Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

# Protecting Network Services against DDoS Attacks

Ms. B. Keerthana[1], Mr. C. Muthubharathi[2]
[1]MCA., MPhil., Research Scholar, Department of Computer Science,SSM College of Arts and Science, Komarapalayam, Tamilnadu, India
[2]M.Sc., MCA., MPhil., Assistant Professor, Department of Computer Science,SSM College of Arts and Science, Komarapalayam, Tamilnadu, India

**Abstract - S**ervice providers facilitate shared services to execute user required functionalities. Denial of Service (DoS) attacks is raised to disrupt the resources under the service providers. Processor, memory and network bandwidth are violated by the attackers. DoS attacks are initiated by single users. Distributed Denial of Service (DDoS) attacks are initiated by a group of users. It is a complex task to identify the DDoS attacks with minimum loses of network resources. Attacker groups are referenced as botnets in the DDoS attack scenario.

Bot master manages the command and control request supply operations within the bots. Simple Mail Transfer Protocol (SMTP) and Internet Relay Chat (IRC) protocols are used to transfer the bot master instructions. Network firewalls control the SMTP and IRC communication for network security reasons. Domain Name Services (DNS) are adapted to distribute the host name and address details. Bot masters use the DNS for command and control messaging process. Exponentially Distributed Query and Piggybacking Query attacks are detected using the markov chain analysis and statistical analysis mechanism. DNS tunneling methods are employed to perform stream based communications. Attack detection latency is increased in the statistical analysis method.

The network service protection scheme is constructed with attack and botnet communication discovery mechanism. Privacy ensured domain name query analysis model is adapted in the security scheme. DDoS attack detection is carried out using anomaly based method. Network traffic and domain query operations are analyzed with Navy bayesian classification technique. Domain Name Service parameters are secured with RSA algorithm. Detection latency is minimized in the attack and communication discovery process.

## I. Introduction

A denial of service attack is characterized by an explicit attempt by an attacker to prevent legitimate users of a service from using the desired resources. The distributed format adds the “many to one” dimension that makes these attacks more difficult to prevent. A distributed denial of service attack is composed of four elements. First, it involves a victim, i.e., the target host that has been chosen to receive the brunt of the attack. Second, it involves the presence of the attack daemon agents. These are agent programs that actually conduct the attack on the target victim [12]. Attack daemons are usually deployed in host computers. These daemons affect both the target and the host computers. The task of deploying these attack daemons requires the attacker to gain access and infiltrate the host computers. The third component of a distributed denial of service attack is the control master program. Its task is to coordinate the attack. Finally, there is the real attacker, the mastermind behind the attack. By using a control master program, the real attacker can stay behind the scenes of the attack.

Although it seems that the real attacker has little to do but sends out the “execute” command, he/she actually has to plan the execution of a successful distributed denial of service attack [10]. The

 

attacker must infiltrate all the host computers and networks where the daemon attackers are to be deployed [3]. The attacker must study the target's network topology and search for bottlenecks and vulnerabilities that can be exploited during the attack. Because of the use of attack daemons and control master programs, the real attacker is not directly involved during the attack, which makes it difficult to trace who spawned the attack.

## II. Related Work

The Domain Name System (DNS) has been increasingly being used by attackers to maintain and manage their malicious infrastructures. As a result, recent research on botnet detection has proposed number of approaches that leverage the distinguishing features between malicious and benign DNS usage [4]. The first study is focused on DNS data for analyzing malicious behavior. The results of the passive DNS analysis showed that malicious domains that are used in Fast-Flux networks exhibit behavior that is different than benign domains. Similarly, Kui Xu et al. [5] performed passive monitoring to identify DNS anomalies. In their paper, although they discuss the possibility of distinguishing abnormal DNS behavior from benign DNS behavior, the authors do not define DNS features that can be used to do so.

In general, botnet detection through DNS analysis follows two lines of research: The first line of research tries to detect domains that are involved in malicious activities [7]. The goal is to identify infected hosts by monitoring the DNS traffic [1]. The second line of research focuses on the behaviors of groups of machines in order to determine if they are infected.

