Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis

Meghana B. Shetty[1] Krishnamoorthy K[2]
[1]M.Tech. Student (Computer Network Engineering) [2]Assistant Professor
[1,2]Department of Computer Science & Engineering
[1,2]Mangalore Institute of Technology & Engineering Moodbidri, Karnataka, India 575025

Abstract— The threat of cyber-attacks systems, such as web servers, databases servers and cloud computing, the most common and aggressive media, DoS can cause a serious impact on the computer systems. DoS attack detection system for a specific network traffic used by the geometric relationship between feature extraction network traffic in the characterization of MCA. The system detects DoS attacks, using anomaly recognition principle MCA. The solution can effectively detect known and unknown DoS attack, study legal network traffic patterns. The purpose of the triangular space technology is to improve and accelerate the MCA process. Effectiveness of the proposed tracking system is the use of KDD Cup 99 dataset and affects both non-standard, standard measures data of the proposed tracking system.

Key words: DOS (Denial of Service), KDD Cup 99 Dataset, MCA Process

I. INTRODUCTION

Network infrastructure should be protected from malevolent traffic, such as scanning, malicious code propagation, spam, and DDoS attacks. The problems caused from these are ranging from operational, political damage to companies, simple annoyance to severe financial, organizations, and critical infrastructure Network infrastructure should be protected from malevolent traffic, such as scanning, malicious code propagation, spam, and DDoS attacks. The problems caused from these are ranging from operational, political damage to companies, simple annoyance to severe financial, organizations, and critical infrastructure.

Mechanization, mainly with the complexity of zombie networks and its platform for attacks synthesis to launch significantly increased in the past year volume. Victims avoid malicious traffic is a daunting task coordination with different or additional components, including what technical solutions (application / network layer) non-technical demands. Screening is an important part of this effort. To prevent DDoS traffic, in order to meet with the current DDoS filtering be Internet service provider customers. Another identifies a performance of the malicious code and compromise before combining ISP proactive and block traffic to reach vulnerable hosts. In both cases, the basic process is what filter must be the network. ACL filtering function is used in the presence of the router.

TCAM can usually store ACL, but allows concurrent access, and reduce the number of searches for the transmitted packet. Although TCAM is expensive, it consumes more space and power. The number of TCAM size and cost reduction filter, which is no change in the situation. One cannot hope to save the ISP to block cookies attack or digital path instead of the filter. Currently control is sent to the victim host by an attacker to prevent the use of resources and services to legal be users (see Figure 1.1). The attack was blocked in the ISP's gateway router to protect customers. To stop the flow of traffic from the source, filter is installed.

Comparatively filters are lesser than attack sources, hence aggregation has been made use. This means that only one filter is used for the entire source address prefix. This process will increase the number of filters to trap all the traffic or even prevent legitimate traffic from blocking, reduce block prefix damage results, unexpected effect desired this loss is called "collateral damage." Filter can be seen as the source goal to prevent attacks and at least collateral damage can limit optimization problem with the limited number of filters. Quantitative studies have shown that malicious sources in space and time are based on prefix filter. It convey a general framework for studying source prefix filtering as a resource allocation problem. As far as our knowledge is concerned, the Ordinance so far, the focus has been more discussed about filtering protocols and architectural features. Within this structure, it resolves five practical source-address filtering problems, depending on the attack scenario and the operator's policy and constraints.

Two fold Contributions: 1.Theoretical –Filter selection optimization leads to novel variations of the multidimensional knapsack problem. The special structure of each problem, optimal design and computationally efficient algorithms are exploited. 2. Practical – Cost effective algorithms are provided which could be used both by operators to block undesired traffic and by routers to optimize the consumption of TCAM.

