Volume: 05, Issue: 04, Pages:1996-2001 (2014) ISSN : 0975-0290

Optimization of Malicious Traffic in Optimal Source Based Filtering

P.MOHANRAJ
Department of Computer Science M S College of Commerce & Science
Bharathiar University, Coimbatore-641 046
Email:mohawondrous@gmail.com
C.N. MANOKARAN
Asst. Professor Department of Computer Science M S College of Commerce & Science
Bharathiar University, Coimbatore-641 046
Email:cnmanokaran@gmail.com
M.DHARAMALIGAM
Asst. Professor Department of Computer Science,
Nandha Arts and science College. Bharathiar University, Coimbatore-641 046
Email:emdharma@rediffmail.com

----------**ABSTRACT**----------

Traffic and spam are the main problems in the data transmission through the network. Many traffic filtering systems have been proposed to find and filter the traffic over the network. The system Optimal Source Filtering (OSF) has implemented a new and optimal filtering mechanism. The new mechanism named as DROP, which monitors and filters the spam and malicious traffic over a network effectively. Traffic filtering systems have been proposed to detect the spammer and malicious traffic, using the optimal rules and policies.

Further these systems are highly ineffective when they encounter malicious traffic. The proposed system introduced OSF protocol, which helps to improve the efficiency of the firewall and filters based on the user rule. The proposed filtering scheme provides TFS false filtering when the flash crowd occurred. The protocol verifies users and firewall rules and policies with the data priority model, which makes the filtering process more robust and fastest manner.

The Proposed spam detection project identifies and eliminates unwanted messages by monitoring outgoing messages. The spam detection is the main challenging task in the network. In the existing system spam detection has implemented after the data received. According to the user rule and request the current system identifies the spam and zombies by monitoring every outgoing message from the sender

Keywords- **Traffic Filtering Systems, OSF, DROP,DB Algorithm.**

Date of Submission: February 22, 2014 Date of Acceptance: April 08, 2014

I. INTRODUCTION

Traffic filtering is a method used to enhance network security by filtering network traffic based on many types of criteria. Several filters and rules have been stored to filter unwanted contents. Active Internet Traffic Filtering systems are the mechanisms for filtering highly distributed environmental attacks such as DOS (Denial of Service) and DDOS (Distributed Denial of Services). Several traffic filtering techniques has been proposed to block a million undesired flows and data's. Traffic filters are also prevents abuse by malicious nodes seeking to disrupt other nodes' communications.

Packet filtering is a method of enhancing network security by examining network packets as they pass through routers or a firewall and determining whether to pass them on or what else to do with them. Packets may be filtered based on their protocol, sending or receiving port, sending or receiving IP address, or the value of some status bits in the packet. There are two types of packet filtering. One is static and the other is dynamic. Dynamic is more flexible and secure as stated below. Does not track the state of network packets and does not know whether a packet is the first, a middle packet or the last packet. It does not know if the traffic is associated with a response to a request or is the start of a request. Tracks the state of connections to tell if someone is trying to fool the firewall or router. Dynamic filtering is especially important when UDP traffic is allowed to be passed. It can tell if traffic is associated with a response or request. This type of filtering is much more secure than static packet filtering. In source routing; packets contain header information describing the route they are to take to the destination. Source routing is a security concern when

Volume: 05, Issue: 04, Pages:1996-2001 (2014) ISSN : 0975-0290

an attacker may gain access to a network that has access to yours without going through your firewall.

Source routing should be disabled on network routers, especially at the network perimeters. Hackers may be able to break through other friendly but less secure networks and get access to your network using this method. The world is becoming more interconnected with the advent of the Internet and new networking technology. There is a large amount of personal, commercial, military, and government information in networking infrastructures worldwide. Network security is becoming of great importance because Computer networks are very important and ever present technology, even though the networking has more security issues. Yet the increased complexity of computer networks combined with the cleverness of attacker's means that they remain vulnerable to expensive attacks from worms, viruses, Trojans, and other malicious software, which we simply refer to as malware. Network traffic filtering is one of many security methods available to network administrators. Network traffic filters provide protection by sampling packets or sessions and either comparing their contents to known malware signatures or looking for anomalies likely to be malware. Filtering capabilities have begun to be integrated into routers themselves, so as to reduce hardware deployment costs and to allow for more adaptive security Future traffic filters are expected to be configurable, networked, and even autonomous. Our objective in this paper is to investigate the deployment and configuration issues of such devices within an optimization framework

II. PROBLEM DEFINITION

Given a policy comprising a set of rules, the goal is to discover and eliminate troublesome rules and find an ordered list of consistent ones while performing the minimum number of comparisons. Formally, the ANOMALY- detection and filter selection problem is defined as follows.

