FAST DETECTION OF DDOS ATTACKS USING NON-ADAPTIVE GROUP TESTING

Huynh Nguyen Chinh[1], Tan Hanh[2], and Nguyen Dinh Thuc[3]

[1]Faculty of Information Technology, University of Technical Education Ho Chi Minh City (UTE-HCMC), HCMC, Vietnam

[2]Faculty of Information Technology, Posts and Telecommunications Institute of Technology (PTIT), HCMC, Vietnam

[3]Faculty of Information Technology, University of Science (UoS), HCMC-VNU, Vietnam

ABSTRACT

Network security has become more important role today to personal users and organizations. Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks are serious problem in network. The major challenges in design of an efficient algorithm in data stream are one-pass over the input, poly-log space, poly-log update time and poly-log reporting time. In this paper, we use strongly explicit construction d-disjunct matrices in Non-adaptive group testing (NAGT) to adapt these requirements and propose a solution for fast detecting DoS and DDoS attacks based on NAGT approach.

KEYWORDS

Denial-of-service attack, ditributed denial-of-service attack, Group testing, Non-Adaptive Group testing, d-disjunct matrix.

1. INTRODUCTION

1.1 Denial-of-Service attacks

Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks have become a serious problem in network. In these attacks, attackers sent a very large number of packets to victims in a very short amount of time. They aim to make a service unavailable to legitimate clients. They are easily done for attackers to launch but are difficult for target users to defend [3]. Network detection and mitigation is necessary to mitigate such malicious attacks. Internet service providers (ISPs) can help customers defend against bandwidth attacks by deploying appropriate filtering rules at routers, or alternatively using routing mechanisms to filter packets to drop malicious packets.

Routers receive and process a lot of packets in network. Every packet has a destination IP address. If there are many packets passing through router which have the same IP destination, it may be a DoS attack.

Our solution aims to provide early warning and tracking DoS or DDoS attacks by collecting IP packets and finding Hot-IPs (hosts appear with high frequency in network and they also called hot

DOI : 10.5121/ijnsa.2013.5505

items, attackers or victims). In our solution, router acts as the sensor. When packet arrives at router, the IP header is extracted and put into groups. Based on the embedded source and destination IP addresses, the analysis is done. This method is much faster than one-by-one testing.

1.2 Group Testing

In the World War II, the millions of citizens of USA join the army. At that time, infectious diseases such as syphilis are serious problems. The cost for testing who was infected in turn was very expensive and it also took several times. They wanted to detect who was infected as fast as possible with the lowest cost. Robert Dorfman [6] proposed a solution to solve this problem. The main idea of this solution is getting N bloods samples from N citizens and combined groups of blood samples to test. It would help to detect infected soldiers as few tests as possible. This idea formed a new research field: Group testing.

Group testing is an applied mathematical theory applied in many different areas [8]. The goal of group testing is to identify the set of defective items in a large population of items using as few tests as possible. There are two types of group testing [11]: Adaptive group testing and non-adaptive group testing (NAGT). In adaptive group testing, later stages are designed depending on the test outcome of the earlier stages. In non-adaptive group testing, all tests must be specified without knowing the outcomes of the other tests. Many applications, such as data streams, require the NAGT, in which all tests are to be performed at once: the outcome of one test cannot be used to adaptively design another test. Therefore, in this paper, we only consider NAGT.

In data stream, it is very efficient way to detect hot items. Cormode and Muthukrishnan [4] set the scenario that millions of packets went through a router. They want to find the "hot items" (Hot-IPs) in data stream. To achieve this goal, they build a matrix $M_{t\times N}$ that could support up to $N = 2^{32}$ users and detect at most d hot items. After that, each j^{th} user (or each IP) was assigned to one column of this matrix (denoted M_j and it was unique). Router would store a vector counter $C_{t\times 1} = (c_1, c_2, ..., c_t)^T$, and if j^{th} user appeared, it increased $C = C + M_j$. They also set a threshold to convert the vector counter to Boolean vector. However, this scheme could not be used in their model because they did not have a strongly explicit construction and the time to decode this matrix was expensive $O(tN)$. Alternately, they used the techniques based on Group Testing and left these problems for the future works.