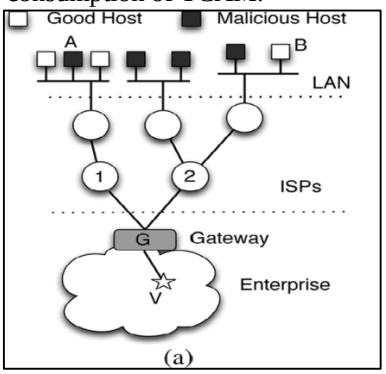

Fig. 1: Attacking example

II. LITERATURE SURVEY

Analysis of seeing Internet connection, a subset of the address space, it is estimated that the structural characteristics of the target IP address. These features can be used, such as routing and congestion control algorithms of the IP address to solve the global basic research. Addresses, by a Cantor fractal pink two parametric models. The model can simulate in your favorite real IP address is useful. Synthesis and characterization developed address structure, and totally responsible for the initiative, and ask prefix. Characterization of stability over short time scales in a given

area, the site has a few distinct characterizations, such as characterization is useful to see the fingerprints of traffic on a website. In addition, changes in traffic conditions, such as worms significant change spreading this fingerprint. In [1] S. Venkataraman, S. Sen, O. Spatscheck, P. Haffner, and D. Song, Symbolized a framework which performs an wide analysis of IP addresses, aggregates provided by 'network: aware' clusters in order to investigate properties that can distinguish the bulk of the legitimate user and malevolent. Also finds that the bulk of the spam comes from IP prefixes that send mostly spam and are persistent. First perform an extensive measurement study in order to understand some IP-based properties of legitimate users.

"Understanding spammer's network layer behavior," the researchers on the network level document spammer behavior, includes: sending spam most of the range of IP addresses, a common means of spam (eg, the road BGP hijacking, robot), insisted on every time the host is spam, botnets and spam characteristics. In the above answer these questions by analyzing 17 months in the Internet "sinkhole junk" more than 10 million spam gathered messages examine the basis of these data and the correlation with the results of research on the blacklist IP, TCP passive fingerprinting information, routing information, and "command and control" tracks botnets. And "found that most spam from the region sent a couple of IP address space, spammers seem short-lived and to be" robot ", just send e-mail pieces very short period of time. Finally, of the corresponding short BGP route hijacking is usually the IP address prefix received a small, but not insignificant, the number of spam. These trends suggests that the development of algorithms to determine on the basis of botnets, network-level properties (it is not quite variable email content) of the e-mail filter assembly, and to improve the infrastructure of the Internet routing security, you can be very effective in the fight against spam.

In [8] Fabio Soldo, KaterinaArgyraki, and Athina Markopoulou, Proposed Optimal filter malicious traffic sources. Consider the source of malicious network - based traffic filtering to prevent problem. In particular, filtered through the ACL. The router is now available, but they are a scarce resource because they are stored in the ternary content addressable memory (TCAM). Polymerization (prefix filter source, rather than a single IP address) can reduce the number of filters, but it also blocks legitimate traffic filter prefix costs. How to choose the best source prefix for various attack scenarios and realistic policy of carriers filter. In each scenario, calculate the optimization effectively, but is designed. Use logs from Dshield.org, evaluate and practically demonstrate significant benefits algorithm.

III. IMPLEMENTATION

In implementation part the following filter algorithms are implemented:

- BLOCK-ALL
- BLOCK-SOME
- TIME-VARYING BLOCK-ALL/SOME

A. Block-All

A network operative has a blacklist BL of size N, a whitelist WL, and a weight assigned to each address that indicates the amount of traffic originating from that address. Fmax is total number of available filters. The first practical area the operator may have is to install a set of filters that block all bad traffic so as to minimize the amount of good traffic that is blocked. Optimal algorithm is designed solve the problem at the lowest achievable complexity (linearly increasing with N).

1) Block All Algorithm (LCP Tree)

Block All algorithm using Longest Common Prefix (LCP) tree.

Given a set of IP addresses denoted as A, define the Longest Common Prefix tree of A, denoted by LCP-tree (A), as the binary tree with the following properties:

Each leaf represents a different address in A, and there is a leaf for each address in A.

Each intermediate (non leaf) node represents the longest common prefix between the prefixes represented by its two children.

The LCP-tree can be constructed from the complete binary tree by removing the branches that do not have addresses in A, and then by removing nodes with a single child. Each node in the LCP tree represents a prefix that can be blocked, hence can represent a filtering solution as the pruned version of the LCP tree, whose leaves are all and only the blocked prefixes.