Firewall policy management is a challenging task due to the complexity and interdependency of policy rules. This is further exacerbated by the continuous evolution of network and system environments. It's also related to knapsack problems. Filter selection belongs to the family of multidimensional knapsack problems. The general KP problem is well-known to be NP-hard. The most relevant variation is the knapsack with cardinality constraint. This study the practical problem of distributed filtering against a flooding attack. The proposed system proves that the problem can be decomposed into several FLOODING problems, which can be solved in a distributed way. Studying filter selection as a resource allocation problem. There are different filters for different events. It deals filter selection optimization leads to novel variations of the multidimensional knapsack problem, malicious traffic finding and unwanted message filtering dynamically, existing system does not Finds and protects a trusted network from an un-trusted network ,Solutions are hardware based. And Time and cost was so high.

The Traffic filtering systems proposed earlier for detecting the malicious traffic and spammer, deducts the attack path and the data over the system but fail to preclude data loss. Several filters have used to filter the malicious traffic. But filter selection generated many delay and communication overhead. The data are transmitted in the form of packets from sender to the receiver, at the same time as transmission packet loss occurs and this leaves the system more vulnerable. This allows the spammer to hack the data through packet loss with ease. When the system encounters encrypted traffic, these systems become highly ineffectual. In existing system the filtering systems have implemented on hardware's. So implementation cost was too high. Access control lists (ACLs) can selectively block traffic based on fields of the IP header. Filters (ACLs) are already available in the routers today but are a scarce resource because they are stored in the expensive ternary content addressable memory (TCAM).

III. RELATED WORK

Computer networks have become an ubiquitous but vulnerable aspect of corporate, university, and government life. Yet the increased complexity of computer networks combined with the ingenuity of attackers means that they remain susceptible to expensive attacks from worms, viruses, Trojans, and other malicious software, which we simply refer to as *malware* [1],[2]. Network traffic filtering is one of many security methods available to network administrators. Network traffic filters provide protection by sampling packets or sessions and either comparing their contents to known malware signatures or looking for anomalies likely to be malware. Filtering capabilities have begun to be integrated into routers themselves, so as to reduce hardware deployment costs and to allow for more adaptive security Future traffic filters are expected to be configurable, networked, and even autonomous. The objective in this paper is to investigate the deployment and configuration issues of such devices within an optimization framework.

A related and more studied area of research is network monitor placement for traffic measurement. In this paper we make use of the framework introduced by Cantieni et. al. The monitor placement problem. In the mentioned paper, the authors set up various optimization problems using the sum of the squared relative errors of traffic flow sizes as the convex objective function for minimization problems involving constraints on sampling rates and capacity. Another relevant paper on the monitor placement problem takes a similar approach, but uses more sophisticated cost models involving discrete variables indicating where

