Mitigation of Application Traffic DDoS Attacks with Trust and AM Based HMM Models

S.Prabha
Research Scholar,
Research and Development Centre,
Bharathiar University,
Coimbatore

Dr. R. Anitha
Professor & Director, Department of M.C.A.,
KSR College of Technology,
Tiruchengode

ABSTRACT

The increase in Internet-based transactions and communications offers new opportunities for hackers to disrupt business operations with DDoS attacks. Organizations that are not adequately protected risk losing customers, revenue, and their good reputations. This thesis discusses the challenges of identifying, countering, and avoiding crippling DDoS attacks. With the proposed comprehensive Self-Defending Network, organizations can deploy layers of defense to detect and mitigate the effects of DDoS attacks. The convenience, efficiency, and global reach of e-business benefit both consumers and businesses. But the accessibility of today's business operations brings increased security challenges. Legions of malicious hackers target e-commerce sites, online banks, partner networks, and Internet or e-mail servers seeking revenge or profit.

DDoS attack quickly overwhelms a company's server, router, firewall or network link with traffic, if successful, the attack floods the network or its resources so completely that legitimate traffic cannot be processed, and the company cannot function. The results are disastrous frustrated customers place orders elsewhere, service-level agreements are violated, and corporate reputations are damaged. Meanwhile, all IT and security resources focus on responding to the attack. Unfortunately, their efforts are usually too late and only partially effective. A security strategy must instantly identify and respond to DDoS threats, while maintaining the availability of critical network resources for custoers, partners, and employees.

The proposed model develops counter mechanism to mitigate the potency of the resource attacks and evaluate the efficacy.

The proposed access matrix captures the spatial-temporal patterns of a normal flash crowd. The anomaly detector based on hidden Markov model (HMM) is proposed to describe the dynamics of Access Matrix (AM) and to detect the attacks. Numerical results based on real Web traffic data are presented to demonstrate the effectiveness of the proposed method. Asymmetric attack overwhelms the server resources, by increasing the response time of legitimate clients from 0.1 seconds to 10 Seconds. Under the same attack scenario, HMM model limits the effects of false-negatives and false-positives and improves the victims' performance to 0.8 seconds.

1. INTRODUCTION

A distributed denial of service attack (DDoS attack) is a large-scale, coordinated attack on the availability of services of a victim system or network resource, launched indirectly through many compromised computers on the Internet [1]. DDoS attacks can seriously impair the Internet service. There have been a number of proposals and solutions to the DDoS attacks. However there is still no comprehensive solution which can protect against all known forms of DDoS attacks.

Basically DDoS attacks can be divided into two categories bandwidth Attack and resource attack. A bandwidth attack simply tries to generate packets to flood the victim's network so that the legitimate requests can not go to the victim machine. A resource attack aims to send packets that misuse network protocol or malformed packets to tie up network resources so that resources are not available to the legitimate users any more.

1.1 Application Layer DDoS Attack (Recent Trends)

Distributed denial of service (DDoS) attack has caused severe damage to servers and will cause even greater intimidation to the development of new Internet services. Traditionally, DDoS attacks are carried out at the network layer, such as ICMP flooding, SYN flooding, and UDP flooding, which are called Net-DDoS attacks. The intent of these attacks is to consume the network bandwidth and deny service to legitimate users of the victim systems. Since many studies have noticed this type of attack and have proposed different schemes (e.g., network measure or anomaly detection) to protect the network and equipment from bandwidth attacks, it is not as easy as in the past for attackers to launch the DDoS attacks based on network layer. When the simple Net DDoS attacks fail, attackers shift their offensive strategies to application-layer attacks and establish a more sophisticated type of DDoS attacks.

To circumvent detection, they attack the victim Web servers by HTTP GET requests (e.g., HTTP Flooding) and pulling large image files from the victim server in overwhelming numbers. In another instance, attackers run a massive number of queries through the victim's search engine or database query to bring the server down [4]. These attacks are called as application-layer DDoS (App-DDoS) attacks. The MyDoom worm [5] and the CyberSlam [6] are all instances of this type attack.

On the other hand, a new special phenomenon of network traffic called flash crowd [7], [8] has been noticed by researchers during the past several years. On the Web, "flash crowd" refers to the situation when a very large number of users simultaneously accesses a popular Website, which produces a surge in traffic to the Website and might cause the site to be virtually unreachable. Because burst traffic and high volume are the common characteristics of App-DDoS attacks and flash crowds, it is not easy for current techniques to distinguish them merely by statistical characteristics of traffic. Therefore, App-DDoS attacks may be stealthier and more dangerous for the popular Websites than the general Net-DDoS attacks when they mimic (or hide in) the normal flash crowd.

