Timing is Almost Everything: Realistic Evaluation of the Very Short Intermittent DDoS Attacks

Jeman Park
University of Central Florida
parkjeman@knights.ucf.edu

DaeHun Nyang
InHa University
nyang@inha.ac.kr

Aziz Mohaisen
University of Central Florida
mohaisen@cs.ucf.edu

***Abstract*— Distributed Denial-of-Service (DDoS) is a big threat to the security and stability of Internet-based services today. Among the recent advanced application-layer DDoS attacks, the Very Short Intermittent DDoS (VSI-DDoS) is the attack, which can bypass existing detection systems and significantly degrade the QoS experienced by users of web services. However, in order for the VSI-DDoS attack to work effectively, bots participating in the attack should be tightly synchronized, an assumption that is difficult to be met in reality. In this paper, we conducted a quantitative analysis to understand how a minimal deviation from perfect synchronization in botnets affects the performance and effectiveness of the VSI-DDoS attack. We found that VSI-DDoS became substantially less effective. That is, it lost 85.7% in terms of effectiveness under about 90ms synchronization inaccuracy, which is a very small inaccuracy under normal network conditions.**

***Index Terms*—DDoS, time synchronization, evaluation**

I. Introduction

The Internet is one of the pillars of our modern society, offering many conveniences through connectivity and creating a large and multifaceted ecosystem. The security of the Internet, however, has been challenged over the years by various threats, affecting its stability and the various applications relying on it. Notably, the Distributed Denial-of-Service (DDoS) attack, a well-known attack for several decades, has re-emerged recently as one of the most challenging threats [1], [2].

Up until recently, DDoS attacks have been mainly performed by sending a large number of packets to target servers to deplete their resources, whether they are bandwidth or memory. An enormous number of packets generated using a set of infected machines in a botnet saturate the available resources of the targeted server instantly [3], [4], making it unavailable to legitimate users and causing a denial of service [5], [6]. As DDoS attacks have become more sophisticated, utilizing new attack vectors, exhibiting new characteristics, and increasing in size and frequency over time [7], defending against them has become a priority, resulting in multiple defenses from the academic and industrial communities. For example, link-saturation DDoS attacks have been addressed by various defenses, including blacklisting, filtering, early detection, and utilizing resource mobilization through proactive analysis of adversarial capabilities for efficient attack containment.

In parallel with the progress made in traditional DDoS attack defenses, new types of low-rate attacks, which deviated from the high volume-based attacks, have emerged [8], [9], [10]. Among the many studies on the low-rate DDoS attacks, Shan *et al.* [11] recently introduced the Very Short Intermittent (VSI) DDoS attack, which is difficult to detect in existing systems but can significantly degrade the Quality of Service (QoS) that legitimate users experience. Unlike the traditional flooding-based DDoS attacks which aim at making the service itself impossible to reach through exhaustion of server resources for a long period of time (e.g., a few seconds, minutes, or even hours), the VSI-DDoS attack temporarily saturates the server with packets concentrated in a short period of time (a few milliseconds) and forces the server to respond to legitimate users' requests with very long delay.

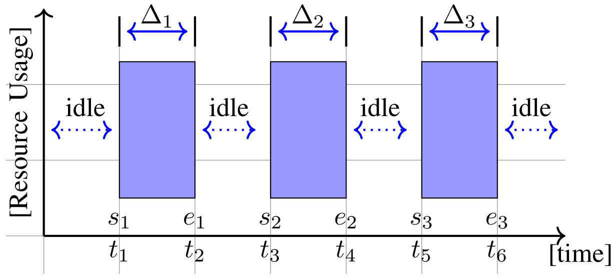


Fig. 1: An illustration of the VSI-DDoS attack cycles, where a large number of packets is sent over a short period of time (e.g., $e_1 - s_1$) resulting in high attack intensity. In our evaluation, the intensity Δ, is characterized by the degree of concentration of HTTP requests from bots, where each chunk is 100 HTTP requests.

For example, as shown in Fig. 1, the adversary in the VSI-DDoS attack proceeds by sending a large number of packets over a very short period of time; e.g., $t_2 - t_1$, $t_4 - t_3$, etc. (measured in milliseconds) upon which the adversary goes in an idle state. Even a reasonably small number of packets, e.g., $p = 1000$ packets, in a VSI-DDoS cycle (bounded by a start s_i and an end e_i in Fig. 1; e.g., 10ms) results in a large attack intensity. For the above example, for instance, and assuming a packet size p_s of 12,000 bit (1500 Byte, the size of typical TCP packet on the Internet), the VSI-DDoS attack would result in an instantaneous intensity Δ of about 1.2Gbps (i.e., $\Delta_i = (p/|e_i - s_i|) \times (1000 \times p_s)$ bps). This large intensity overwhelms the server for a short period of time, and thus increases the latency when responding to legitimate users.

978-1-5386-7493-2/18/$31.00 ©2018 IEEE

The adversary repeats the attack after the system recovers from the “shell-shock” effect of the VSI-DDoS cycle.

To evaluate the impact of the VSI-DDoS attack, Shan *et al.* experimented with the RUBBoS benchmark [12] to show how concentrated HTTP requests over a short period of time can degrade servers’ performance. In doing so, they assumed that all packets originate from fully synchronized bots and arrive at the server almost concurrently (within a 50ms time window). The key premise of the attack success, and upon which the evaluation is conducted, is that the bots used for launching the VSI-DDoS attack are tightly synchronized.

In this paper, we revisit the key premise of the VSI-DDoS attack in the wild. In particular, we are interested in assessing the impact of the VSI-DDoS attack when evaluated under more realistic assumptions of bots’ synchronization. The key motivation for this assessment is that in distributed systems in general, and in botnets (as a special case of distributed systems) in particular, time-synchronization is a non-trivial task. In other words, when multiple machines gather together to form a botnet, it can be practically difficult for any traffic, such as HTTP requests, originating from all bots to arrive at a given server in a short period time (a few milliseconds).

Even if a high-level synchronization technique, such as the Network Time Protocol [13], is applied, a time difference of several tens or even several hundreds of milliseconds may still occur due to the stochastic nature of the network condition, affecting latency. As such, we measure the VSI-DDoS attack in a loosely-synchronized environment of the botnet to understand its real, potential, and possible difficulty in deploying it. It can be intuitively inferred that the weak synchronization of the botnet will reduce the damage of the VSI-DDoS. However, this study is meaningful in that it empirically measured the impact of the VSI-DDoS attack on the target server under the various levels of synchronization. Through the analysis, we also can quantitatively evaluate the potential risk of the VSI-DDoS to the Internet. Furthermore, considering that the VSI-DDoS is a type of low-rate DDoS attack, this study shows why the synchronization between the bots is a prerequisite for the successful low-rate DDoS attack.

Organization. In section II, we introduce the concept of the VSI-DDoS attack and its effect presented in the previous work. In section III, we elaborate the system setup and experimental scenarios for the evaluation. In section IV, we present the results of experiments that reflect actual scenarios. In section V, we discuss the options that an attacker can choose for VSI-DDoS attacks. In section VI, we introduce related works about low-rate DDoS attack. In section VII, we provide a summary from this work.