## III. Problem Statement

Botnet command-and-control (C&C) channel refers to the protocol used by bots and botmaster to communicate to each other, or to submit stolen data. A C&C channel for a botnet needs to be reliable, redundant, noncentralized, and easily disguised as legitimate traffic. Many botnet operators used the Internet Relay Chat protocol (IRC) or HTTP servers to pass information [11]. Botnet operators constantly explore new stealthy communication mechanisms to evade detection. HTTP-based command and control is difficult to distinguish from legitimate web traffic. The feasibility of email as a stealthy botnet command and control protocol was studied by researchers. In this systematically investigate the feasibility of solely using Domain Name System (DNS) queries for botnet command and control. DNS provides a distributed infrastructure for storing, updating and disseminating data that conveniently fits the need for a large-scale command and control system. The HTTP protocol is for the end-to-end communication between a client and a server. In comparison, DNS provides not only a means of communication between computers, but also systematic mechanisms for naming, locating, distributing and caching resources with fault tolerance. These features of DNS may be utilized to fulfill a more effective command-and-control system than what HTTP servers may provide.

Botnet controllers use stealthy messaging systems to set up large-scale command and control requests. A C&C channel for a botnet needs to be reliable, redundant, noncentralized and legitimate traffic. Domain Name Service (DNS) provides a distributed infrastructure for storing, updating and disseminating data. DNS is targeted as a stealthy botnet command-and-control channel. Malicious DNS activities are hiding at the network level. Exponentially Distributed Query and Piggybacking Query attacks are detected using the markov chain analysis and statistical analysis mechanism. Probability distribution based analysis model is used to detect automatic domain flux attacks. DNS tunneling technique is used for transmitting arbitrary data via DNS protocol. The following drawbacks are identified in the existing system.

- DNS sensitive data access is not controlled

 

- Data leaks are not accurately detected
- User intention is required for anomaly detection
- Detection latency is high

## IV. Bayes Classifier for Attack Detection

The domain name query based attacks for service providers are detected using the bayes classifier algorithm. In simple terms, a naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable [2]. For example, a fruit may be considered to be an apple if it is red, round, and about 3" in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of the presence or absence of the other features. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting [6]. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; in other words, one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations [8]. An analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests [9]. An advantage of Naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

In simple terms, a naive Bayes classifier assumes that the presence or absence of a particular feature is unrelated to the presence or absence of any other feature, given the class variable. For example, a fruit may be considered to be an apple if it is red, round, and about 3" in diameter. A naive Bayes classifier considers each of these features to contribute independently to the probability that this fruit is an apple, regardless of the presence or absence of the other features. For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting. In many practical applications, parameter estimation for naive Bayes models uses the method of maximum likelihood; one can work with the naive Bayes model without accepting Bayesian probability or using any Bayesian methods.

Despite their naive design and apparently oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations. In 2004, an analysis of the Bayesian classification problem showed that there are sound theoretical reasons for the apparently implausible efficacy of naive Bayes classifiers. Still, a comprehensive comparison with other classification algorithms in 2006 showed that Bayes classification is outperformed by other approaches, such as boosted trees or random forests. An advantage of Naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix.

## V. Network Services Protection Against DDoS Attacks

The Domain Name Service based attack detection system is designed to handle command and control message process over DNS query values. The DNS query analysis is performed with statistical

 

analysis. The system analyzes the DNS query values to identify the insertion of DDoS attack detection information. Machine learning approach is used to detect the DDoS attacks. The system is divided into five major modules. They are request observer, statistical analysis, Bayesian analysis, command and control request handler and security and privacy process. The request observer module is designed to collect service requests from the clients. Statistical analysis module is designed to detect DDoS attacks using statistical methods. Bayesian approach based attack detection mechanism uses the classification process. The command and control handler is designed to manage the DNS query based botnet communication activities. Security and privacy module is designed to provide security and privacy for DSN parameters.

### 5.1. Request Observer

Stream requests are collected from various clients. The server assigns session instances to new stream requests. The stream requests are processed by the web server. The responses are redirected to the clients.

### 5.2. Statistical Analysis

The statistical analysis model is used to detect the Service attacks. The request flow and its similarity are analyzed with frequency and interval values. The user session patterns are learned from the request flow analysis. Attack decisions are made with the support of DNS query format. The requests are assigned with normal or attack labels in the analysis.