2. RELATED WORKS

The DDoS attack detection approaches in different scenario can be categorized as
Net-DDoS attacks versus stable background traffic, Net-DDoS attacks versus flash crowd (i.e., burst background traffic), App-DDoS attacks versus stable background traffic, and App-DDoS attacks versus flash crowd. The first two scenarios have been well studied and can be dealt with by most existing DDoS detection schemes while the other two groups are quite different from the previous ones. Besides the flooding attack pattern, App-DDoS attacks may focus on exhausting the server resources such as Sockets, CPU, memory, disk/database bandwidth, and I/O bandwidth.

With increasing computational complexity in Internet applications and larger network bandwidth, the server resources may become the bottleneck of those applications. Thus, the App-DDoS attacks may cause more serious problems in the high-speed Internet than in the past. The first characteristic of App-DDoS attacks is that the application- layer requests originating from the compromised hosts are indistinguishable from those generated by legitimate users. Unlike the Net-DDoS attacks, App-DDoS attacks do not necessarily rely on inadequacies in the underlying protocols or operating systems, they can be mounted with legitimate requests from legitimately connected network machines.

App- DDoS attacks utilize the weakness enabled by the standard practice of opening services such as HTTP and HTTPS (TCP port 80 and 443) through most firewalls to launch the attack. Many protocols and applications, both legitimate and illegitimate, can use these openings to tunnel through firewalls by connecting over a standard TCP port 80 (e.g., Code Red virus) or encapsulating in SSL tunnels (HTTPS). Attack requests aimed at these services may pass through the firewall without being identified. Furthermore, attackers may request services to the point where other clients are unable to complete their transactions or are inconvenienced to the point where they give up trying.

To handle the third scenario of APP-DDoS attacks Vs stable background traffic, four issues have to be considered:

a) Net-DDoS attacks detection methods are unable to collect enough offensive signals for detecting the App-DDoS attacks because they belong to different layers respectively,

b) TCP anomaly detection mechanisms can hardly identify the App-DDoS attacks launched by HTTP requests based on successful TCP connections,

c) in order to establish the TCP connection, attackers have to use the legitimate IP addresses and IP packets, which makes the anomaly detection mechanisms for IP packet become invalid, and

d) implied premise of most current detection schemes is that the characteristics of DDoS attack traffic differ from normal traffic, which might fail because App-DDoS attacks may mimic the access behaviors of normal users. However, because the background traffic of this scenario is assumed to be stable, some simple App-DDoS attacks (e.g., Flood) still can be monitored by improving existing methods designed for Net-DDoS attacks, e.g., we can apply the HTTP request rate, HTTP session rate, and duration of user’s access for detecting.

The second characteristic of App-DDoS attacks is that the attackers aiming at some special popular Websites are increasingly moving away from pure bandwidth flooding to more surreptitious attacks that masquerade as (or hide in) normal flash crowds of the Websites. Since such Websites become more and more for the increasing demands of information broadcast and electronic commerce, network security has to face a new challenge: how to detect and respond to the App-DDoS attacks if they occur during a flash crowd event, i.e., the fourth scenario of our clusters for DDoS attacks. The difficulties of dealing with such scenario include both the flash crowd and App-DDoS attacks are unstable, bursty and huge traffic volume and attack nodes may arrange their vicious Web traffic to mimic the normal one by HTTP synthetic tools, [10] so the malicious requests differ from the legitimate ones in intent but not in traffic characteristics. Therefore, most current detection mechanisms (e.g., those based on traffic characteristics) become invalid.

It is difficult to associate the amount of resources consumed to a client machine and attack nodes consisting of a large number of geographically widespread machines are increasingly belong to known client clusters. Thus, they cannot be filtered on the IP prefix. Other existing defense methods may be those based on man–machine interaction, e.g., puzzles, passwords, and the CAPTCHAs. However, as Kandula [3] and Ranjan [9] have pointed out, those schemes are not effective for the DDoS attack detection because they may annoy users and introduce additional service delays.