1.3 Related Works

In 2009, Khattab et al. [10] proposed system-based "live baiting" defend scheme by applying group testing theory to application DoS detection. They based on a "high probability d-disjunct" matrix.

In 2010, Ying Xuan et al. [1] presented Group Testing based approach deployed on backend servers. They use t virtual servers as testing pools and N clients, in which d clients are attackers. Consider the binary matrix $M_{t\times n}$, the clients can be mapped the columns and virtual servers into rows in M, where $m_{ij} = 1$ if and only if the requests from client j are distributed to virtual server i. With regard to the test outcome column V, they have $V[1] = 1$ if and only if virtual server i has received malicious requests from at least one attacker, but they cannot identify the attackers at once unless this virtual server is handling only one client. Otherwise, if $V[i] = 0$, all the clients

assigned to server i are legitimate. The d attackers can then be captured by decoding the test outcome vector V and the matrix M.

In 2013, Dayanandam et al. [2] combine the approach of Ying Xuan et al. [1] and password based scheme to defend DDoS attacks. One of the main limitations of these methods was not construct strongly explicit d-disjunct matrix. In 2010, Indyk, Ngo and Rudra proved that they can fast decoding group testing matrix. It means that we can apply this method to fast detecting Hot-IPs in network. They also proved that NAGT can adapt requirements for data stream algorithm: one-pass over the input, poly-log space (the matrix is not stored directly), poly-log update time and poly-log reporting time [5].

1.4 Paper Organization

We begin with some preliminaries in Section 2. In Section 3 gives the system setup. We describe our experimentation in Section 4. The last Section is conclusion.

Our Main Results

In this paper, we present a solution for fast detecting Hot-IPs in network using Non-adaptive group testing approach. We implement strongly explicit d-disjunct matrix in our experimentation.

2. PRELIMINARIES

2.1 Non-Adaptive Group Testing

The basic problem of NAGT can be described as follows. Given a population of N items which contains at most d "positives" items, we want to identify the positives as quickly as possible using t simultaneous "test". Each test is a subset of items, which returns "positive" if there's at least one positive item in the subset. We want to "decode" uniquely the set of positives given the results of the t simultaneous tests.

NAGT can be represented by a $t \times N$ binary matrix M, where the columns of the matrix correspond to items and the rows correspond to tests. In which, $m_{ij} = 1$ means that the j^{th} item belongs to the i^{th} test, and vice versa. We assume that we have at most d defective items. It is well known that if M is a d-disjunct matrix, we can show all at most d defectives.

Definition **(d-disjunct matrix)**.[11] *A binary matrix M with t rows and N columns is called d-disjunct matrix if and only if the union of any d columns does not contain any other column.*
Here is an example of d-disjunct matrix:

$$M_{9\times 7}=\begin{pmatrix}0&0&1&0&0&1&1\\0&0&1&0&1&0&0\\0&0&1&1&0&0&0\\0&1&0&0&0&1&0\\0&1&0&0&1&0&1\\0&1&0&1&0&0&0\\1&0&0&0&0&1&0\\1&0&0&0&1&0&0\\1&0&0&1&0&0&1\end{pmatrix}, d=2, N=7, t=9$$

There are two methods to construct d-disjunct matrices: random and non-random constructions. The advantage of non-random construction is that we can generate any column of matrix as we need. In this paper, we only consider the non-random construction of d-disjunct matrix. Non-random d-disjunct matrix is constructed by concatenated codes [9]. The codes concatenation between Reed-Solomon code and Identity matrix is represented below.

2.2 Reed Solomon and Concatenated Codes

- **Reed Solomon [7]**

For a message $\bar{\mathbf{m}}=(\bar{\mathbf{m}}_0,...,\bar{\mathbf{m}}_{k-1})\in \mathrm{F}_q^k$, let P be a polynomial

$$P_{\bar{\mathbf{m}}}(X)=\bar{\mathbf{m}}_0+\bar{\mathbf{m}}_1X+...+\bar{\mathbf{m}}_{k-1}X^{k-1}$$