### 5.3. Bayesian Analysis

The Bayesian analysis model is used to detect DNS query based attacks using machine learning approaches. The classification model is used for the request category identification process. The Bayesian classification algorithm is used for the attack detection process. The attacker requests are rejected by the service provider.

### 5.4. Command and Control Request Handler

The command and control requests are exchanged between the botmaster and bots. The DNS query values are used to exchange the command and control requests. The DNS query responses are verified with the domain name server details. Command and control instructions are dropped with reference to their category.

### 5.5. Security and Privacy

The DNS query values are submitted from the clients in the networks. The DNS responses are prepared by the domain name server. The user can add additional parameters to the DNS responses. The RSA algorithm is used to protect the DNS parameter values. Sensitive attribute values are protected with cryptography methods.

 

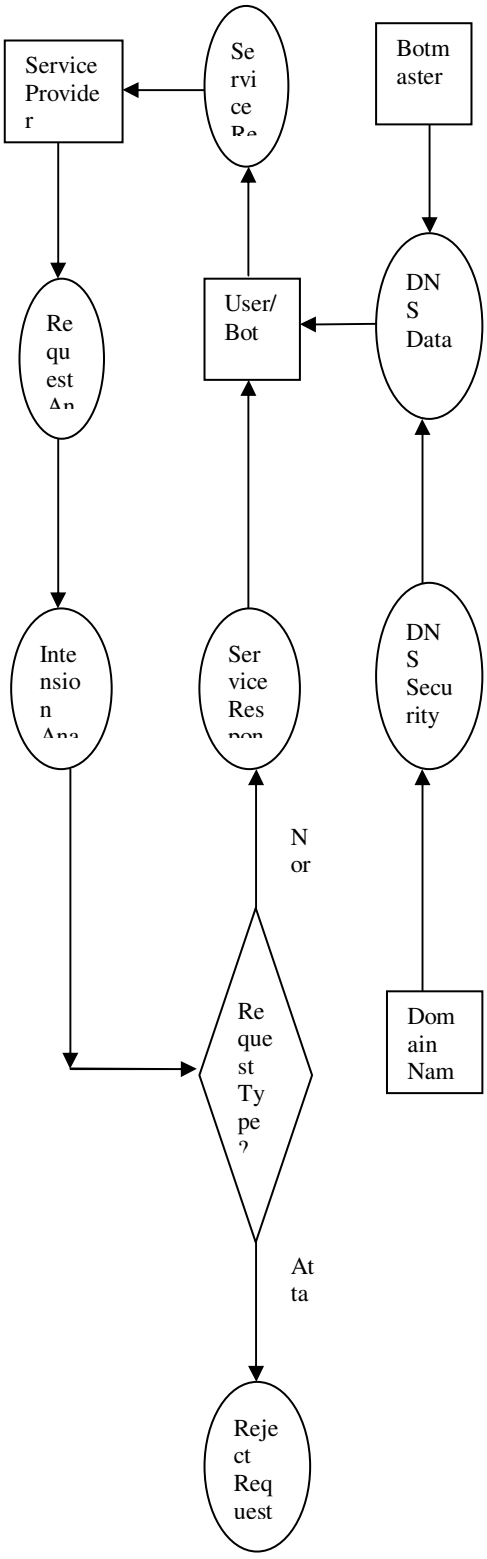


**Fig. No: 5.1. Attack Detection Model for Network Services**

## VI. Results and Discussion

The Domain Name Service (DNS) based Command and Control (C&C) mechanism is used to carry out the bots and botmaster communication process. The service providers are attacked with the support of communication information. The system operations are carried out in two ways. They are DNS communication detection process and Botnet attack detection process. The DNS communication detection process is performed to identify the bot and botmaster communications. The domain name and host information are provided in a Domain Name Server. Client requests are processed in the Domain Name Server. Bot and Botmaster communications are also carried out using the domain name service query mechanism. The attack detection process is used to protect the service providers. The service requests are raised by the bots. Clients are also submits the service requests to the service providers. The service provider request information is analyzed by the attack detection mechanism.

The DNS communication detection and botnet attack detection system is developed to secure the Domain Name Server and the Service Providers. The DNS communication detection scheme controls the communication between the bots and the botmaster. The Statistical Analysis Model (SAM) and the Naïve Bayesian Classifier (NBC) methods are used to detect the communication process. The service provider is also monitored and secured with the support of the Statistical Analysis Model (SAM) and the Naïve Bayesian Classifier (NBC) methods. In the Statistical Analysis Model the system integrates the markov analysis, piggybacking model and Exponential Distribution analysis methods. The Naïve Bayesian Classifier is used for machine learning based detection mechanism.

