An Approach to Meta-Alert Generation to Reduce Analyst Workload

Deeksha Kushwah
Department of CSE & IT
Madhav Institute of Technology and Science, Gwalior, India
deekshakushwah0@gmail.com

Rajni Ranjan Singh Makwana
Department of CSE & IT
Madhav Institute of Technology and Science, Gwalior, India
rrsingh@mitsgwalior.in

***Abstract*—**This is the era of Technology. Digitization becomes a trend today. Organizations are trying to become more digitized than one another because of the growth in a number of internet users. The Internet is a popular place among criminals too. They perform criminal activities for their benefit over the internet. These actions are called cybercrimes. The reason for increased cybercrime is the enhanced use of the internet. These days millions of attacks are being performed each year. The prodigious number of attacks makes the data under potential infringement. So as far as concerned, cybersecurity becomes the most important issue these days. In recent years intrusions are increasing rapidly that result in compromise with the protection of user data. Intrusion detection systems are used to detects and prevents these intrusions. Port Scanning is a common type of intrusions. Generally, it takes place as the first step of an intrusion. Port-scanner is software intended to probe a machine for open ports. Port scanning attacks result in huge amount of network alerts which is manually analyzed by the network administrator. In the proposed work, a novel approach is presented to minimize this huge amount of the similar alerts of stealth port scanning attack by generating meta-alert. To validate the performance of the proposed method an experiment has been carried out using MACCDC dataset. And it is observed that 99.65% alerts are significantly reduced.

***Keywords*—** Alert Merging; alert reduction; Stealth port scanning; intrusion detection system**.**

I. Introduction

From the very beginning of the internet, nobody could have imagined that the development of technologies would lead to radical changes in everyday life. Online services have changed the lifestyle of peoples. Peoples are depending on the internet for every need. From online shopping to communication everyone is dependent on the internet. Because of its accessibility to everyone, the criminally minded peoples are also using it for completing their ruthless intentions. There is a big community of cybercriminals who try their best to find out loopholes on the internet and take advantage of that. Each day, over the web, hundreds of attacks are being performed. Due to this reason there exist an uncalculated risk of data infringement, privacy violation, and fraud over the web.

Intrusions are common over the internet. An attempt to invade a system or violate security aspect is called intrusion. Intrusion is detected and managed by the intrusion detection systems. An IDS is a device or software which is used to monitor suspicious activity. It can be deployed at the host level or network level. The capabilities of IDS also depend upon the complexity of the system. The IDS can be classified by detection methods or by analyzed activities. Here we briefly discussed the type of IDS-

i) Based on Analyzed Activity

The classification of IDS based on the deployment method. There are two types of IDS based on this method.

a) Host Intrusion Detection System – A HIDS monitors the incoming and outgoing traffic from a device and alerts the administrator if any inconsistency is found in the traffics. HIDS runs on the individual hosts in the network.

b) Network Intrusion Detection System –

NIDS is placed on different point or points in a network. it performs analysis of entire subnet’s traffic and matches it against the library of known attacks.

DOI-10.18486/ijcsnt.2018.7.2.02
ISSN-2053-6283

ii) Based on the Detection method

The classification of IDS based on detection and processing methods. There are two types of IDS based on the detection method.

a) *Anomaly-based Intrusion detection System* -

Anomaly-based IDS monitors network traffic and classifies it as either abnormal or normal. It detects both computer and network intrusions. This type of IDS suffers from a huge number of false positives.

b) Signature-based Intrusion detection system –

Signature-Based IDS constantly look for specific patterns in the network traffic and match them with the signature of a different attack. If a match is found it issues an alarm for the network administrator.

The problem with the current intrusion detection system, such as Snort, is that they generate a high volume of low-level alerts which are manually analyzed by an administrator. So it becomes a difficult task to analyze and manage this huge amount of alerts. The same type of alerts can be merged together to reduce the number of alerts. So, the alerts can be processed more quickly.

