Hybrid Intrusion Detection

by Slamet A.

1 Introduction

Speed of internet development invites new problems in network security. It becomes an important subject to be researched and improved, so it does not become a serious problem for humans due to the possibility of attacking in the network [1].

In-depth insight into network attacks from campus network users, including detecting a number of incomplete and unclear network data, vague data from campus network users can help provide comprehensive evidence for school administrators in making decisions. It allows preventive actions against unwanted network attacks. In a broader perspective, this condition can create a learning environment that is beneficial for students and will have a major influence on the environment of higher education [2].

Intrusion detection systems (IDS) are usually used to prevent network attacks. Based on how to detect, there are two ways of intrusion detection, namely anomaly-based detection and misuses-based detection [3].

In misuses-based detection [4], there is a database that contains many known signatures of attacks. The content of the database in IDS is compared with many known signatures data collected by the IDS. A notification will be generated if a match is found. However, if there is an event that does not match one of the attack models, the event will be considered as part of legitimate activity. The advantage of misuses-based systems produces very few positive errors. But the disadvantage is cannot detect attacks that have never been known before, and cannot even detect new variations of known attacks.

In addition, another detection model is anomaly-based detection [4]. This detection model is behavior-based. The behavior-based means that all activities are assumed to be dangerous activities and all attacks are part of abnormal activities. After that, this model builds a normal model of system behavior, it looks for anomalous activities that are not in accordance with the specified model.

However, because it is not possible to describe all user activities in the system that lead to activities with a relatively high false-positive rate, and most IDSs currently use one of two detection methods [4], we combine the method of misuse-based detection with anomaly detection to improve the performance of IDS into the latest research on hybrid IDS.

2 Related Works

In [5], Peng et al. propose two stages in hybrid intrusion detection and visualization system that utilizes the ability of signature and anomaly-based detection methods. This hybrid system can identify known and unknown attacks on system calls. However, the results of evaluating the system disappeared in the paper. This work is more like an introduction to how to implement several stages of intrusion detection to improve IDS detection capabilities. Based on the idea of integrating the excess

false positives of low IDS-based signatures and the advantages of anomaly intrusion detection systems to detect new attacks or unknown attacks, Hwang et al. proposed a new hybrid intrusion detection system (HIDS) in [6].

Evaluation of the three IDSs used by Wang et al. concluded that the low computing resources used by Snort and the rules succeeded in accurately classifying legitimate and malicious network traffic. Researchers have evaluated the performance of three IDSs in a simulated environment consisting of physical and virtual computers. Snort has a negative impact on network traffic using more than two other IDSs tested in the experimental results [7].

Snort IDS was used to conduct experiments by Bulajoul et al. [8] in designing real networks. The results show Snort IDS weaknesses in processing packets at high speed and it easily dropping packets without analyzing them accurately. The conclusion of this study is that Snort IDS failed to process network traffic at high speeds and higher packet reduction rates. As a solution to reduce the decline rate of the packet, the researchers introduced parallel IDS. A dynamic traffic awareness histogram is used to improve the performance of IDS Snort. The most effective way has been discussed in this study is using the order of attack signature rules and sequence of rules. The approach is to use a histogram in predicting the next signature rule and the order in the field. The simulation shows that the proposed approach can significantly improve the performance of Snort [9].

Regardless of the amount of research conducted to date, there are still fewer works that investigate the performance of network IDS in campus networks. To this extent, this paper has made further use of Snort IDS as a system capable of detecting known attacks and C45 data mining techniques are used to detect unknown attacks. Thus, the performance of the monitoring and assessment process throughout the campus network can be improved in the direction of developing learning mechanisms for detecting unknown attacks.

3 Hybrid Intrusion Detection System

There are two functions in Hybrid IDS, the anomaly detection technique detects unknown attacks, and the signature detection technique detects known attacks.