### 6.1. Experimental Setup

The performance of Botnet attack detection scheme is carried out with a set of experiments. The experiments are conducted under an Intel Core 2 Duo 2.8 GHz speed and 2 GB of memory. Java front end and Oracle back end are installed in the Windows 7 platform environment. The system is developed to test the Domain Name Server, Service Provider, Client, Bots and Botmaster activities. The Domain Name Server is constructed with the real data values. The service provider is tested with different service requests that are collected from the Internet. The Bots and Botmaster communications are tested with user level requests. The service requests are also generated by the client and the bots. All the applications are integrated with the same system environment. The user can access al the applications in the same environment. The system also designed to support user level and machine level based request submission mechanism. The user can test the system with all types of requests.

### 6.2. Dataset

The DNS command and control communication detection system is tested with real and synthetic datasets. The domain names and associated host information are collected from the real domain name server from www. whois.internic.net. The DNS details are collected and updated in the local database environment. The DNS entries are used for the client request handling process. Domain Name Server attribute details are listed in table 6.1. The domain name, IP addresses, registrar name and activities date information are maintained in the DNS environment. The DNS queries are used to collect information about the host name.

| S. No | DNS Attribute Name | Attribute Description |
|---|---|---|
| 1 | dname | Domain Name |

| | | |
|---|---|---|
| 2 | hname | Host name |
| 3 | ipaddr | IP address |
| 4 | rname | Registrar name |
| 5 | wserver | Web server |
| 6 | refurl | Referred url |
| 7 | udate | Updated date |
| 8 | cdate | Created date |
| 9 | edate | Expiration date |
| 10 | param | Parameter value |

**Table No: 6.1 Attribute details for the Domain Name Service**

### 6.3. Experimental Analysis

The DNS communication detection and attack detection system is tested with different request levels. The system is tested with different request count levels for the DNS and the service provider. User level and machine level requests are submitted in the system. Command and control detection and attack detection operations are verified by the system. Attack detection accuracy is verified with three performance metrics. They are false positive rate, false negative rate and detection latency metrics. The false positive and false negative rates are used to estimate the accuracy level for the attack detection system. The detection time is analyzed under the detection latency analysis. The DNS communication detection is verified with communication detection ratio parameter. Statistical Analysis Model (SAM) and Naïve Bayesian Classifier (NBC) methods are used for the communication and attack detection process.

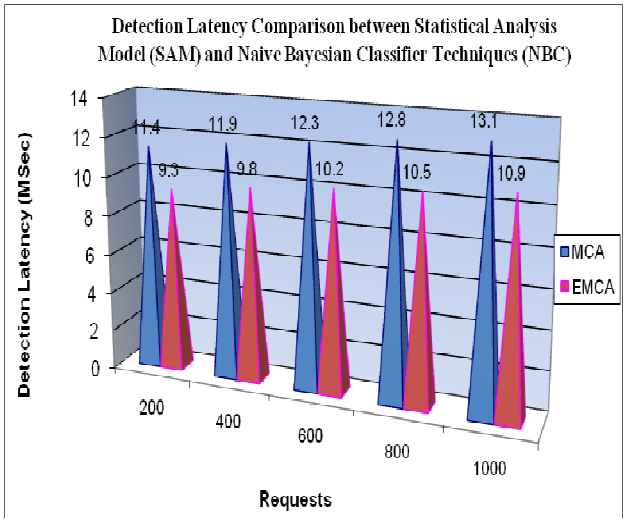


**Figure 6.1 Detection Latency Comparison between Statistical Analysis Model (SAM) and Naive Bayesian Classifier Techniques (NBC)**

The detection latency comparison is carried out between the Statistical Analysis Model (SAM) and Naïve Bayesian Classifier techniques. Average attack detection duration is measured as detection latency for the system. Figure 6.1. show the detection latency comparison between the SAM and NBC

techniques. The analysis result shows that the Naïve Bayesian Classifier reduces the detection latency 20% than the Statistical Analysis Model technique.

