Endpoint Mitigation of DDoS Attacks Based on Dynamic Thresholding

Daewon Kim, Byoungkoo Kim, Ikkyun Kim, Jeongnyeo Kim, and Hyunsook Cho

Cyber Convergence Security Research Department
Electronics and Telecommunications Research Institute
218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Korea
{dwkim77,bkkim05,ikkim21,jnkim,hscho}@etri.re.kr

Abstract. Socially and economically, the distributed denial-of-service (DDoS) attacks have been serious threats in the cyber world. Despite of many researches, current defense methods can be vulnerable to the DDoS attacks of unknown traffic pattern to avoid the methods. That is because most of the defense policies configured for the methods are fixed thresholds that were mainly determined by the learning of traffic volume. To overcome the problem caused by the fixed thresholds, we introduce the endpoint mitigation method based on the dynamic thresholding of DDoS defense policies according to the usage changes of system resources. We focused on the fact that the usage changes of system resources show the abnormal statuses of server if the failure/delay of service is occurred by the DDoS attacks that have not been blocked by current defense thresholds. The proposed method detects the server overload as measuring the usage changes of system resources and automatically adjusts current defense thresholds in conjunction with the strength of usage change. As the result, the service problem caused by the DDoS attacks can be gradually mitigated by the automatic threshold controlling of our method.

Keywords: cyber threat, network security, distributed denial-of-service attack, intrusion detection system, intrusion prevention system.

1 Introduction

Internet services have been rapidly developed enough to cover most of our lives [1], [2]. However, these important internet services are always exposed to various attacks millions time a day [3]. The recent attacks are mainly focused on financial and political demands [4], [5]. To achieve these goals, the attackers use the distributed denial-of-service (DDoS) attacks with the zombie PCs which have been infected with malicious codes due to security vulnerabilities. Although many researches [6] to enhance security vulnerabilities have been studied, new vulnerabilities are still being discovered and new malwares are still being propagated to infect new zombie PCs. In recent, the DDoS attack that caused extensive damage to South Korea mobilized approximately 150,000 zombie PCs [7].

To prevent the servers from DDoS attacks, various DDoS defense methods have been researched. These works can be divided into the ways to change current network

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 381–391, 2012.

infrastructure [8], [9] and the ways to maintain the infrastructure [10], [11], [12]. The formers to modify network protocols or network configurations are effective for the defense of DDoS attack. However, the methods are difficult to spread the technology because they need to change current network environments. After all, the latters to maintain current network environments are leading research trends on the defense of DDoS attack. A research trend recently appeared is visualization techniques [14]. However, the techniques have a disadvantage that the information for visualization has to be sampled due to the performance problem.

The researches [10], [11], [12] to maintain current network environments learn the traffic patterns of normal and attack, and configure their defense policies as the differences of analyzed traffic patterns. After that, they compare the measurement result of incoming traffic to the defense policies configured in advance. However, the defense policies with fixed thresholds based on the learned traffic patterns may pass many attack packets if weak policies. On the other hand, if strong policies, the fixed thresholds may block many normal packets. As the reason, current DDoS defense systems cannot arbitrarily apply the strong policies to prepare for new DDoS attacks of the future with unknown traffic patterns. Because of this limitation, new DDoS attacks to circumvent the policies of fixed thresholds will be continuously appeared, and the denials of service by the new attacks will be repeated as well.

In order to solve the problem of defense policy with the fixed threshold, our method monitors the usage changes of server resources such as CPU, memory and network session, and automatically controls current thresholds depending on the strength of usage change whenever the resource usages show abnormal patterns. That is possible because service troubles can be detected with the abnormal usage patterns of server resources. Finally, if any defensive action leads the abnormal usages of server resources to the normal usages, the server service will be stabilized. In the paper, our method gradually and automatically adjusts the fixed thresholds for the defense mechanisms, which are embedded in some security systems, as applying the analysis results of current abnormal resource usages.

The rest of the paper is structured as follows. In Section II, we briefly introduce the overview for the automatic control of DDoS defense thresholds, and Section III describes the detailed operations of our method. In Section IV, the paper introduces the *SecureNIC* which is a FPGA-based network interface card for endpoint DDoS defense and presents the experiment results that our prototype program automatically adjusts the thresholds of defense mechanisms embedded in the *SecureNIC*. Finally, we conclude the paper in Section V.