3.1 Component of Hybrid Intrusion Detection System

The Hybrid Intrusion Detection System must be able to detect a known and unknown attack on the campus network. The components that must exist in the Hybrid Intrusion Detection System include Snort, rule module, alert module, and C4.5 Algorithm Detector.

3.2 Intrusion Detection System (IDS)

Judging from the way of working in analyzing whether the data packet is considered as infiltration or not, IDS is divided into 2 based: knowledge-based or misuses detection and behavior-based or anomaly detection [10].

Knowledge-based IDS can recognize data flow on a computer network by tapping a data packet, then comparing it with the rules in the IDS database that contain signs of an attack packet. If the captured data packet has the same pattern or at least one pattern in the IDS database rules, then this packet will be considered as an attack. However, if the data packet captured does not have the same pattern as in the pattern of the IDS rule database, then this data packet is not considered as an attack in the network [10].

Behavior-based or anomalies based can detect data flow by observing irregular relationships in a network system, or observe any deviations from normal conditions. For example, there is a sharp increase in memory usage of Server, or there is an IP Address with multiple connections using a huge capacity of bandwidth at the same time and same place. This condition is considered a deviation which is then based on the type of IDS anomaly considered as an attack.

While seen from the ability to detect intrusions on the network, IDS is divided into two based, namely: host-based and network-based. Host-based is able to detect only the host where IDS is implemented, while network-based IDS is able to detect all hosts that are in a network with hosted IDS implementation. This paper specifically uses network-based IDS and knowledge-based [11].

4 Research Methods

The steps used in completing this study are as shown in Fig. 1.

The research method that used is the Security Policy Development Life Cycle (SPDLC) method [12]. With SPDLC, the network system development life cycle is defined in a number of phases, including analysis, design, implementation, enforcement, and enhancement.

4.1 Stage of Analysis

The SPDLC model begins its network system development cycle at the analysis stage. At this stage, the system specifications will be analyzed, the tools needed such as software and hardware needed for the IDS system.