Volume: 05, Issue: 04, Pages:1996-2001 (2014) ISSN : 0975-0290

monitors will be placed .The same paper also considers constraints requiring that some minimum benefit be provided while a cost metric is minimized. While the malware filter placement problem has not been studied using an optimization framework similar to those discussed above, it has been analyzed from a game theoretical perspective. Kodialam and Lakshman [3] consider the most difficult filter placement scenario where the attacker has complete awareness of the network topology and can choose the path that malignant traffic will take. A Markov game between an attacker and an intrusion detection system (IDS) is considered. The attacker selects nodes to attack from and nodes to target while the IDS chooses links on which to deploy traffic filters. Yet another approach to the malware filter placement problem is currently being pursued by researchers at Ben- Gurion University in Israel. This approach involves centrality measures, which originated in social network analysis. Recent developments allow for these measures to be calculated quickly [4].Filtering capabilities are already available at routers today via access control lists (ACLs). ACLs enable a router to match a packet header against predefined rules and take predefined actions on the matching packets and they are currently used for enforcing a variety of policies, including infrastructure protection . For the purpose of blocking malicious traffic, a filter is a simple ACL rule that denies access to a source IP address or prefix. To keep up with the high forwarding rates of modern routers, filtering is implemented in hardware: ACLs are typically stored in ternary content addressable memory (TCAM), which allows for parallel access and reduces the number of lookups per forwarded packet. However, TCAM is more expensive and consumes more space and power than conventional memory. The size and cost of TCAM puts a limit on the number of filters, with thousands or tens of thousands of filters per path, an ISP alone cannot hope to block the currently witnessed attacks, not to mention attacks from multimillion-node botnets expected in the near future. An attacker commands a large number of compromised hosts to send traffic to a victim (say a Web server), thus exhausting the resources of and preventing it from serving its legitimate clients. The ISP of tries to protect its client by blocking the attack at the gateway router . Ideally, should install one separate filter to block traffic from each attack source. However, there are typically fewer filters than attack sources, hence aggregation is used, i.e., a single filter (ACL) is used to block an entire source address prefix. This has the desired effect of reducing the number of filters necessary to block all attack traffic, but also the undesired effect of blocking legitimate traffic originating from the blocked prefixes (we will call the damage that results from blocking legitimate traffic “collateral damage”). Therefore, filter selection can be viewed as an optimization problem that tries to block as many attack sources with as little collateral damage as possible, given a limited number of filters. Furthermore, several measurement studies have demonstrated that malicious sources exhibit temporal and spatial clustering [5] a feature that can be exploited by prefix-based filtering.

IV .PROPOSED MODEL

The presents about the proposed system. Basic concepts of protocols and filtering technique and (Data Blocking) algorithm (DROP protocol) are discussed.

Protecting a victim (host or network) from malicious traffic is a hard problem that requires the coordination of several complementary components, including nontechnical and technical solutions (at the application and/or network level). Several mechanisms have been proposed. So implementing firewall and access control rules are very tedious because the network has so many vulnerabilities and security issues. The proposed system introduces a new protocol which is named as DROP (Decentralized Rule Optimized Protocol). The decentralized approach provides effective rule matching and verification process in the network while data transmission. Access Control List has also applied in order to maintain black and white list of users and nodes for effective data restriction. The importance of the DROP protocol is facilitating a solution against filter selection problem.

Data blocking algorithm with spam detection system

1: An outgoing message arrives at spam monitor locale
2: Get IP address of sending machine m
3: Get the rules of receiver machine R
4:Let n be the message outbox of machine m
5: Read every policy and match the message
6: If the data match with the policies of users and firewall then do step 7
7: Let Xn = 1 if message is spam, Xn = 0 otherwise
8: if (Xn == 1) then
7: measure the total blocked messages and store Cs
9: if (CS>threshold) then block the user
10: else non spam
11: end if
12: Check Black and white list which denoted as BL and WL respectively
13: if ($m==BL_m$) then
14: Machine m is blacklisted. data terminates for m.
15: else if ($m==WL_m$) and Xn = 0 then
16: Machine m is normal. Test is reset for m.
17: else if ($m==WL_m$) and Xn = 1 then
18: Test continues with new observations
19: else
20: end if

Testing of the firewall rules verifies whether the security policy is correctly implemented by a set of firewall rules and user rules. A security policy is a document that sets the basic mandatory rules and principles on information security. Such a document should be specified in every transaction. The firewall

Volume: 05, Issue: 04, Pages:1996-2001 (2014) ISSN : 0975-0290

rules are intended to implement the directives defined in the security policy. The idea is to translate the security policy into firewall rules (or vice versa)

V. IMPLEMENTATION

This phase is to design and implement a program that runs the rules and policies by analyzing the corresponding data packets and information about the traffic. If the reaction of the traffic filter fits the expectation, the firewall behaves as suggested, otherwise it report the irregularities.