II. VSI-DDoS Attack: Overview and Assumptions

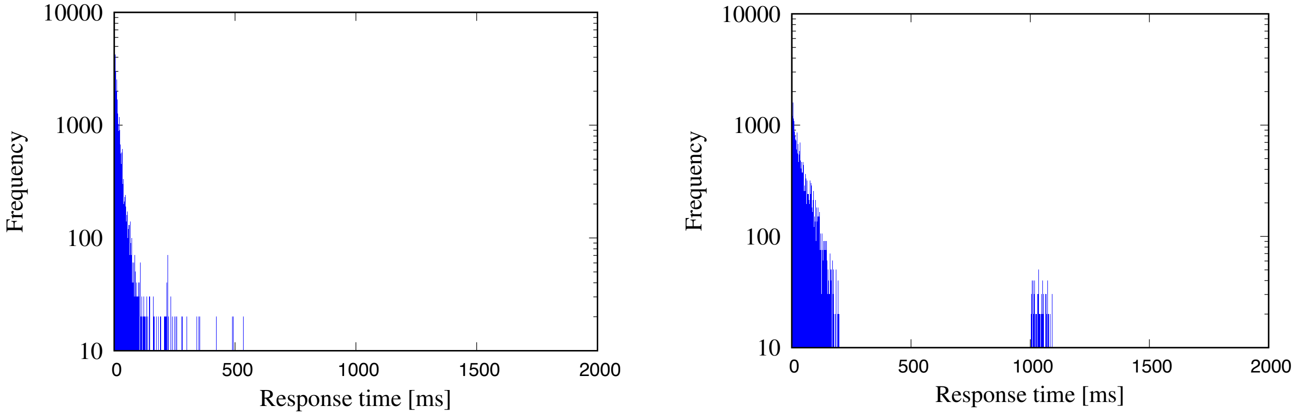
A. *VSI-DDoS: An Overview*

Latency as a Target. In web application services in general, the QoS is considered an important notion that measures the overall performance of the system through various easily to interpret measures [14]. Users’ experience when using Internet services, which is affected by QoS measures, has a significant impact on their behavior. For example, service providers’ revenue can be severely affected by the QoS that users experience when using web services [15]. As such, many web service providers, including Google and Amazon, are making a lot of efforts to reduce tail latency to a level that does not inconvenience users. This means, in other words, that simply causing a degradation of QoS without causing the attack to completely disable the service can lead to serious damage. By making the response time to be delayed, the user may feel fatigued in using the service, thereby preventing the target service from being used.

VSI-DDoS Objective and Operation. The VSI-DDoS attack, as described in section I, is a new type of application-layer low-volume DDoS attack aiming at the quality of service of web services. By causing very short bottlenecks (VSBs) on web application servers, the VSI-DDoS attack degrades the legitimate users’ QoS [11]. Unlike the traditional DDoS attacks which exhaust server resources for a long period of time, the VSI-DDoS attack causes the transient saturation of resources and the delays to response to legitimate user’s request. For example, when a number of HTTP requests suddenly concentrate within a few milliseconds and exceed the server’s queue limit, the legitimate user’s request is dropped, resulting in a very long response time (VLRT), as the TCP retransmission occurs. The occurrence of TCP retransmission remarkably aggravates the user experience, since the user cannot use the service until the response of the retransmitted request arrives after the TCP retransmission timeout of the dropped packet. Fig. 2 shows how the VSI-DDoS attack can adversely affect the user’s experience. In general, the most responses for HTTP requests are quickly returned to the users within 100ms (as in Fig. 2a). However, under the VSI-DDoS attack, some of the responses are significantly delayed more than one second due to the attack (as in Fig. 2b). As such, the VSI-DDoS attack can be a major threat to web application services, affecting their QoS.

Difficulty of Detection. Since the transient server resource exhaustion caused by the VSI-DDoS attack occurs mostly for a few milliseconds, it is difficult to detect using the current second-level monitoring systems such as *sar, vmstat*, and *top*. This is because those monitoring systems have a low frequency (i.e., the check the utilization of resources at a granularity of seconds compared to the millisecond operation realm of VSI-DDoS). As shown in Fig. 3, although the CPU usage of each server under the VSI-DDoS attack is slightly higher (as in Fig. 3b) than the case without the attack (as in Fig. 3a), however, it still does not to be saturated from the second-level monitoring system’s perspective. In other words, the VSI-DDoS attack operates in a way that is difficult to detect with the current DDoS detection mechanisms and may continue to impact the convenience of users of web application services without being detected. While there has not been a report of damages caused by low-rate DDoS, such as VSI-DDoS, many security agencies are warning about the danger of stealthy and sub-saturating attack [16].

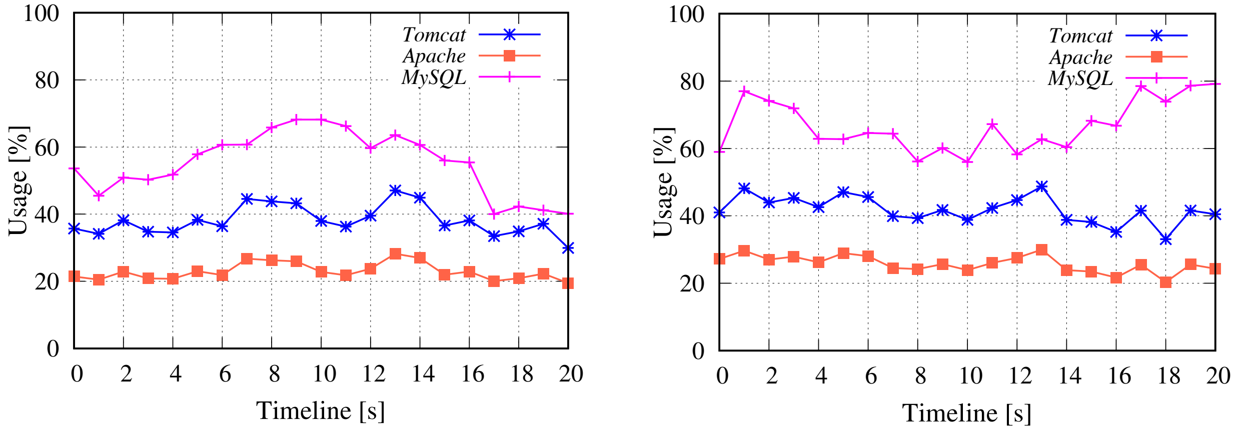
Results and Assumption. To validate the attack, Shan *et al.*



(a) Response time without the VSI-DDoS attack

(b) Response time under the VSI-DDoS attack

Fig. 2: The distribution of the HTTP response time experienced by the legitimate users. Under the VSI-DDoS attack, the some of the users may experience the very long response time (over than 1 second) for their requests due to the TCP retransmission (right), while the users can get the most responses within 100ms in general (left).



