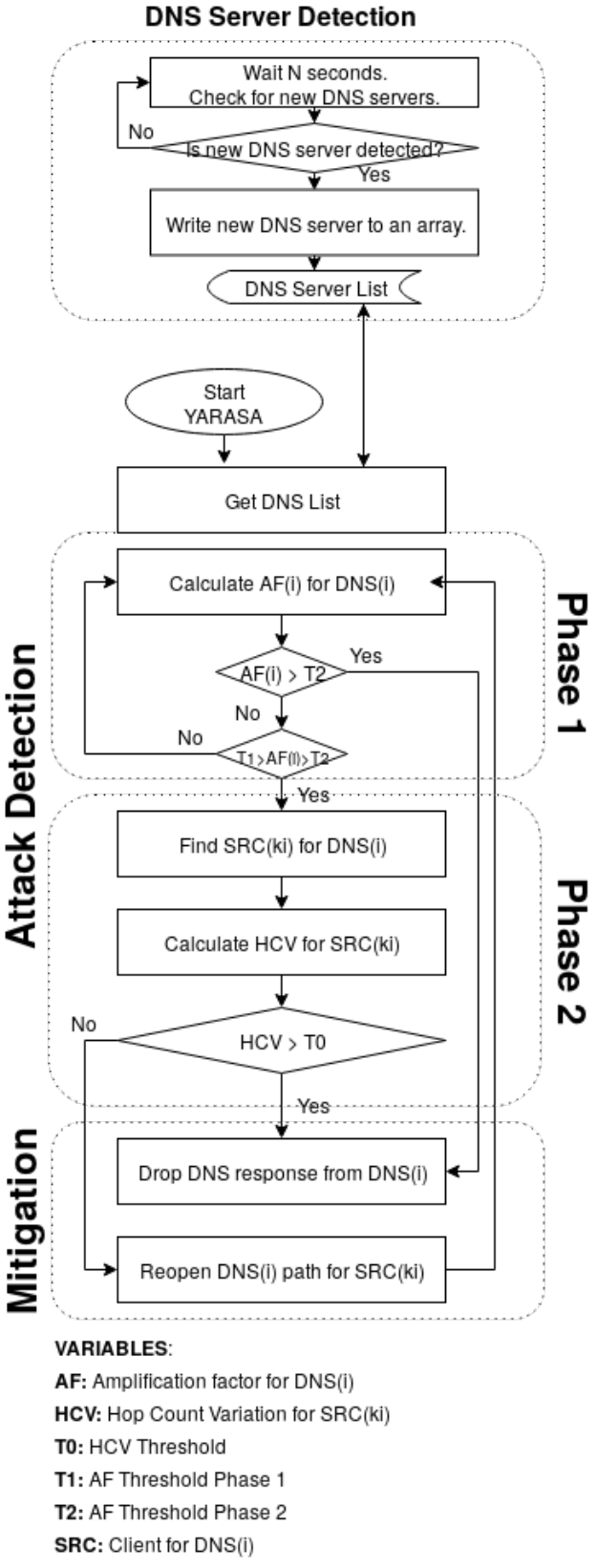

Fig. 4. Flowchart of proposed system.

