Performance Enhancement of a Malware Detection System using Score Based Prioritization of Snort Rules

Pritpal Singh, Research Scholar, Deptt. of CSE, SBSSTC, Ferozepur, Punjab, India
Sunny Behal, Assistant Professor, Deptt. of CSE, SBSSTC, Ferozepur, Punjab, India
Krishan Kumar, Associate Professor, Deptt. of CSE, SBSSTC, Ferozepur, Punjab, India

Abstract—Snort is an open source Intrusion Detection System (IDS) that uses a rule-based approach to detect different kinds of malware, online attacks, vulnerabilities, etc. The performance of a Malware Detection System (MDS) deployed in a large network depends on the nature and type of rules stored in its database. As the number and type of attacks are increasing, more number of rules are appended in the MDS database. This increase in the size of rule database itself becomes the bottleneck in the performance of the MDS. This paper proposes a rule scoring based mechanism for prioritizing the snort rules so as to optimize the number of rules in the MDS database. Only those rules are retained in the database whose total score is greater than the computed threshold value. The results show that the performance of MDS has enhanced remarkably.

Index Terms—Anamoly, Intrusion, Signature, False Positive, Alert, Malware.

1 Introduction

The rapid development of the internet in the past few decades facilitates an increase in the incidents of online attacks [1]. Fig. 1 shows the most significant attacks experienced last year as reported by the Arbor Inc. From the report, it is seen that Distributed Denial of Service (DDoS) attacks against customers is the number one operational threat nowadays. Basically a DDoS attack is a malicious attempt from the multiple hosts to cause the victim, site, or node to deny service to its customers [2] and can be launched using a mechanism called botnet [3] through a network of controlled computers. Recently in March 2015, GitHub suffered the largest DDoS attack in the website's history till date with an estimated size of more than 400 Gbps [4]. According to Akamai's Q1 2015 "The state of the internet" security report, there has been 35 % increase in DDoS activities against customers as compared to 2014, which is double the number of attacks recorded a year ago [5]. It is clear from the report that the attacks on frequently accessed websites evidently increased, with government websites becoming a common target [6]. To address this rapid increase in attacks, the organizations around the world are turning to Malware Detection Systems (MDS) which provides better protection for the network of computers than fixed-rule firewalls [7]. The work done in paper [8], [9] proposes signature based detection of malware using snort, but no emphasis given on number of rules stored in MDS database. According to K.Tomar [10] "The MDS is installed in one or more nodes of the network, check each packet that is routed over this node, and also check the node for abnormal behaviour; like connection from disallowed networks, and unauthorized login." Each packet is classified whether it is malicious or not and the files, processes, and connections are observed for abnormal behaviour [11]. Network Intrusion Detection system (NIDS) [12] which consists of several sensors/tools can produce better log generation and alert analysis using signature based traffic analysis. There are many malware detetion engine's available which provide the facility to write rules like Bro [13], Suricata [14] and Snort [15] etc. In the proposed work we use snort as an MDS engine for which a rule scoring mechanism is proposed to compute the score of the rule. This is the first attempt to calculate score for rules present in MDS.

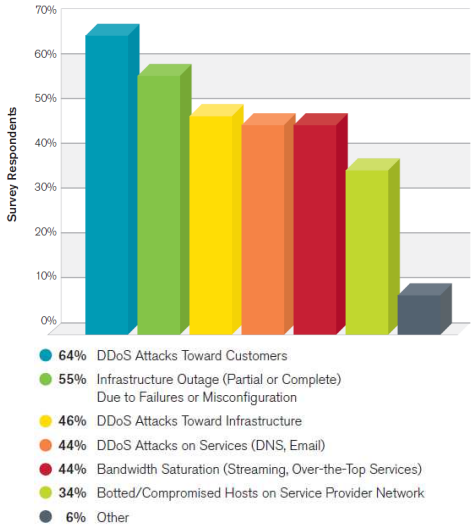


Fig. 1: Most Significant Malware

978-1-4673-7910-6/15/$31.00 ©2015 IEEE

The Major contributions of this paper are:

- The originality of the approach lies in applying the score based approach to optimize the number of rules in a snort based MDS.
- A score based method is proposed to calculate the score of a rule using identified parameters.
- Threshold value is calculated as the part of the work to decide whether the rule should be part of the MDS database or not.
- Based on the threshold value, the number of rules in MDS database are prioritized.

This paper explores how the performance of an MDS like Snort can be enhanced to detect different kinds of malware, online attacks, vulnerabilities, etc in the network in which it is deployed. The performance throughout the network largely depends on traffic monitoring i.e obtaining valuable information about the incoming traffic from the different parameters of the packet received on the network. Following is the list of parameters that can be used to examine network traffic:

- Source and Destination IP addresses;
- Different types of Protocols;
- Source and Destination port numbers;
- Packet size;
- Flags that indicate the priority associated with packet;
- Packet payload;

On the basis of these parameters, rules can be defined in MDS to meet the network demands.
The remainder of this paper is organized as follows. Section-2 presents the related work done in the performance enhancement of a malware detection system. In section-3, the proposed methodology of rule scoring mechanism is given. Section-4 describes the experimental topology setup and results obtained are discussed in section-5. Finally, section-6 concludes the work by highlighting the future work.

2 Related Work

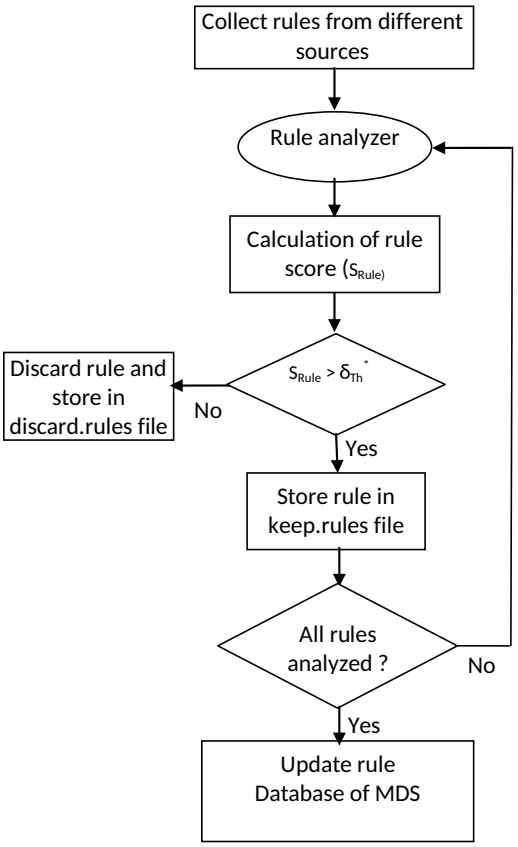
Detection of intrusions and anomalies are the two major issues in a computer network. A number of methods have been developed to classify intrusion detection system events into attack sequences [16], [17], [18], [19]. Snort is an open source [15] intrusion detection and prevention system and has been used extensively for the detection of different kinds of malware in a network. The proposed work uses rule scoring mechanism to compute the score of rules available in snort database. K. Zhao [20] uses the concept of dynamic adjustment algorithm to optimize the performance of MDS. The authors applies the algorithm to all the rules stored in MDS database. The efficiency of MDS can further be enhanced if the algorithm is applied to those rules that are appropriate to a particular network. SNORTRAN [21] is an optimizing compiler for snort rule sets that incorporates ideas of pattern matching compilation based on cost-optimized decision trees with set wise string search algorithms. Dmitry S. Kazachkin [22] propose a signature based optimization method for improving the performance of intrusion detection and prevention system which only reduces the processing time of the incoming packet on the network. A mechanism named WIND-Workload aware Intrusion Detection proposed in [23] improves the performance of snort by developing a high performance and memory efficient packet inspection strategy on the basis of different attack signatures. Similarly the work done in paper [24] discusses an approach to improve the performance of intrusion detection system through optimizing the order of the attack signature rules as well as the order of the rule fields. All of the work done in this field tries to improve the performance of MDS using different techniques.
As we all know, the performance of MDS largely depends on the number and type of rules present in it and may vary from network to network because of different network characteristics in terms of generated traffic. This means that the performance of MDS can be enhanced if the optimized set of rules according to the network demand are present in MDS database.

3 Proposed Methodology

The proposed methodology works on the basis to enhance the performance of a malware detection system. As we know performance of an MDS is very important issue for its efficient working. That's why there is a need to optimize the number of rules which helps in generating alerts every time a malicious activity takes place. If number of rules are more, every time MDS will have to scan traffic for specific pattern corresponding to the rules present in the MDS database. So, number of rules is a critical issue to handle. More the number of rules, more time will be taken by MDS to scan the network traffic which may result in poor efficiency and vice versa. So the question arises that what should be the optimal number of rules in the MDS database for its efficient working ?

Rule scoring based mechanism is proposed in this paper to prioritize the rules in the MDS database. Only those rules are stored in MDS database that are appropriate for the network. The proposed mechanism calculates the "score of a rule" depending on the different function parameters. Threshold value is calculated on the basis of different rule properties to make the decision whether the rule should be a part of the database of MDS or not. Based on the threshold value, the number of rules in MDS database are prioritized i.e if score of a rule is greater than threshold value then it will be kept otherwise discarded, so as to increase the performance of overall MDS. The methodology of proposed work is shown in Fig.2. The different features that should be taken into account for assigning a score to a rule:

- The most important one is the searching of a specific content in the incoming packet. How far and from where to start searching of a specific pattern with in a particular depth is better than searching anywhere in a packet.
- As the packet payload contains variety of source and destination ports like specific one, any ports i.e wildcard value and port ranges etc. So how to deal with the ports is the another feature to score a rule.
- The size of packet payload. Looking for a specific pattern within the packet size is better.



* δ_{Th} – Threshold value.

Fig. 2: Proposed Methodology

- class-types is used to classify the rules into different attack classes. Certain very high priority attack classes are shown in Table 1.
- Flowbits is the another feature in rule that can track the state of the different protocols.
- PCREs improves performance of a rule.

The proposed malware detection system examines incoming network data packets for the content that matches known attacks by using the methodology proposed. In proposed methodology, firstly we have to split the various fields of rule header and store in the various parameters named as :

<proto>= type of protocol(TCP/UDP)
<src>= source ip address
<srcport>= source port
<way>= way of communication(uni / bidirectional)
<dst>= destination ip address
<dstport>= destination port
<payload>= content of rule

After splitting the different parts of the rule, the score of the rule can be calculated on the basis of different function parameters named as NumberOfPorts(NP), PacketSize(PS), Content(C), Classtype(Ct), FlowBits(FB), Message(M), Pcre. The total score (S_{Rule}) of the rule is the sum of weights associated with function parameters

TABLE 1: Snort: Classes of Signatures

Classtype	Description
attempted-admin	Attempted Administrator Privilege Gain
attempted-user	Attempted User Privilege Gain
inappropriate-content	Inappropriate Content was Detected
policy-violation	Potential Corporate Privacy Violation
shellcode-detect	Executable code was detected
successful-admin	Successful Administrator Privilege Gain
successful-user	Successful User Privilege Gain
trojan-activity	A Network Trojan was detected
unsuccessful-user	Unsuccessful User Privilege Gain
web-application-attack	Web Application Attack

multiplied by their respective scores :-

$S_{Rule}= NP_w* NP_s+ PS_w* PS_s+ C_w* C_s+ Ct_w* Ct_s+ FB_w* FB_s+ M_w* M_s+ Pcre_w* Pcre_s$

Where S_{Rule}= Total score of rule,

Parameters:

NP_w= weight of NumberOfPorts;
PS_w= weight of PacketSize;
C_w= weight of Content;
Ct_w= weight of ClassType;
FB_w= weight of FlowBits;
M_w= weight of Message;
$Pcre_w$= weight of Pcre;

Similarly the score of various function parameters are:

NP_s= score of NumberOfPorts;
PS_s= score of PacketSize;
C_s= score of Content;
Ct_s= score of ClassType;
FB_s= score of FlowBits;
M_s= score of Message;
$Pcre_s$= score of Pcre;

Initially the score values of function parameters can be initiated to zero and can be set accordingly if the rule contains following:

1. NP_s= 0 # value can vary from 0 in case of all ports to 1 in case of specific one because there exist different number of ways for specifying the the port numbers which are discussed as follows:

1. Static port i.e specific one;
2. Any ports;
3. Port ranges;
4. Negation;

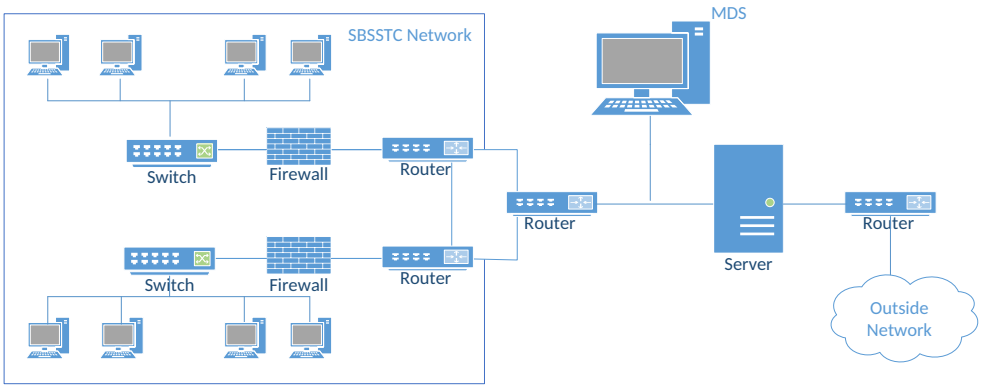


Fig. 3: Experimental Topology

2. C_s= 0 # value can be :

1. =0, when there is no checking;
2. =20, maximum value;
3. =2, checking of single byte within particular depth for specific pattern;
4. =20, checking of ten bytes within particular depth for specific pattern;

3. PS_s= 0 # value can be vary as following:

1. =0, when size is not mentioned;
2. =1, when faithful size is mentioned;

4. Ct_s= 0 # value can be set to 1 for high priority signature classes of snort;
5. FB_s= 0 # value can be set to 1 because rule containing flowbits can never be ignored;
6 M_s= 0 # value can be set to 1 if keyword "POLICY" found in message;
7 $Pcre_s$= 0 # value can be set to 1 because it improves performance of a rule if present;

In proposed mechanism to compute the rule score, only content parameter is used for calculating the score of a rule so weight of content parameter is 1 i.e Cw =1 and all other weights are initialized to 0 and the program is easily modifiable by assigning different weights to different parameters.

4 Experimental Setup

This section provides the description about the experimental setup. The MDS is deployed as shown in Fig.3 between the SBSSTC network and internet server to monitor the internet traffic. The SBSSTC network is composed of 500 no of client PCs, 1 sniffer which act as MDS, 1 server , 3 router and 35 switches. Various hardware and softwares which have been used are mentioned below:

Softwares and Hardware used:

Softwares Packages:
1. Apache;
2. my-sql server;
3. Snort 2.8;
4. Python 2.7.8;
5. Ms Access database 2007.

Hardware:
1. 500 nos of client PCs;
2. 1 server;
3. 1 sniffer having Snort installed on it which act as MDS between internet server and client PCs network;
4. 3 Routers.
5. 35 Switches.

5 Results and Discussions

The proposed MDS have been run in SBSSTC network for a period of 5-6 weeks. The results obtained during the live run of MDS are summarized in Table 2. The results obtained are based on the assumption that the same set of events occur in the network under consideration over the way.