(a) CPU usage without the VSI-DDoS attack

(b) CPU usage under the VSI-DDoS attack

Fig. 3: The CPU usage of each server with/without the VSI-DDoS attack captured by *collectl* with the sampling rate of 1Hz. Notice that even under the VSI-DDoS attack, the CPU usage of each server is not saturated, which makes it difficult to detect.

conducted the experiments showing how varying parameters of the attack affect the response latency to legitimate users' requests. With multiple bots fully synchronized, they noted that tail latency is significantly increased when generated packets arrive at the server within a very short time window (a few milliseconds). By assuming an almost complete synchronization, they experimentally demonstrated that the 95th percentile response time of the target server increases by more than 1 second, which is the timeout of TCP retransmission, under the VSI-DDoS attack. In other words, they demonstrated that the VSI-DDoS attack can be a threat to many web application services, by degrading their QoS guarantees.

B. Synchronization in Botnet

Time synchronization in distributed systems is a challenging problem, and clock skew is, by default, an intrinsic feature of those systems. Particularly, given the strong synchronization assumption under which the VSI-DDoS attack works, it is unclear how violating this assumption, even slightly, would affect the performance of the attack. In this section, we revisit the assumption of the VSI-DDoS attack operation by highlighting botnets time synchronization in the wild, and challenges associated with their tight synchronization.

Botnet Time Synchronization. To understand the time synchronization of botnets in the wild, we use evidence from our prior measurement studies in [17], [18]. In this line of work, we demonstrated through measurements that the request time of domains registered using domain generation algorithms (DGAs) preceded their registration, resulting in non-existent domains (NXDomain) responses.

One of the ways for establishing a command and control in botnets is through DGAs, which are algorithms used for domain name registration by the botmaster, and taking the current time into account. When bots want to communicate with the command and control, they similarly execute the

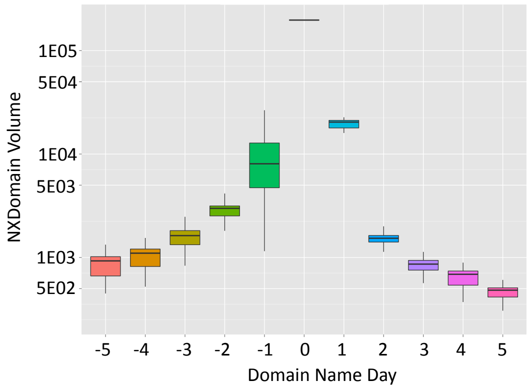


Fig. 4: The Conficker NXDomain DNS lookups over time, highlight the lack of synchronization between bots and botmaster, and across bots, in the wild.

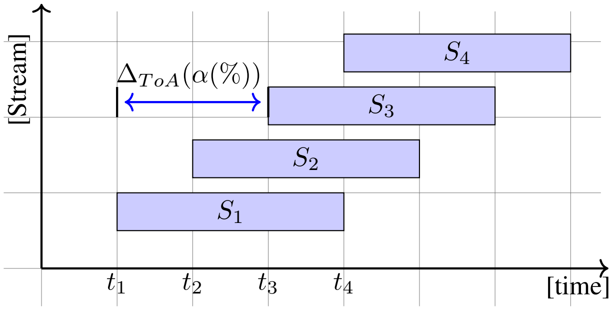


Fig. 5: An illustration of botnet synchronization in [19]. Notice that S_i means the stream from each bot i and t_i means the arrival time of the first packet of each stream. $\Delta_{ToA}(\alpha(\%))$ corresponds to the maximum time difference between the first packets of the majority ($\alpha\%$) of attack streams. For example, the expression, $\Delta_{ToA}(90\%) = 50ms$, means that the first packets of 90% of attack streams arrive within 50ms.

DGA and connect to the domain generated by issuing a regular DNS query. The key insight in using DGAs is that both the botmaster and bots are using the same clock, and are tightly synchronized. As a result, the botmaster would register the domain name using the DGA right at the time (or just before) any bot could query the domain for command and control.

Fig. 4 shows the number of NXDomain requests (plotted as a box plot), issued from /24 network addresses for different domains generated by the DGA for the Conficker botnet over time. The time, plotted on the x-axis characterizes the domain name day: zero day indicates the day of domain name registration (when generated using the time of the day), a negative day indicates that a query is being sent to the domain name generated by the DGA before its registration, and a positive day indicates that the query is sent to the domain name after its registration. As we can see in this figure and its associated analysis, 1) a large number of queries are issued for the domains before their registration, resulting in NXDomain responses, and indicating a clock skew between the botmaster and the several bots it controls, and 2) the first query issued by the multiple bots is not simultaneous, indicating a clock skew across the different bots. Such a time skew can at the granularity of days (omitted for the lack of space).

On the Difficulty of Synchronization. The results above demonstrate the chaotic nature of the botnet in the wild with respect to time synchronization. However, one may argue that using an off-the-shelf time synchronization, such as the Network Time Protocol (NTP) [13] may bring an order to bots, to meet the assumptions of the VSI-DDoS attack. However, in reality, such an approach would face various shortcomings. First, while NTP works nicely in local area networks, achieving up to a millisecond level of accuracy under ideal conditions, it provides worse performance (tens to hundreds of milliseconds) on the Internet due to congestion and path asymmetry. Second, connecting to NTP servers on the Internet, while common, can be used as an indicator to trace bots back and detect them, pronouncing the approach unfeasible.

Recently, Ke *et al.* [19] introduced *CICADAS* which performs the highly sophisticated botnet synchronization in 2016. To amplify the impact of pulsating attack, a kind of low-rate DDoS attack, they designed a synchronization technique in the decentralized system. To evaluate the level of synchronization, they also introduced the concept of the differential time of arrival, denoted by Δ_{ToA}. As shown in Fig. 5, $\Delta_{ToA}(\alpha) = t_\alpha$ means that the first packet of $\alpha(\%)$ of the attack streams arrive at the target within t_α. From the Internet-wide experiments, they got the result that $\Delta_{ToA}(75\%)$ and $\Delta_{ToA}(90\%)$ are about 40ms and 60ms, respectively.

Despite outstanding results, CICADAS still demonstrates that perfect synchronization among the bots is a very difficult goal to achieve. Per their approach, if there are four machines attacking the server with 50ms of burst, for example, and considering the worst-case scenario, the server is attacked from three (75%) machines at the same time for only 10ms, that is 50ms - 40ms which values correspond to burst length and $\Delta_{ToA}(75\%)$, respectively. Furthermore, the accuracy of their approach enforces a lower-bound on the burst time: if the burst ($e_i - s_i$ in Fig. 1) gets shorter, the time period that the packets from multiple sources concentrate on the target will be shortened also, and even may not exist. That is, although the HTTP requests sent from bots in the VSI-DDoS attack should be concentrated in a very short period of time, in fact, the packets arriving at the server may have a large time difference. Note that for the approach in to work, at least 25 seconds are required for achieving a synchronization of accuracy less than 40 ms, making the approach inappropriate for the time-scale of VSI-DDoS in the first place.