The attack detection process accuracy is calculated using the false positive and false negative rates. The false positive rate indicates the falsely assigned positive results against the negative one. Figure 6.2. show the false positive rate comparison between the SAM and NBC techniques. The analysis result shows that the Naïve Bayesian Classifier reduces the false positive rate 10% than the Statistical Analysis Model technique.

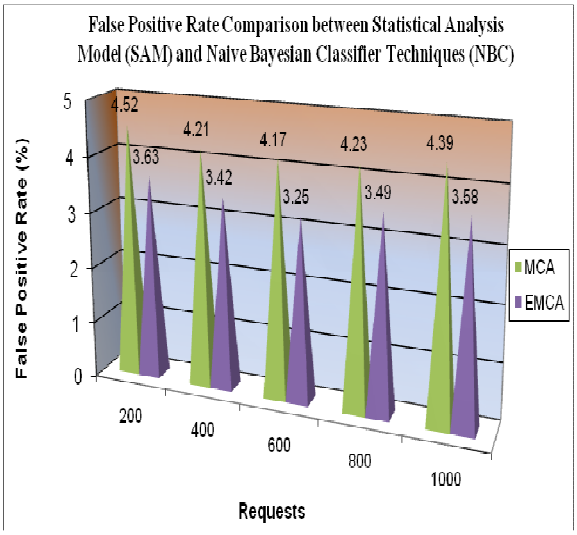


**Figure 6.2 False Positive Rate Comparison between Statistical Analysis Model (SAM) and Naive Bayesian Classifier Techniques (NBC)**

The false negative rate indicates the falsely assigned negative results against the positive one. Figure 6.3. show the false negative rate comparison between the SAM and NBC techniques. The analysis result shows that the Naïve Bayesian Classifier reduces the false negative rate 10% than the Statistical Analysis Model technique.

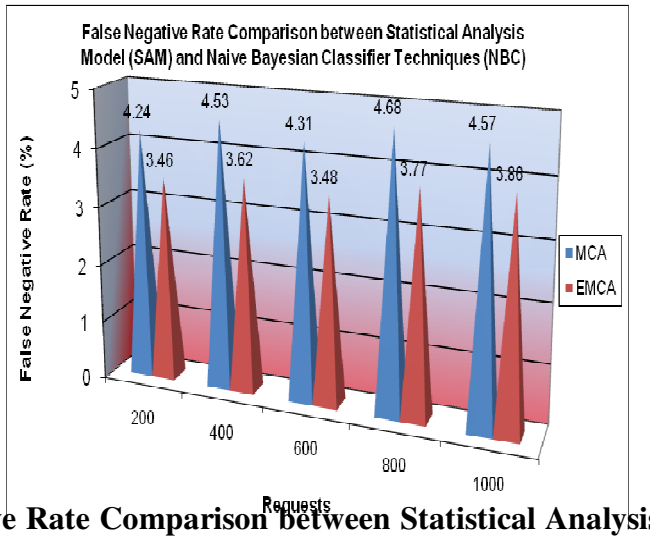


**Figure 6.3 False Negative Rate Comparison between Statistical Analysis Model (SAM) and Naive Bayesian Classifier Techniques (NBC)**

The DNS query values are used to fetch the doain name details from the Domain Name Server. Bots and Botmaster communication are also conducted in the DNS query model. The communication

detection process is used to identify the bots communications. The detection ratio comparison is carried out between the Statistical Analysis Model (SAM) and Naïve Bayesian Classifier techniques. Average attack detection rate is measured as detection ratio for the system. Figure 6.4. show the detection ratio comparison between the SAM and NBC techniques. The analysis result shows that the Naïve Bayesian Classifier increases the detection ratio 15% than the Statistical Analysis Model technique.