After calculating the score of each rule, the total score (S_{Rule}) is then compared to sample threshold value which is taken as 5 initially. If S_{Rule} is greater than threshold value then it will be stored in keep.rules file else it will be discarded and store in discard.rules file. Each keep.rules file generated from the proposed rule scoring algorithm at different threshold values is then updated to the proposed MDS for its live run for one week and then the threshold value is incremented by 2, again the total score of each rule is calculated and new keep.rules and discard.rules files are generated and again updated to the proposed MDS. In this way, the number of alerts generated corresponding to each threshold value is stored in Table 2. This variation in threshold value and number of alerts generated are shown in Fig. 4.

TABLE 2: Comparison of Alerts at different Threshold values

Threshold value	No of Rules	Keep	Discard	Alerts
5	30688	27935	2753	135400
7	30688	27030	3658	130300
9	30688	25256	5432	125500
11	30688	22195	8493	121425
13	30688	15780	14908	73255

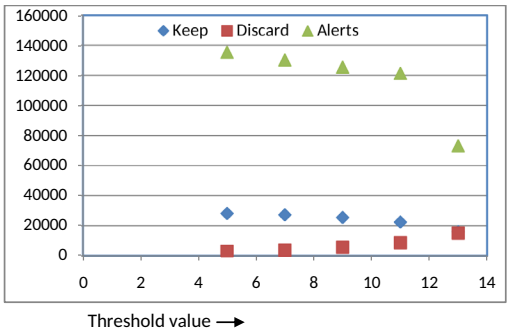


Fig. 4: Variation in Threshold Values

In Fig. 4, it is shown that there is change in number of alerts generated with the increase in threshold value. There is steep decrease in number of alerts when threshold value is changed from 11 to 13 as shown in Fig. 5 which indicates that these set of rules are mandatory for malware detection and should not be discarded. In this way threshold values are computed and used for prioritization of snort rules in MDS. So, by using the proposed approach the number of rules in snort based MDS can be prioritize according to the requirement of a particular network. Further the proposed MDS uses SnortSam [25] and Oinkmaster [26] plug-ins of snort to dynamically update the rules. The results obtained have shown that the performance of a snort based MDS has been enhanced remarkably.

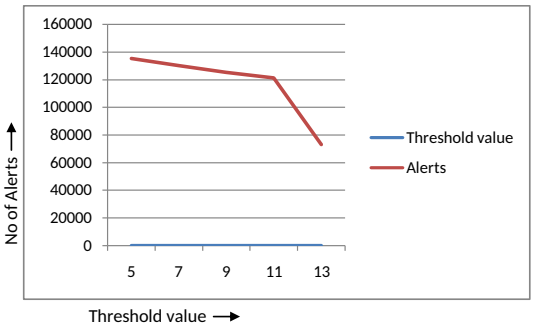


Fig. 5: No of Alerts vs Threshold Value

6 Conclusion and Future Work

Different types of MDS are available nowadays for detection of online attacks and many solutions have been proposed to enhance the performance of these systems which have been validated using experimentation's based on simulation, emulation and other techniques. These systems work on the basis of various rules present in their engine to detect malware. So the kind of rules that should be present in MDS to detect malware is a very critical issue. The efficient working of the MDS largely depends on the various kinds of rules it possesses which ultimately generates the alerts when some malicious activity takes place. To deal with the different kinds of malware, the rules of MDS have to be modified, updated and disabled according to the network in monitor. A rule scoring mechanism has been proposed in this paper, which is based on different function parameters for prioritization of rules like searching of specific content or pattern in the incoming packet, source and destination port numbers, size of packet payload, attack class-types, flow bits and PCREs. It has been finally concluded that the performance of MDS has been enhanced remarkably by using the proposed rule based scoring mechanism. Further, the proposed technique can be combined with network traffic analysis optimization technique for better results. The proposed mechanism can also further be extended by assigning different weights to different parameters used for calculating the optimum threshold value.

References

[1] L. Garber, "Denial-of-service attacks rip the internet," *Computer*, no. 4, pp. 12–17, 2000.