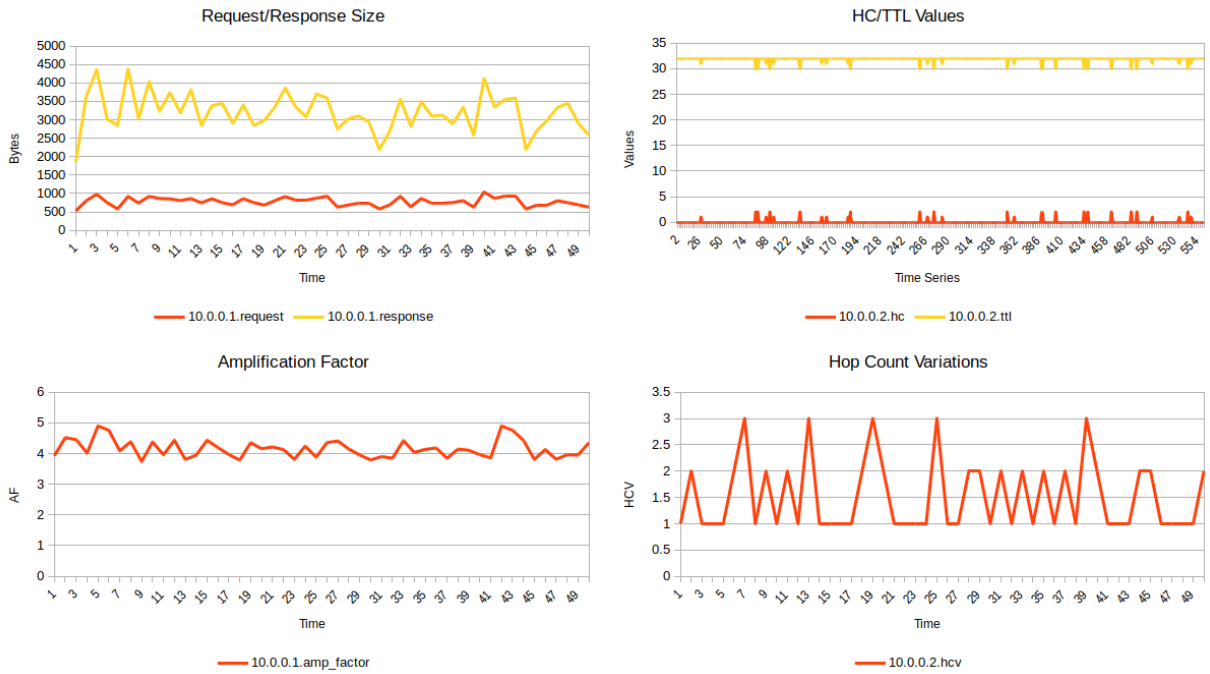

Fig. 5. Legitimate Traffic Metrics

IV. Experiments And Results

In this section, the infrastructure of the algorithm is detailed. The infrastructure is then tested against network traffic that is generated using real life data. The response of the system as well as its effectiveness against potential attacks are tested. Particularly, AF and HCV values were studied to understand the performance of the system against DNS amplification attacks.

A. Infrastructure

First, mininet and Open vSwitch are used for the network infrastructure. As a SDN controller, Python programming language was used to programming RYU controller, Bind9 as DNS server, Scapy to generate packets. The data aggregate with the controller was written to influxdb (a time-series database) and Chronograph was used for instant graphical monitoring. AF and HCV were calculated with the queries made to Influxdb and the necessary mitigation operations were performed with the OpenFlow interface of RYU. To PCAP file analysis, wireshark, termshark and tshark were used.

B. Datasets

AmpPot [26] dataset was used as the basis for assault experiments. This data set was taken from the honeypod capturing DRDoS. The data set contains the traffic data (PCAP) captured by AmpPot. This dataset includes PCAP files of an AmpPot from May 31st, 2015 to January 6th, 2015. Data includes only query packets of CharGen, DNS, NTP, SSDP. In 2013, Booter [27] contains approximately 250 GB of real-world DDoS traffic. This data set was used to test the proposed system and provide normal DNS data.

C. Experiments

The data provided in [27] was used to generate legitimate traffic using the datasets mentioned above. Then, when we look at the graphs, we see Figure 5, a regular request/response ratio and HC/TTL variation. The above graphs are combined graphs of the metrics used, and all the AF and HCV values calculated using these metrics. After that, we only need to look at the AF and HCV charts. Suppose phase 1 threshold is 5. This traffic will never enter the mitigation system. Suppose phase 1 threshold is 5, phase 2 threshold is 20 and HCV threshold is 6.

When we generate a traffic with a high traffic TXT query and set AF threhold as 4, we see the graphs in Figure 6. The AF threshold was crossed several times, even though it was at the threshold. However, no mitigation mechanism has been observed. Because if there is an attack, we expect the HCV value to change. That's why a system that only looks at AF has not been developed. There is no need to look after a certain value of AF, as the importance of this system emerges in such intermediate values. For example: The DNS infrastructure may generate high AF values. But that doesn't mean there's an attack.

Take a look at the attack and running a mitigation system in a 1-minute window in Figure 7. In fact, it's a 10-minute graph, but the moment of attack and blocking is visible in a 1-minute window. After the increase in AF, passed to phase 2 and the Open Resolver DNS traffic dropped immediately because the HCV is higher than the threshold. Here, the central structure of SDN and high adjustable feature, it is possible to take immediate action against the detected attack. The real