The below Fig 2 shows the example of LCP tree. In this tree the numbers are denoted as 0/29, 4/30, 8/30 etc. Here the first number represents the IP address and second number represents the level of tree. Here define Whitelist (WL) which contains good nodes and Blacklist (BL) which contain malicious nodes. In Fig 2 the shaded circle represents the malicious node and white filled circle represents the good nodes.

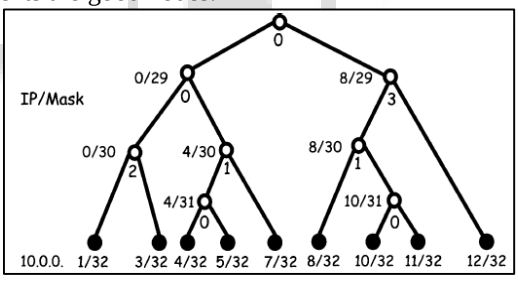

Fig. 2: LCP tree

Given a blacklist BL, a whitelist WL, and the number of available filters Fmax, select filters that block all bad traffic and minimize collateral damage.

To reduce the numbers of filters required have to reduce the number of malicious nodes by aggregation. Formulate this problem by making two adjustments to the general framework. First, (1) becomes (5), which expresses the goal to minimize the collateral damage. Second, (3) becomes (7), which enforces the constraint that every bad address should be blocked by exactly one filter, as opposed to at most one filter in (3).

Hence it is proved that, for each feasible solution to BLOCK-ALL, there exists another feasible solution 'S' that:

Can be represented as a pruned sub tree of LCP-tree and

Whose collateral damage is smaller or equal to S's.

BLOCK-ALL algorithm consists of two steps. First, build the LCP tree from the input blacklist BL. Second, in a bottom–up fashion, compute the minimum

collateral damage needed to block all bad addresses in the sub tree of prefix p using at most F filters. Following a dynamic programming (DP) approach, can find the optimal allocation of filters in the sub tree rooted at prefix p by finding a value n and assigning F-n filters to the left sub tree and n to the right sub tree, so as to minimize collateral damage. The fact that BLOCK-ALL needs to filter all bad addresses (leaves in the LCP tree) implies that at least one filter must be assigned to the left and right sub tree. It returns an optimal solution and required set of filters.

1: build LCP-tree($\mathcal{BL}$)
2: **for** all leaf nodes leaf **do**
3: $z_{\text{leaf}}(F) = 0\ \forall F \in [1, F_{\max}]$
4: $X_{\text{leaf}}(F) = \{\text{leaf}\}\ \forall F \in [1, F_{\max}]$
5: **end for**
6: level = level(leaf) − 1
7: **while** level ≥ level(root) **do**
8: **for** all node p such that level(p) == level **do**
9: $z_p(1) = g_p$
10: $X_p(1) = \{p\}$
11: $z_p(F) = \min_{n=1,\ldots,F-1}\{z_{s_l}(F-n)+z_{s_r}(n)\}\forall F \in [2, F_{\max}]$
12: $X_p(F) = X_{s_l}(F-n) \cup X_{s_r}(n) \forall F \in [2, F_{\max}]$
13: **end for**
14: level = level − 1
15: **end while**
16: **return** $z_{\text{root}}(F_{\max}), X_{\text{root}}(F_{\max})$

B. *Block-Some*

A blacklist and a whitelist are as assumed before, but the operative is now willing to block only some, instead of all, bad traffic, so as to reduce the amount of good traffic blocked at the expense of leaving some bad traffic freed. The goal now is to block only those prefixes that have the highest impact and do not contain sources that create a lot of good traffic, so as to minimalize the total cost. Optimal, lowest-complexity (linearly increasing with N) algorithm is designed to solve the problem.

1) Block Some Algorithm

Given a blacklist BL, a whitelist WL, and the number of available filters Fmax, the goal is to select filters so as to minimize the total cost of the attack.