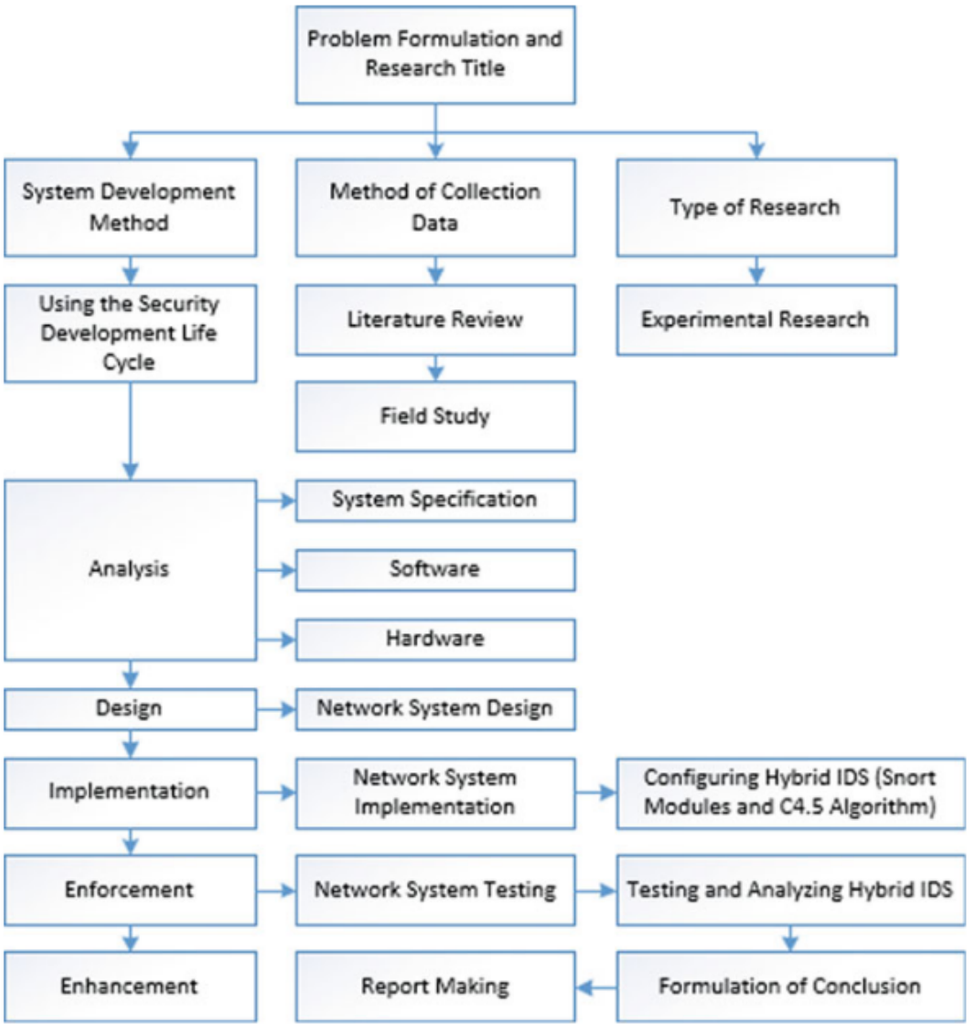

Fig. 1 Research flow chart

4.2 Stage of Design

This design is based on concepts and descriptions that explain the actual device. Intrusion detection system developed by the Network Intrusion Detection System (NIDS) type. This is because this type of IDS is placed in a strategic place/point or a point in a network to supervise the traffic that leads to and originates from all devices (devices) in the network. All scanning from the outside and inside the network is carried out by the scanning process ideally. The following (Fig. 2) is the design of topology when applied to Hybrid IDS.

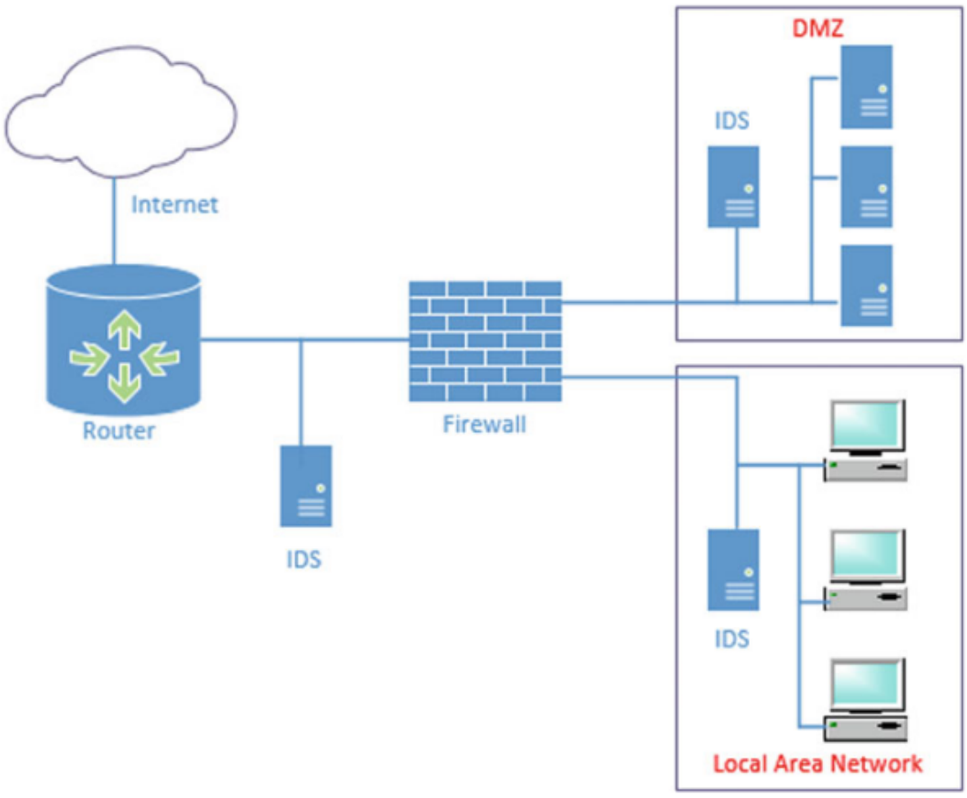

Fig. 2 Network topology design

4.3 Stage of Implementation

The next phase is the implementation of a detailed topology design and system design. Design details are used as instructions or guidelines for the implementation stage so that the system built becomes relevant to the system that has been designed.