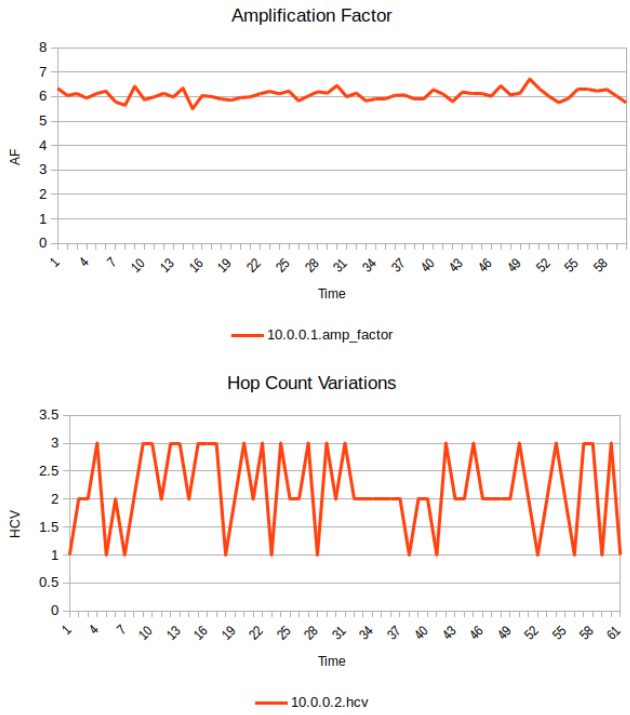

Fig. 6. Legitimate TXT query Traffic Metrics

reason for the mitigation speed here is that an attack occurs above the phase 2 threshold value greater than defined in the system. Thus, it is possible to proceed directly to the mitigation stage without the need to calculate HCV.

Since the characteristics of the UDP-based amplification attacks are similar, we have also been able to test the NTP attack in the dataset we have. And in a very similar way, we achieved to mitigate the attack. This suggests that if the correct thresholds for the proposed system can be determined, it can mitigate many DRDoS attacks.

V. Conclusion and Future Works

In conclusion, the proposed system has been tested in real datasets and it has been found that it performs a fast detection and mitigation in SDN. Since system did not perform a DNS-specific examination, it was found suitable for UDP-based amplification attacks. The system does not maintain a TTL table for each DNS server client. It only performs a simple TTL variation/entropy calculation. Thus, it can prevent these attacks by spending fewer resources. With its two-phase structure, it consumes resources only when needed. Also on the SDN controller, YARASA only decodes packages up to Layer4, consumes fewer resources and does not indicate application-level packages. In summary, YARASA is a lightweight system that reactively detects and mitigates UDP-based amplification/reflection attacks.

As can be seen in the literature [5] [12], ML-based systems are quite high. An ML-based detection scheme can be studied in the future by implementing ML-based prediction system for detection phase. All experiments performed in this study were performed for IPv4. No IPv6-specific work has yet been performed. Its reactive structure allows it to work without any prior knowledge. Presenting a more active structure, initiating mitigation with rate-limiting is in the future works. A function that works better for calculating the TTL variation can be found. As a future job, it can be integrated into an already written and used IDS. In the future, the integrity of the proposed SDN-based detection and mitigation system will be more studied against large network datasets.

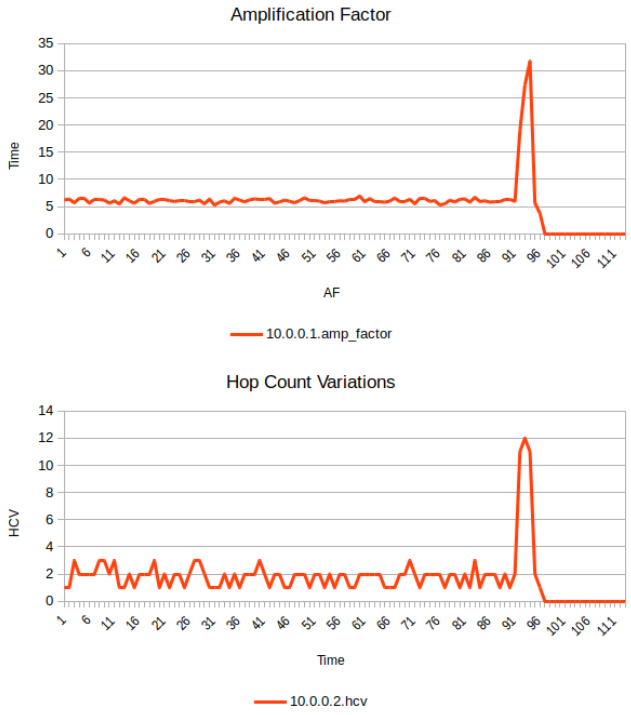

Fig. 7. DNS Amplification Traffic Metrics

References

[1] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 623–654, 2015.