Even when existing approaches provide a perfect synchronization accuracy, network conditions are stochastic, where the interarrival time between packets sent from different hosts to the same server may vary depending on the network condition and paths those packets traverse. Such variance in time makes it very difficult to ensure a full synchronization, alluding to less effectiveness of VSI-DDoS. For this reason, we pose the following question: under realistic conditions, with worse

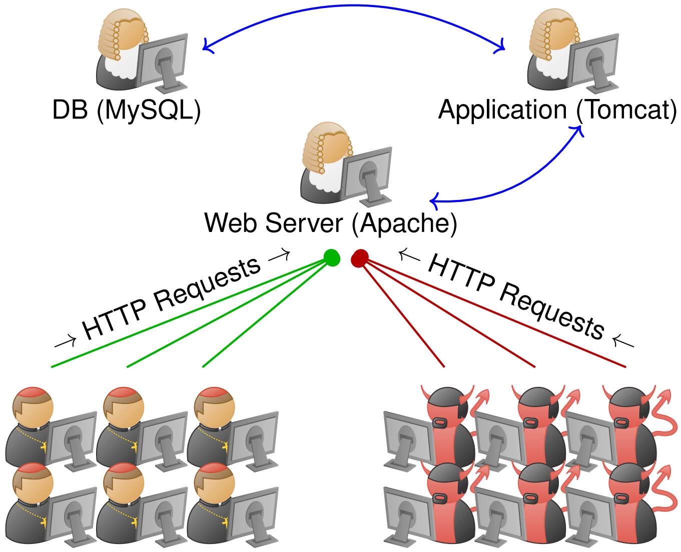


Fig. 6: The overview of the communication among RUBBoS benchwork (consisting of RUBBoS 3-tier architecture; front-end web server, back-end DB server, and application server), legitimate users, and botnet.

synchronization inaccuracy, how effective is the VSI-DDoS attack? We answer this question in the rest of this paper.

III. Experimental Setting

To measure the impact of a more realistic time synchronization in botnets, taking a slight deviation from ideal scenarios into account, we build a testbed similar to the one in [11]. While it is difficult to match the exact specifications of the hardware used in their experiments (partly due to the lack of details), in our work we replicated the behavior of servers, legitimate users, and bots, which are more crucial to the reproducibility of their results. Particularly, because the work at hand is mainly concerned with understanding the effect of the VSI-DDoS attack depending on the degree of synchronization, we focus on how the change in synchronization level, rather than individual values, affects the performance of the VSI-DDoS attack. As a result, in the rest of this work, and given that other parameters (e.g., packet size) are the same in all of our experiments, we use the burst time (the difference between start and end of a VSI-DDoS attack cycle) to characterize the attack intensity. We note that given the main goal is to compare the relative latency for different burst times, the exact matching of the hardware specifications as in [11] is unnecessary.

A. System Setup

As shown in Fig. 6, our evaluation system consists of the RUBBoS 3-tier architecture for the server-side setup (consisting of a web server, an application server, and a DB server), bots, legitimate users. In the following, we elaborate on each of those system components.

Web Application Server. As shown in Fig. 6, we build a server system that provides web services using RUBBoS, a popular n-tier web application benchmark [12]. Following the typical 3-tier architecture, an Apache web server, a Tomcat application server, and a MySQL database server were deployed on the Vultr cloud [20]. Each server is created as an independent instance with the same specification of a 2.4GHz single core virtual CPU and a 1.8GB of Random Access Memory (RAM). The servers interconnect using 1Gbps links.

Legitimate users. The behavior of the legitimate user is imitated using the workload generator of RUBBoS. Three cloud instances mimic the behavior of a total of 1,050 legitimate users, each for 350 users. According to the configuration of RUBBoS, the users surf web pages following a Markov chain model. Behaviors such as searching the bulletin board, writing a new article, leaving a comment, etc. are continuously generated following the probability model of Markov process.

Bot for the VSI-DDoS Attack. Apache Bench (AB) [21] is used to create a bot that performs the VSI-DDoS attack. The purpose of bots is to intermittently send the given number of HTTP requests to the server and to trigger transient saturation, thereby delaying responses by the server to legitimate users.

In the default setting, AB does not use keep-alive, which means that a new TCP connection is created for each HTTP request. Therefore, every HTTP request behaves like a different user and tries to connect to the server to use the service.

In addition, we made the botnet operate on the single machine rather than be deployed over the multiple cloud instances. In this research, since it is not our focus to implement the sophisticated synchronization of botnet itself, we implemented the bot on a single device so that the degree of synchronization can be accurately adjusted by simply modifying the configuration. The average of the round trip time (RTT) between the bot instance and the server is about 0.5 ms (measured using *ping*), which means that the network latency in one-way packet transmission is about 0.25 ms. This latency is sufficiently small. We also measure the actual concentration of botnet packets with the latency through preliminary experiments in section III-C.

B. Intensity of Bot's Attack

In our experiments, the concentration of HTTP requests made by bots is set as a variable parameter, incorporating a variable accuracy in time synchronization between the originating bots. As shown in Figure 1, the VSI-DDoS attack is made in such a way that HTTP bursts are sent from bots to the server intermittently (i.e., within a very short period of time separating the sending of the first packet from the bot and the arrival of the last packet to the server). In the figure, each chunk (blue rectangle) means the given number of HTTP requests (e.g., 100 HTTP requests in our experiments) is sent from the bot periodically (resulting in intensity Δ_i, as described in section I), with an ideal time between every two consecutive cycles (e.g., 2 seconds in our experiments).

To express the concentration of packets from the bot, in this paper, the time difference between the first request and the last request in the same chunk is denoted as *intensity* for convenience. For example, if the intensity is 45ms, it means that all 100 HTTP requests are sent to the server within 45ms.

A high intensity means that the attack is concentrated within a shorter period of time; i.e., the bots are well synchronized.

C. Generating Intensity Values

As described in section III-A, we used AB for launching the VSI-DDoS attack. AB supports sending the given number of HTTP requests in a short time to a URL to check the performance of the web server hosting the given URL. As such, we can control this process by feeding the number of total packets sent from the bot to the server, the concurrent level of transmission, and time limit, all as options [21]. However, AB does not fully respond to the input variables due to the issues such as the limited network resource. In our study, in order to analyze the correlation between the degree of synchronization and the impact of the VSI-DDoS attack, the packet transmission from bots is key variable we need to accurately control. Thus, we conducted a preliminary experiment to figure out the behavior of AB in detail.

While running the AB and changing the input option value of the concurrent level, we captured the actual transmission time of 100 HTTP packets using `tcpdump`. Splitting the entire transmission into multiple tasks (e.g., two AB instances where each sends 50 requests) using shell script is also used to generate various intensity levels. We conducted the measurements with six different settings, and each setting used in the measurement is shown in Table I. The burst length was calculated by measuring the time of the packets captured by `tcpdump` with each setting. In order to minimize the impact of the external network environment, the web server and the bot are configured to be instances located in the same locale in the Vultr (perhaps locally connected).