In the proposed work, Snort IDS have been utilized. Snort is a free and open source network intrusion detection system (NIDS) in [1]. It is libpcap [10] based packet sniffer. Snort is able to perform packet detection, analysis, and logging in real-time. Snort performs content searching and matching, and protocol analysis. The program can also be used to detect stealth port scans, probes or attacks, server message block probes, buffer overflows, semantic URL attacks, and operating system fingerprinting attempts. Mainly, it is configured in three modes: network intrusion detection, packet logger, and sniffer. In intrusion detection mode, the program will detect network traffic and analyze it against a rule set defined by the administrator. The program will then perform a specific action based on what has been identified. In packet logger mode, the program will log packets to the disk. In sniffer mode, the program will read network packets and display them on the console.

Port Scanning is an application created to probe a host for open ports. Port scanning is used perform by network administrators for monitoring and troubleshooting in the network. Although, this amenity becomes a huge vulnerability when it is used by an attacker for probing the network for finding loopholes and if found, infiltrate cyber assets. NMAP [9] is software used for port scanning. Port scanning is classified into three types based on packets utilized for this purpose.

i) Half open - in half-open scanning a full connection is initiated but terminated in the middle of three hand-shake i.e. do not complete the three-way handshake and leaves the connection half open. This type of scanning attacks is detected by a firewall because it uses known TCP flags.

ii) Full open - Full open a full connection is initiated usually using three-way hand-shaking utilizing connect() call functionality. The limitation of this attack is that the scanning attempt is stored by the destination and easily detected when the logs are examined by an administrator.

iii) Stealth Port-Scanning - Any scan that bypass firewall, filters, routers, and appear as casual network traffic. Most common stealth attacks are completed by setting individual flags or no flags or all flags in a packet.

The presented work is concerned with Xmas, FIN and NULL scan of stealth scanning. These scans are briefly described below-

i) FIN Scan – In the probe packet of FIN scan, only FIN bit is set. If the response contains RST packet then the port is considered closed if it no response is received then the port is considered open and if an ICMP unreachable error is received then the port is considered filtered.

ii) XMAS Scan - In the probe packet of XMAS scan FIN, PSH and URG bits are set. If the response contains RST packet then the port is considered closed if it no response is received then the port is considered open and if an ICMP unreachable error is received then the port is considered filtered.

iii) NULL Scan - In the probe packet of NULL scan no bit is set i.e. TCP flag header is 0. If the response contains RST packet then the port is considered closed if it no response is received then the port is considered open and if an ICMP unreachable error is received then the port is considered filtered.

A port scanning attack produces a lot of same types of alerts. As RFC 793 [11] states that if incoming packets do not contain ACK, RST or SYN bits set, it will return no response if the port is open or RST if the port is closed. Thus, when stealth Scan is performed against a system it sends so many repeated packets on different ports to another machine that resulted in a lot of packets at the destination. The destination machine's IDS, store all of these packets for analysis by a human administrator. This becomes a tedious and time-consuming task for the administrator.

. In this paper, a method is presented by us which is able to detect the stealth port scanning attacks more efficiently while minimizing the similar packets. So instead of analyzing the whole lot of alerts in a great amount of time, we can analyze them in less time and in a hassle-free way.

II. Literature Survey

Chyssler *et al.* [2] architecture are proposed in which intrusion detection systems is combined as sensors. The alarms, provided by this architecture, are improved from the point of view of both quality and quantity.

Farhadi *et al.* [3] presented a system that contains two major components. First is the AESA algorithm. The work of this algorithm is to mine the stream of alerts for attacks scenario. Second is HMM. HMM used to predict the next action of the intruder to correlate intrusion alerts.

Valeur *et al.* [4] presented a method for correlating alerts. This approach consists five-step process for correlating alerts: preprocessing, reconstructing attack session, attack prioritization, effect analysis and reporting intrusion.

Siraj and Vaughn [5] dissert the alert correlation side of sensor alert fusion in a multi-sensor environment. In this work, an abstract incident modeling is developed for alert correlation with generalized security events.

Treinen and Thurimella [6] focused on automatically generate pattern matching rule from inbound traffic. They utilize a data set which contains over 1000 of sensor's output.

Julisch and Dacier [7] use a data mining approach for reduction of false positives via rules, which generate automatically and clustering.