Implementation of Hybrid Intrusion Detection System. The main modules and supporting modules are needed to build the functional requirements of the Hybrid Intrusion Detection System. The main modules are: snort engine, snort rule, C4., and alert module. While supporting modules are: ACID (event management) and Webmin (rule management).

The target of implementing Hybrid Intrusion Detection System on Ubuntu Linux systems is 18.04. The block diagram of the Hybrid Intrusion Detection System designed as follows.

Intrusion detection system model is designed in this paper, combined with the advantages of misuses detection and anomaly detection technology, instead of the single detection technology. As shown in Fig. 3, it includes misuse detection module based on snort, anomaly detection module based on the Decision Tree C4.5, and alarm log. Packet data (known attack) is used by the Snort to detect known malicious

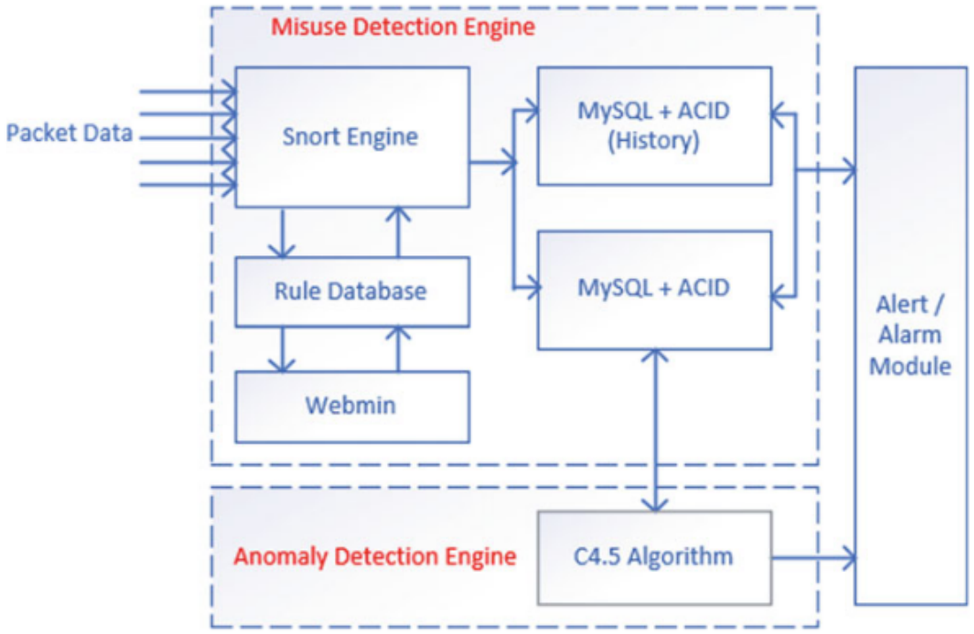

Fig. 3 Hybrid intrusion detection system infrastructure

attacks, and traffic classification and detection of unknown attack used for Decision Tree C4.5 [13]. The detailed testing process as depicted in Fig. 4.

When detecting, matching the characteristic of network traffic with the rule database, once matched, we considered it is an intrusion behavior, in this way, the

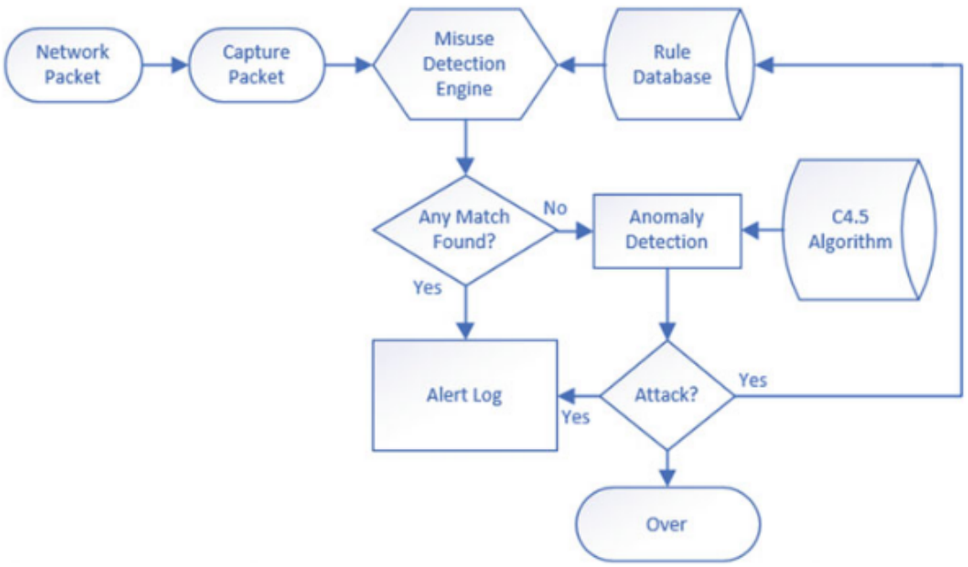

Fig. 4 The flow of the testing process

known malicious attacks can be detected rapidly and accurately. Once found malicious behaviors, will immediately alert, when don't match, the data packets will send to the anomaly detection module, if discovery it is unusual, and alarm, at the same time to save the network events into the feature database; if not, then add data to the training set in the database, to ensure the real-time update of the database. It is able to accurately detect known attacks, but also can discover the new, unknown attacks, and achieve the goal of all-round protect the network security.

Rule Database. This module provides rules in the form of pattern type attacks. This rule is a text file that is compiled with certain rules [14, 15].

Snort Engine. This module serves to read data packages and compare them with rule databases, if the data package is punished as an intrusion/attack, then the Snort engine will write it to an alert (in the form of a log file) and to the database (which is used in this experiment is a MySQL database) [14, 15].

Alert. This section is a record of attacks on a log file.

Webmin. Webmin (http://www.webmin.com/) which has been added to the snort rule module is used to manage the rule. Which Rule will be enabled and disabled can be set via Webmin, it can even be used to add rules manually with a web-based editor.

ACID (Analysis Console for Intrusion Databases). ACID (http://www.cert.org/kb/acid) is used to manage security event data, the advantages of using ACID include: log-logs that were hard to read become easy to read, data can be searched and filtered accordingly with certain criteria, Managing Large Alert Databases (Deleting and Archiving), and for certain cases can refer alerts on database security sites such as Securityfocus, CVE, arachNIDS.

C4.5 Algorithm. Intrusion detection algorithm based on C4.5 is divided into three steps [13]:

Step 1: Make a decision tree, Algorithm: C4.5 Trees produce decision trees from the provided training data. Input: T training sample set, candidate attribute collection, attribute-list. Output: A Decision tree.

(a) Create N root node.
(b) If T belongs to the same category C, then return N as a leaf node then mark it as class C.
(c) If the remaining sample T is less than the given value or the list of attributes is empty, return N as the leaf node, then mark it as the most frequently occurring category.
(d) Calculate the information acquisition ratio for each attribute in the attribute-list.

(e) If test attributes are attributes that have the highest information acquisition ratio in the attribute list, noting that the test attribute is the N test attribute.
(f) Find the division threshold if the test attribute is continuous.
(g) For each new leaf node grown by node N, calculate the classification error rate of each node, and then prune the tree.