In the experiment of sending 200 chunks (each chunk of 100 HTTP requests), the burst length ($e_i - s_i$) has the distribution as in the Fig. 7. In the figure, Δ_S (blue) is the length of the burst measured at the sender, a bot, and Δ_R (yellow) is the burst length measured at the receiver, the web server. Each box shows the distribution from the upper 25% to the lower 25% of the burst length, while the black line in the box means the average value of all results (200 chunks). Two horizontal lines above and below the box represent the maximum and minimum values, respectively. In the figure, we can see that the packet distribution in the sender and the receiver is similar under the strictly controlled environment. The average burst lengths with each setting measured at the bot were 11.2ms, 45.64ms, 63.37ms, 79.05ms, 88.84ms, and 129.53ms, while the average values were 11.27ms, 45.51ms, 63.53ms, 79.15ms, 88.93ms, and 130.01ms at the web server. The maximum value of the standard deviation of all distributions was 4.22 (in the receiver with the setting S_2), the other values are less than 4, which means that the packets from the bot arrive at the server in a similar pattern under the same configuration. In the rest of the paper, for convenience, the intensity values with each setting are denoted by i_1, i_2, i_3, i_4, i_5, and i_6, respectively.

To understand the result where the degree of synchronization is more realistic, we compared the response time experienced by legitimate users for each case with the intensity value as shown above. Based on previous studies on synchronization, among these values, i_1 (about 10ms) can be considered a very difficult level of synchronization to be achieved by the attacker (e.g., impossible to achieve by NTP [13] and state-of-the-art botnet synchronization [19]), i_2 (about 45ms) and i_3 (about 65ms) can be considered difficult levels, i_4 (about 80ms) and i_5 (about 90ms) can be considered achievable levels, and i_6 (about 130ms) and above can be considered more realistic levels even with the network dynamics.

TABLE I: The various settings used in the preliminary experiment. S_{id} corresponds to the index of each setting, o_c corresponds to the input option of concurrency parameter (option -c in AB), $\#_{AB}$ corresponds to the number of AB instances, and $\#_{req}$ corresponds to the number of HTTP requests that each AB instance sends to the server.

S_{id}	o_c	$\#_{AB}$	$\#_{req}$
S_1	100	1	100
S_2	50	2	50
S_3	25	4	25
S_4	20	5	20
S_5	10	10	10
S_6	5	20	5

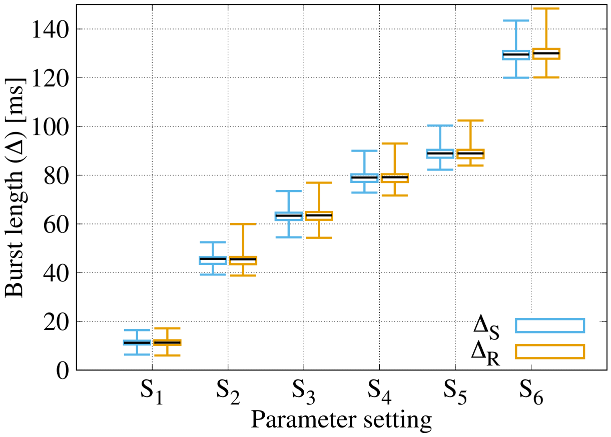


Fig. 7: Measurements of actual burst length with different setting; 200 repeated measurements for each setting. Δ_S and Δ_R correspond to the measured burst lengths from the sender (bot) and the receiver (web server), respectively. The box represents the distribution from the upper quartile to the lower quartile, and the black bar represents the mean value.

IV. Measurement and Results

A. Response Latency Measurement

In order to investigate the effect of the VSI-DDoS attack on the QoS, we performed each simulation for the same time (50 seconds) in all scenarios with different intensity values. Ten experiments were repeated (totally 500 seconds) for each setting to ensure a sufficient amount of data. During each simulation, we measured the response time at the legitimate users from the web service and counted the number of packets

TABLE II: Statistics of HTTP requests and responses; $e_i - s_i$ corresponds to the intensity of requests, c_1 corresponds to the total number of HTTP request-response pairs between all legitimate users and the server, and c_2 corresponds to the number of HTTP responses with over 1 second time.

$e_i - s_i$	c_1	c_2
i_1	69,090	9,643
i_2	71,530	6,080
i_3	73,006	2,644
i_4	72,744	1,655
i_5	73,574	1,463
i_6	73,049	1,097

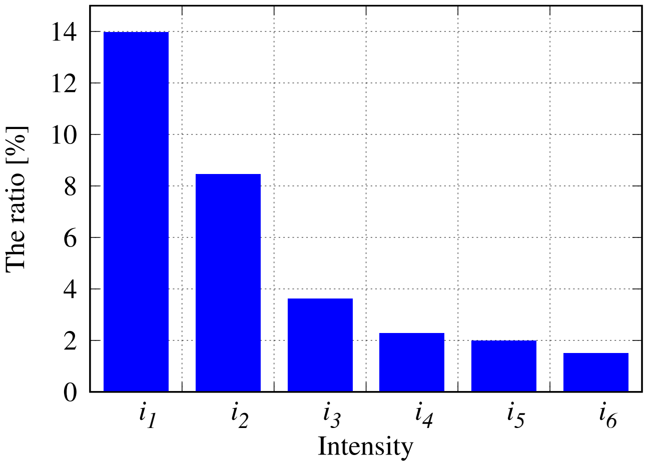


Fig. 8: The percentage of legitimate users response with time over 1 second out of the total HTTP responses; an indicator of QoS degradation due to the VSI-DDoS attack. Their decay with time shows ineffectiveness of the attack.

with response time longer than 1 second, which indicates the higher latency due to the VSI-DDoS attack. Table II represents the total number of HTTP request-response pairs between the web server and all legitimate users.

Because the simulation was conducted for the same time period, and as shown in the table, the total number of HTTP request-response pairs (c_1) remained similar except for the cases with the i_1 and i_2 that have tight intensities. The reason for the two relatively small values can be found in the delayed answer that occurs under the VSI-DDoS attack. As in the table, both i_1 and i_2 intensities lead to a larger number of delayed responses (c_2). Therefore, in the case of an HTTP request in which the response is not returned to the legitimate client during the experiment period of 50 seconds due to the delay caused by the VSI-DDoS attack, it is not included in the c_1 value shown in the table. In other words, low c_1 values with the i_1 and i_2 intensities are the result of frequent latency occurred by tightly synchronized attack not reflected in statistics.

Fig. 8 shows the degradation of QoS provided to the legitimate user as a latency guarantee when the intensity of the bot attack is changed. In particular, the figure plots $c_2/c_1 \times 100$ for the different intensity values in the table. When the HTTP request intensity of the bot is i_1, about 14.0% of legitimate users' responses received from the server exceed 1 second, which corresponds to an ideal setup of the VSI-DDoS attack. As the intensity of the attack gradually loosened, the effect of the VSI-DDoS attack quickly decreased.

This is, the ratios of HTTP requests that have a response time over 1 second are 8.4% for the intensity of i_2,3.6% for i_3, 2.3% for i_4, 1.9% for i_5 and 1.5% for i_6. This shows that if the intensity of HTTP requests sent from the bot change slight, to reflect loosely synchronized bots, it is difficult for the attacker to obtain the desired result. For example, when the intensity is i_5, the legitimate user's VLRT rate (over 1 second) is only about half the rate at i_3, which means that the attacker should use about more number of machines to get the desired effect on the server.