F. Cuppens [12] presented a cooperation module for intrusion detection environment. The cooperation module consists of three functions i.e. Alert management function, alert clustering function and alert merging function. In these modules first, the alerts are collected through the internet then the clustering is performed on the collected alerts after that for each cluster is global alert is generated.

F Cuppens & Miege [13] developed a cooperation module that consists of five functions but only one is discussed in this paper. The two functions are alert correlation and intention correlation. Here only alert correlation is described. In alert correlation, the global alerts, from the first three functions, is correlated in order to provide the administrator with more information.

Ning *et al.* [14] demonstrated a method that constructs attack scenario through correlating alerts. This is done by using conditions and results of attacks. The idea behind the method is that by examining a range of attacks the author concluded that the attacks are not distinct but connected with each other at some stage.

B. Morin et al. In [15] proposed a data model M2D2 for intrusion detection alert correlation. Authors supplies M2D2 with four information type i.e. related with observed events, related with tools utilized for monitor, related with vulnerabilities, and related with monitored information system's characteristics.

III. Dataset Description

In this work, MACCDC dataset is utilized for validation of the work. The MACCDC is an acronym for Mid-Atlantic Collegiate Cyber Defense Competition [8]. It is a competition organized for university or college Students to assay their cybersecurity knowledge and skills in a challenging environment. The description of the utilized dataset is given below-

Table 1

Dataset Description

Dataset	MACCDC
Size	1 GB
Format	Tcpdump
Total Captured Packet	8635943
Total TCP Packets	8485246
TOTAL ALERTS GENERATED	588

IV. Proposed Work

In the proposed method we reduce the number of alerts by merging the same type of alerts and generate meta-alert. The snort rule is developed for this purpose. In this work, the redundant packets of the stealth scan are reduced and alerts are generated by merging the same type of alert. The generated meta-alert contains a high level of abstraction which shows common information contained in all alerts.

The proposed model is shown in figure 1. The whole process is divided into four phases. Here, the phases of the process are described step by step.

Phase 1: The network traffic is loaded into IDS to scrutinize valuable and relevant packets.

Phase 2: IDS scrutinize all traffic and match every packet with the given rule set. After this, packets are marked as relevant and irrelevant packets. Relevant packets are sent for processing while irrelevant packets are dropped.

Phase 3: All the packets that are marked relevant are stored into alert log.

Phase 4: the alerts logs pass through the correlation engine in which merging of similar alerts is taken place. The output of the correlation engine is called meta-alerts.

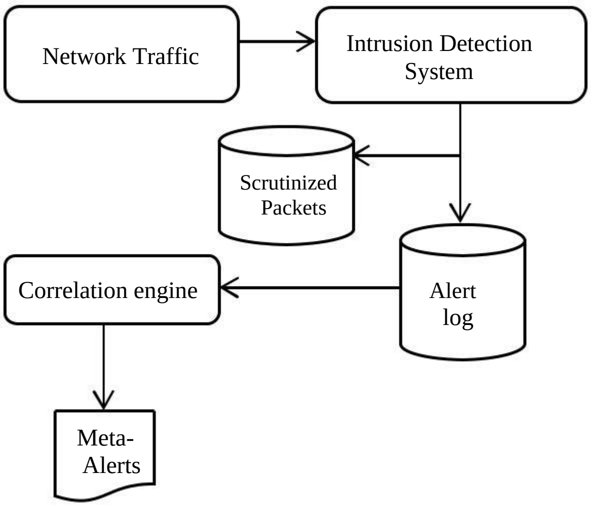

FIGURE 1: Proposed Model

The generated meta-alerts are negligible in compares to the original alerts. The meta-alerts contain following fields of information- Source IP, and Destination IP, total rows, Time, RuleId and Rule message. these fields are described in brief-

i) Total rows: this field contains the number of total alerts a meta-alert represent.

ii) Rule ID: This is a unique number given to each rule. By this, the meta-alerts are separated from each other.

iii) Rule Message: rule Message is a message given to each rule at the time of rule generation.

iv) Time: This field shows the time at which an alert occurs.

v) Source IP: This field contains the IP address of the source of an alert.

vii) Destination IP: This field shows the IP address of the destination of an alert.

