Yu, Shui and Zhou, Wanlei 2008, Entropy-based collaborative detection of DDOS attacks on community networks*, in Proceedings of the 6th Annual IEEE International Conference on Pervasive Computing and Communications*, IEEE, Piscataway, N.J., pp. 566-571.

Sixth Annual IEEE International Conference on Pervasive Computing and Communications

Entropy-Based Collaborative Detection of DDOS Attacks on Community Networks

Shui Yu and Wanlei Zhou

School of Engineering and Information Technology
Deakin University, Burwood, VIC 3125, Australia
{syu, wanlei}@deakin.edu.au

Abstract

A community network often operates with the same Internet Service Provider domain or the virtual network of different entities who are cooperating with each other. In such a federated network environment, routers can work closely to raise early warning of DDoS attacks to void catastrophic damages. However, the attackers simulate the normal network behaviors, e.g. pumping the attack packages as Poisson distribution, to disable detection algorithms. It is an open question: how to discriminate DDoS attacks from surge legitimate accessing. We noticed that the attackers use the same mathematical functions to control the speed of attack package pumping to the victim. Based on this observation, the different attack flows of a DDoS attack share the same regularities, which is different from the real surging accessing in a short time period. We apply information theory parameter, entropy rate, to discriminate the DDoS attack from the surge legitimate accessing. We proved the effectiveness of our method in theory, and the simulations are the work in the near future. We also point out the future directions that worth to explore in the future.

1. Introduction

The Internet is an open architecture susceptible to various forms of network attacks, a prime example of which is the Distributed Denial-of-service (DDoS) attack. The early attacks to the well-known web sites, such as CNN, Amazon, Yahoo, in early 2000 brought the normal services of victims stopped for hours [9]. A study showed that the number of DDoS attacks increased by 50% per year [11], and the attacks were also increased in sophistication and severity. There are lots of attack methods, such as DDoS, DRDoS [17]. Flooding packages are the most common and effective attack tool among all the attack methods.

To against DDoS attacks, researchers have designed and implemented a number of countermeasures. In general, the counter DDoS attacks fall in three folds: detection, defense (or mitigation), and IP trace-back. It is obvious that detecting DDoS attacks in real time is the first step of combating DDoS attacks. [4] surveyed the methodologies that used in DDoS detection, such as, activity profiling [15] [8], sequential change-point detection [2] [20] [3] [7] , wavelet analysis [1], chi-square/entropy detector [8] [13], and so on. However, fighting DDoS is a constant spy versus spy game. The defender will try to defend against all the known attacks, and the attacker will try to disguise their attacks to stay under the radar. Every time, when a new defending method is invited, the attacker will design a counter defending method to attack, e.g. attackers may simulate the normal network behaviors to disable the detection.

It is a open question and a considerable challenge to discriminate DDoS flooding attacks from sudden increase in legitimate traffic or flash events [12] [4] [7]. [12] tried to use three dimensions, traffic patterns, client characteristics and file reference characteristics, to discriminate the flash event from DoS attacks. This counter attack method can not follow the ever changing attack methods, as the attack patterns are changing from time to time, moreover, the attacker will simulate the network traffic pattern of flash event, et al, which will disable the detector quickly. The entropy detector mentioned in the survey [4] came from paper [8], which used entropy very shallowly. It can raise the alarm for a sudden massive accessing, however, it can not discriminate the DDoS from the surge of legitimate accessing. [7] suggested to separate flash crowd from DDoS flooding by "other" performance metrics (except the change-point detection method) to capture their difference, however, there is no clue about the possible metrics in the paper at all.

An effective and critical method against DDoS attacks is to identify attacks as early as possible. Most of the current DDoS defense schemes are based on detecting sustained congestion on communication links [14], running out of half-open SYN queue, or imbalance between incoming and outgoing traffic volume on routers [4]. These detecting methods, unfortunately, have to wait for the flooding becomes widespread, consequently, they make the defense scheme ineffective to fence off the DDoS timely. An ideal scenario is to eliminate the DDoS before the attack packages reach the target.

The community network offers the convenience of network administration. A community network often operates with the same Internet Service Provider domain or the virtual network of different entities who are cooperating with each other. In such a federated network environment, routers can work closely to raise early warning of DDoS attacks to void catastrophic damages.

0-7695-3113-X/08 $25.00 © 2008 IEEE
DOI 10.1109/PERCOM.2008.12

In this paper, we motivated by identifying DDoS attacks at the early stage in community networks, and eliminating attack packages before they reach the target. The second motivation is to counter the attacks which simulate normal network accessing patterns to fly under the radar, namely, discriminate DDoS attacks from the surge of legitimate traffic. In order to achieve these goals, we take use of information theory parameter, *entropy*, which is a measure for the randomness of a process, to raise the alarm for the potential attacks. We define the packages which share the same destination address as a *flow*, and entropy will be employed to measure the randomness of flows on a given router. Once an alarm is raised for a flow, we will employ, *entropy rate*, which is the rate of growth of entropy, on the path of the flow to the destination. If the flow is a DDoS attack, the entropy rates on different routers are the same or very close (in case of the normal accessing as 'noise').

Our contributions in this paper are as follows:

- Raising DDoS attack alarm at the edge routers in community networks. We collect flow samples on routers of community networks, especially the edge routers, for a time window. If the entropy on routers changed dramatically, a DDoS attack alarm is raised at the router. However, the entropy itself can not discriminate DDoS attacks from the surge of legitimate traffic.
- Discriminating DDoS attacks from the surge of legitimate traffic effectively. In order to further confirm the raised DDoS attack alarm, entropy rates for the suspected flow are calculated at the neighbored routers. If the entropy rates are the same or very similar, then we can confirm the DDoS attack.
- Eliminating DDoS attack packages before they reach the target. Once the attack is confirmed, the router will discard the suspected flow, namely, the attack flow. In curtain community networks, the detection and elimination could be done before the attack packages reach the target.

The remainder of the paper is organized as follows: Section 2 presents the related work of DDoS attacks, and the preliminaries of information theory that we will deploy in the following analysis. We analyze and model the detection of DDoS attacks in Section 3. The detection algorithm is designed in Section 4. Finally, we discuss the future work in Section 5.

2. Related Work and Background

A. Related work of DDoS attacks

DDoS attacks target on exhausting the victim's resources, such as, network bandwidth, computing power, operating system data structures, and so on. To launch a DDoS attack, malicious users first establish a network of computers that they will use to generate the volume of traffic needed to deny services to computer users. To create this attack network, attackers discover vulnerable sites or hosts on the network. Vulnerable hosts are usually those that are either running no antivirus software or out-of-date antivirus software, or those that have not been properly patched. Vulnerable hosts are then exploited by attackers who use their vulnerability to gain access to these hosts. The next step for the intruder is to install new programs (known as *attack tools*) on the compromised hosts of the attack network. The hosts that are running these attack tools are known as *zombies*, and they can carry out any attack under the control of the attacker. Many zombies together form what we call an *army* [17].

Attackers can use different kinds of techniques (referred to as *scanning techniques*) in order to find vulnerable machines [19] [21] [22]. Such as *random scanning*, the machine that is infected by the malicious code (such a machine can be either the attacker's machine or the machine of a member of their army, such as a zombie) probes IP addresses randomly from the IP address space and checks their vulnerability. *Hit-list scanning:* Long before attackers start scanning, they collect a list of a large number of potentially vulnerable machines. In their effort to create their army, they begin scanning down the list in order to find vulnerable machines. *Topological scanning:* Topological scanning uses information contained on the victim machine in order to find new targets. *Local subnet scanning:* This type of scanning acts behind a firewall in an area that is considered to be infected by the malicious scanning program. *Permutation scanning:* In this type of scanning, all machines share a common pseudorandom permutation list of IP addresses.

The attackers have three categories of methods to propagate their malicious codes, the attack tools, such as *Central source propagation*: In this mechanism, after the discovery of the vulnerable system that will become one of the zombies, instructions are given to a central source so that a copy of the attack toolkit is transferred from a central location to the newly compromised system; *Back-chaining propagation*: In this mechanism, the attack toolkit is transferred to the newly compromised system from the attacker; *Autonomous propagation:* In this mechanism, the attacking host transfers the attack toolkit to the newly compromised system at the exact moment that it breaks into that system.

There are two categories of DDoS attacks, typical DDoS attack and DRDoS attacks. In a typical DDoS attack, the army of the attacker consists of *master zombies* and *slave zombies*. The hosts of both categories are compromised machines that have arisen during the scanning process and are infected by malicious code. The attacker coordinates and orders master zombies and they, in turn, coordinate and trigger slave zombies. More specifically, the attacker sends an attack command to master zombies and activates all attack processes on those machines, which are in hibernation, waiting for the appropriate command to wake up and start attacking. Then,

master zombies, through those processes, send attack commands to slave zombies, ordering them to mount a DDoS attack against the victim. In that way, the agent machines (slave zombies) begin to send a large volume of packets to the victim, flooding its system with useless load and exhausting its resources. Unlike typical DDoS attacks, in DRDoS attacks the army of the attacker consists of master zombies, slave zombies, and reflectors [10]. The scenario of this type of attack is the same as that of typical DDoS attacks up to a specific stage. The attackers have control over master zombies, which, in turn, have control over slave zombies. The difference in this type of attack is that slave zombies are led by master zombies to send a stream of packets with the victim's IP address as the source IP address to other uninfected machines (known as *reflectors*), exhorting these machines to connect with the victim. Then the reflectors send the victim a greater volume of traffic, as a reply to its exhortation for the opening of a new connection, because they believe that the victim was the host that asked for it.

The defense DDoS attacks is a catch-me-if-you-can game. From the beginning, all legitimate users have tried to respond against these threats. University communities and software corporations have proposed several methods against the DDoS threat. Despite the efforts, the solution remains a dream. The attackers manage to discover other weaknesses of the protocols and—what is worse—they exploit the defense mechanisms in order to develop attacks. They discover methods to overcome these mechanisms or they exploit them to generate false alarms and to cause catastrophic consequences. The basic discrimination is between *preventive* [5][16]and *reactive* [18] defense mechanisms.

B. The preliminaries of Information Theory

In this section, we summarize the information theory concepts and theorems that will be used in our later exploration. We first review the concepts of entropy, conditional entropy, mutual information and entropy rate, which will be used to answer the question raised in Section 3. Then we introduce Fano's inequality [6], which will be used to answer the question of accuracy of DDoS detection.

1) Entropy, Conditional Entropy, Mutual Information:

Definition1: The entropy of a discrete random variable X is defined as

$$H(X) = -\sum_{x \in X} \Pr[X = x] \log \Pr[X = x]$$

The entropy of a random variable X measures the uncertainty of X, in the unit of bits.

Definition 2: The conditional entropy of a random variable X conditioned on another random variable Y is defined as

$$H(X \mid Y) = -\sum_{x \in X} \sum_{y \in Y} (\Pr[X = x, Y = y] \cdot \log \Pr[X = x \mid Y = y])$$

The concept of conditional entropy arises when we are interested in estimating the value of X, which can not be observed directly, using the observation of a related random variable Y. The conditional entropy, $H(X \mid Y)$, measures how much uncertainty remains for X given our observation of Y.

Definition 3: The mutual information $I(X;Y)$ is defined as $H(X) - H(X \mid Y)$.

Note that before the observation of Y, the uncertainty of X is $H(X)$. After the observation, this uncertainty goes down to $H(X \mid Y)$. Therefore, the mutual information measures the amount of information we learn about X from observing Y.

Definition 4: The entropy rate is the growth rate for a random process. For a stationary stochastic process $\{X_i\}$, there are two quantities of entropy rate, given as follows

$$H(\chi) = \lim_{n \to \infty} \frac{1}{n} H(X_1, X_2, ..., X_n)$$

$$H'(\chi) = \lim_{n \to \infty} H(X_n \mid X_{n-1}, X_{n-2}, ..., X_1)$$

and $H(\chi) = H'(\chi)$. If $\{X_i\}$ is a stationary Markov chain with stationary distribution μ and transition matrix P, and let $X_1 \sim \mu$. Then the entropy rate is $H(\chi) = -\sum_{ij} \mu_i P_{ij} \log P_{ij}$ [6]

2) Fano's inequality:

In our analysis, we would like to estimate the value of X (the attack packages) based on the observation of Y, which includes the real attack packages and the normal legitimate accessing packages. Recall that the conditional entropy $H(X \mid Y)$ measures how much uncertainty remains for X given our observation of Y. Intuitively, the smaller this conditional entropy value is, the more accurate the estimation that can be made is. This intuition is captured by Fano's inequality [6].