In which the degree of $P_{\bar{\mathbf{m}}}(X)$ is at most k-1. RS code $[n,k]_q$ with $k\le n\le q$ is a mapping RS: $\mathrm{F}_q^k\rightarrow \mathrm{F}_q^n$ is defined as follows. Let $\{\alpha_1,...,\alpha_n\}$ be any n distinct members of F_q

$$RS(\bar{\mathbf{m}})=(P_{\bar{\mathbf{m}}}(\alpha_1),...,P_{\bar{\mathbf{m}}}(\alpha_n))$$

It is well known that any polynomial of degree at most $k-1$ over F_q has at most $k-1$ roots. For any $\bar{\mathbf{m}}\neq\bar{\mathbf{m}}'$, the Hamming distance between $RS(\bar{\mathbf{m}})$ and $RS(\bar{\mathbf{m}}')$ is at least $d=n-k+1$. Therefore, RS code is a $[n,k,n-k+1]_q$ code.

- **Code Concatenation [9]**

Let C_{out} is a $(n_1,k_1)_q$ code with $q=2^{k_2}$ is an outer code, and C_{in} be a $(n_2,k_2)_2$ binary code. Given n_1 arbitrary $(n_2,k_2)_2$ code, denoted by $C_{in}^1,...,C_{in}^{n_1}$. It means that $\forall i\in[n_1]$, C_{in}^i is a mapping from $\mathrm{F}_2^{k_2}$ to $\mathrm{F}_2^{n_2}$. A concatenation code $C=C_{out}\circ(C_{in}^1,...,C_{in}^{n_1})$ is a $(n_1n_2,k_1k_2)_2$ code defined as follows: given a message $\bar{\mathbf{m}}\in \mathrm{F}^{k_1k_2}=(\mathrm{F}^{k_2})^{k_1}$ and let $(x_1,...,x_{n_1})=C_{out}(\bar{\mathbf{m}})$, with $x_i\in \mathrm{F}_2^{k_2}$ then $C_{out}\circ(C_{in}^1,...,C_{in}^{n_1})(\bar{\mathbf{m}})=(C_{in}^1(x_1),...,C_{in}^{n_1}(x_{n_1}))$, in which C is constructed by replacing each symbol of C_{out} by a codeword in C_{in}.

Here is an example of a matrix constructed by concatenated codes.

$$C_{out}:\begin{bmatrix}0&1&2&0&1&2\\0&1&2&1&2&0\\0&1&2&2&0&1\end{bmatrix} \quad C_{in}:\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

$$C_{out}\circ C_{in}:\begin{bmatrix}1&0&0&1&0&0\\0&1&0&0&1&0\\0&0&1&0&0&1\\1&0&0&0&0&1\\0&1&0&1&0&0\\0&0&1&0&1&0\\1&0&0&0&1&0\\0&1&0&0&0&1\\0&0&1&1&0&0\end{bmatrix}$$

2.3 Algorithm and Analysis

We will look into the connection between group testing and hot items problem established by Cormode and Muthukrishnan [4]. Let consider a $t \times N$ d-disjunct matrix M, where N is the number of items and $t = O(d^2 \log^2 N)$. Note that we have a strongly explicit construction. Besides this, we maintain total number of input m. The initialization and update process as well as reporting problem are shown as following:

Initialization: $m = 0, C_i = 0,$ for $1 \le i \le t.$

Update: $m = m + 1, C_i = C_i + 1$ if and only if $m_{ij} = 1.$

Reporting hot items: given $C_i (1 \le i \le t)$ and $m,$ output all hot items indices $j.$

The problem of reporting hot items turns out to be the decoding problem of group testing. We briefly present two decoding algorithms as follows.

Given a sequence of m IPs from $[N]$, an item is considered "hot" if it occurs more than $m/(d+1)$ times. Note that there can be at most d hot items. Given the matrix $M_{t \times N} = (m_{ij})$ d-disjunct, $m_{ij} = 1$ if IP_j belong to group test i^{th}. Using counters $c_1, c_2, \ldots, c_t, (c_i \in [t])$, when an item $j \in [N]$ arrives, increment all of the counters c_i such that $m_{ij} = 1$. From these counters, we defined a result vector $r \in \{0,1\}^t$ as follows: Let $r_i = 1$ if $c_i > m/(d+1)$ and $r_i = 0$, otherwise. A test's outcome is positive if and only if it contains a hot item.