[2] C. Douligeris and D. N. Serpanos, Eds., *Network security: Current Status and Future Directions*. A John Wiley & Sons, Inc., Publication, 2007.

[3] K. K. Sunny Behal, "Extrusion: An outbound traffic based approach to detect botnets," *International Journal of Information and Telecommunication Technology*, vol. 2, pp. 71–76, 2010.

[4] "Large scale ddos attack on github.com." [Online]. Available: https://github.com/blog/1981-Large-Scale-DDoS-Attack-on-github-com.

[5] "Akamai's quarterly [state of the internet]/ security report q1 2015 by akamai technologies." [Online]. Available: https://www.akamai.com/us/en/about/news/press/2015-press/akamai-first-quarter-2015-state-of-the-internet-report.jsp.

[6] "Cloudflare, "2011: The year of the ddos"." [Online]. Available: http://blog.cloudflare.com/2011-the-year-of-the-ddos.

[7] K. Hwang, Y. Chen, and H. Liu, "Defending distributed systems against malicious intrusions and network anomalies," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. IEEE, 2005, pp. 8–pp.

[8] S. Behal and K. Kumar, "An experimental analysis for malware detection using extrusions," in *Computer and Communication Technology (ICCCT), 2011 2nd International Conference on*. IEEE, 2011, pp. 474–478.

[9] S. Behal, A. S. Brar, and K. Kumar, "Signature-based botnet detection and prevention," *http://www. rimtengg. com/iscet/proceedings/pdfs/advcom p/148. pdf*, 2010.

[10] K. Tomar, S. Tyagi, and R. Gupta, "Dynamic rule based traffic analysis in nids."

[11] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.

[12] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *International Journal of Information and Computer Security*, vol. 1, no. 1-2, pp. 169–184, 2007.

[13] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[14] "Suricata." [Online]. Available: http://en.wikipedia.org/wiki/SuricataUserGuide
[15] "Snort." [Online]. Available: https://www.snort.org.
[16] S. A. Hesham Altwaijry, "Bayesian based intrusion detection system," *Journal of King Saud University Computer and Information Sciences*, pp. 1–6, 2012.
[17] H. Altwaijry and S. Algarny, "Multi-layer bayesian based intrusion detection system," *training*, vol. 9, no. 228, pp. 311–029, 2011.
[18] K. Hwang, M. Cai, Y.-K. Kwok, S. Song, Y. Chen, and Y. Chen, "Dht-based security infrastructure for trusted internet and grid computing," *International journal of critical infrastructures*, vol. 2, no. 4, pp. 412–433, 2006.
[19] Y. Wu, L. Shi, B. Wang, P. Wang, and Y. Liu, "Research on intrusion detection based on sequential pattern mining algorithms," *Energy Procedia*, vol. 13, pp. 505–511, 2011.
[20] K. Zhao, J. Chu, X. Che, L. Lin, and L. Hu, "Improvement on rules matching algorithm of snort based on dynamic adjustment," in *Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on*. IEEE, 2008, pp. 285–287.
[21] S. Egorov and G. Savchuk, "Snortran: An optimizing compiler for snort rules," *Fidelis Security Systems*, 2002.
[22] D. S. Kazachkin and D. Y. Gamayunov, "Network traffic analysis optimization for signature-based intrusion detection systems," in *Proceedings of the Spring/Summer Young Researchers Colloquium on Software Engineering*, no. 2, 2008.
[23] S. Sinha, F. Jahanian, and J. M. Patel, "Wind: Workload-aware intrusion detection," in *Recent Advances in Intrusion Detection*. Springer, 2006, pp. 290–310.
[24] Z. Trabelsi and S. Zeidan, "Ids performance enhancement technique based on dynamic traffic awareness histograms," in *Communications (ICC), 2014 IEEE International Conference on*. IEEE, 2014, pp. 975–980.
[25] "Snortsam plugin." [Online]. Available: http://www.snortsam.net/
[26] "Oinkmaster plugin." [Online]. Available: http://oinkmaster.sourceforge.net/