2 The Overview

Fig. 1 shows the operation overview of our method under DDoS attacks. In Fig. 1, (1) when unknown DDoS attacks are incoming into the server via the *SecureNIC*, (2) if the embedded defense mechanisms are applying the policies with wrong thresholds or the *SecureNIC* has not defense mechanisms to detect the new attack traffic patterns, (3) the DDoS attacks flow into the server without some attack detection. (4) The server has service troubles by the attack packets and the service troubles are measured with various server usages such as CPU, memory, and network session. The proposed method analyzes the measured resource usages and (5) depending on the results, the method adjusts current defense thresholds or triggers new defense mechanism being prepared. The prepared defense mechanism means the defense method that is more effective if it is activated only in cases of some special attacks. (6) By the proposed method, next attack packets are gradually blocked through the changes of defense thresholds and the activation of prepared defense mechanism. (7) Finally, the server status is stabilized and the service troubles are disappeared as well.

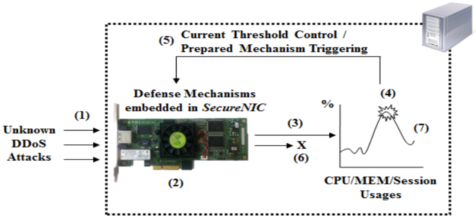


Fig. 1. The overview

3 Endpoint Mitigation of DDoS Attacks Based on the Dynamic Thresholding

3.1 The Types of Server Loads

Fig. 2 shows the typical server loads that can be occurred as time passes. In Fig. 2, the server load means the current usage ratio (%), which is max 100%, of resources that have direct impacts on the server service. In our prototype program, it includes the usage ratios of CPU, memory, and network session. Additionally, as a service-wide, it can include the usage ratios of PPS, BPS, and SYN_RECV for each protocol and port. Large server load means that the values of these usage ratios are high.

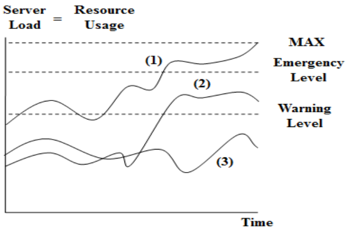


Fig. 2. The typical server loads

Like Fig. 2, normal server loads can be classified into the three types of (1) *rising*, (2) *surging*, and (3) *vibrating*. For the classification, the level configuration of resource usage ratios is required. From the perspective of DDoS attacks, the case of *rising* means that the attack transfers a small amount of traffic at a low speed to avoid the threshold policies of DDoS defense mechanisms. On the other hand, the case of *surging* means that the attack transfers a large amount of traffic at a high speed in a short time to paralyze the server with the weak DDoS defense policy. From the perspective of a server, the cases of *rising* and *surging* have a high probability that the service troubles will be occurred and on the other hand, the *vibrating* will be occurred routinely.

Therefore, on the endpoint to mitigate the service troubles caused by DDoS attacks, our method analyzes the changes of server load and detects the cases of *rising* and *surging*. After that, the method automatically controls the thresholds of current defense mechanisms to return the current status of *rising* and *surging* to the normal *vibrating*. It also analyzes the fluctuation intensity of server load to determine the strength of automatic control.

3.2 The Analysis of Server Loads

If some troubles are happened to the services by DDoS attacks, the resource usages will show abnormal changes. Thus, if monitoring the server resources, we can determine whether the service troubles are occurred or not. In the paper, based on the fact, the method of paper periodically monitors the resource usage ratios of CPU, memory, and network session to determine whether the services are going smoothly. For example, in the case of TCP web service, for reliable service to users, the service has normally the maximum concurrent TCP session number depending on the performance of server system. As monitoring the current usage ratio of TCP sessions, the method in advance can detect the overload statuses causing service failures.

Excepting for the cases targeting the vulnerable codes of operating system and service program among various DDoS attacks, most of DDoS attacks use the attack techniques with the excessive or irregular service requests to occur the service failures of target server. For example, if the 100 percent utilization of resources is monitored, the service troubles on the server can be detected in advance. Our method periodically monitors the usage ratios of resources and detects the service troubles. Then, the method controls the thresholds of defense policies depending on the change strength of resource usages. Therefore, our method can respond to the new and unknown attacks that are pointed to the shortcomings of existing methods based on the learned traffic volumes.