They may deny search engines access to the Web site, and the machine hosting authentication mechanism may be easy to become the new attack targets. Finally, compared with the consumption of resources such as CPU, memory, and database, App-DDoS attacks may not need to consume a lot of network bandwidth. Therefore, the traditional DDoS detection schemes designed for bandwidth exhausting attacks become ineffective.

3. TRUST AND HMM FOR RESISTING APPLICATION DDOS ATTACKS

The proposal of this work develops a novel scheme to capture the spatial-temporal patterns of a normal flash crowd event and to implement the App-DDoS attacks detection. Since the traffic characteristics of low layers are not enough to distinguish the App-DDoS attacks from the normal flash crowd event, the objective of this paper is to find an effective method to identify whether the surge in traffic is caused by App-DDoS attackers or by normal Web surfers. A flash crowd is a large spike or surge in traffic to a particular Web site. Proposed contribution of this work is

a) Define the Access Matrix (AM) to capture spatial-temporal patterns for normal flash crowd and to monitor App-DDoS attacks during flash crowd event;

b) Use hidden Markov model (HMM) to describe the dynamics of AM and to achieve numerical and automatic detection,

c) Present the monitoring scheme and validate it by a trust model for four strategic
Application DDoS attacks.

Web user behavior is mainly influenced by the structure of Website (e.g., the Web documents and hyperlink) and the way users access web pages. The proposed monitoring scheme considers the App-DDoS attack as anomaly browsing behavior. Investigate the characteristic of Web access behavior plots the HTTP request number and the user number per 5s during the burst Web workload. From the maximum correlation coefficient 0.9986, between the series of request numbers and that of the user numbers, we can see that the normal flash crowd is mainly caused by the sudden increment of user amount.

The entropy of the aggregate access behavior against our model does not change much during the flash crowd event, which implies that both the main access behavior profile of normal users and the structure of Website do not have obvious varieties during the flash crowd event and its vicinity area. The users' access behavior profile can be used to detect the abnormal varieties of users' browsing process during the flash crowd. Since the document popularity has been widely used to characterize the user behavior and improve the performance of Web server and Internet cache

3.1 Access Matrix

The access matrix model is the policy for user authentication, and has several implementations such as access control lists (ACLs) and capabilities. It is used to describe which users have access to what objects. The access matrix model consists of four major parts a list of objects, a list of subjects, a function T which returns an object's type and the matrix itself, with the objects making the columns and the subjects making the rows. In the cells where a subject and object meet lie the rights the subject has on that object. Some example access rights are **r**ead, **w**rite, e**x**ecute, **l**ist and **d**elete.

An access matrix has several standard operations associated with it:

- Entry of a right into a specified cell
- Removal of a right from a specified cell
- Creation of a subject
- Creation of an object
- Removal of an subject
- Removal of an object

The two most used implementations are access control lists and capabilities. Access control lists are achieved by placing on each object a list of users and their associated rights to that object. An interactive demonstration of access control lists can be seen.

Capabilities are accomplished by storing on each subject a list of rights the subject has for every object. This effectively gives each user a key ring. To remove access to a particular object, every user (subject) that has access to it must be "touched". A touch is an examination of a user's rights to that object and potentially removal of rights. This brings back the problem of sweeping changes in access rights.

Access restrictions such as access control lists and capabilities sometimes are not enough. In some cases, information needs to be tightened further, sometimes by an authority higher than the owner of the information. For example, the owner of a top secret document in a government office might deem the information available to many users, but his manager might know the information should be restricted further than that. In this case, the flow of information needs to be controlled secure information cannot flow to a less secure user.

3.2 Hidden Markov Model

A hidden Markov model (HMM) is a statistical model in which the system being modeled is assumed to be a Markov process with unobserved state. An HMM can be considered as the simplest dynamic Bayesian network. In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters.

In a hidden Markov model, the state is not directly visible, but output dependent on the state is visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. The adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model. Even if the model parameters are known exactly, the model is still hidden. Hidden Markov models are especially known for their application in temporal pattern recognition such as application layer DDoS attack mitigation, bio informatic, speech recognition etc.,

4. EXPERIMENTATION OF TRUST AND HMM MODEL ON TARGET AND ATTACK SERVERS

The proposed attack mitigation model of Access matrix based HMM deployed in the target server make the following two assumptions.

a) Under session flooding attacks, the bottleneck of a server is the maximal number of simultaneous session connections, called as MaxConnector. It depends not only on the bandwidth of the server, but also on other resources of the server, e.g. CPU, memory, maximal database connections.

b) Without attacks, the total number of session connections of the server should be much smaller than MaxConnector, e.g., smaller than 20% of MaxConnector, as a server would set the threshold much higher to tolerate the potential burst of requests, e,g., flash crowds on websites.

