International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:5, 2014

Techniques Used in String Matching for Network Security

Jamuna Bhandari

***Abstract*—**String matching also known as pattern matching is one of primary concept for network security. In this area the effectiveness and efficiency of string matching algorithms is important for applications in network security such as network intrusion detection, virus detection, signature matching and web content filtering system. This paper presents brief review on some of string matching techniques used for network security.

***Keywords*—**Filtering, honeypot, network telescope, pattern, string, signature.

I. Introduction

ALONG with the rapid development of network technology, demands for anti-attack and security protection are now facing a drastic increase in almost all network applications and systems. Hence network security arises as a big issue to be discussed. String matching is a widely used concept for network security. Whenever we talk about malicious attacks, suspected information or some keywords or signatures passing over various nodes of the networks, we need to match them or search them for many security reasons. This matching process is done with the help of various string matching or pattern matching algorithms. Widely deployed network intrusion detection and prevention systems often use signature-based method to detect possible malicious attacks. String matching has recently proven useful for deep packet inspection to detect intrusions in networks, scan for virus's protection, and refine internet content. Although string matching algorithms are used for many other purposes but this paper focuses only for string matching techniques for network security. With the help string matching algorithms, various types of attacks have been detected.

Various algorithms have been designed [1]-[21] and done hardware implementation to speed up the inspection, lower pattern space complexity, and perform operations efficiently.

The paper is organized as follows; Section II discussed various string matching techniques for network security. Section III discussed areas of network security with string matching process. Finally conclude in Section IV with conclusion and future work.

Jamuna Bhandari is with the Manipal Universiy Jaipur, Rajsathan, India (Phone: 09829931897; e-mail: bunu17_bhandari@yahoo.com).

II. String Matching Techniques Used in Network Security

There are many techniques which have been designed so far for network security; this paper discussed four most commonly used techniques along with their limitations.

A. Filtering Technique

A filtering technique is used for internet security purpose. Using filtering techniques, one can block various applications which you do not want to perform. This technique applies string matching process and neglects the regions of text T in which maximum matches cannot occur, and apply dynamic programming computation to the remaining portions of text T. [2] Dynamic programming computation creates the rigid separation between the filtering phase and checking phase as shown in Fig. 1. Dynamic programing computation improves the filtering process and rapid up the security by merging the filtering and the checking phase. It measures the statically derived filter information during the checking phase, strengthening it by information determined dynamically.

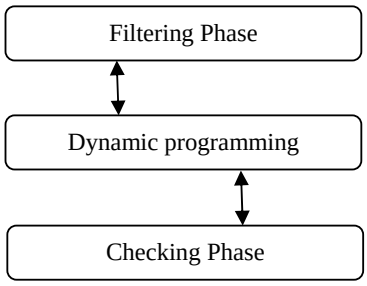

Fig. 1 Merging of filtering phase and checking phase via dynamic programming computation

B. Honeypots Techniques

Honeypots are exciting technology with many potential for the network security community. The concepts were first introduced by Cliff Stoll [5]. It creates a virtual machine, which makes attackers and is a highly flexible tool; this tool of honeypots can work from detecting encrypted attacks in IPv6 networks to capturing the latest on-line credit card fraud and other online transactions. Honeypot worked as an information system resource whose value lies in unauthorized or outlaw use of the resources.

Honey pots collect information in small amounts instead of logging a GB of data per day; they can log only one MB of data a day. Instead of yielding thousands alerts a day, they can generate only 10 alerts per day [6]. This makes matching algorithms for honeypots to work effectively. Another good

advantage of honeypots is that it is only capture bad activity; any interaction with a honeypot is most likely unauthorized or malicious activity. Honeypots cut down 'noise' by collecting only small sets of data, but information of high value. Most security technologies (such as IDS systems) honeypots work fine in encrypted or IPv6 environments. It does not matter what the hackers thrown in honeypot, it will just detect and capture it. Honeypots techniques are very simple. It does not need to have developed any fancy algorithms, any state tables to keep up, or any signatures to update. This simpler technology makes it less likely happen for mistakes or any misconfiguration since data is collected in small scale.

The limitation or drawbacks of honeypots are it does not solve a specific problem like firewalls or Intrusion detection systems. Instead, it used as tool that are of many different shapes and sizes. It can only track and capture activity that has interacted honeypot directly. It will not capture attacks against other connected systems, until and unless the attacker or threat is interacts also with the honeypots.

Honeypots has the risk for firewalls of being penetrated, information encryption has the risk of being broken, Sensor of Intrusion Detection System have the risk of failing to detect attacks are some of main risk of honeypots. Other important risk of honeypots is of being taken over by the hackers and being used to harm other systems. This risk varies for different honeypots types, depending on the type of honeypot; it can have no more risk than an IDS sensor.

C.Figures

Network telescopes techniques: A network telescope is also known as darknet and is an unused address range [7]. It neither offers Internet services nor does it uses. A network telescope is a part of routed IP address space in which little or no existence of legitimate traffic. Using matching process it monitors an unexpected traffic arriving at a network telescope provides the opportunity to view remote network security events such as various forms of attacks, like internet worms caused infection of host, and network scanning [7] and this is done via matching algorithms.

Browsing a data on the Internet becomes first-rate task now a day. It is either impractical or impossible to collect data in enough locations to build a global view of this dynamic, unstructured system. A network telescope [8] has emerged as the dominant mechanism for measuring internet security process such as denial-of-service (DoS) attacks and network worms. Traditional network telescopes deduce remote network behaviour and events in an entirely passive way by examining spurious traffic arriving for non-existent hosts at a third-party network.

Beside these advantages a network telescope has some limitations such as – in most cases; it cannot track "reflector" DoS attacks for the reason that they cause systems to respond to the target. Another issue is the size of telescope matters a lot. Bigger the telescopes better the security. Smaller telescopes monitor smaller set of addresses lead to underestimate the peak intensity of an attack and detect it later than a bigger telescope. Moore[8] suggest that home and small-office users on DSL (digital subscriber line) and cable modem connections played a big role in spreading Code Red and are the targets of many DoS attacks. In addition, many of the systems infected and inadvertently helped to spread Code Red and Code Red 2 were on DSL and cable modem accounts. Home users and most small businesses don't have full-time network administrators to update software and take other steps to keep up security.

D.Signature Matching Techniques

Network devices are increasingly employing deep packet inspection to enable advanced services such as intrusion detection, traffic shaping, and quality of service [9]. Signature (keyword) matching is an important part of deep packet inspection and involves matching pre-supplied signatures to network payloads at line rates. It is widely observed that signature matching is the most processing-intensive part and the chokepoint to increased performance. [10] Introduce an efficient data structure called Extended Bloom Filter (EBF) and the corresponding string matching algorithm to perform the multi-pattern signature matching. They also present a technique to support long signature matching so that only need to maintain a limited number of supported signature lengths for the EBFs. Fig. 2 shows the improve algorithm [9], when the number of signatures increased the absolute saving becomes bigger.

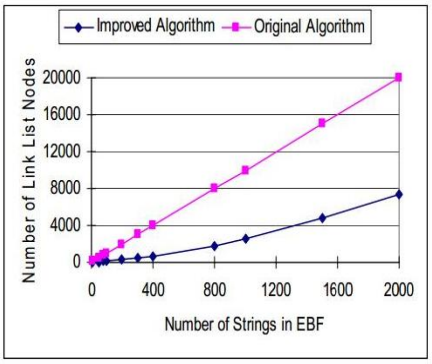

Fig. 2 Memory saving with improve algorithm

Signature based Network Intrusion Detection System collects these signatures and scans the payload of the internet packets supplied for them to find the contained such malicious behaviors [10]. An efficient and fast solution needed to adopt for the most prominent signature set as per today's situation and to sustain the real-time processing of the high-speed network.

Like in pattern P and text T in exact string matching, here suppose given packet payload T of length n and a set of signatures of variable length for intrusion detection. The concept behind the signature-matching problem is to figure any exact match of signature S[i] and a substring of T. In Network security, signature matching is an important part and decides the overall performance of connected systems. While the network band width and the size of the signature set keep

growing, to do real-time detection which is still away from real world. Some of best known algorithms adopt for matching is Boyer-Moore algorithms, which scan for a signature patterns with various observations. Another well know algorithm is Aho-Corasick, this work with the finite state automaton supporting multi-pattern string matching. The major drawback noticed is that its excessive consumption of memory. A modified algorithm of Aho-Corasick [11] reduces the amount of memory and improves its performance. Third best algorithm is Wu-Manber [12]; this uses a hash table and the bad character heuristics to accelerate the searching speed.

All these algorithms designed for software implementation. The various study of experimental results shows no such algorithm is fast enough for real-time string matching in high-speed network.

Nevertheless, the limitation drawn to signature matching techniques is if any change to the attack may lead to missed events generates false negatives results, also if signature patterns are not unique, than results may lead to high false-positive rates. It is also limited to inspect only single packet and does not apply well to the stream-based nature of network traffic such as HTTP traffic [9], [10].

Apart from above mentioned four techniques the approaches to detect the deep packet inspections are automata based, heuristic based and simple filtering based. Automata approach matches the patterns by state transition of deterministic finite automata (DFA) or non-deterministic finite automata (NFA). It has linear execution time and also consumes more memory if the data structure is not compressed.

Heuristic approach checks the block of characters in the window, it moves to the next position if the match is not found.

Filtering approach searches text for necessary patterns and drops if the content does not contain the pattern. But it has been noticed that Heuristic and Filtering approaches are memory efficient but suffers in worst case.

III. Areas in Network Security with String Matching Process

This section discussed about the major areas in the field of network security which used string matching process and its mechanism.

For the low-cost hardware-based intrusion detection systems [13], [16], it proposes a memory-efficient parallel string matching plan. Long target patterns divided into sub patterns with a fixed length; deterministic finite automata build with the sub patterns. Using the pattern dividing, the variety of target pattern lengths may relieve, so that memory usage in string matches becomes proficient. Two-stage sequential matching method proposed for the consecutive matches with sub patterns in order to find each original long pattern being divided. Investigational results show that total memory requirements decrease on average by 47.8 percent and 62.8 percent [13].

Traffic volumes of Internet are growing constantly; string matching using the Deterministic Finite Automaton will be the performance bottleneck of Deep Packet Inspection [14]. The recently proposed bit-split string matching algorithm suffers from the unnecessary state transitions problem, limiting the efficiency of deep packet inspection of network security. The cause behind the fact that each tiny DFA of the bit-split algorithm only processes a k-bit substring of each character input, but cannot verify whether the entire character belongs to the set of original alphabet of signature rules [15], [17] proposes a byte-filtered string matching algorithm, where bloom filters used to pre-process each byte of every incoming packet payload to check whether the input byte belongs to the derived set of alphabet or not, before processing bit-split string matching. The experimental results show that compared to the bit-split algorithm, [10] byte-filtered algorithm enormously decreases the time of string matching as well as the number of state transitions of tiny DFA on both synthetic and real signature rule sets. A Memory Efficient Multiple Pattern Matching Architecture for Network Security has been proposed by Tian Song, Wei Zhang [22]. In this effort, a pattern matching architecture for tens of thousands of signatures is proposed. The first idea is to use an algorithm based on a novel model, namely cached DFA (CDFA), to express the pattern set more efficiently.

The second idea, next state addressing (NSA), is to store transition rules of finite automata using less memory. It is achieved by taking states as addresses and employing feature of the state acting as the next state in DFA or CDFA. These two ideas both increase the memory efficiency. Moreover, the architecture for multiple pattern matching is given with some optimizations for reducing critical path and the memory utilization (refer [21] for more details).

IV. Conclusion

The applications of string matching are widely useful in many areas based on string matching for network security. There are many scopes for research in activist assault through cyber, medicinal skills such as biological analyses, huge area of online and offline data transfer, library sciences are already rolling in many directions, different types of anti-viruses are endlessly promoted based on their effectual and quicker exposure nature. So many more areas covered by research on such matching concepts. There are a large amount of variety and interesting research are still needs to adopted by researcher to extend the applications on string matching.

Most important factor of string matching is its application and is not limited, its requirements and improvements being done often. The study shows there are many scopes and areas are covered by pattern matching techniques in the field of network security. Much work has been done, many are available and many more are yet to be projected. This is a very immeasurable area with plenty of future scale and plenty of work left for researcher. The various application are emerging very swiftly, the dispute for network security becomes very essential, so numerous algorithms and techniques are likely to be continuous in research.

REFERENCES

[1] G. O. Young, "Synthetic structure of industrial plastics (Book style with paper title and editor)," in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
[2] Young H. Cho and William H. Mangione-Smith. A Pattern Matching Co-processor for Network Security. Anaheim, California, USA. 2005 ACM 1-59593-058-2/05/0006, 2005
[3] Robert Giegerich, Frank Hischk. A General Technique to Improve Filter algorithms for Approximate String Matching. Fourth South American Workshop on String, 1997
[4] Le Zhang, Jingbo Zhu, Tianshun Yao. An Evaluation of Statistical Spam Filtering Techniques. ACM Transactions on Asian Language Information Processing, Vol. 3, No. 4, , Pages 243-269, 2004
[5] Ernest Romanofski. A Comparison of Packet Filtering Vs Application Level Fire wall Technology. Global Information Assurance Certification Paper SANS institute, 2002
[6] Kyi Lin LinKyaw. Hybrid Honeypot System for Network Security. World Academy of Science, Engineering and Technology, 2008
[7] Honeypot Design for Intrusion Detection, American U. of Beirut, 2004
[8] Jean-Pierre van Riel, Barry Irwin. InetVis, a Visual Tool for Network Telescope Traffic Analysis. AFRIGRAPH 2006, Cape Town, South Africa, ACM 1-59593-288-7/06/0001, 2006
[9] David Moore et al. Network Telescopes: Technical Report support for this work is provided by NSF Trusted Computing Grant CCR-0311690, Cisco Systems University Research Program, DARPA National Institute of Standards Grant 60NANB1D0118, and a generous gift from AT&T, 2004
[10] Haoyu Song, John W. Lockwood. Multi-pattern Signature Matching for Hardware Network Intrusion Detection Systems. 0-7803-9415-1/05. IEEE, 2005
[11] R. Smith, S.Jha 'XFA: Faster Signature Matching with Extended Automata' IEEE Symposium on Security and Privacy (Oakland), May 2008
[12] A. Aho and M. J. Corasick. Efficient string matching: An aid to bibliographic search. Communications of the ACM, 1975
[13] U. Manber and S. Wu, " A fast algorithm for multi-pattern searching, "in T ech.Report TR-94-17, CS Dept., University of Arizona, 1994
[14] HyunJin Kim 'A Memory-Efficient Bit Split Parallel String Matching Using Pattern Dividing for Intrusion Detection Systems' IEEE, vol. 13, no. 12, 2009
[15] Prasad. R, Agarwal. S, 'A new parameterized string matching algorithm by combining bit-parallelism and suffix automata' IEEE International Conference on Computer and Information Technology, 2008
[16] Kun Huang, Dafang Zhang 'A Byte-Filtered String Matching Algorithm for Fast Deep Packet Inspection' IEEE 9TH Young Computer Scientists, 2008
[17] Tian Song, Wei Zhang, Dongsheng Wang, YiboXue. A Memory Efficient Multiple Pattern Matching Architecture for Network Security. This work is supported by National Natural Science Foundation of China (No.60673145). Processing, V alparaiso, Chile,1997
[18] Jiong Zhang and Mohammad Zulkernine. A Hybrid Network Intrusion Detection Technique Using Random Forests. First International Conference on Availability, Reliability and Security (ARES'06) 0-7695-2567- IEEE, 2006
[19] Tian Song, YiboXue and Dongsheng Wang, 'An algorithm of large-scale approximate multiple string matching for network security', 1-4 244-0463-0/06/IEEE, 2006
[20] R. Sidhu and V. K. Prasanna. Fast regualr expression matching using fpgas. In FCCM, 2001
[21] J. Moscola, J. Lockwood, R. P. Loui, and M. Pachos. Implementation of a content-scanning module for an Internet firewall. In FCCM, 2003
[22] X Tian Song, Wei Zhang et al, A Memory Efficient Multiple Pattern Matching Architecture for Network Security, This work is supported by National Natural Science Foundation of China (No.60673145)

Jamuna Bhandari (M'2012–Student Member IEEE, M'2014- IAENG) born on 1st January 1986, Sikkim, India. Pursuing Full-Time PhD on computer science from Manipal University Jaipur, Rajasthan, Registered on Jan 2012. In 2011, completed Post-Graduation with the degree on Master of computer application from Manipal Institute of Technology, Sikkim, India. 2008, completed graduation degree in Bachelor of computer application from University of North Bengal, Siliguri, India. Area of interest is on algorithm designing for string matching techniques, Design and analysis of algorithms, Applications of string matching techniques in network security.

Currently, pursuing full time PhD and worked as a teaching assistant 2012-2014. There is no job experience since further education is continued after PG. Published one international and one national conference paper and four papers in international journal.

Jamuna Bhandari, IEEE Student member with membership number 92533695. Member of IAENG, non-profit international association for the engineers and computer scientists, with membership no. 142390.