Step 2: Extract rules of classification.
In the decision tree, each branch will represent the test output and each leaf node will represent the category or distribution category. Follow each path from the root node to the leaf node. Conjunctions of each attribute value are antecedents of rules, while leaf nodes are a consequence of rules. Thus, decision trees can be easily converted to IF-THEN rules.
Step 3: Determine patterns of network behavior.
New behavior patterns in the network are determined by patterns that are classified as intruders or not based on classification rules.

4.4 *Stage of Enforcement*

The SPDLC computer network system development model categorizes enforcement at the testing stage. The testing process is needed to ensure the system built is in compliance with the design specifications and meets the needs of the problems at the Institut Bisnis dan Informatika Stikom Surabaya.

Modeling the Attack. An attack on the network requires a target server that is running FTP, HTTP, and SSH services as shown in the network topology (Fig. 2). This experiment produces seven types of malicious traffic and legitimate traffic as shown in Table 1. This traffic is intentionally injected into the IDS to be attacked, and each IDS will check all existing traffic, whether legitimate or malicious traffic. When input traffic matches the rules set, it will trigger an alarm to carry out its function. Snort accuracy in classifying network traffic will be determined by the number of alarms (true positive, false positive, and false-negative). This malicious traffic with various exploits and payloads is generated using the Metasploit framework.

Table 1 Number of rule set

No	Type of malicious traffic and rules	Rule set number
1	ARP	25
2	Dos/Ddos	70
3	ICMP	130
4	Scan	35
5	SSH	10
6	FTP	80
7	HTTP	120

In running an exploit requires information about the target attack system such as information about the operating system and what services are being ru[2]. This information can be searched and collected using a port scanning application or other exploitation tools. This Metasploit is modular and can be mixed or matched with different exploits to achieve th[2] required results. The following is an example of the Snort IDS rule using the same syntax in this case. A general Snort rule is: alert ICMP any any; any any (msg:"ICMP Packet"; sid:476; rev:4;). This rule indicates that there is "ping traffic" or an ICMP packet.

Experiment Scenario 1: Ping Attack (ICMP Attack). Hybrid IDS is implemented on a network router that connects intranet and DMZ networks. In this test, large ICMP packages were sent so that they were categorized by IDS as a DOS attack (denial of service).

The following tests are carried out through the client on the internal network.

Ping 172.25.83.30 -l 5000 -t
Pinging 172.25.83.30 with 5000 bytes of data
Reply from 172.25.83.30: bytes = 5000 time = 10 ms *TTL = 63*
Reply from 172.25.83.30: bytes = 5000 time = 10 ms *TTL = 63*
Reply from 172.25.83.30: bytes = 5000 time = 10 ms *TTL = 63*
Reply from 172.25.83.30: bytes = 5000 time = 10 ms *TTL = 63*
Ping statistics for 172.25.83.30
Packets: Sent = 4, Received = 4, Lost = 0 (0 Approx round trip times in m-seconds
Minimum = 0 ms, Maximum = 10 ms *, Average = 3* ms.

This DOS attack will be detected immediately by the snort engine, then the snort engine will send alerts to alert logs, MySQL ACID and MySQL ACID history. The IDS engine reads alerts on the MySQL ACID and then instructs the firewall to update the rule by adding a rule to block access from detected IP attackers. Observation of this experiment was carried out in 2 places: in the client where the attack was carried out and on the IDS system.

Experiment Scenario 2: Nmap Port Scanning Attack. In this case, the author will simulate and analyze the types of port scanning activities using Nmap, which are carried out from both the attack machine, internal (Client) and external attackers.

The first step is to make rules/signatures to define this type of activity. Based on the results of traffic analysis, the author defines Nmap ping as follows:

alert icmp any any-> any any (msg: "ICMP PING NMAP attack"; dsize: 0; itype5: rev: 1; sid: 1003;).