Suppose, given an observation of Y, our estimation of X is $\hat{X}$. We denote P_e as the probability that this estimation is incorrect, e.g. $P_e = \Pr[\hat{X} \neq X]$. Fano's inequality [6] states the following.

$$\mathrm{H}(P_e) + P_e \log_2 | \chi - 1 | \geq H(X \mid Y)$$

Here, $\mathrm{H}(P_e)$ is the "overloaded" to stand for the entropy of the indicator random variable $1_{\{\hat{X} \neq X\}}$.

3. System Modeling and Analysis

In a community network, we can manage and configure our routers, therefore, the routers can cooperate with each other to detect the possible attacks in the early stage. Figure 1 indicates a possible attack tree in a network attack.

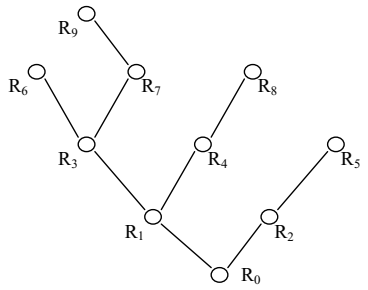

Figure 1. An attack tree in a community network

In this case, R_6, R_9, R_8 and R_5 are the routers at the edge of the community network, we call them edge routers, and R_0 is the router which connected with the victim. In this paper, we use this tree topology to explain our detection ideas and methods. We suppose the attack packages enter the community network via all the edge routers and attack follows will merge at the joint routers, e.g. router R_3 and R_1.

We have a few assumptions in order to make the discussion clear and easy to understand.

- The attackers use the same function to generate attack packages at the zombies, and it is stationary stochastic process.
- There is only one server in the community network is under attacking or massive accessing at a time.
- The researched system is linear and stable (this is quite reasonable for a short time period).

Detection algorithms are running on the routers of the community network. The edge routers monitor the network traffic by router entropy, which is the randomness of the flows at the router. In the non-attack case, the router entropy stays in a stable range. When there is an attack or a surge accessing, the router entropy drops dramatically, because there is one flow dominating the routers. In this case, the edge routers treat the dominant flow as a DDoS attack suspect, and start to calculate the entropy rate of the suspected flow, at the same time, edge routers notice their downstream router to calculate the entropy rate of the suspected flow. If the entropy rates on the routers are the same or very similar, e.g. the difference is less than a given value, then the suspected flow will be confirmed as a DDoS attack, otherwise, it is a surge of legitimate accessing.

Suppose in an attack scenario, the attacker uses a random variable X to control the generation speed of the attack packages. For example: using a constant speed to generate the packages, namely, $P\{X = C\} = 1$, and C is a constant; Increasing the number of attack packages according to attack time t, $X = a \cdot t + b$, a and b are constants; simulating the network accessing as Poison process, namely, $P\{X = k\} = \frac{\lambda^k e^{-\lambda}}{k!}, k = 0,1,...$ and λ is a constant; and so on.

We monitor the traffic on each edge router. Similar to paper [7], we focus on the *flow*, which is the packages who share the same destination address at a router.

We use a random variable X to represent the rand process of the flows on a router during a time slot. Let $X = \{x_1, x_2, ..., x_n\}$, and $x_i, i = 1,2,...n$ denotes the number of packages of different destinations during the time slot, respectively. We use the frequency of each flow to represent the possibility of that flow. Then we have

$$P(x_i) = \frac{x_i}{\sum_{j=1}^{n} x_j}, \quad i = 1,2,...n \qquad \text{……..…..(1)}$$

We use $H_f(X)$ to represent the entropy of a random variable X. According to [6], we then obtain:

$$H_f(X) = -\sum_{i=1}^{n} p(x_i) \log p(x_i) \quad \text{…….. (2)}$$

Generally speaking, $H_f(X)$ is stable with minor variations. In a DDoS attack scenario, the frequency of the flow which targets on the victim is extremely greater than the frequencies of the other flows, responsively. As a result, $H_f(X)$ decreases dramatically. However, this happens as well when there is a surging of legitimate accessing to one server, rather than a DDoS attack.

