# Improving Automatic Response Model System for Intrusion Detection System

Dandy Pramana Hostiadi[1], I Made Darma Susila[2]
{dandy@stikom-bali.ac.id[1], darma_s@stikom-bali.ac.id[2]}

STIMIK STIKOM BALI, Bali - Indonesia, Jl. Raya Puputan No. 86 Renon – Denpasar[12]

**Abstract.** Intrusion Detection System is a system used to detect attacks on a network. IDS can be divided into two types: host-based IDS and network-based IDS. NIDS is mostly used because it consist of several sub-network nodes and more easily to control the host. The problems occurred on the NIDS is not integrated yet with system response and consume time to handling the incident. In this research, we proposed a model of automatic response model through SMS sending to handling system based on detection information. The automated response system works by detecting incoming messages from user and executing instructions against the attacked system based on NIDS detection. We use Snort IDS as NIDS machine to produce alert intrusion. The experimental results showed that the proposed model on attack detection mechanism was successed of response of at least 80% based on the instructions of user done through short message send (sms).

**Keywords:** NIDS, automatic response, short message send.

## 1 Introduction

Intrusion Detection System (IDS) is a security system that runs on computer networks. When the intrusion of alert detection founds, it will be classified as a normal traffic or abnormal traffic [1]. There are two types of IDS: Host Intrusion Detection system (HIDS) and Network Intrusion Detection System (NIDS). HIDS work independently on a personal system that applied only to one device or on a single system, while the NIDS runs on the network mode and mostly used to improving the security on network scope [2]. By the concept, IDS is divided into two types: misuse and anomaly-based detection. Here, misuse-based IDS works by using a database of malicious behavior / pattern of attacks. This type of IDS relies on the availability of attack patterns in the database because the system is unable to detect unknown attacks with unknown behaviors. Differently, anomaly-based IDS makes use of the approach of normal activity detection. Each network activity whose pattern is different from the normal one, will be regarded as anomaly.

NIDS commonly used in system security works as a detection of anomaly activities or attacks on the network. When detect an anomaly activities, the NIDS system requires rules of detection mechanism from network administrator to produce the information as alert information detection and need response being attacking happened. This process will be consumed much time when the NIDS produce large numbers of records in a short time. To handle the attacks that occur, manually administrators need time to process the next steps to deal with an attack. Automatic response to the detection of anomaly activities on the IDS system is an automatic handling mechanism performed by the IDS when it detects an attack without

ICCSET 2018, October 25-26, Kudus, Indonesia

DOI 10.4108/eai.24-10-2018.2280575

requiring analysis from the network administrator. In technically, manual interactions on server devices require time to analyzed in the attack handlers that occur [3]. Which is expected in the security system is the existence of a fast handler during an attack on the network. For the example, we can changing the status of the server into sleep or shutdown device to secure the systems being attacking.

In this research, we proposed an automatic response systems model on the intrusion detection machine with the aim to take action when the IDS system detects anomaly on a network. Automatic system executed in real time when detection is obtained, and the security system will provide notification as short message send and sending to administrator. The aim of this system is to reduce the time response from administrators by processing the incoming messages. The network administrator need to replay the message notification as alert information. The message from administrator will be responded as instruction to handling the system without waiting manually access to server. The automatic response system can reduce the activity and time analysis to handling system being attack by shutting down the system.

## 2 Related work

Several studies on IDS snort are performed [4]–[7] to show the snort working on rulebased system and has ability to detect some types of attacks such as U2L, R2L Probe and DoS. Snort is an open source software built by Martin Roesch using the C programming language in 1998 [8]. To construct the rule of the snort, snort has two main parts : the rule header and the rule option[7].

The development of the Intrusion Response System model known as IRs is discussed by comparison of its method [3], [9]. The core of the development of the IRs model [3] explains that a response model mechanism requires an analysis of the relationship between passive, reactive and proactive responses using attack time frames. In his research has not been realizing the real implementation to run the technical response model.

The utilization of short message send, was applied in network security [10]–[14]. Utilization of sms can be useful as a notification system using web interface, as an early warning system with the type of flooding data attacks, or an information system to the network administrator. In its implementation, information from sending alerts is functioning as a security information system and has not been integrated yet into response system in real time mode.

Based on related research that has been done by the researchers, we see an opportunity to build an automatic response model that utilizes a short message send as a medium of communication between systems that run with network administrators to follow up the existence of information attacks provided by NIDS. Our system doesn’t requiring direct physical contact between the server and network administrator in the event of an attack on critical systems, handling has been made directly when the network administrator reply sms notifications attacks.