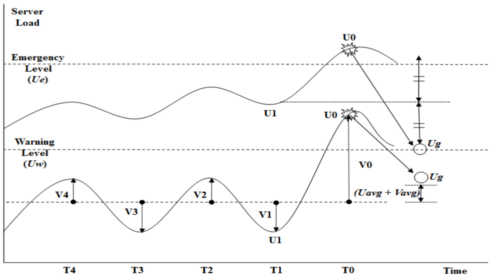


Fig. 3. Detection and threshold control

Fig. 3 shows the detection of service troubles depending on the server loads, which are the resource usages. If the resource usages in normal situation exceed the emergency level percentage *Ue*, in general the administrator extends the service performance with the upgrades and replacements of current systems. Therefore, if the service is smoothly dealing with the service requests of normal users, current usage ratio *U0* will not exceed *Ue*, except for special situations. In contrast, if *U0* often exceeds *Ue* by attacks, it means that the server will be out of service soon. Eventually, if *U0* exceeds *Ue* as the cases of *rising* and *surging* on the server loads, the adjustments of current defense thresholds will be needed at *T0*. (*Emergency Detection:* $if\ U0 > Ue$).

On the other hand, if *U0* exceeds the warning level percentage *Uw*, the additional analysis about usage changes is required to finally determine whether the services has troubles by attacks because this can even occur to normal situations. Our method has individual FIFOs (First-In First-Out) on each resource to save, on every second, the usage ratio *U* of maximum *n* number. The average usage ratio $Uavg = \frac{\sum_{i=0}^{i=n-1}|Ui|}{n}$ and the average usage variations $Vavg = \frac{\sum_{i=0}^{i=n-1}|Vi|}{n}$ are calculated for each resource on every period. The *Warning Detection* can determine the *surging* type attack. (*Warning Detection:* $if\ (U0 > Uw)\ and\ (U0 > Uavg)\ and\ (V0 > Vavg)$). The reason that current surged *U0* needs to be compared to *Uavg* and *Vavg* is for excluding the surge cases of usage changes that can often happen under normal situations.

3.3 The Load Measurement of Internal Processes

In the cases of CPU and Memory, the server overload can be temporarily occurred by the internal processes unrelated to the services for external users. For example, under the condition that the attack traffic does not flow into the server, the server can be overloaded by the programs for server maintenance such as log managements. When adjusting the defense thresholds in this situation, the normal user traffic can be blocked by the *SecureNIC*. To avoid the problem, if the server overload is detected as *Warning* or *Emergency*, it is necessary to determine whether the cause of overload is the internal processes or not.

To do this, our method sorts the processes in descending order of CPU and MEM usages. After that, the decision is given by

$$If\ \frac{\sum_{i}^{Np} Up_i}{U0} > Uint, Not\ Attack\ and\ No\ Threshold\ Control$$

Where: *Np* = the process number to be selected; Up_i = the CPU or MEM usage (%) of *i*-th process; *Uint* = the set usage to be compared.

3.4 The Selection of Defense Threshold to Be Adjusted

If the service troubles occurred by external attack traffic are detected by analyzing the resource usages, the analysis to find the cause of theses troubles is worked to determine how to defend the attacks. By the analysis, our method determines which policies should be adjusted among the defense mechanisms mounted to the *SecureNIC*. The work uses the traffic statistics of each defense mechanism. Each of the defense mechanisms has individual thresholds for blocking the attacks, and the incoming traffic that exceeds the threshold is blocked. As mentioned earlier, if the server overload situation occurs by normal service requests, the administrator will upgrade the system performance. Thus, assuming that the server performance is enough for accepting all normal service requests, if the service troubles happen in situation that each of the defense mechanisms is normally working with each threshold, the reasons of troubles can be consider as three cases.

The first is the case that the service trouble occurs under the situation blocking the attack traffic over the threshold. The second is the case that the service trouble occurs by the attack traffic under the threshold, and the third is the case that the defense mechanism is none for responding the attacks. The point that the resource usage of server is abnormally increasing means that any kinds of traffic volumes are unusually increasing. Therefore, the method of this paper selects the defense mechanism closest to each threshold among traffic statistics related to each defense mechanism and adjusts its threshold stronger. Through this analysis, in the cases of first and second the proposed method can mitigate the service trouble with the adjustment of policy threshold, and in the case of third, the method can activate the prepared defense mechanism.

3.5 The Threshold Control for Attack Defense

If the defense mechanism to be adjusted is selected, our method adjusts the threshold for strengthening the attack response. The purpose of threshold adjustment is to stabilize the abnormally overloaded usages through the gradual blocking of specific traffic determined as attacks. The existing defense methods fail to block the attack traffic when the defense threshold was wrong, and the server service may be out of control because there is no time to fundamentally analyze the attack traffic such as the generation of defense rule and signature. Therefore, the large damage will be happened economically and socially because the server needs a lot of time to recover the service.