***Algorithm 1**:* naïve decoding

- *Input:* Given **M** be d-disjunct $t \times N$ matrix and $\mathbf{R} \in \{0,1\}^t$
- *Output:* Hot-IPs

With each r_i=0 do

for i=1 to N do

if (m_{ij})=1 Then

IP:=IP\{j}

Return IP, the set of remaining items

***Algorithm 2**:* Fast decoding

To construct an efficiently decodable group testing matrix, the main idea is to stack on top of one another a "filtering" matrix and an "identification" matrix. The filtering matrix is used to identify quickly a "small" set of candidate items which include all the positives. Then, the identification matrix is used to pinpoint precisely the positives.

Input : Given **M** be d-disjunct $t \times N$ matrix and $\mathbf{R} \in \{0,1\}^t$.

Output: Hot-IPs.

Step 1. Finding a set $S_i = \{ j \in \mathrm{F}_q \mid r_i(j) = 1 \}$ contains at most d "Hot-IPs".

Step 2. Using the naïve algorithm to find "Hot-IPs" based on this set.

Analysis of the Algorithm:

- *One-pass requirement*: we use non-adaptive group testing. Therefore, the algorithm for the hot items can be implemented in one pass. If adaptive group testing is used, the algorithm is no longer one pass.

- *Poly-log space requirement*: we can represent each counter in $O(\log n + \log m)$ bits. This means we need $O((\log n + \log m)t)$ bits to maintain the counters. With $t = O(d^2 \log^2 N)$ and $d = O(\log N)$, we need the total space to maintain the counters is $O(\log^4 N(\log N + \log m))$. The d-disjunct matrix is constructed by concatenated codes and we can generate any column as we need. Therefore, we do not need to store the matrix M.

- *Poly-log update time*: Since Reed-Solomon code is strongly explicit, the d-disjunct matrix is strongly explicit.

  We construct d-disjunct matrix M by concatenated codes $C^* = C_{out} \circ C_{in}$, where C_{out} is a $[q,k]_q$-RS code and C_{in} is an identify matrix I_q.

Recall that codewords of C^* are columns of the matrix M. The update problem is like an encoding, in which given an input message $\bar{\mathbf{m}} \in \mathrm{F}_q^k$ specifying which column we want (where $\bar{\mathbf{m}}$ is the representation of $j \in [N]$ when thought of as an element of F_q^k), the output is $C_{out}(\bar{\mathbf{m}})$ and it corresponds to the column $M_{\bar{\mathbf{m}}}$. Because C_{out} is a linear code, it can be done in $O(q^2 \times poly \log q)$ time, which means the update process can be done in $O(q^2 \times poly \log q)$ time. Since we have $t = q^2$, the update process can be finished with $O(t \times poly \log t)$ time.

- *Reporting time*: The naïve decoding algorithm (*algorithm 1*) on a d-disjunct matrix can identify all the positives in $O(tN)$ time. In 2010, P. Indyk, Hung Q. Ngo and Rudra [5] proved that the *algorithm 2* can be decoded in $poly(d) \cdot t \log^2 t + O(t^2)$.

3. SET UP

Router stores $[q-1, k]_q$-RS codes which defined in section 2. Router can construct d-disjunct matrix $M_{t \times N}$ ($N \le q^k$). For initialization, j^{th} IP will be identified by assigning a unique column M_j of M. Router stores a counter vector $C_{t \times 1} = 0_{t \times 1}$. If there has any packet coming to j^{th} IP, $C = C + M_j$. We assume that the total of the frequency of N items is m, and we have at most d "Hot" items. A value is common if its frequency is smaller than $\frac{m}{d+1}$. We use the method was proposed by Cormode and Muthukrishnan [4]. We will convert C into Boolean vector using the following rules:

- If $C(i) > \frac{m}{d+1}$, i^{th} test outcome is 1.
- If $C(i) \le \frac{m}{d+1}$, i^{th} test outcome is 0.

4. EXPERIMENTATION

We use a server (IBM X3650, CPU 3.0Ghz, RAM 4GB, OS CentOS) acting as the router in our system and some software at clients to generate any number of packets. We implement the algorithm in C program, using "pcap" library to capture packets through router. Each packet captured, the IP header is extracted. Based on the embedded destination addresses, the analysis is done.