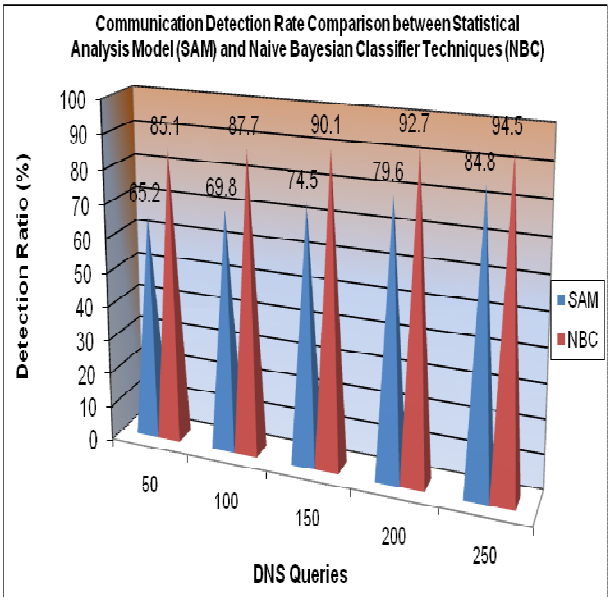


**Figure 6.4 Communication Detection Rate Comparison between Statistical Analysis Model (SAM) and Naive Bayesian Classifier Techniques**

## VII. Conclusion and Future Enhancement

The DNS query based attack detection scheme is enhanced to provide privacy preserved data traffic analysis. Automated anomaly detection is adapted to the system. Naiva Bayesian classification technique is integrated to the system. Small query analysis mechanism is integrated with the system. The system performs botnet communication detection and control operations. DDoS attack detection mechanism is included in the system. The system improves the detection accuracy with minimum latency. DNS parameter security is also provided by the system. The system can be enhanced with the following futures. The system can be enhanced to handle attack detection under distributed server environment. The DNS parameter security model can be improved with data integrity verification methods. The system can be improved to perform attack detection under load balancing process. The intrusion detection system can be enhanced to provide message alert to all service providers.

## References

1. H. Zhang, W. Banick, D. Yao and N. Ramakrishnan, "User Intention-Based Traffic Dependence Analysis for Anomaly Detection," Proc. Workshop Semantics and Security (WSCS), vol. 5, no. 3, pp. 415 – 423,May 2012.

2. Jingtang Luo, Xiaolong Yang, Jin Wang and Keping Long, "On a Mathematical Model for Low-Rate Shrew DDoS" IEEE Transactions On Information Forensics And Security, Vol. 9, No. 7, July 2014

3. K. Xu, H. Xiong, D. Stefan, C. Wu and D. Yao, "Data-Provenance Verification for Secure Hosts," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 173-183, Mar.-Apr. 2012.

4. Kui Xu, Patrick Butler and Danfeng (Daphne) Yao, "DNS for Massive-Scale Command and Control", IEEE Transactions On Dependable and Secure Computing, Vol. 10, No. 3, May/June 2013.

5. L. Bilge, C. Kruegel and M. Balduzzi, "Exposure: Finding Malicious Domains Using Passive DNS Analysis," IEEE Transactions On Network and Distributed System Security, vol. 2, no. 2, pp. 235-242, Feb. 2011.

6. P. Butler, K. Xu and D. Yao, "Quantitatively Analyzing Stealthy Communication Channels," Proc. Ninth Int'l Conf. Applied Cryptography and Network Security (ACNS '11), pp. 238-254, 2011.

7. Robert Mitchell and Ing-Ray Chen, "Adaptive Intrusion Detection of Malicious Unmanned Air Vehicles Using Behavior Rule Specifications" IEEE Transactions On Systems, Man, And Cybernetics: Systems, Vol. 44, No. 5, May 2014

8. Shui Yu, Weijia Jia, Yong Xiang and Feilong Tang, "Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient" IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 6, June 2012.

9. Udi Ben-Porat, Anat Bremler-Barr and Hanoch Levy, "Vulnerability of Network Mechanisms to Sophisticated DDoS Attacks" IEEE Transactions On Computers, Vol. 62, No. 5, May 2013

10. Xianliang Jiang, Guang Jin and Wei Wei, "RED-FT A Scalable Random Early Detection Scheme with Flow Trust against DoS Attacks" IEEE Communications Letters, Vol. 17, No. 5, May 2013

11. Yajuan Tang, Xiapu Luo and Rocky K. C. Chang, "Modeling the Vulnerability of Feedback-Control Based Internet Services to Low-Rate DoS Attacks" IEEE Transactions On Information Forensics And Security, Vol. 9, No. 3, March 2014

12. Zhiyuan Tan, Priyadarsi Nanda and Ren Ping Liu, "A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis", IEEE Transactions On Parallel And Distributed Systems, Vol. 25, No. 2, February 2014

 