4.1 Legitimate User Model

In contrast to attackers, legitimate users are people who request services for their benefit from the content of the services. Therefore, the inter-arrival time of requests from a legitimate user would form a certain density distribution density(t). The user model is built in the following way

i) Use traces of Internet accesses to build an initial model density0(t), where t is a inter-arrival time and density(t) is the probability a legitimate user will revisit the service after t seconds. Three different data sets are traced in this study, two from academic (i.e., university) environments, and one from a commercial Internet provider.

ii) Rebuild user model densityi+1(t) with the newly collected inter-arrival times of all legitimate users after TMH runs d days under model densityi(t), where d is randomly chosen from [dmin, dmax]. Note that we build the new density distribution using the data of legitimate users, whose requests are accepted by proposed model server. It means that densityi+1(t) is tightly derived from densityi(t) and hence is difficult to be fooled by attackers.

As a practical legitimate user model, it should satisfy the following properties, firstly, it should converge fast to the users' accesses interval distribution, secondly, it should be dynamic as the distribution may change from time to time, and finally it should be lightweight to be easily implemented and monitored in the defense mechanism. The user model we proposed in this section can satisfy the first two requirements as the density function is updated regularly, and it is lightweight as the update to density distribution is incremental and it does not try to capture the complicated reasons for the changes reflected.

In this initial density distribution model, there are a number of peaks in the user request arrival intervals, with the most prominent ones corresponding to intervals of one minute, one hour and one day. The mean inter-arrival time was 25.4 hours with a median of 1.9 hours and a standard deviation of 49.6 hours.

4.2 Attacker Model

The goal of session flooding DDoS attack is to keep the number of simultaneous session connections of the server as large as possible to stop new connection requests from legitimate users being accepted. Therefore, an attacker may consider using the following strategies when he controls a lot of zombie machines or can misuse P2P network as an attack platform

i) Send session connection requests at a fixed rate, without considering the response or the service ability of victim.

ii) Send session connection requests at a random rate, without considering the response or the service ability of victim.

iii) Send session connection requests at a random rate and consider the response or the service ability of victim by adjusting request rate according to the proportion of accepted session connection requests by the server.

iv) First send session connection requests at a rate similar to legitimate users to gain trust from server, then start attacking with one of the above attacking strategies.

The tradeoff of these strategies is between cost and ability to avoid the detection.
Strategy 1 and 2 are easy to implement, but they are also easier to be detected, strategy 3 and 4 are more complicated as they consider the server responses or modeling legitimate users. Strategy 4 requires long-term preparation of attackers in order to gain a high trust level. This strategy needs attackers being more patient. In session flooding attacks, attackers cannot spoof their IPs or change them within a session, because a session is set up on TCP connection which requires a three-way handshake. Since attackers cannot hide themselves through modifying IPs, they would prefer using strategy 3 and 4 to mimic behavior of legitimate users, to evade detection. The simulation is carried out to each strategy.

4.3 Resistance for Application DDoS attacks

The functional attributes of the AM based HMM model for resisting the application DDoS attacks are, AM applied at the server for incentive and performance reasons, reduce the processing delay and to avoid being a new target of attacks, easy to deploy and independent to the details of servers. The defense mechanism need not know what services the server runs or what configuration it uses. The resistance model, adaptive to the server's resource consumption and differentiate between concurrent requests. To evaluate the visiting history of clients effectively use trust model. The client who behaves better in history will obtain higher degree of trust. The trust models are