B. Network Statistics

Fig. 9 shows the number of incoming/outgoing packets per second (pps) captured at the database server by *collectl* under VSI-DDoS with the intensities of i_1, i_3, and i_6. Among ten iterations with each setting, we carefully selected one experiment that has the most similar total number of packets for fair comparison and compared the number of packets.

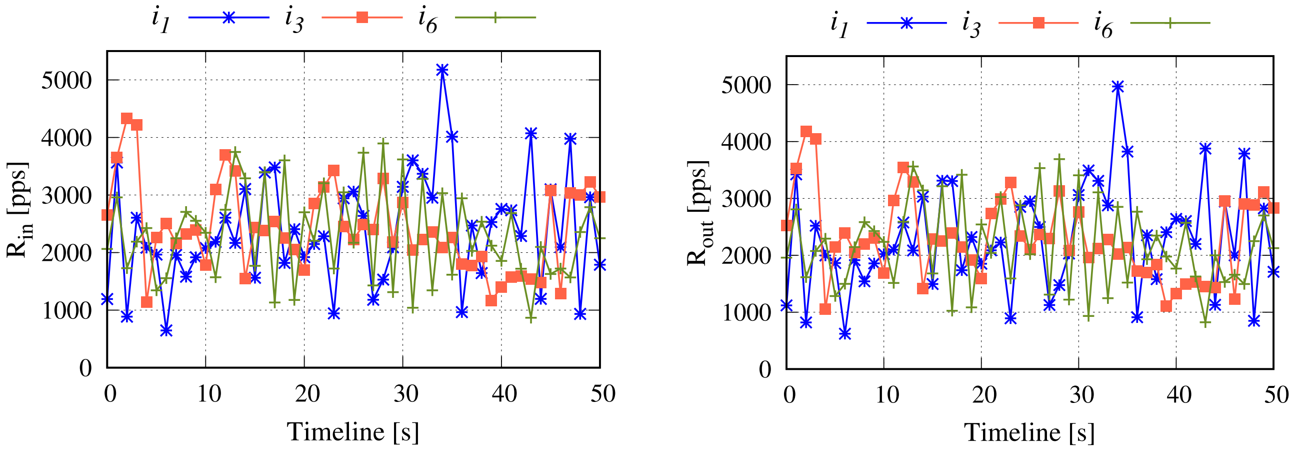
As shown in the figure, most of both incoming and outgoing rates are between from 2,000 to 3,000 pps. As shown in III, the average rates of the whole experiment were 2,386 pps, 2,425 pps, and 2,295 pps, respectively, in each experiment using i_1, i_3, and i_6 intensity values, and that average values are similar one another. The difference between the largest and the smallest of the average values is about 5%, so this difference seems not large.

However, it can be seen that there is a big difference in standard deviation for each intensity. The difference between the smallest standard deviation, 752.2 (δ_{in} in i_6), and the biggest value, 955.6 (δ_{in} in i_1), is 203.4 which is about 20% of the biggest one. In addition, we also can find a large difference in standard deviations in the case of outgoing packets (δ_{out} of i_1 and i_6) which is about 20% as well.

The difference in these standard deviations can be clearly seen in Fig. 9. In the case of i_1 in both Fig. 9a and Fig. 9b, the fluctuation is more intense every second. Given that there are only 100 HTTP requests sent from the bot, it is difficult to justify that those fluctuations are the influence of HTTP requests itself generated by the botnet every two seconds. Rather, this can be interpreted as a result of the queue drop caused by the saturation under the VSI-DDoS attack.

C. HTTP Response Time Distribution

While the results above demonstrate the key insight of our experiments, Fig. 10 shows the HTTP response time that the legitimate users received from the server. In the graph, the x-axis represents HTTP response time (ms), and the y-axis represents the number of packets having the corresponding time. Because the total number of HTTP request-response pairs is different in each case, for an accurate comparison, the



(a) The rate of incoming packets.

(b) The rate of outgoing packets.

Fig. 9: The number of incoming/outgoing packets under the VSI-DDoS attack with intensity values of i_1, i_3, and i_6 at the database server captured by *collectl* with the sampling rate of 1Hz. Notice that R_{in} and R_{out} correspond to the rate of incoming and outgoing packets, respectively.

TABLE III: Analysis of the incoming/outgoing packets at the database server under VSI-DDoS with intensities i_1, i_3, and i_6. Notice that μ_{in}, δ_{in}, and $\#_{in}$ correspond to the average rate (pps), the standard deviation, and the total number of incoming packets, respectively. μ_{out}, δ_{out}, and $\#_{out}$ correspond to the notions of outgoing packet in the same order.

i	μ_{in}	δ_{in}	$\#_{in}$	μ_{out}	δ_{out}	$\#_{out}$
i_1	2,386	955.6	121,672	2,293	922.8	116,966
i_3	2,425	793.5	123,660	2,314	757.3	118,020
i_6	2,295	752.2	117,046	2,169	730.9	110,606

graph was created by normalizing the ratio of observed results in the experiment to a total of 100,000 HTTP packets. As shown in the graph, in all scenarios the distribution increases from 1 second of response time due to TCP retransmission under the VSI-DDoS attack. Considering that all graphs are plotted on a log scale, the actual amount of change will vary highly depending on the intensity of the bot's attack.

From the same figure, we notice that when the response time is equal to or greater than 1,000 ms, the number of HTTP request-response pairs increases more sharply as the intensity is higher (e.g., i_1, and i_2). The degradation of QoS with the various values of intensity can be seen more clearly in Fig. 11. In the CDF representation of the response time experienced by legitimate users, the degradation of QoS caused with each intensity is significantly distinguished.

This means that HTTP requests that are concentrated on the server within a short period of time saturate the server more frequently and cause more frequent response delay to legitimate users. Conversely, as packets from the bot arrive over a wider interval of time, the actual effect of the VSI-DDoS attack is weakened.

VSI-DDoS in the Wild. Based on the results of the experiments, we conclude that the VSI-DDoS attack is less effective even under moderate synchronization imperfections. Those moderate imperfections do not represent reality by any means. For example, where the clock skew between different bots could be at the scale of hours or days, as shown in Fig. 4, the minor lack of synchronization we examined the VSI-DDoS attack under could be achieved only using sophisticated state-of-the-art synchronization approaches.

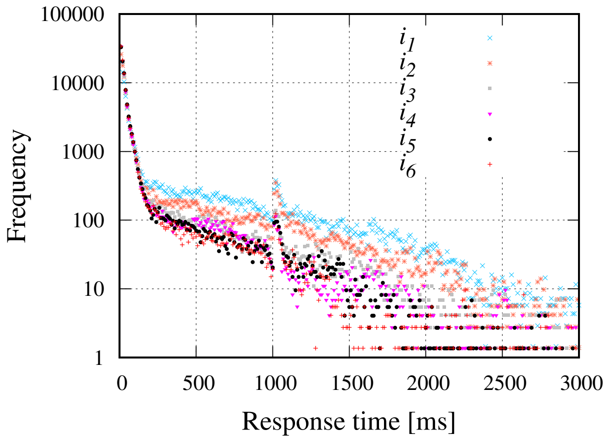


Fig. 10: The distribution of response times with different VSI-DDoS attack intensity values. Notice that the number of packets with the delay bigger than 1 second significantly decreases as bots become less synchronized.