There are five basic actions of the program have to perform:

1. Generation. Build the rules and policies.

2. Transmission. Perform the data transmission.

3. Capture. Capture the packets and traffic information

4. Analysis. Detect uncommon events (packets that should be blocked are passed through the firewall or vice versa).

5. Logging. Log the irregularities. Maintain the reports

This section sheds light on the implementation of the malicious traffic detection tool. This describes the control flow, the different functional modules and discusses the structure of the source code. The implementation puts into practice what theoretically designed in the previous section. This also means to face reality and adapt the model as it hits unforeseen problems. Some difficulties in the implementation phase are illustrated and the solutions to overcome the problems are presented.

The thesis has used C#.Net for developing the front end of this software and SQL Server for the back end. The reason for using C#.Net is its flexibility. This can add or remove any features without editing the whole code. This separated the standalone functions like port matching and IP address matching in separate functions which are reused again and again. For the back end this needed a freely distributed and powerful database so SQL Server was a good choice. Whole of the rule list is stored in the database. All fields except the Rule No. are stored as the Strings. They are accessed and parsed according to the use, edited if necessary and stored again in the String form.

VI.RESULTS AND DISCUSSION

In Traffic Filtering system consider two performance metrics: latency and throughput. Latency is the time between the existing of a request at a server and the completion of the request. Throughput is the number of requests completed per second. The latency and throughput results of the three models are in a 16-node application server. The results are measured as the parameter of the verification for the incoming requests and data.

Since routing systems are much faster than the domain name service in detecting failures and responding to it, network and server failures have only temporary impact on the any cast based server location and load distribution scheme. Additionally, it has no single point of failure or bottleneck as is the case for a connection router. The deficiency of an any cast based scheme, as compared to a connection router, is that it cannot distribute load based on precise information. Achieving these performance benefit in the domain of server Traffic Filtering concept is not a small task, even the load has increased the performance will be effectively.

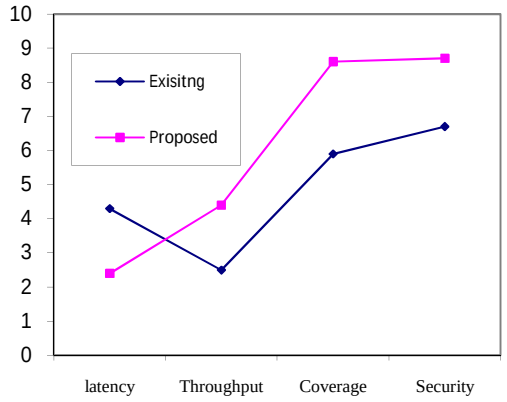

FIG:1 Traffic Filtering

The performance impact of Traffic Filtering can be measured in four key areas:

A.Latency,B.Throughput,C.Coverage, D.Security

The above figure describes the performance comparison between the existing approaches such as optimal source filtering and DROP protocol with the proposed system. That result shows the effectiveness of the proposed system by using three parameters such as latency, throughput and security. The following indicates the detailed results of the proposed system performance.

Volume: 05, Issue: 04, Pages:1996-2001 (2014) ISSN : 0975-0290

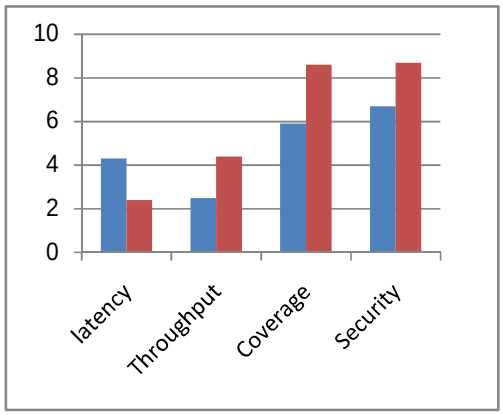

Figure2: performance comparison

A.LATENCY:

In practice, hosts are added to a Network Traffic Filtering cluster in proportion to the request rate as the client load increases. When this is the case, the server may respond later. This will affect the client. This system propose to minimize the latency when the client requesting a file. This can be done by Traffic Filtering scheme which regulates user request and makes the prompt response. Fig2. a shows the average request latency, throughput, coverage and security measurements with the EXISITNG, proposed DROP, and other Traffic filtering modals. EXISITNG shows the worst performance since subsequent requests from a client are not likely to be forwarded to the same server that caches the previous session information of the client. EXISITNG cannot yield good performance.