The above signatures or rules will generate Snort alerts if they detect access to the ICMP protocol originating from external or internal network segments, through any port to any port 172.25.83.254 (machine server): statement rules: "ICMP PING NMAP attack"; 0 byte packet size; use ICMP type 5; First revision rules: ID number rules 1003.

The second step is to apply these new rules/signatures by placing them in the rules directory Snort (/etc./snort/rules). In this study, the author keeps this signature with the name localrules. After that, the Snort process must be restarted, so Snort can detect, read, and apply the new rules to the core code.

Experiment Scenario 3. This technique was analyzed using a KDD Cup99 Network Intrusion Dataset [16] carried out by the Lincoln Laboratory at MIT. This data is a standard dataset has been reviewed and includes training and testing sets. The training set is about 7 GB of binary TCP chuck data that has been compressed from 7 weeks of network traffic with around 3 million connections. The test set was taken from three weeks of network traffic with around 3 million connections [17] (Table 2).

In this experiment, we use 295,078 records from the KDD data set (corrected.zip). The number of samples is shown in Table 3. From this data, 10% of the data was extracted by sampling, 34% was dedicated to the test set and 66% of this new set belonged to the training set. After the training set process, 23 types of attacks were found in 37 types of attacks available in the KDD Cup dataset. Therefore, this test set can be used to predict the ability to detect unknown attacks or new attacks.

The following metrics can be used to measure attack detection [18]: 1. False-positive (FP) or a false alarm, when normal behavior that is incorrectly classified as intrusive by the IDS; 2. False-negative (FN), when an attack that is missed by the IDS, and classified as normal; 3. True positive (TP), when an attack that is

Table 2 The experiment of attack categories

Category of attack	Method of attacking
DoS (Denial of Services)	Udpstorm, teardrop, smurf, processtable, neptune, mailbomb, apacha2, backland
Probing	Nmap, satan, mscan, Ipsweep, portsweep, saint
R2L	Worm, sendmail, named, ftp-write, imap, guess password, warezmaster, multihop, snmpgetattack, spy, warezclient, xsnoop, snmpguess
U2R	Xterm, rootkit, perl, Buffer_overflow, oadmodule, ps, sqlattack, httpptunnel
Normal	Normal

Table 3 Number of samples in the dataset

Category of attack	Number of samples
DoS (Denial of Services)	215,000
Probing	4500
R2L	15,500
U2R	178
Normal	59,900
Total	295,078

Table 4 Result of C4.5 algorithm for 200 and 30% records

Parameter	200 record	30% record
Accuracy (%)	95.5	95.15
False alarm rate (%)	9.35	9.56
Detection rate (%)	99	99

successfully detected by the IDS; and 4. True negative (TN), when normal behavior that is successfully labeled as normal by the IDS.

Detection rates and false alarm levels measure the accuracy of the intrusion detection system.

The two terms used to calculate the efficiency of the IDS system are: (1) Detection Rate: is the percentage of attacks detected between all attack data, with the following formula: Detection rate $= TP/(TP + TN) \times 100$; (2) False alarm level: is the percentage of normal data that is incorrectly recognized as an attack, with the following formula: False alarm level $= FP/(FP + TN) \times 100$. We get the results with the appropriate values for C4.5 Algorithms for 200 and 30% records are shown below in Table 4.

4.5 Stage of Enhancement

In this phase, the activities include improvements to the system that has been built. Enhancement phase through a series of improvement processes carried out for a number of purposes: (a) Correcting a number of errors found in the previous system implementation (existing system). (b) Add functionality to specific components or the latest additional features to complement the shortcomings in the previous system. (c) Adapting a system that has been built on new platforms and technologies in overcoming a number of developments in new problems that arise. (d) Thus, the repair phase can effectively guarantee the reliability of the performance of the IDS.

5 Conclusions and Recommendations

The Hybrid Intrusion Detection System has functions: detection of known attacks and unknown attacks. A hybrid of the C4.5 Detection and Snort algorithms can increase detection rates 99% of 200 records, and also reduce false alarm levels 9.35% of 200 records from the Intrusion Detection System.