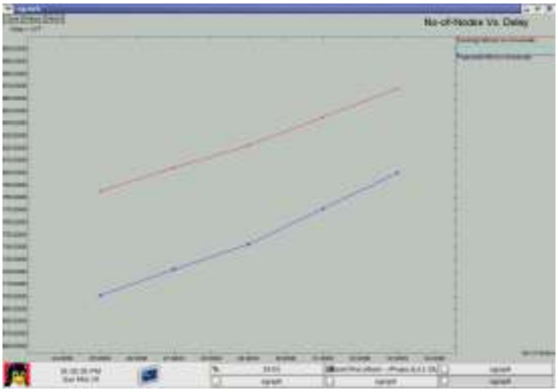
Short-term trust Ts, estimate the recent behavior of a client. It is used to identify those clients who send session connection requests at a high rate when the server is under session flooding attacks. Long-term trust Tl, estimate the long-term behavior of a client. It is used to distinguish clients with normal visiting history and those with abnormal visiting history. Negative trust Tn, cumulating the distrust to a client, each time the client's overall trust falls below the initial value T0. It is used to penalize a client if he is less trustworthy than a new client. Misusing trust Tm, cumulate the suspicious behavior of a client who misuses its cumulated reputation. It is used to prevent vibration attacks by repeatedly cheating for high trust. Trust T, representing the overall trustworthiness of a client, which takes into account all of his short-term trust, long-term trust, negative trust and misusing trust.

To reduce the processing overhead, a short-term blacklist should be implemented. The blacklist records the list of clients whose trust values are too low. When a client's trust T drops below some threshold, he is recorded. Clients are used to represent both legitimate users and malicious attackers. The components of AM based HMM and its communication with other modules is into the blacklist with an expiration time. By then banned from accessing the server until blacklist record expires. The mitigation mechanism is deployed at the server. A session connection request first reaches AM and it checks whether the client is blacklisted; if not, it computes the trust to the client and use trust-based scheduling to schedule the connection request for the server.

5. PERFORMANCE EVALUATION

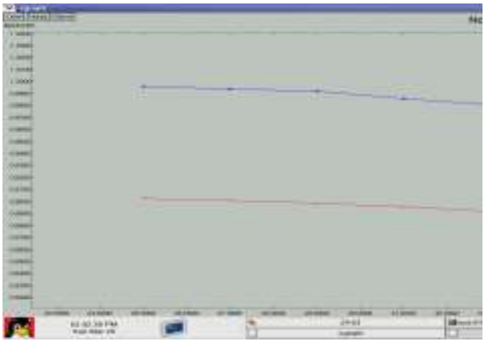
The simulation is set up in a local area network with 100Mbps links. We simulated 100 legitimate users, varying number of attackers and a server protected by the proposed model of AM based HMM. Clients request the server for HTTP sessions. The server directly responds to them if they pass the verification and get scheduled by AM based HMM. Constrained by the server's memory and other resources, Max Connector is set to 1000. That is, the server can serve maximally 1000 concurrent sessions, beyond that, the session requests will be dropped. In our simulation, legitimate users follow the model described in Section 3.1, we set dmin=15 and dmax=20; while attackers attack with different strategies. The life time of a session follows an exponential distribution with mean equals to 20 seconds.

The performance of the HMM model with bandwidth threshold is depicted in Graph 1. It lists the delay level against the varying node size on the network in effective communication between the source destinations pairs in the typical laid out network structure of ISPs. When number of nodes increases, delay also increases. When compared to non threshold scheme, the delay is low in the bandwidth threshold method.



Graph 1: Client nodes Vs Response delay

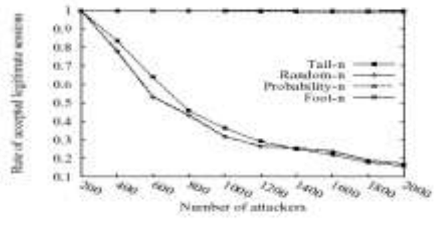
The graph 2 shows the result of the DDoS resistive mechanism function with number of nodes against the bandwidth capacity of the target servers. As the number of nodes in the network of the source or intermediate junction increases, consumption of bandwidth decreases. When compared to non threshold trust scheme, the bandwidth is high in the proposed HMM scheme



Graph 2: Number of nodes Vs Bandwidth

AM based HMM uses default values of alpha, beta and gamma in the computation of trust. It issues license to new users with density (now - LT) and density (AT) set to be 0.1. After it verifies a license and updates the trust, it schedules the requests using the policies. For comparison, we also implemented two simple scheduling policies Tail-n: drop the n = µ £ N requests that arrive last in a time slot and Random-n randomly drop n = µ £ N requests in a time slot. A time slot is one second.

The performance evaluation of access matrix for resisting application DDoS attacks is done by comparing scheduling policies by plotting the acceptance rate of legitimate sessions in graph 3. The number of legitimate users is 100 and the proportion of each kind of attackers who adopt one of the four attack strategies is 25%. It is seen that under trust-based scheduling strategies, the acceptance rate of legitimate sessions keeps at a high level and is insensitive to the number of attackers.