Based on router entropy $H_f(X)$, we can not identify the surging of legitimate accessing from DDoS attacks. We, therefore, need to find new features to solve this problem.

We suppose there are *n* zombies once the attack software is installed , and there is a "standard" zombie, which generation speed of the attack packages is exactly X, and X is a random variable. Because of the CPU difference and network delay difference to the victim, the real attack speed is different from each zombie. It is easy to obtain that it is a linear relationship of CPU speed ratio and network delay to the "standard" zombie as we suppose the network is linear and stable during a short time period. Then, we use X_i to represent the attack speed of zombie i

$$X_i = f(X) = a_i \cdot X + b_i, \quad a_i, b_i = C, \quad i = 1,2...,n \qquad \text{……(3)}$$

Theorem 1 For random variable X, and $Y = f(X)$, if $f(\cdot)$ is a linear function, then the entropy $H(X) = H(Y)$.

Proof:
Suppose X is a discrete variable, and $X \in \{x_1, x_2, ..., x_n\}$, then $Y \in \{f(x_1), f(x_2), ..., f(x_n)\}$. Because of the mapping is a one-to-one mapping, therefore, the possibilities of each pair in the two domains are the same, respectively.

$$p(x_1) = p(f(x_1)), p(x_2) = p(f(x_2)),..., p(x_n) = p(f(x_n))$$

Therefore,

$$H(Y) = -\sum_{i=1}^{n} p(y_i) \log\ p(y_i) = -\sum_{i=1}^{n} p(f(x_i)) \log\ p(f(x_i))$$
$$= -\sum_{i=1}^{n} p(x_i) \log p(x_i) = H(X) \cdots\cdots\cdots\cdots (4)$$

In the case of continuous random variable X, the proof is similar.

Theorem 1 shows clearly that the entropy of attack packages generation speed of each zombie in the army is the same, although the CPU and the network delay are different among the zombies.

Theorem 2 For random variable X, $Y_i = f_i(X), \quad i = 1,2,...n$, $f_i(\cdot)$ are n different linear functions, and $Y = \sum_{i=1}^{n} Y_i$, the entropy $H(X) = H(Y)$

Proof:

Because $Y_i = f_i(X), \quad i = 1,2,...n$ is a one-to-one mapping, we suppose $y_i = a_i \cdot x + b_i$, $\quad i = 1,2,...n$, a_i and b_i is given for each mapping, then

$$y = \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} (a_i \cdot x + b_i) = (\sum_{i=1}^{n} a_i) \cdot x + \sum_{i=1}^{n} b_i = a' \cdot x + b'$$

Hence, we can rewrite it as follow

$Y = f(X)$, $f(X)$ is a linear function for random variable X. Based on Theorem 1, the entropy $H(X) = H(Y)$, therefore, we obtain:

$$H(\sum_{i=1}^{n} X_i) = H(X_i) = H(X) \cdots\cdots\cdots (5)$$

Theorem 2 indicates that on the attack path, the intermediate routers aggregate multiple zombies' attack packages, the entropy of the attack packages pumping speed on the intermediate routers is the same.

Theorem 3 For a stationary stochastic process $\{X_i\}$, $Y = f(X)$, if $f(\cdot)$ is a linear function, then the entropy rates of the two random process are the same, namely,

$$\lim_{n\to\infty} \frac{1}{n} H(X_1, X_2,..., X_n) = \lim_{n\to\infty} \frac{1}{n} H(Y_1, Y_2,..., Y_n) \cdots\cdots (6)$$

Proof.

As a linear system, we have $P(x_1) = P(f(x_1) = P(y_1)$ then we have $H(X_1) = H(Y_1)$. Furthermore, for the stationary stochastic, then we have $P(x_2 \mid x_1) = P(y_2 \mid y_1)$, and then $H(X_2 \mid X_1) = H(Y_2 \mid Y_1)$.

$$H(X_1, X_2) = H(X_1) + H(X_2 \mid X_1)$$
$$= H(Y_1) + H(Y_2 \mid Y_1) = H(Y_1, Y_2)$$

Apply the chain rule for entropy[6],

$$H(X_1, X_2,..., X_n) = \sum_{i=1}^{n} H(X_i \mid X_{i-1},..., X_1)$$

We obtain

$$H(X_1, X_2,..., X_n) = H(Y_1, Y_2,..., Y_n)$$

Therefore, $\lim_{n\to\infty} \frac{1}{n} H(X_1, X_2,..., X_n) = \lim_{n\to\infty} \frac{1}{n} H(Y_1, Y_2,..., Y_n)$.

Theorem 3 showed that the entropy rates of attack rate at difference routers in the community network are the same.

Theorem 4. Given X is the random variable of the attack rate, and Y is the random variable of observation of packages to the victim (which includes legitimate packages and the attack packages). Let Pe be the possibility of incorrect estimation. Then

$$| \chi | \geq 2^{-H(P_e)} + 1$$

Where $| \chi |$ is the number of different samples of the observation.

Proof:

According to the definition of entropy, $\mathrm{H}(P_e) = -p_e \cdot \log_2 p_e - (1 - p_e) \cdot \log_2 (1 - p_e)$

In the case of attacking, the observation Y definitely includes the attack packages X, then $H(X \mid Y) = 0$. Combine with Fano's inequality, we have

$$\mathrm{H}(P_e) + P_e \log_2 | \chi - 1 | \geq 0$$

It is easy to obtain the following result

$$| \chi | \geq 2^{-H(P_e)} + 1 \ \cdots\cdots (7)$$

Theorem 4 expresses the samples that we need to guarantee the accurate of our detection.

Detection algorithm at routers of the community networks

1. for a time window, initiate the parameters
2. Counting the number of packages to different destinations.
3. calculate the router entropy $H_f(X)$ according to formula (1) and (2)
4. if $H_f(X)$ is less than a given threshold.
5. start calculating the entropy rate, $H(\chi) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2,..., X_n)$ for the suspected flow, and notice the downstream routers to calculate the entropy rate.
6. if the entropy rate are the same or the difference is less than a given threshold, then the attack is confirmed, and discard the attack packages.

Figure 2 DDoS detection algorithm based on information theory

4. Algorithms

Based on the analysis of the previous section, we have the corresponding algorithm, which includes two parts: calculate the router entropy and the suspected flow entropy rate. The algorithm is listed as Figure 2.

We expect to identify DDoS attacks as early as possible. The theorem 4 can be applied to compromise the accuracy of the detection and delay of confirming the attacks.

5. Conclusion and Future Work

In this paper, we focus on detection of DDoS attacks in community networks. Our motivation comes from discriminate the DDoS attacks from surge legitimate accessing, and identify attacks at the early stage, even before the attack packages reaching the target server. The entropy of flows at a router, router entropy, is calculated, if the router entropy is less than a given threshold, then a attack alarm is raised; the routers on the path of the suspected flow will calculate the entropy rate of the suspected flow. If the entropy rates are the same or the difference is less than a given value, then we can confirm that it is an attack, otherwise, it is a surge of legitimate accessing.

We have proven that combine the router entropy and the entropy rate of flows, we can discriminate DDoS attacks from surge legitimate accessing, moreover, we can identify attacks at the early stage. Extensive simulations are planed in the very near future to test our proposed methods. The comparisons of our algorithm and the existing detection algorithms will be conducted as well. The future work on the detection theory is listed as follows:

- The compromise of detection accuracy and the time of confirming is a critical aspect in the battle with DDoS attacks, how to obtain an optimal solution for this is quite demanded in practices;
- Identify DDoS attacks when there are quite number of the legitimate accessing to the same server. In this case, the entropy rate of the suspected flow is not the same. We can treat the attack packages as the 'signal', which we expect to indentify, and the legitimate accessing is treated as 'noise'
- Attackers may using multiple attack package generation functions in one attack, trying to fool our detection algorithm, e.g. they may use random functions with different seeds, however, we believe, there are definitely rules behind the attack distributions if they are 'man-made' rules. This is also a direction that we are going to explore with high interest.

Acknowledgement:
The authors would like to thank Australian Research Council for their support with the ARC Discovery grant (Project number DP0773264).

References

[1] P. Barford et al., "A Signal Analysis of Network Traffic Anomalies," Proc. ACM SIGCOMM Internet Measurement Workshop, ACM Press, 2002, pp. 71–82.
[2] R.B. Blazek et al., "A Novel Approach to Detection of 'Denial-of-Service' Attacks via Adaptive Sequential and Batch-Sequential Change-Point Detection Methods," Proc. IEEE Workshop Information Assurance and Security, IEEE CS Press, 2001, pp. 220–226.
[3] R.R. Brooks, Disruptive Security Technologies with Mobile Code and Peer-to-Peer Networks, CRC Press, 2005.
[4] G. Carl, G. Kesidis, R. Brooks, and S. Rai, "Denial-of-Service Attack Detection Techniques," IEEE Internet Computing, January 2006.
[5] http://www.cert.org/advisories/CA-1996-21.html
[6] Thomas M. Cover and Joy A. Thomas, Elements of Information Theory, second edition, 2007.
[7] Yu Chen and Kai Hwang, "Collaborative Change Detection of DDoS Attacks on Community and ISP Networks, the IEEE International Symposium on Collaborative Technologies and Systems (CTS 2006), 2006. pp401-410.
[8] L Feinstein et al., "Statistical Approaches to DDoS Attack Detection and Response," Proc. DARPA Information Survivability Conf. and Exposition, vol. 1, 2003, IEEE CS Press, pp. 303–314.
[9] L. Garber, Denial-of-Service attacks rip the Internet, IEEE Computer, 33(4): 12-17, April 2000.
[10] Steve Gibson, "Distributed Reflection Denial of Service Description and Analysis of a Potent, Increasingly Prevalent, and Worrisome Internet Attack," February 2002.
[11] J. Howard, "An Analysis of Security Incidents on the Internet," PhD thesis, Carnegie Mellon University, Aug., 1998.
[12] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial-of-Service Attacks: Characterization and Implications for CDNs and Web Sites," Proceedings of International World Wide Web Conferences, ACM Press, 2002.
[13] K. Kumar, R.C., Joshi, and K., Singh, "A Distributed Approach using Entropy to Detect DDoS Attacks in ISP Domain," the International Conference on Signal Processing, Communications and Networking, 2007. ICSCN '07. Feb. 2007 pp:331 – 337.
[14] R. Mahajan, S. Floyd and D. Wetherall, "Controlling high-bandwidth flows at the congested router," Proceeding of ACM 9th International Conference on Network Protocols, Nov. 2001.
[15] D. Moore, G.M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," Proc. Usenix Security Symp., Usenix Assoc., 2001.
[16] Charalampos Patrikakis, Thomas Kalamaris, Vaios Kakavas, "Performing Integrated System Tests Using Malicious Component Insertion," Electronic Notes in Theoretical Computer Science, Volume 82 No. 6 (2003).
[17] Charalampos Patrikakis, Michalis Masikos, and Olga Zouraraki, "Distributed Denial of Service Attacks," The Internet Protocol Journal, Volume 7, Issue 4, December, 2004.
[18] http://www.paypal.com/html/computerworld-011402.html
[19] Kevin Tsui, "Tutorial-Virus (Malicious Agents)," University of Calgary, October 2001.
[20] Haining Wang, Danlu Zhang and Kang G. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," IEEE Transactions on Dependable and Secure Computing, Vol. 1, NO. 4, October-December 2004. pp193-208.
[21] Nicholas Weaver, "Warhol Worms: The Potential for Very Fast Internet Plagues," http://www.iwar.org.uk/comsec/resources/worms/warhol-worm.htm. 2001.
[22] Nicholas Weaver, U.C. Berkeley BRASS group, "Potential Strategies for High Speed Active Worms: A Worst Case Analysis," February 2002.