B.THROUGHPUT:

Throughput is the average rate of successful message delivery over a communication channel. Network throughput is the sum of the data rates that are delivered to all terminals in a network. Throughput to clients, which increases with additional client traffic that the cluster can handle prior to saturating the cluster hosts (higher is better).

Network Traffic Filtering simultaneously delivers incoming packets to all cluster hosts and applies a filtering algorithm that discards packets on all but the desired host. Filtering imposes less overhead on packet delivery than re-routing, which results in lower response time and higher overall throughput. Network Traffic Filtering scales performance by increasing throughput and minimizing response time to clients. When the capacity of a cluster host is reached, it cannot deliver additional throughput, and response time grows non-linearly as clients awaiting service encounter queuing delays. Adding another cluster host enables throughput to continue to climb and reduces queuing delays, which minimizes response time. As customer demand for throughput continues to increase, more hosts are added until the network's subnet becomes saturated. At that point, throughput can be further scaled by using multiple Network Traffic Filtering clusters and distributing traffic to them using Round Robin DNS.

C.COVERAGE:

Dealing the client requests efficiently even the serer load capacity exceeds is more important for every Traffic Filtering scheme. But in the existing proposals existing and ssl_session methods are considering only a limited set of client request. This makes the performance better than the other two schemes.

D.SECURITY:

Sharing the files in the network makes every file available in the sub server. So that the sub server can respond to their clients more effectively. But the security issues may create problems by using sub servers. Preventing those files from the security threads is more important, in this system the files are shared and stored after the encryption, so that security is high than the existing schemes.

The performance of other policies is similar to each other. The efficiency never exceeds 50% of the average load and is below 30% in most cases. Although the number of active connections is a good measure of server load, the amount of data transfer is a more appropriate metric for network load.

The proposed system model shows the performance advantages between the existing system models. The result defines the impact and efficiency of the proposed system. The above topics discussed with the consideration of comparison where the followings are the evaluation of the proposed technique.

From the above results it can observe that EXISITNG shows the worst performance since subsequent requests from a client are not likely to be forwarded to the same server that caches the previous session information of the client. Thus, CPU cycles are wasted to re-authenticate and negotiate keys between a client and a server. The results of EXISITNG show that the techniques of filter selection setup procedure are the main bottleneck in application servers.

Like the latency result, the throughput of existing filter selection is much lower compared to the proposed DROP models. The DROP model also yields a better throughput compared to the existing system as the load increases.

VI.CONCLUSION

We are proposed a optimal filter and algorithm which can detect the malicious traffic and spam before transmission which made on the network against the security issues through the help of DROP concept for enhancing filtering malicious data transmission

Volume: 05, Issue: 04, Pages:1996-2001 (2014) ISSN : 0975-0290

security. This application was developed in such a way that it can detect spammers, malicious nodes and reports ,then and in the mean while was made to produce appropriate notifications along with the log of the system. The ultimate enhancement of the project was the impact of customized Rule based filtering with client side in order to bring down the data transmission load of the network. The system has also managed to address the false alarm during the period of flash crowd by proper monitoring of the attacks.

REFERENCES

[1] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security & Privacy Magazine*, vol. 1, pp. 33–39, 2003.

[2] D. Moore, C. Shannon, and K. Claffy, "Code-red: a case study on the spread and victims of an internet worm," in *Proc. of ACM SIGCOMM Workshop on Internet measurement*, Marseille, France, pp. 273–284,2002.

[3] M. Kodialam and T. V. Lakshman, "Detecting network intrusions via sampling: A game theoretic approach," in *Proc. of 22nd IEEE Infocom*, vol. 3, San Fransisco, CA, USA, pp. 1880–1889,2003.

[4] U. Brandes, "A faster algorithm for betweenness centrality," *Journal of Mathematical Sociology*, vol. 25, no. 2, pp. 163–177, 2001.

[5] M. Collins, T. Shimeall, S. Faber, J. Janies, R.Weaver,M.De Shon, and J. Kadane, "Using uncleanliness to predict future botnet addresses," in *Proc. ACM Internet Meas. Conf.*, San Diego, CA, , pp. 93–104,2007.

.