V. Discussion

Our findings have shown it is difficult to obtain sufficient performance of the VSI-DDoS attack in an environment where synchronization between bots is imperfect. To this end, we envision various approaches to cope with this limitation:

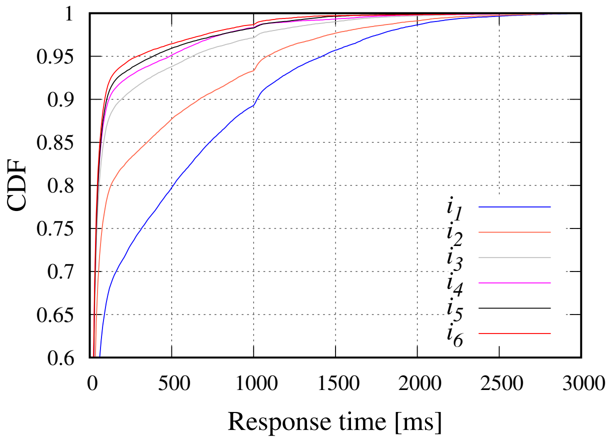


Fig. 11: The CDF of response times under VSI-DDoS attack. Notice that the response time of less than 1 second is about 87% with the intensity of i_1 but 98% with i_6, which means that high concentration of packets makes QoS worse.

A. Improve Bots' Synchronization

Improving the bots' synchronization is perhaps the most obvious option. The better synchronization can be achieved by applying highly advanced techniques, and by placing machines in geographically similar spaces to minimize time differences. However, considering that VSI-DDoS attack should concentrate HTTP requests in a very short period of time, and the limitations of the state-of-the-art time synchronization approaches that go beyond the desired timescale for VSI-DDoS, this approach has very little potential in addressing the problem at hand. In the section VI-B, we highlight the modern approach to achieve the sophisticated level of synchronization.

B. Increase HTTP Requests from Each Bot

The second approach to address the problem is to increase the HTTP requests generated by each bot, which would result in higher attack intensity. Since the packets generated on one machine are likely to arrive at the server almost at the same time, they can be effective regardless of the synchronization of the botnet. However, when the number of HTTP requests per bot increases, repetitive sending of similar packets can be easily detected by the current Intrusion Detection System (IDS) as an anomalous activity.

C. Increase The Number of Bots Participating in Botnet

The third option the adversary can choose to launch the VSI-DDoS attack is increasing the number of bots participating in the botnet. As more bots participate in and more attack streams are created, more packets can arrive at the server concurrently under the same synchronization level. While this approach can circumvent the anomaly detection, it would lead to two side effects: 1) increasing the cost of the attack, and 2) making it more difficult to apply any (loose) time synchronization among a large number of bots.

VI. Related Work

A. Low Rate DDoS Attack

Related works to the VSI-DDoS attack are the low-rate DDoS attacks, including both on application and network layer. Kuzmanovic and Knightly [22], [23] proposed the Shrew attack, a low-rate TCP attack that exploits the TCP retransmission time-out for DDoS. The pulsing attack was proposed by Luo and Chang [24], exploiting the TCP congestion control window and TCP timeout. Gourgouis *et al.* proposed Reduction of Quality (RoQ) attack, which operates in a similar manner to QoS reduction attacks by attacking Internet resources [25], end-systems [26], and dynamic load balancers [27]. Zhang *et al.* [28] pointed low-rate DDoS attacks on Internet routing mechanism, while Luo *et al.* [29] proposed models for estimating the impact of TCP-targeted low-rate attack. Maciá-Fernández *et al.* [30], [31] proposed low-rate DoS Attack against application servers (LoRDAS) which forecasts the free position in service queue at application server and sends traffic to drop legitimate user's requests. Jung *et al.* [32] presented a DDoS mimicking flash crowds.

B. Synchronization

NTP is the most widely used method for time synchronization [33], [13]. Since then, although many studies have shown that the accuracy of NTP steadily improves as the network develops [34], [35], [36], [37], however, tens or hundreds of milliseconds error in time synchronization still remains. PTP (Precision Time Protocol) is another approach for the systems that require the highly precise time synchronization such as power system [38], [39]. Although PTP ensures the nanoseconds-level synchronization, but it does not directly mean that current botnet can employ this advanced technique due to the dependency for supporting hardware. Similarly, Datacenter Time Protocol (DTP) with the sub-microsecond accuracy also have the hardware dependency, which makes it only for the specific purpose (synchronization in the datacenter), but not for personal devices that make up most of botnet [40]. Even if we assume that the above techniques ensure the tight synchronization (within a few milliseconds) of system time, it does not ensure the synchronization of the arrival time from distributed bots due to the continuous change in network condition [41].

C. Botnet Detection

A lot of studies have been conducted on various technologies for detecting botnet. Gu *et al.* [42] proposed BotSniffer, the system for detecting centralized botnet by identifying Internet Relay Chat (IRC) or Hypertext Transfer Protocol (HTTP) based command & control (C&C) channel. The BotSniffer does not need a large number of bots, and they can even detect a single member botnet. Choi *et al.* [43] introduced the botnet detection system called BotGAD, which focuses on group activities, not the traffic contents nor the signature.

Zhao *et al.* [44] focused on detecting Peer-to-Peer (P2P) botnets by identifying network traffic behavior using machine learning technique. Hang *et al.* [45] proposed Entelechesia, an approach for detecting decentralized botnet by analyzing the social behavior of bots.

VII. Conclusion

In this paper, we analyzed the impact of loose bots synchronization on the impact of VSI-DDoS, an advanced application-layer DDoS attack that is capable of bypassing existing defenses. Through the preliminary experiment, we fixed the six degrees of synchronization of the botnet and apply it to quantitatively evaluate the effect of the actual vsi attack. As a result, we demonstrate that even a moderate imperfections in bots time synchronization would degrade the impact of the VSI-DDoS attack, and pronounce it ineffective. Specifically, the effect of the VSI-DDoS attack at realistic synchronization level seems to make adversary's goal difficult to be achieved. Mitigation to our main finding is possible, although it can creates a clear trade-off between the attack, its cost, and detection. In the future, we will theoretically and empirically explore analyzing the effort of the adversary to make the VSI-DDoS attack successful, including the minimum number of bots needed for such intensities in real world.

Acknowledgement. This work is supported in part by NSF grant CNS-1809000 and NRF grant 2016K1A1A2912757.

References

[1] A. Wang, A. Mohaisen, W. Chang, and S. Chen, "Delving into internet ddos attacks by botnets: Characterization and analysis," in *Proc. of IEEE DSN*, 2015.

[2] ——, "Capturing ddos attack dynamics behind the scenes," in *Proc. of DIMVA*, 2015.

[3] J. M. Smith and M. Schuchard, "Routing around congestion: Defeating ddos attacks and adverse network conditions via reactive bgp routing," in *Proc. of IEEE S&P*, 2018.