On the other hand, the proposed method, by controlling the defense thresholds, manages gradually the server load to avoid the out-of-control service during the attacks are continued. The first reason that the gradual threshold adjustment is required is because the majority of normal users may be blocked by the strong threshold adjustment at a time, and the second is because the method is no need to adjust the defense threshold by force if the server performance is available apart from the attack traffic that is incoming.

Fig. 3 represents the example to determine the strength of threshold adjustment according to the resource usage change of *rising* and *surging*. In the case of *rising* type the attack traffic increases slowly and in the future we can expect the slow increasing of server load. Thus, without the strong threshold adjustment, our approach precisely controls the current defense threshold for simply deviating from the emergency level. When the server load exceeds the emergency level, our method adjusts the current threshold to come *U0* to *Ug*. New threshold *Pn* is shown to Eq. (1).

$$Pn = \frac{P0 \times \{U0 - 2 \times (U0 - U1) \times Ru\}}{U0} \quad \text{Eq. (1)}$$

Where: *P0* = current threshold value; *Ru* = constant ratio that reflects the difference of *U0* and *U1* (If *Ru* > 1, weak adjustment).

In the case of *surging* type the attack traffic increases rapidly. Thus, in the future we can expect the rapid increasing of server load, and if there is no defense for that situation, the server will be out-of-control soon. With the strong threshold adjustment, it is necessary to urgently stabilize the current server load to the average server load. When the server load rapidly exceeds the warning level, our method adjusts the current threshold to come *U0* to *Ug*. New threshold *Pn* is shown to Eq. (2).

$$Pn = \frac{P0 \times (Uavg + Rv \times Vavg)}{U0} \quad \text{Eq. (2)}$$

Where: *Vavg* = average usage variation; *Uavg* = average usage (%); *Rv* = constant ratio that reflects *Vavg* (If *Rv* > 1, weak adjustment).

3.6 Attack Mitigation

Although our approach adjusted the thresholds to stabilize the server overload, the server load may not be reduced in contrary to our prediction. The first reason is because the current usage *U0* for determining new threshold *Pn* and the current threshold *P0* applied already to the defense mechanism are not proportional relation. Therefore, this case can fail to decrease the usage because the attacks were not blocked as much as the expected through the threshold adjustment. The second reason is due to the wrong selection of defense mechanism although the cause analysis of service trouble was performed. The third is the case that there is no defense mechanism for blocking the attacks. The reason that the subsequent attack mitigation is required, even after the current threshold was adjusted by our method, is because the additional defense is needed for the first and second situation to stabilize the server load even except for the third case.

To reduce current overload, our approach firstly adjusts the threshold of defense mechanism selected by the cause analysis to the new threshold *Pn* determined by Eq. (1) and Eq. (2). After the threshold changing, if the server load is reduced or maintained by monitoring the load at the next time, the threshold adjustment of selected defense mechanism is considered as a success. On the other hand, after the threshold changing, if the server load is increased, our method adjusts the thresholds of all mechanisms embedded in the *SecureNIC* because the reason of adjustment failure is one among the above first and second. At this time, our method decreases the thresholds of all defense mechanisms to 10% rather than conforming to Eq. (1) and Eq. (2). Such the processes of subsequent defenses are lasted until the server overload is released. In the situation of subsequent defense, if the server load is decreased under the warning level, our method considers as the termination of attack situation, and returns all thresholds to their original thresholds.

3.7 The Selection and Continuous Block of Attack Traffic

The purpose of proposed method is to mitigate the damage of out-of-service caused by an overload condition due to the attacks. The proposed method cannot choose and block only attacks. The reason that the method has not the detailed functions is because the work implemented by software aggravates the server load on the endpoint system. To complement this problem, the *SecureNIC* includes an intelligent IP-based ACL (access control logic) which are based on the arrival time gap of request packets between the attack and normal traffic.

If the thresholds by detecting the server overload are adjusted, the attack and normal traffic that excesses the thresholds are blocked, and then the remote IP addresses are registered to the ACL list. In general, the requests by attack programs are transferred automatically and quickly, and the requests by normal users are transferred manually and slowly. Using the fact, in the first step, if the packet of ACL-registered IP address flows into the *SecureNIC* within the block time of one second, it is blocked as requests of attack program and the block time of the ACL entry is doubled as the consideration of attack IP. If the packets of ACL-registered IP address do not flow into the *SecureNIC* within the block time, the IP address is deleted from the ACL as the consideration of normal user IP.