## 3 Methodology

The use of NIDS in a network must be capable of accurately detecting attacks. One of the commonly used NIDS machines is SNORT. In this study, we use SNORT NIDS to detect attacks. The accuracy of the detection system on SNORT NIDS depends on the rule based that is used and can be developed in accordance with the desirability of attack model detection. The result of SNORT NIDS attack detection is called alerts information.

Automatic response system model is a system that works automatically against the instructions set provided by the network administrator through the sending of messages (short message send) after detection by SNORT NIDS. In the research, the design of automatic response system model seen in Figure 1.

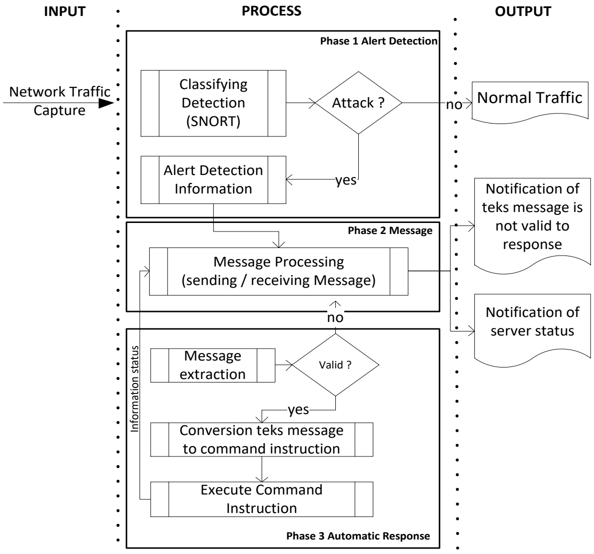

INPUT
PROCESS
OUTPUT
Phase 1 Alert Detection
Network Traffic Capture
Classifying Detection (SNORT)
Attack ?
no
Normal Traffic
Alert Detection Information
yes
Phase 2 Message
Notification of teks message is not valid to response
Message Processing (sending / receiving Message)
no
Notification of server status
Information status
Message extraction
Valid ?
yes
Conversion teks message to command instruction
Execute Command Instruction
Phase 3 Automatic Response

**Fig. 1.** Architecture automatic response system.

In Figure 1 can be explained that the automatic response system architecture consists of 3 Main Phase:

### 3.1 Phase 1 alert detection

In phase alert detection, network traffic from each network will be selected in NIDS. The result of the selection will show that every traffic that runs is an attack or is a normal traffic. When traffic is detected as an illegal activity, NIDS SNORT will issue the result of detection in alert information. In accordance with SNORT working system the results of detected alerts will be stored in database alerts and will be processed in the next phase.

The data stored in the database, carried out the processing of alert information is done by adding information priority attack level. Priority attacks are intended to see how large bytes of data are transmitted and the intensity of activity that occurs in the type of detected attack activity. We count the number of alerts that are similar in the time range of the existing timestamp as the priority level information.

### 3.2 Phase 2 Message

Phase message is the phase used in the process of sending or receiving SMS messages. After the phase detection of attacks stored in the database, the function of the phase message is to retrieve alert information to be sent to the network administrator via sms. Some information taken as sending alert information to network administrators is alert type, Source IP, Destination IP, Port Source, Destination Port, Priority level, and timestamp. After the alert information is received by the network administrator, the network administrator need to replay and send back to the message control system. The text that replied from network administrator will matching with the text pattern known by the message manager section. Known text messaging patterns such as text shutdown, reboot, block ip_src or block port_dst. When the sms sent by the network administrator does not match with the text pattern the messenger manager will notify the network administrator that the sms text pattern is not recognized and administrators must re-send sms in accordance with text patterns known by the messenger manager.

Text patterns that have been identified by the message manager division, will be forwarded to the next phase to be managed as an instructions command, described in phase 3.

### 3.3 Phase 3 automatic response

In phase automatic response, the message that is already known is converted into a command order instruction in accordance with the type of operating system used by the device being attacked. In this study, the attacked device uses the linux operating system, so the text of received messages will be translated according to the command instruction commands known by the linux operating system.

Instructions command that match the commands of the operating system will be forwarded to the attacked device using the phpshell command. Then automatically the attacked device will perform in accordance with instructions submitted by NIDS.

## 4 Result and Discussion

In this study, the network architecture is built and used in realtime. Where automatic response system is integrated on the computer network that runs with the work system described in Figure 2.