Parameters such as Accuracy, Detection Level, and False Alarm Level is done as comparison tools. In the next study, building an effective intrusion detection model with good accuracy and real-time performance is very important. For this

reason, other techniques are needed from the initial processing and other data mining approaches that can be tested for better detection rates in future research in Hybrid IDS System.

Acknowledgements The research is funded by University Malaysia Pahang, UMP Lab2Market Research Fund (UIC170901). This acknowledgment also goes to the Faculty of Electrical and Electronic Engineering for providing us with facilities to conduct this research.

References

1. Öğütçü G, Testik ÖM, Chouseinoglou O (2016) Analysis of personal information security behavior and awareness. Comput Secur 56:83–93
2. Huang L, Wang X (2016) On the construction of university campus culture under the network environment. In: 3rd international conference on education, management and computing technology (ICEMCT 2016)
3. Chun G, Ping Y, Liu N, Luo S-S (2016) A two-level hybrid approach for intrusion detection. Neuro Comput 214:391–400
4. Gisung K, Seungmin L, Sehun K (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst Appl 41(4 Part 2):1690–1700
5. Peng J et al (2006) A hybrid intrusion detection and visualization system. In: Proceedings of the 13th annual IEEE international symposium and workshop on engineering of computer based systems, p 2
6. Peddabachigari S et al (2007) Modeling intrusion detection system using hybrid intelligent systems. J Netw Comput Appl 30(1):114–132
7. Wang X, Kordas A, Hu L, Gaedke M, Smith D (2013) Administrative evaluation of intrusion detection system. In: Proceedings of the 2nd annual conference on research in information technology, RIIT'13. ACM, NY, USA, pp 47–52
8. Bulajoul W, James A, Pannu M (2013) Network intrusion detection systems in high-speed traffic in computer networks. In: 2013 IEEE 10th international conference on e-Business engineering (ICEBE), pp 168–175
9. Trabelsi Z, Zeidan S (2014) IDS performance enhancement technique based on dynamic traffic awareness histograms. In: IEEE international conference on communications (ICC), pp 975–980
10. Vishnu Balan E, Priyan MK, Gokulnath C, Usha Devi G (2015) Hybrid architecture with misuse and anomaly detection techniques for wireless networks. In: International conference on communications and signal processing (ICCSP)
11. Snapp SR, Brentano J, Dias G, Goan TL, Heberlein LT (2017) DIDS (distributed intrusion detection system)—motivation, architecture, and an early prototype. dl.lib.mrt.ac.lk
12. Tuyikeze T, Pottas D (2010) An information security policy development life cycle. In: Proceedings of the South African information security multi-conference (SAISMC)
13. Kosamkar V, Chaudhari SS (2014) Improved intrusion detection system using C4.5 decision tree and support vector machine. Int J Comput Sci Info Technol 5(2):1463–1467
14. SnortTM Users Manual (2019) http://www.snort.org/. The Snort Project
15. Snort FAQ (2019) http://www.snort.org/. The Snort Project
16. http://kdd.ics.uci.edu/databases/kddcup99 (2019)
17. Wu S-Y, Yen E (2009) Data mining-based intrusion detectors. Expert Syst Appl 36(3):5605–5612
18. Caulkins BD, Lee J, Wang M (2005) A dynamic data mining technique for intrusion detection systems. In: Proceedings of the 43rd annual southeast regional conference, vol 2, ACM, pp 148–153

Hybrid Intrusion Detection

ORIGINALITY REPORT

13%	14%	4%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.atlantis-press.com Internet Source	5%
2	mafiadoc.com Internet Source	3%
3	ijcsit.com Internet Source	3%
4	tel.archives-ouvertes.fr Internet Source	3%

Exclude quotes	On	Exclude matches	< 3%
Exclude bibliography	On		