[2] M. Howard. (2019) Ihs markit: Sdn deployed by 78% of global service providers at end of 2018. [Online]. Available: https://techblog.comsoc.org/2019/01/28/ihs-markit-sdn-deployed-by-78-of-global-service-providers-at-end-of-2018/

[3] L. Rudman and B. Irwin, "Characterization and analysis of ntp amplification based ddos attacks," in *2015 Information Security for South Africa (ISSA)*. IEEE, 2015, pp. 1–5.

[4] N. G. Dharma, M. F. Muthohar, J. A. Prayuda, K. Priagung, and D. Choi, "Time-based ddos detection and mitigation for sdn controller," in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*. IEEE, 2015, pp. 550–553.

[5] K. Kalkan, L. Altay, G. Gür, and F. Alagöz, "Jess: Joint entropy-based ddos defense scheme in sdn," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2358–2372, 2018.

[6] K. Kalkan, G. Gür, and F. Alagöz, "Sdnscore: A statistical defense mechanism against ddos attacks in sdn environment," in *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2017, pp. 669–675.

[7] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed ddos detection mechanism in software-defined networking," in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1. IEEE, 2015, pp. 310–317.

[8] L. Li, J. Zhou, and N. Xiao, "Ddos attack detection algorithms based on entropy computing," in *International Conference on Information and Communications Security*. Springer, 2007, pp. 452–466.

[9] K. Kumar, R. Joshi, and K. Singh, "A distributed approach using entropy to detect ddos attacks in isp domain," in *2007 International Conference on Signal Processing, Communications and Networking*. IEEE, 2007, pp. 331–337.

[10] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: an effective defense against spoofed ddos traffic," in *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 2003, pp. 30–41.

[11] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed ip traffic using hop-count filtering," *IEEE/ACM Transactions on Networking (ToN)*, vol. 15, no. 1, pp. 40–53, 2007.

[12] C.-C. Chen, Y.-R. Chen, W.-C. Lu, S.-C. Tsai, and M.-C. Yang, "Detecting amplification attacks with software defined networking," in *2017 IEEE conference on dependable and secure computing*. IEEE, 2017, pp. 195–201.

[13] A. A. Aizuddin, M. Atan, M. Norulazmi, M. M. Noor, S. Akimi, and Z. Abidin, "Dns amplification attack detection and mitigation via sflow with security-centric sdn," in *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*. ACM, 2017, p. 3.

[14] D. Huistra, "Detecting reflection attacks in dns flows," in *19th Twente Student Conference on IT*, 2013.

[15] J. Zheng, Q. Li, G. Gu, J. Cao, D. K. Yau, and J. Wu, "Realtime ddos defense using cots sdn switches via adaptive correlation analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 7, pp. 1838–1853, 2018.

[16] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, "A fair solution to dns amplification attacks," in *Second International Workshop on Digital Forensics and Incident Analysis (WDFIA 2007)*. IEEE, 2007, pp. 38–47.

[17] G. Yao, J. Bi, and P. Xiao, "Source address validation solution with openflow/nox architecture," in *2011 19Th IEEE international conference on network protocols*. IEEE, 2011, pp. 7–12.

[18] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: Flexible and elastic ddos defense," in *24th {USENIX} Security Symposium ({USENIX} Security 15)*, 2015, pp. 817–832.

[19] F. J. Ryba, M. Orlinski, M. Wählisch, C. Rossow, and T. C. Schmidt, "Amplification and drdos attack defense–a survey and new perspectives," *arXiv preprint arXiv:1505.07892*, 2015.

[20] M. Anagnostopoulos, G. Kambourakis, S. Gritzalis, and D. K. Yau, "Never say never: Authoritative tld nameserver-powered dns amplification," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–9.

[21] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Fingerprinting internet dns amplification ddos activities," in *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2014, pp. 1–5.

[22] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, "Dnssec and its potential for ddos attacks: a comprehensive measurement study," in *Proceedings of the 2014 Conference on Internet Measurement Conference*. ACM, 2014, pp. 449–460.

[23] R. Yamada and S. Goto, "Using abnormal ttl values to detect malicious ip packets," *Proceedings of the Asia-Pacific Advanced Network*, vol. 34, pp. 27–34, 2013.

[24] Q. Scheitle, O. Gasser, P. Emmerich, and G. Carle, "Carrier-grade anomaly detection using time-to-live header information," *arXiv preprint arXiv:1606.07613*, 2016.

[25] D. US-CERT, "Amplification attacks–alert (ta13-088a)."

[26] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, and C. Rossow, "Amppot: Monitoring and defending against amplification ddos attacks," in *International Symposium on Recent Advances in Intrusion Detection*. Springer, 2015, pp. 615–636.

[27] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras, "Booters—an analysis of ddos-as-a-service attacks," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 243–251.