4 Experiments

4.1 SecureNIC

Our method has been implemented to the management software for controlling the host-based DDoS defense network interface card, which is the *SecureNIC* in Fig. 4. It is a NIC developed for the DDoS defense of host-level server and includes one giga bit NIC function that supports both the optic and RJ-45 with DDoS defense function. With Xilinx FPGA (Field Programmable Gate Array), the *SecureNIC* supports the SYN proxy function for the defense of attacks related to TCP session, the DDoS defense function of network and application level, and various ACL functions.

Fig. 4. The SecureNIC

Our implementation in the management software monitors the usage ratios (%) of CPU, memory, and service session in the *SecureNIC*-installed server and automatically adjusts the thresholds of defense mechanisms based on the hardware security logics in the *SecureNIC* against the attacks of TCP ESTABLISHED flooding [13], UDP/ICMP flooding, HTTP GET flooding, and so on. We demonstrate the effectiveness of our mechanism through a representative experiment because the way to adjust the thresholds is similar to each of the defense mechanisms on the hardware security logics.

4.2 The Mitigation of TCP ESTABLISHED Flooding Attack

The *SecureNIC* blocks most of the abnormal behaviors related to TCP session with the SYN Proxy feature on FPGA. However, the SYN Proxy cannot response the attacks exhausting the resource of TCP ESTABLISHED session [13]. Every second, our implementation monitors the usage ratio (%) of current ESTABLISHED number to maximum allowable ESTABLISHED number. If the usage ratio shows abnormal patterns, our method sends RST packets to the attack hosts for disconnecting the ESTABLISHED sessions in order of a large number of ESTABLISHED per IP address.

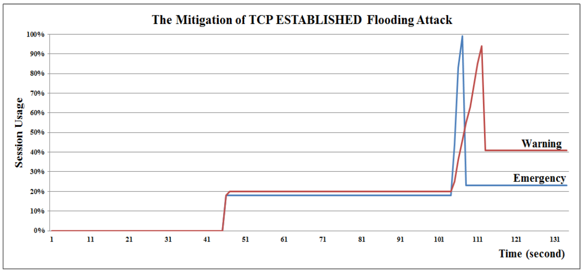


Fig. 5. Experiment of TCP established flooding attack

The number to disconnect the ESTABLISHEDs is *P0 – Pn*. In Section III.E, *P0* is the maximum allowable service session number and *Pn* is the allowable service session number to be adjusted. The difference of threshold adjustment compared to other defense mechanisms is the point that *Pn* is not applied to new threshold and the attack IPs are forced to ACL on FPGA to block the traffic of attack IPs.

Fig. 5, on the *SecureNIC*, shows the experimental results of session defense function in our approach. The environment is as follows:

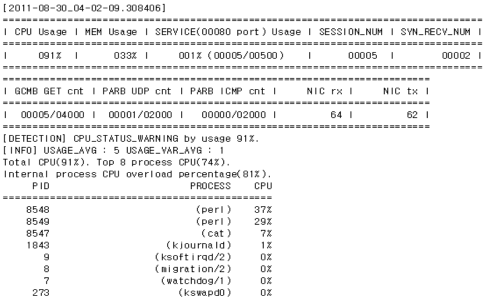
- Apache web server in CentOS 5.6.
- The maximum allowable TCP ESTABLISHED number is set to 1000.
- The emergency level is 99% and the warning level is 89%.
- The normal average ESTABLISHED number is about 200 from 47 to 105 seconds. The attack is started from 106 seconds.

In the case of *Emergency*, the ESTABLISHED attack for exhausting the service session resource was performed with full speed of attack tool and only in 2 seconds the session usage ratio became 100%. When the usage ratio exceeded 99%, our method disconnected the attack sessions to be the previous normal average ratio of about 20%. In the case of *Warning*, we increased about 200 ESABLISHED sessions with the attack tools every second. That means that the average usage ratio also increases. At the time of 113 seconds, our method detected 95% session usage and disconnected the attacks sessions to be the measured average usage of about 40% because it was matched to the condition of warning level detection.

4.3 The Load Analysis Log of Internal Processes

Fig. 6 presents a part of log file saved by our program of the *SecureNIC* operated in a real web hosting server. From the log, when the CPU average usage (*USAGE_AVG*) is 5%, the warning (*CPU_STATUS_WARNING*) is detected with the current usage

91%. The top CPU usage processes except for the service process *httpd* used about 74%. *Uint* had been 50%, our program did not adjust the thresholds because the total usage ratio of top processes usage occupied 81% of current usage 91% load.



```
[2011-08-30_04-02-09.308406]
==================================================================================
| CPU Usage | MEM Usage | SERVICE(00080 port) Usage | SESSION_NUM | SYN_RECV_NUM |
==================================================================================
|    091%   |      033% |     001% (00005/00500)    |      00005  |        00002 |
==================================================================================
=========================================================================
| GCMB GET cnt | PARB UDP cnt | PARB ICMP cnt |      NIC rx |      NIC tx |
=========================================================================
|  00005/04000 |  00001/02000 |   00000/02000 |          64 |          62 |
=========================================================================
[DETECTION] CPU_STATUS_WARNING by usage 91%.
[INFO] USAGE_AVG : 5 USAGE_VAR_AVG : 1
Total CPU(91%). Top 8 process CPU(74%).
Internal process CPU overload percentage(81%).
    PID                          PROCESS    CPU
===============================================
   8548                           (perl)    37%
   8549                           (perl)    29%
   8547                            (cat)     7%
   1843                      (kjournald)     1%
      9                    (ksoftirqd/2)     0%
      8                    (migration/2)     0%
      7                     (watchdog/1)     0%
    273                        (kswapd0)     0%
```

Fig. 6. The log of internal process load

5 Conclusion

Most of the existing DDoS Defense systems determine large thresholds to reduce the false alarms of normal situation because their systems apply the defense thresholds of fixed type. It means that under the thresholds the probability of successful attack is high as well. To solve the problem, we focused on the fact that the service troubles can be detected from the usage ratios of system resources.

Based on the fact, we developed the *SecureNIC* and the management software which includes the method of paper. Finally, in the paper we suggested the automatic threshold adjustment method of DDoS defense mechanisms embedded in the *SecureNIC* of server system through the analysis of current server loads. The effectiveness of our method was presented with the automatic control experiment of *SecureNIC* developed by our project team.

References

1. Internet World Stats. Internet Growth Statistics, `http://www.internetworldstats.com/emarketing.html`
2. The Internet Economist. The Internet Economy 25 years After.com, `http://www.itif.org/files/2010-25-years.pdf`
3. Symantec. Internet Security Threat Report-Volume XV, `http://eval.symantec.com/mktginfo/enterprise/white_papers/b-whitepaper_internet_security_threat_report_xv_04-2010.en-us.pdf`
4. Cisco. Cisco 2010 Annual Security Report, `http://www.cisco.com/en/US/prod/collateral/vpndevc/security_annual_report_2010.pdf`

5. Symantec. Symantec's monthly state of spam report (October 2008), http://eval.symantec.com/mktginfo/enterprise/other_resources/b-state_of_spam_report_10-2008.en-us.pdf
6. Lee, J.-H., Sohn, S.-G., Chang, B.-H., Chung, T.-M.: PKG-VUL: Security Vulnerability Evaluation and Patch Framework for Package-Based Systems. ETRI Journal (2009)
7. Hauri. 7.7 DDos Virus Report, http://www.maxoverpro.org/77DDoS.pdf
8. Liu, X., Yang, X., Xia, Y.: NetFence: Preventing Internet Denial of Service from Inside Out. In: ACM SIGCOMM (2010)
9. Argyraki, K., Cheriton, D.: Scalable Network-layer Defense Against Internet Bandwidth-Flooding Attacks. ACM/IEEE ToN 17(4) (2009)
10. Carl, G., Kesidis, G., Brooks, R.: Denial-of-Service Attack-Detection Techniques. IEEE Internet Computing 10, 82–89 (2006)
11. Vijayasarathy, R., Raghavan, S., Ravindran, B.: A system approach to network modelling for DDoS detection using a Naive Bayesian classifier. In: Communication Systems and Networks, COMSNETS (2011)
12. Yu., S., Zhou, W., Dross, R., Jia, W.: Traceback of DDoS Attacks Using Entropy Variations. IEEE Transactions on Parallel and Distributed Systems 22 (2011)
13. The Open Web Application Security Project. OWASP HTTP Post Tool, https://www.owasp.org/index.php/OWASP_HTTP_Post_Tool
14. Chang, B.-H., Jeong, C.: An Efficient Network Attack Visualization Using Security Quad and Cube. ETRI Journal (2011)