There are two differences from BLOCK-ALL. First, the goal is to minimize the total cost of the attack, which involves both collateral damage $g_{p/l}$ and the filtering benefit $b_{p/l}$ $g_{p/l}$.

Second, states that every bad address must be filtered by at most one prefix, which means that it may or may not be filtered.

The algorithm that solves BLOCK-SOME is similar to BLOCK-ALL algorithm, in that it relies on the LCP tree and a dynamic programming (DP) approach. The main difference is that not all bad addresses need to be filtered. Hence, at each step, can assign n=0 filters to the left and/or right sub tree, So for Block some can replace line number 11 in Algorithm 1, n=1,2……F-1 by n=1,2…F.

C. *Time-Varying Block-All/Some*

Bad addresses may change over time. New sources may send malicious traffic and, conversely, previously active sources may disappear (e.g., when their vulnerabilities are patched). One way to solve the dynamic versions of BLOCK-ALL (SOME) is to run the algorithms. Static versions for the blacklist/whitelist pair is proposed at each time slot. However, given that subsequent blacklists typically exhibit significant overlap, it may be more efficient to exploit this temporal correlation and incrementally update the filtering rules. It is showcased that it is possible to update the optimal solution, as new IPs are inserted in or removed from the blacklist, in log N time.

1) Time Variant Algorithm

a) Inputs

1) A blacklist BL and whitelist WL, and
2) The number of available filters Fmax.
3) The corresponding solution to BLOCK-ALL (SOME), denoted St.
4) Another blacklist BL1 and whitelist WL1, obtain the solution to BLOCK-ALL (SOME) for the second blacklist/whitelist, denoted St1.

Consider that the whitelist remains the same and assume the focus is on the changes in the blacklist.

b) Addition:

First, consider that the two blacklists differ only in a single new bad address, which does not appear in BL, but appears in BL1. There are two cases, depending on whether the new bad address belongs to a prefix that is already filtered in St. If it is, no further action is needed, and St=St1. Otherwise, modify the LCP tree that represents St to also include the new bad address.

c) Deletion:

Then, assume that two blacklists differ in one deleted bad address, which appears in BL but not in BL1. In this case, it modifies the LCP tree that represents St to remove the leaf node that corresponds to that address as well as its parent node (since that node does not have two children anymore), and recomputed the optimal filter allocation for all and only the node's ancestors.

The below Fig 3 shows the example for block-all and time varying block-all.

Consider a blacklist of 10 bad IP addresses as shown in Fig 5.2,

BL = {10.0.0.3, 10.0.0.10, 10.0.0.15, 10.0.0.17, 10.0.0.22, 10.0.0.31, 10.0.0.32, 10.0.0.33, 10.0.0.57, 10.0.0.58}

The table next to each node p shows the minimum cost Zp(F) (Refer Appendix) computed by the Dynamic programming (DP) algorithm for Block-All for F=1…. number of leaves in sub tree. The optimal solution to BLOCK-ALL consists of the four prefixes highlighted in black. When a new address, e.g. 10.0.0.37 is added to the blacklist, a leaf node is added to the tree and TIME-VARYING needs to update all and only the ancestor nodes in LCP-tree (BL), indicated by the dashed lines. Moreover, a new node is created to denote the longest common prefix between 10.0.0.37 and 10.0.0.32 (or 10.0.0.33). Note that all other nodes corresponding to the longest common prefixes between 10.0.0.37 and other addresses in BL are already in the LCP tree.

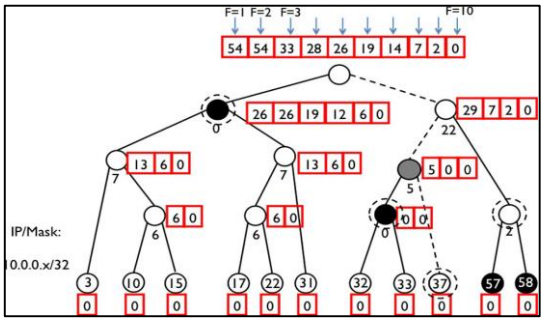

Fig. 3: Example for block-all and time varying block-all.