[4] M. Schuchard, A. Mohaisen, D. Foo Kune, N. Hopper, Y. Kim, and E. Y. Vasserman, "Losing control of the internet: using the data plane to attack the control plane," in *Proc. of NDSS*, 2011.

[5] M. S. Kang, V. D. Gligor, and V. Sekar, "SPIFFY: inducing cost-detectability tradeoffs for persistent link-flooding attacks," in *Proc. of NDSS*, 2016.

[6] M. S. Kang, S. B. Lee, and V. D. Gligor, "The crossfire attack," in *Proc. of IEEE S&P*, 2013.

[7] Kaspersky, "Kaspersky lab report on ddos attacks in q1 2017: The lull before the storm," https://bit.ly/2tDUWXC, 2017.

[8] J. Luo and X. Yang, "The newshrew attack: A new type of low-rate tcp-targeted dos attack," in *Proc. of IEEE ICC*, 2014.

[9] R. Rasti, M. Murthy, N. Weaver, and V. Paxson, "Temporal lensing and its application in pulsing denial-of-service attacks," in *Proc. of IEEE S&P*, 2015.

[10] A. Shevtekar and N. Ansari, "Is it congestion or a ddos attack?" *IEEE Communications Letters*, vol. 13, no. 7, 2009.

[11] H. Shan, Q. Wang, and Q. Yan, "Very short intermittent ddos attacks in an unsaturated system," in *Proc. of SecureComm*, 2017.

[12] RUBBoS, http://jmob.ow2.org/rubbos.html.

[13] D. L. Mills, "Internet time synchronization: the network time protocol," *IEEE Transactions on Communications*, vol. 39, no. 10, pp. 1482–1493, 1991.

[14] D. A. Menascé, "Qos issues in web services," *IEEE Internet Computing*, vol. 6, no. 6, pp. 72–75, 2002.

[15] R. Kohavi and R. Longbotham, "Online experiments: Lessons learned," *IEEE Computer*, vol. 40, no. 9, 2007.

[16] Corero, "Short, stealthy, sub-saturating ddos attacks pose greatest security threat to businesses," https://bit.ly/2N9ciEe, 2017.

[17] J. Spaulding, J. Park, J. Kim, and A. Mohaisen, "Proactive detection of algorithmically generated malicious domains," in *Proc. of ICOIN*, vol. 2018.

[18] M. Thomas and A. Mohaisen, "Kindred domains: detecting and clustering botnet domains using dns traffic," in *Proc. of ACM WWW*, 2014.

[19] Y.-M. Ke, C.-W. Chen, H.-C. Hsiao, A. Perrig, and V. Sekar, "Cicadas: Congesting the internet with coordinated and decentralized pulsating attacks," in *Proc. of ACM ASIACCS*, 2016.

[20] Vultr, https://www.vultr.com/.

[21] Apache HTTP server benchmarking tool, https://httpd.apache.org/docs/2.4/en/programs/ab.html.

[22] A. Kuzmanovic and E. W. Knightly, "Low-rate tcp-targeted denial of service attacks: the shrew vs. the mice and elephants," in *Proc. of ACM SIGCOMM*, 2003.

[23] ——, "Low-rate tcp-targeted denial of service attacks and counter strategies," *IEEE/ACM Transactions on Networking (TON)*, vol. 14, no. 4, pp. 683–696, 2006.

[24] X. Luo and R. K. Chang, "On a new class of pulsing denial-of-service attacks and the defense." in *Proc. of NDSS*, 2005.

[25] M. Guirguis, A. Bestavros, and I. Matta, "Exploiting the transients of adaptation for roq attacks on internet resources," in *Proc. of ICNP*, 2004.

[26] M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang, "Reduction of quality (roq) attacks on internet end-systems," in *Proc. of IEEE INFOCOM*, 2005.

[27] ——, "Reduction of quality (roq) attacks on dynamic load balancers: Vulnerability assessment and design tradeoffs," in *Proc. of IEEE INFOCOM*, 2007.

[28] Y. Zhang, Z. M. Mao, and J. Wang, "Low-rate tcp-targeted dos attack disrupts internet routing." in *Proc. of NDSS*, 2007.

[29] J. Luo, X. Yang, J. Wang, J. Xu, J. Sun, and K. Long, "On a mathematical model for low-rate shrew ddos," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1069–1083, 2014.

[30] G. Maciá-Fernández, J. E. Díaz-Verdejo, P. García-Teodoro, and F. de Toro-Negro, "Lordas: A low-rate dos attack against application servers," in *Proc. of CRITIS*. Springer, 2007.

[31] G. Maciá-Fernández, J. E. Díaz-Verdejo, and P. García-Teodoro, "Evaluation of a low-rate dos attack against application servers," *computers & security*, vol. 27, no. 7, pp. 335–354, 2008.

[32] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for cdns and web sites," in *Proc. of ACM WWW*, 2002.

[33] D. L. Mills, "On the accuracy and stablility of clocks synchronized by the network time protocol in the internet system," *ACM SIGCOMM Computer Communication Review*, vol. 20, no. 1, pp. 65–75, 1989.

[34] J. D. Guyton and M. F. Schwartz, "Experiences with a survey tool for discovering network time protocol servers." Technical Report CU-CS-704-94, Tech. Rep., 1994.

[35] D. L. Mills, A. Thyagarjan, and B. C. Huffman, "Internet timekeeping around the globe," University of Delaware, Tech. Rep., 1997.

[36] N. Minar, "A survey of the NTP network," MIT, Tech. Rep., 1999.

[37] C. D. Murta, P. R. Torres Jr, and P. Mohapatra, "Qrpp1-4: Characterizing quality of time and topology in a time synchronization network," in *Proc. of IEEE GLOBECOM*, 2006.

[38] "IEEE standard for a precision clock synchronization protocol for networked measurement and control systems," *IEEE Std 1588-2008*, 2008.

[39] "IEEE standard profile for use of IEEE 1588 precision time protocol in power system applications," *IEEE Std C37.238-2017*, 2017.

[40] K. S. Lee, H. Wang, V. Shrivastav, and H. Weatherspoon, "Globally synchronized time via datacenter networks," in *Proc. of ACM SIGCOMM*. ACM, 2016, pp. 454–467.

[41] D. Boteanu and J. M. Fernandez, "A comprehensive study of queue management as a dos counter-measure," *International journal of information security*, vol. 12, no. 5, pp. 347–382, 2013.

[42] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic," in *Proc. of NDSS*, 2008.

[43] H. Choi, H. Lee, and H. Kim, "Botgad: detecting botnets by capturing group activities in network traffic," in *Proc. of ICST COMSWARE*, 2009.

[44] D. Zhao, I. Traore, A. Ghorbani, B. Sayed, S. Saad, and W. Lu, "Peer to peer botnet detection based on flow intervals," in *Proc. of the IFIP SEC*, 2012.

[45] H. Hang, X. Wei, M. Faloutsos, and T. Eliassi-Rad, "Entelecheia: Detecting p2p botnets in their waiting stage," in *Proc. of IFIP Networking*, 2013.