To run network scenarios on automatic response systems, some network devices will be used like routers, servers with services running like http and dns, NIDS Server running on SNORT IDS running on raspberry pii, gammu sms gateway and a mobile phone device. System Model Automatic response system is described in the following scenario.

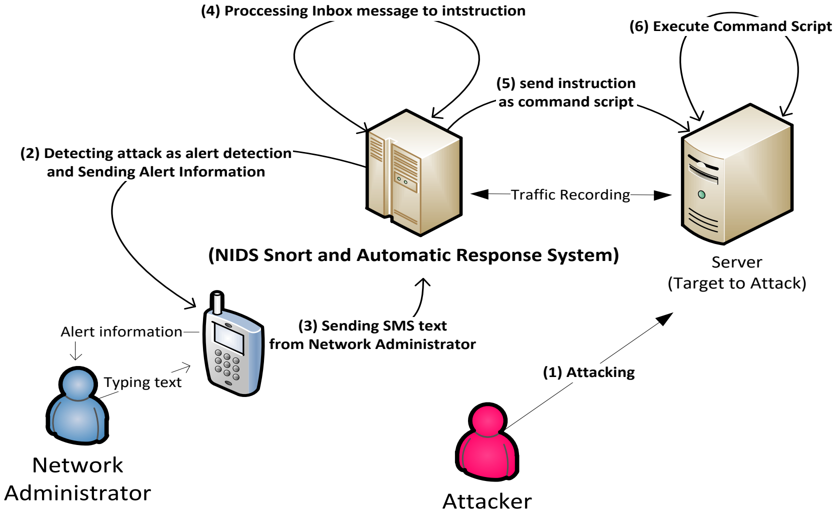

(4) Proccessing Inbox message to intstruction
(6) Execute Command Script
(5) send instruction
as command script
(2) Detecting attack as alert detection
and Sending Alert Information
Traffic Recording
(NIDS Snort and Automatic Response System)
Server
(Target to Attack)
Alert information
(3) Sending SMS text
from Network Administrator
Typing text
(1) Attacking
Network
Administrator
Attacker

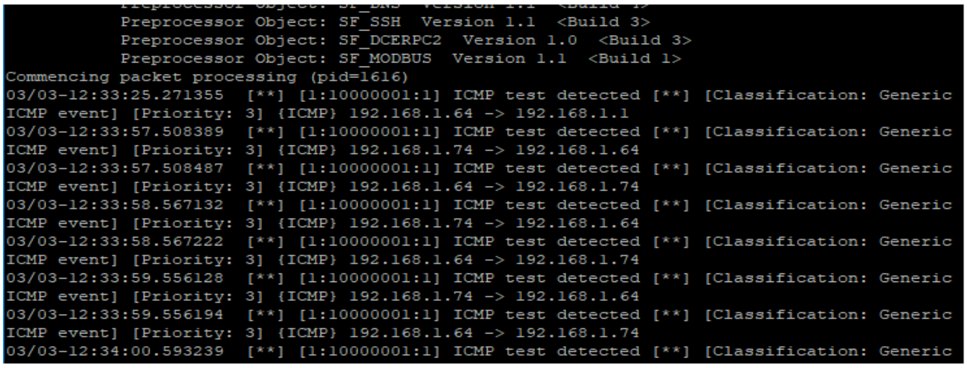
**Fig. 2.** Scenario automatic response system.

SNORT NIDS is one of the most reliable IDS and is often used as a detection of illegal activity in a network or an attack detection in a computer network. SNORT works based on rules that are set according to the desires of detection which will result in detection alerts. One of the simplest rules in SNORT IDS is as follows:

```
alert icmp any any => $HOME_NET any (msg:”TCMP test detected”; GID:1; sid:10000001;
rev:001; classtype:icmp-event:;)
```

```
1 || 10000001 | 001 || icmp-event || 0 || ICMP test detected || url, tools.ietf.org/html/rfc792
```

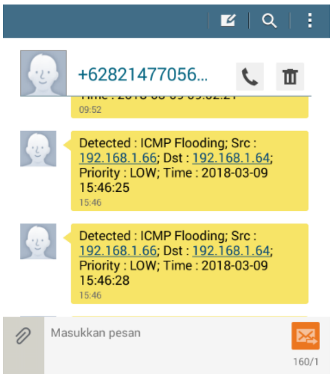
Rules that have been made in the test to check the results of packet selection that runs on the network shown in figure 3.



```
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
Commencing packet processing (pid=1616)
03/03-12:33:25.271355  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.64 -> 192.168.1.1
03/03-12:33:57.508389  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.74 -> 192.168.1.64
03/03-12:33:57.508487  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.64 -> 192.168.1.74
03/03-12:33:58.567132  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.74 -> 192.168.1.64
03/03-12:33:58.567222  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.64 -> 192.168.1.74
03/03-12:33:59.556128  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.74 -> 192.168.1.64
03/03-12:33:59.556194  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
ICMP event] [Priority: 3] {ICMP} 192.168.1.64 -> 192.168.1.74
03/03-12:34:00.593239  [**] [1:10000001:1] ICMP test detected [**] [Classification: Generic
```