We can generate d-disjunct matrices as define in Section 2 and support the number of hosts as much as we want. In our experiments, a matrix is generated from $[31,5]_{32}$-RS code and identity matrix I_{32} can support up to $32^5 = 33554432$ hosts, detect at most $d = \left\lfloor \frac{31-1}{5-1} \right\rfloor = 7$ where the code length is $t = 31 \times 32 = 992$.

We test many cases with different hosts sending packets at the same time and results are described in table 1 (we ignore time to capture packets and only count the time to decode

captured packets). We use 7 hosts acting as attackers and they perform DoS attacks to the network with a very large number of packets forwarded to router.

Table 1. Decoding time for Hot-IPs with $[31,5]_{32}$ - RS code

N IPs	Decoding time (sec)	N IPs	Decoding time (sec)
100,000	14.42	600,000	68.66
200,000	22.68	700,000	81.31
300,000	35.55	800,000	92.51
400,000	47.56	900,000	104.90
500,000	60.20	1,000,000	116.66

In another case, we construct d-disjunct matrices using some RS codes and results are described in Table 2.

Table 2. Decoding time for Hot-IPs with some RS codes

RS codes	d	Decoding time (sec)	N(IPs)
$[7,3]_8$	3	0.000	512
$[15,3]_{16}$	7	0.110	4096
$[31,3]_{32}$	15	3.650	32768
$[63,3]_{64}$	31	142.13	262144

5. CONCLUSION

Early detecting DoS and DDoS attacks in networks are the most important problem in order to mitigate risks on network. In this paper, we present the efficient solution of Non-Adaptive group testing method for fast detecting distributed denial-of-service attacks in network. Our future works are evaluating the solution at ISPs, implement the algorithm with the multi-processing, and combine with distributed model to lowering cost.

REFERENCES

[1] Ying Xuan, Incheol Shin, My T. Thai, Taieb Znati , *"Detecting Application Denial-of-Service Attacks: A Group-Testing-Based Approach"*, Parallel and Distributed Systems, IEEE Transactions on (Volume:21 , Issue: 8), 2010

[2] Dayanandam G., Rao T.V., Pavan Kumar Reddy S., and Revinuthala Sruthi, *"Password Based Scheme And Group Testing For Defending Ddos Attacks",* International Journal of Network Security & Its applications (IJNSA), Vol.5, No.3, May 2013.

[3] Tao Peng, Christopher Leckie, And Kotagiri Ramamohanarao, *"Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems",* ACM Computing Surveys, Vol. 39, No. 1, Article 3, Publication date: April 2007.

[4] Cormode, Graham, and S. Muthukrishnan, *"What's hot and what's not: tracking most frequent items dynamically",* In Proceedings of the twentysecond ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 296-306. ACM, 2003.

[5] Indyk P., Hung Q. Ngo, and Atri Rudra, *"Efficiently decodable nonadaptive group testing"*, In Proceedings of the Twenty-First Annual ACMSIAM Symposium on Discrete Algorithms (SODA'2010), New York, 2010, ACM, pp. 1126-1142.

[6] Robert Dorfman,*"The detection of defective members of large populations"*, The Annals of Mathematical Statistics (1943): 436-440.

[7] Reed I. and Solomon G., *"Polynomial codes over certain finite fields*", Journal of the Society for Industrial and Applied Mathematics, 8 (1960), pp. 300–304.

[8] Ngo Q. Hung, Ding-Zhu Du, *"A survey on combinatorial group testing algorithms with applications to DNA library screening",* Discrete mathematical problems with medical applications 55 (2000): 171-182.

[9] Forney Jr, G. David, *"Concatenated codes*", No.TR-440. Massachusetts Inst Of Tech Cambridge Research Lab Of Electronics, 1965.

[10] Khattab S., Gobriel S., Melhem R., and Mosse D., *"Live Baiting for Service-level DoS Attackers",* INFOCOM 2008.

[11] Du D.Z. and Hwang F. K., *"Combinatorial group testing and its applications, volume 12 of Series on Applied Mathematics*", World Scientic Publishing Co.Inc., River Edge, NJ, second edition, 2000.