V. Experimental Analysis And Results

In the proposed model Snort IDS is utilized. Snort was configured in NIDS mode (option –c is utilized for this purpose). Snort rules are applied for this purpose is shown in table 2-

Table 2: Snort Rules

Rule 1
alert tcp any any -> any any (msg:"NULL Scan"; flags; 0; sid:100000016;)
Rule 2
alert tcp any any -> any any (msg:"FIN Scan"; flags; F; sid:100000034;)
Rule 3
alert tcp any any -> any any (msg:"XMAS Scan"; flags; FPU; sid:100000037;)

Here, MACCDC dataset is utilized. The tcpdump file of this dataset is retrieved from the internet. Then we processed this tcpdump with Snort. After processing, the alerts which contain stealth scanning attack packets logged into a separate log file. These log files send these alerts in correlation engine for further processing. The correlation engine merged these alerts on the basis of similarity.

The resulted meta-alerts are shown in table 3 and 4.

Table 3: Meta-Alert 1

Total rows	293
Rule ID	100000016
Rule Message	NULL Scan
Time	128 distinct values
Source IP	5 distinct values
Destination IP	62 Distinct values

Table 4: Meta-Alert 2

Total rows	295
Rule ID	100000037
Rule Message	XMAS Scan
Time	134 distinct values
Source IP	5 distinct values
Destination IP	63 Distinct values

In our work, the generated Meta-alert contains Total rows i.e. a total number of alerts it contains, Rule ID, Rule Message, Time, Source IP, Destination IP. The impact of alert is shown in table 5.

Table 5: Impact Of Process

	MACCDC
Input Packets	8640136
Output Alerts	588
Meta-Alert	2
Reduction	99.65%

VI. Conclusion

In the Presented work a model has been constructed to reduce the similar alerts of stealth Scanning attack i.e. Null Scan, FIN Scan, and XMAS scan. The objective of the work is to decrease the count of packets contains similar alerts in order to mitigate the workload of the network administrator. An experiment has been carried out utilizing MACCDC data set and it is observed that 99.65% alerts are reduced. In the future, we plan to conduct many experiments using various datasets like DARPA 1998, DARPA 1999, Honeynet datasets and real network traffic.

VII. References

[1]. Roesch, Martin. (1999). Snort - Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX conference on System administration. 229-238.

[2] T. Chyssler, S. Burschka, M. Semling, T. Lingvall and K. Burbeck, "Alarm Reduction and Correlation in Intrusion Detection Systems," in GI Special Interest Group SIDAR Workshop, DIMVA, Dortmund, Germany, 2004.

[3] H. Farhadi, M. AmirHaeri, and a. M. Khansari, "Alert Correlation and Prediction Using Data Mining and HMM," The ISC International Journal of Information Security, pp. 1-25, 2011.

[4] F. Valeur, G. Vigna, C. Kruegel and R. A. Kemmerer, "Comprehensive approach to intrusion detection alert correlation," in IEEE Transactions on Dependable and Secure Computing, vol. 1, no. 3, pp. 146-169, July-Sept. 2004. doi: 10.1109/TDSC.2004.21

[5] A. Siraj and R. B. Vaughn, "Alert Correlation with Abstract Incident Modeling in a Multi-Sensor Environment," International Journal of Computer Science and Network Security, pp. 8-19, 2007.

[6] J. J. Treinen and R. Thurimella, "A framework for the application of association rule mining in large intrusion detection," in Recent Advances in Intrusion Detection, Berlin Heidelber, Springer-Verlag, 2006, pp. 1-18.

[7] K. Julisch and m. Dacier, "Mining intrusion detection alarms for actionable knowledge," Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 366--375, 2002.

[8] Mid-Atlantic Collegiate Cyber Defense Competition (MACCDC) (2018, may 20) Retrieved from http://www.netresec.com/?page=MACCDC

[9] Lyon, G., 2009. Nmap-free security scanner for network exploration & security audits.

[10] Jacobson, V., leres, C. and Mccanne, S., 1994. Libpcap, Alwrence Berkeley Laboratory, Berkeley, CA. Initial Public Release

[11] Postal, J., 1981. Transmission Control ProtocolRFC793.