Graph 3: Acceptance of legitimates sessions against multiple attackers

Even when the number of attackers is 2000, the acceptance rate of legitimate user sessions is still 99.1% and 99.7% with *Probability-n* and *Foot-n* scheduling policies respectively. However, it is only 16.0% using *Tail-n* policy and 17.2% using *Random-n* policy.

6. CONCLUSION

To defend against application DDoS attacks is a pressing problem of the Internet. Motivated by the fact that it is more important for service provider to accommodate good users when there is a scarcity in resources, we present a mechanism to mitigate session flooding attack using trust evaluated from users' visiting history. We verify its effectiveness with simulations under different attack strategies. Comparing to other defense mechanism, proposed HMM is, independent to the service details, adaptive to the server's resource consumption and extendable to allow collaboration among servers.

The dynamic DDoS threat mitigation solution arms the service provider with a tool to counter at the root of the problem. By helping application service customers to clean their process operations, the end user will get a greater application service experience, and service provider will be rewarded by increase loyalty and reduced issues. The resistance mechanism mitigates session flooding attack using trust evaluated from users' visiting history. Its effectiveness is verified with simulations under different attack strategies.

The proposed model of access matrix based Hidden Markov Model for application traffic DDoS attack mitigation with traffic flow filters (HMM) and trust scheme allow service providers to take a more proactive role in protecting broadband subscribers and enterprises against attacks. Enterprises protect themselves by placing firewalls at the perimeter of the network, and just recently they have started to add unified access control (UAC) technologies to take control and offer remediation support for compromised PCs also from within their own network. Against DDoS attacks, however, the companies depend on service providers to take actions.

REFERENCES

[1] N. Long S. Dietrich and D. Ddittrich, “Analyzing distributed denial of service tools: the shaft case,” in Proceedings of the LISA XIV.

[2] Gary C. Kessler, “Defenses against distributed denial of service attacks,” http://www.garykessler.net/library/ddos.html, November 2000.

[3] Celeste Biever, “How zombie networks fuel cybercrime,” http://www.newscientist.com/article.ns?id=dn6616, November 2004.

[4] K. Poulsen, “FBI Busts Alleged DDoS Mafia,” 2004. [Online]. Available: http://www.securityfocus.com/news/9411 Authorized licensed use limited to: K Duraiswamy. Downloaded on July 29, 2009 at 09:20 from IEEE Xplore.

[5] “Incident Note IN-2004-01 W32/Novarg. A Virus,” CERT, 2004. [Online]. Available: http://www.cert.org/incident_notes/ IN-2004-01.html

[6] S. Kandula, D. Katabi, M. Jacob, and A. W. Berger, “Botz-4-Sale: Surviving Organized DDoS Attacks that Mimic Flash Crowds,”MIT, Tech. Rep. TR-969, 2004 [Online]. Available: http://www.usenix.org/events/ nsdi05/tech/ kandula/kandula.pdf

[7] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long, “Modeling, Analysis and Simulation of Flash Crowds on the Internet,” Storage Systems Research Center Jack Baskin School of Engineering University of California, Santa Cruz Santa Cruz, CA, Tech. Rep. UCSC-CRL-03-15, Feb. 28, 2004 [Online]. Available: http://ssrc.cse.ucsc.edu/, 95064

[8] J. Jung, B. Krishnamurthy, and M. Rabinovich, “Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites,” in Proc. 11th IEEE Int. World Wide Web Conf., May 2002, pp. 252–262.

[9] S. Ranjan, R. Karrer, and Knightly, “Wide area redirection of dynamic content by Internet data centers,” in Proc. 23rd Ann. Joint Conf. IEEE Comput. Commun. Soc., Mar. 7–11, 2004, vol. 2, pp. 816–826.

[10] J. Cao, W. S. Cleveland, Y. Gao, K. Jeffay, F. D. Smith, and M.Weigle, “Stochastic models for generating synthetic HTTP source traffic,” in Proc. IEEE INFOCOM, 2004, vol. 3, pp. 1546–1557.

[11] L. Limwiwatkul and A. Rungsawangr, “Distributed denial of service detection using TCP/IP header and traffic measurement analysis,” in Proc. Int. Symp. Commun. Inf. Technol., Sappoo, Japan, Oct. 26–29, 2004, pp. 605–610.