The new optimal solution consists of the four prefixes indicated by the dashed circles. Here F denotes the Filters. F=1, 2, 3…. There are the numbers of Filters that are assigned.

D. Block-All Versus Block-Some:

There is an interesting connection between the two problems. The latter can be regarded as an automatic way to select the best subset from BL and run BLOCK-ALL only on that subset. If the absolute value of weights of bad addresses is significantly larger than the weights of the good addresses, then BLOCK-SOME degenerates to BLOCK-ALL.

IV. CONCLUSION

Implements a framework for optimal source prefix-based filtering. The framework is rooted at the theory of the knapsack problem and provides a novel extension. Within it, formulate five practical problems, presented in increasing order of difficulty. For each problem, designed optimal algorithms that are also low-complexity (linear or pseudo-polynomial in the input size). Algorithms over Dshield.org logs and demonstrate that they bring significant benefit compared to non-optimized filter selection or to generic clustering algorithms. A key insight behind that benefit is that our algorithms exploit the spatial and temporal clustering exhibited by sources of malicious traffic.

ACKNOWLEDGMENT

Dedicating the paper work to our esteemed guide, Mr. Krishnamoorthy K, whose interest and guidance helped in the completion of work successfully. And also extending the gratitude to Mr. Krishnamoorthy K, (Assistant Professor, Computer Engineering Department) who has facilitated exploring the subject with more enthusiasm.

REFERENCES

[1] S. Venkataraman, S. Sen, O. Spatscheck, P. Haffner, and D. Song, "Exploiting network structure for proactive spam mitigation," presented at the USENIX Security Symp., Boston, MA, and Aug. 2007.
[2] G. Pack, J. Yoon, E. Collins, and C. Estan, "On filtering of DDoS attacks based on source address prefixes," presented at the SecureComm, Istanbul, Turkey, and Sep. 2006.
[3] E. Kohler, J. Li, V. Paxson, and S. Shenker, "Observed structure of addresses in IP traffic," in Proc. ACM SIGCOMM Workshop Internet Meas., Marseille, France, pp. 253–266, Nov. 2002.
[4] Z. Chen, C. Ji, and P. Barford, "Spatial-temporal characteristics of internet malicious sources," in Proc. IEEE INFOCOM Mini-Conf.,Phoenix, AZ,pp. 2306–2314,May 2008.
[5] R. Mahajan, S. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Schenker, "Controlling high bandwidth aggregates in the network," Comput.Commun. Rev., vol. 32, pp. 62–73, Jul. 2002.
[6] A. Ramachandran and N. Feamster, "Understanding the network-level behavior of spammers," in Proc. ACM SIGCOMM, Pisa, Italy, pp. 291–302,Sep. 2006.
[7] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How dynamic are IP addresses?," in Proc. ACM SIGCOMM, Kyoto, Japan, pp. 301–312, Aug. 2007.
[8] Fabio Soldo, Student Member, IEEE, KaterinaArgyraki, and AthinaMarkopoulou, Member, IEEE "Optimal Source-Based Filtering of Malicious Traffic".
[9] "Understanding ACL on Catalyst 6500 series switches," Cisco Systems, San Jose, CA, 2003 [Online]. Available: http://www.cisco.com/en/US/products/hw/switches/ps708/products_white_paper09186a00800c9470.shtml
[10] "Dshield: Cooperative network security community—Internet security," Dshield.org [Online]. Available: http://www.dshield.org
[11] H. Kellerer, U. Pferschy, and D. Pisinger, Knapsack Problems. New York: Springer-Verlag, 2004.
[12] G. Varghese, Network Algorithmics: An Interdisciplinary Approach to Designing Fast Networked Devices. San Mateo, CA: Morgan Kaufmann, 2005.
[13] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[14] T. Kanungo, D. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 24, no. 7, pp.881–892, Jul. 2002.
[15] "Protecting your core: Infrastructure protection access control lists," Cisco Systems, San Jose, CA, 2008[Online].Available:http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml

