ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, No 5, May 2013

Intrusion Alert Elimination on Network Attack Alerting System

Mon Mon Zaw

***Abstract*— Network attack alerting system becomes a critical technology to help and assist security engineers and network administrators to secure their network infrastructure. The proposed system implements network attack alerting system based on Network-based and Host-based Intrusion Detection System (IDS). Open source attacking system, Backtrack is used to initiate and launch the attacks. Well-known free open source tools available on Security Onion Linux Distribution are used to distinguish the important network IDS alert types. The system uses existing IDS rules and defines the set of new rules to fetch these attacks. There are the overwhelming alerts generated by IDSs so finding a solution to reduce these alerts is the most important field of IDS. The system eliminates the large numbers of alerts that belong to the same attack type within the defined time window.**

***Index Terms*— intrusion detection system, rules, alerts, attack.**

I. INTRODUCTION

With the increase usage of the network of computers and Internet, the number of network attacks has also risen. To detect against these attacks, intrusion detection systems are greatly used into computer networks. The main purpose of the intrusion detection system is to reveal intrusive events and flag alerts in such an event. An intrusion detection system can be compared with a house burglar alarm: if somebody tries to enter illegally in the house, one of the sensors will detect it and will trigger the alarm bell and alert the house owner and the police. Similarly, if somebody tries to compromise the confidentiality, the integrity or the availability of a computer system or network, or tries to break the security protections, an intrusion detection system will alert the system owner and the security team [1].

Intrusion detection system may raise large number of alerts that are redundant, irrelevant and correlate alerts. The redundant, irrelevant and false alerts are reduced as early as possible for the purpose of reducing the number of processed alerts to enhance the performance. This paper eliminates the redundant alerts that have the similar attributes such as the source IP, destination IP, source Port, destination Port and so on. Threshold count and threshold time are defined to classify the severity level of the alerts. Alert reduction and defining the severity level are important for the system administrators to take appropriate actions. If the alerts reach the highest severity level, the security engineer needs to take down the attack origin.

Manuscript received May, 2013.

***Mon Mon Zaw**, Faculty of Information and Communication Technology, University of Technology (Yatanarpon Cyber City), Pyin Oo Lwin, Myanmar, 09-420731668,*

II. INTRUSION DETECTION SYSTEM

Intrusion Detection System is greatly becoming a vital component to detect various attacks or intrusion activities as an active way and also useful in monitoring attempts to break network security. Intrusion could be in many patterns such as: non-legitimate user attempting to get access to the system resources or network resources, malicious programs that ruin the system resources, declines the system function and legitimate user attempting to gain advanced privileges or access to confidential information, thus compromising the system's security policy. The primary function of IDS is to inform the system administrator about the event of an attack. The typical components of IDS are sensor or agent, management server, database server and console. Intrusion detection systems are classified into two types: Host-based and Network-based Intrusion Detection System.

A. Host-based vs. Network-based Intrusion Detection System

Intrusion detection system was firstly developed for host-based computer systems. Host-based Intrusion Detection System (HIDS) are located in the server computers and check the internal interfaces. It examines attack patterns by revising application logs, system calls, file-system modifications, and other host behaviors that are relevant to the server computers. They are generally applied for checking user behavior and used to trail intrusions happened when legitimate user attempts to get confidential information. HIDSs typically built the extensive log file data that are relevant to detected events. This log data can be applied to endorse the validity of alerts, to explore incidents, and to correlate events between the host-based IDS and other logging sources. The attributes commonly logged by host-based IDSs include the following: Timestamp (usually date and time) , Event or alert type , Rating (e.g., priority, severity, impact, confidence) and Event details specific to the type of event, such as IP address and port information, application information, filenames and paths, and user IDs [3].

With the increased usage of computer networks, IDS gradually shifted toward the network-based IDS. NIDS regards and revises network packets to detect attacks in the network system. It attempts to detect malicious behavior such as denial of service attacks, port scan or even tries to break

into computer by monitoring network traffic. NIDSs typically built the extensive log file data that are relevant to detected events. Data fields commonly logged by network-based IDSs include the following: timestamp (usually date and time), Connection or session ID (typically a consecutive or unique number assigned to each TCP connection or to like groups of packets for connectionless protocols), Event or alert type , Rating (e.g., priority, severity, impact, confidence) , Network, transport, and application layer protocols, Source and destination IP addresses , Source and destination TCP or UDP ports, or ICMP types and codes, Number of bytes transmitted over the connection , Decoded payload data, such as application requests and responses and State-related information (e.g., authenticated username) [3].

B. Misuse-based vs. Anomaly-based Intrusion Detection System

Misuse-based or Signature-based IDS performs pattern matching techniques to be compatible an attack pattern corresponding to known attack patterns in the database and issues very low false positives (FP). The major disadvantage of misuse detection is that it cannot guess new and unknown attacks and has high false alarm rate. The necessity of Misuse-based IDS is to regularly update rules or signatures and it cannot detect unknown attacks.

Anomaly-based IDS creates normal behavior models and automatically detects anomalous behaviors. Anomaly detection techniques identify new types of intrusions as deviations from normal usage, but the drawback of these techniques is the rate of false positives (FP). The advantage of anomaly detection is that it can detect attacks that have been seen or not before. But the drawback of anomaly detection is ineffective in detecting insiders' attacks.

III. RELATED WORK

Since Anderson's report [Anderson 1980], Intrusion detection has been observed for over twenty years. By applying various techniques, the researchers proposed systems that purpose to construct attack scenarios. Dain et al. [7] apply data mining approach to integrate the alerts into attack scenarios in real time. In [Valdes and Skinner], a probabilistic approach is applied to carry out correlation information from diverse sensors, and concentrate on the idea of 'threads' to control links between alerts [8]. Ritchey and Ammann used a model checking technique to identify network vulnerabilities on the basis of prerequisites and consequences of attacks together with hosts and network connectivity information [9]. Humphrey Waita Njogu uses Clustering technique to eliminate the large amount of alerts and to improve the quality of alerts sent to the analysts by verifying alert using the available Supporting Evidence (Vulnerability data, logs and Network Resources) before alerts are clustered [10].

V.SrujanaReddy proposed a new technique based on maximum likelihood approach for the purpose of online alert aggregation based on dynamic, probabilistic model [11]. Safaa O. Al-Mamory use Breadth-First search algorithm to find the related attacks and show the correlation graph CGs that effectively simplify the analysis of large amounts of alerts [12]. H Pao proposed a graphical signature for intrusion detection given alert sequences and identified group of alerts that are frequent and shows novel graph based on dissimilarity measure [13]. This paper emphasizes the elimination of alerts of the same attack based on the attributes values of the attack pattern and shows how many times of these attacks alerts on the defined time window. The severity level of these alerts is classified by defining threshold time and threshold value.

IV. PROPOSED SYSTEM ARCHITECTURE

The network attack alerting system is based on the virtual machine (VM) ware. This system creates network and host attacks using attacking tools on Network Lab Environment. Well-known free open source tools available on Security Onion Linux Distribution are used to detect these attacks. This system uses network-based intrusion detection sensor and host-based intrusion detection sensor to distinguish the important IDS alert types. The alerts are stored in the database to define rules set, reduce the large number of alerts of the same attack and define the severity level of the alert types. This system builds the own database consisting of the necessary information to reduce the same alert types.

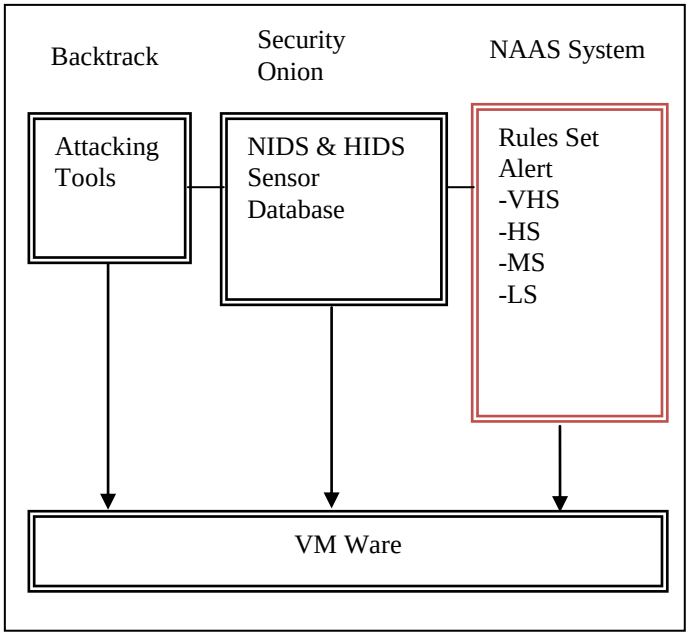

Figure. 1. Network Lab Environment

A. Security Onion

Security Onion is Linux distribution for IDS (Intrusion Detection) and NSM (Network Security Monitoring) that provides full context and forensic visibility into the traffic it monitors. It is based on Xubuntu and contains Snort, Squil, Snorby, Squert, OSSEC and many other security tools [5]. In this system, security onion is used to alert the event of the attacks that are launched from penetration testing tools.

B. Backtrack

Backtrack is Linux distribution designed for the world's leading penetration testers and information security auditing professionals. Backtrack provides users with easy access to a comprehensive and large collection of security-related tools ranging from port scanners to password crackers. The penetration testing tools included in Backtrack can be categorized into: information gathering, network mapping, vulnerability identification, web application analysis, penetration, privilege escalation, maintaining access and so on [6].

C. Snort

Snort is an open source Intrusion Detection System for monitoring and detection of security attacks on networks. A rule-driven language is used in snort. It combines the benefits of signature, protocol and anomaly based inspection method [4].

alert tcp any any -> $HOME_NET any (flags: SF; msg: "SYN-FIN packet detected";)

This signature detects any scan attempt using SYN-FIN TCP packets. The flags field is used to find out which flag bits are set inside the TCP header of a packet. The protocol is TCP and can be issued by the any host on the internet and pointed to any node in HOME_NET.

V. PROPOSED SYSTEM

When suspicious traffic is detected based on applying rules, an Intrusion detection system issues large number of alerts. Some are duplicate, dissimilar, unrelated, frequent, non-frequent, important and non important alerts. Some alerts are fragmentary attribute information. The proposed system is used the network-based IDS sensor and host-based IDS sensor for detecting intrusion or attack and constructs a log file database , in order to save all the reports issued for future references. In intrusion detection system, some alerts are getting from the same type of attack patterns, some are not similar alert patterns and various attribute values but are the same group and some alerts are the consequence of the previous alerts. This system eliminates the number of alerts that are relevant to the same attack pattern. Alert reduction and defining the severity level are important for the system administrators to take appropriate actions.

A. Algorithm for Defining Severity Level

For defining the severity level, threshold value and threshold time based on the occurrence of the attacks within time interval are predefined. The algorithm for defining severity level is shown above.

TI: Time Interval; Threshold Time: TT;
Threshold Count: TC;Alert Count : A_c ;
if ((TI (A_i) < TT && A_c > TC) or (TI (A_i) > TT && A_c > TC)) then
  Alarm 'severity level: HIGH';
else if (TI (A_i) < TT && A_c < TC) then
  Alarm 'severity level: MEDIUM';
else (TI (A_i) < TT && A_c > TC) then
  Alarm 'severity level: LOW';

B. Algorithm for ICMP Alert

In this system, each alert A is mostly considered on the attributes A = (TS, SID, Proto, srcIP, srcPort, destIP, destPort), where the time stamp attribute expresses the frequent time of the alert, the SID attribute states the signature ID that issued the alert, and the Proto attribute reveals the protocol type of the network traffic that initiated the alert. The srcIP, srcPort, destIP, and destPort attributes describe the source IP address, source port, destination IP address, and destination port of the traffic. The attributes (SID, Proto, srcIP, destIP) are used to reduce the alerts of the same attack because the protocol ICMP does not consist of ports. If necessary, type and code of ICMP is used to check the alerts.

List of attributes on Alert (A): [TS, SID, Proto, srcIP, destIP]
TS: Time Stamp; SID: Signature ID;
srcIP : SourceIP; destIP: Destination IP;
Alert Count : A_c = 0;
while Proto (A_i) = "ICMP"
{
  if (SID (A_i) == SID (A_{i+j}) and srcIP (A_i) == srcIP (A_{i+j}) and destIP (A_i) == destIP (A_{i+j})) where i,j = 1,2,3,.. n.
  {
  A_c ++;
  j++;
  }
TI (A_i) = TS (A_{i+j}) – TS (A_i);
}
  Alert A_i (TS, SID, Proto, srcIP, destIP);
  Proceed Severity Level Algorithm.

C. Algorithm for Port Scanning Alert

If the attacker tries to scan ports (Port Scanning) what services are running on the victim host, the protocol type is TCP and the attributes of alerts (TS, SID, Proto, srcIP, srcPort, destIP, destPort) are considered. The signature id of the alerts that corresponds the same attack is mostly same. But in this attack, the signature id of these alerts is not the same. When the attacker (same source IP) tries to connect the same destination IP for port scanning, the attack comes from random same source port to different destination ports with the same sequence number on the one time session. In elimination of the port scanning alerts case, need to check source IP, destination IP, source port, destination port and sequence number. There is no payload information on port scanning attack alert. In some attack case, if necessary, the payload information of the alerts is considered.

List of attributes on Alert (A): [TS, Proto, srcIP, destIP, srcPort, destPort, seqno]
TS: Time Stamp; srcIP : Source IP; destIP: Destination IP ;
srcPort : Source Port; destPort: Destination Port;
seqno: sequence number;
Alert Count : A_c = 0;
while Proto (A_i) = "TCP"
{

if (srcIP (A_i) == srcIP(A_{i+j}) and destIP (A_i) == destIP (A_{i+j}) and srcPort (A_i) == srcPort (A_{i+j}) and seqno (A_i) == seqno(A_{i+j})) where j = 1,2,3,…,n then
{
A_c ++;
j++;
destPort (A_i) = destPort (A_i) U destPort (A_{i+j});
}
TI (A_i) = TS (A_{i+j}) – TS (A_i);
}
Alert A_i (TS, Proto, srcIP, destIP, srcPort, destPort);
Proceed Severity Level Algorithm.

D. Algorithm for Direct Flooding Alert

The attacker tries to make a machine or network resources unavailable to its legitimate user launching the Flooding attack. When the attacker sends a SYN packet, the victim server must open a connection and keep it alive until the connection ends. Using unlike spoofed IP addresses, the attacker can send large number of SYN packets until the target machine is incapable to accept any more connections. This algorithm shows the elimination of the direct flooding alert using one spoofed IP address.

List of attributes on Alert (A): [TS, Proto, srcIP, destIP, srcPort, destPort, Flag]

TS: Time Stamp; srcIP : Source IP; destIP: Destination IP ; srcPort : Source Port; destPort: Destination Port; Flag: Flag;
Alert Count : A_c = 0;
while Proto (A_i) = "TCP" && Flag = "SYN" or Proto (A_i) = "TCP" && Flag = "RST"
{
if (srcIP (A_i) == srcIP(A_{i+j}) and destIP (A_i) == destIP (A_{i+j}) and srcPort (A_i) == srcPort (A_{i+j}) and destPort (A_i) == destPort (A_{i+j})) where j = 1,2,3,…,n then
{
A_c ++;
j++;
}
TI (A_i) = TS (A_{i+j}) – TS (A_i);
}
Alert A_i (TS, Proto, srcIP, destIP, srcPort, destPort);
Proceed Severity Level Algorithm.

VI. CONCLUSION

This system shows the importance alerting system to help and assist the security engineers and network administrators to secure their network infrastructure. It also generates the alerts which are useful for the security engineers to take down the attack origin definitely. This system removes the duplicate alerts of the same attack and then shows one alert. To eliminate the alerts, it only considers that the attacks caused from the same source IP to same destination IP and how many times the one source IP creates the same attack. Later, the system will built the profile-based signature database to check the attack occurring day-by-day. And then correlation technique is used to consider the relevance of the alerts.

ACKNOWLEDGMENT

I would like to be grateful my thesis advisor, Dr. Thandar Phyu for pointing my paper and providing many valuable comments and suggestions to improve this paper. I would also like to thank my family for always being there for me.

REFERENCES

[1] R. & Mell P, "Intrusion Detection Systems". NIST Special Publication,pp. 800-31,2001.
[2] Prahanthi , Radha Devi & K.Sandhya Rani "Analysis of Intrusion Detection System & Emergence of Online Alert Aggregation", Vol. 2, Issue 2,Mar-Apr 2012, pp.1483-1487 1483 | P a g e
[3] K. Scarfone, and P. Mell, Guide to Intusion Detection and Prevention Systems‖ ,National Institute of Standards and Technology NIST. Computer Security, 2007.
[4] Snort Users Manual, http://www.snort.org.
[5] http://code.google.com/p/security-onion/.
[6] http://en.wikipedia.org/wiki/BackTrack.
[7] Dain O.M. and Cunningham R. K, "Fusing a heterogeneous alert stream into scenarios", Proceedings: the 2001 ACM Workshop on Data Mining for Security Applications, 2001, pp. 1-13.
[8] Valdes A. and Skinner K., "Probabilistic alert correlation", Proceedings: Recent Advances in Intrusion Detection, LNCS 2212, 2001, pp. 54-68.
[9] Ritchey, R. and Ammann, P. 2000, "Using model checking to analyze network vulnerabilities", In Proceedings of IEEE Symposium on Security and Privacy. 156–165.
[10] Humphrey Waita Njogu and Luo Jiawei, "Using Alert Cluster to reduce IDS Alerts".
[11] Safaa O. Al-Mamory and Hong Li Zhang, "Scenario Discovery Using Abstracted Correlation Graph".
[12] H Pao, C- Mao and H- Ming Le, "An Intrinsic Graphical Signature Based on Alert Correlation Analysis for Intrusion Detection, Journal of Information Science and Engineering 28, 243-262 (2012).