**Fig. 3.** SNORT Detection sample.

After SNORT NIDS runs in detecting attack activity by the attacker, the alert information is sent to the network administrator. The administrator will receive alert information shown in Figure 4:



+62821477056...

09:52

Detected : ICMP Flooding; Src : 192.168.1.66; Dst : 192.168.1.64; Priority : LOW; Time : 2018-03-09 15:46:25
15:46

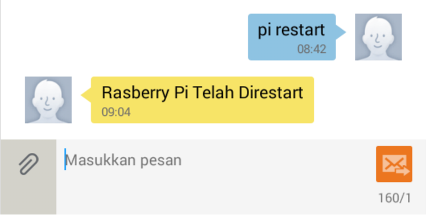
Detected : ICMP Flooding; Src : 192.168.1.66; Dst : 192.168.1.64; Priority : LOW; Time : 2018-03-09 15:46:28
15:46

**Fig. 4.** SMS Alert Information.

In figure 4 shows that the message received by the network administrator has information of alert information. The information is a type of detection in the form of ICMP Flooding, with the IP Source information 192.168.1.66, for the IP destination it is 192.168.1.64 while for the ICMP Flooding the priority is LOW, and the ICMP Flodding Time is March 9, 2018 at 15:46:28 (15th hour, 46th minute, 28th seconds). Information the port address (source port and destination port) is not displayed because the type of attack performed runs in the ICMP protocol. In this research, the example of ICMP Floding attack is done manually with high intensity with command.

```
Ping –l 7000 192.168.1.64 -t
```

To limit the presence of stacked messages, the same message restriction setting in 30 seconds. If there is a different type of attack with the same IP header (ip source and destination IP) information, then the sms will be sent to the network administrator. After receiving SMS alert information, network administrator will answer the message shown in figure 5.

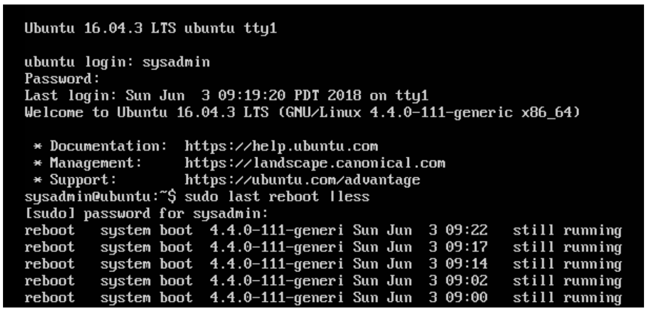


pi restart
08:42

Rasberry Pi Telah Direstart
09:04

**Fig. 5.** Sending SMS text from network administrator.

To make sure the command runs on servers that are the attacker's target, it can be seen in the activity log of the server shown in Figure 6.



```
Ubuntu 16.04.3 LTS ubuntu tty1

ubuntu login: sysadmin
Password:
Last login: Sun Jun  3 09:19:20 PDT 2018 on tty1
Welcome to Ubuntu 16.04.3 LTS (GNU/Linux 4.4.0-111-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage
sysadmin@ubuntu:~$ sudo last reboot |less
[sudo] password for sysadmin:
reboot   system boot  4.4.0-111-generi Sun Jun  3 09:22   still running
reboot   system boot  4.4.0-111-generi Sun Jun  3 09:17   still running
reboot   system boot  4.4.0-111-generi Sun Jun  3 09:14   still running
reboot   system boot  4.4.0-111-generi Sun Jun  3 09:02   still running
reboot   system boot  4.4.0-111-generi Sun Jun  3 09:00   still running
```

**Fig. 6.** Log reboot from server.

In our research, we have to test the automatic response model system by implemented on real network infrastructure that runs in a company for 12 weeks and evaluates against detected attacks. The system will send the alert information to the network administrator in case of an attack. Text messages received by the automatic response system as a text replaying from administrator and converted as command instructions. The evaluation results are shown in table 1.

**Table 1.** Evaluation result.

| **Evaluation Type** | | Alert Detection | Unique Alert | Alert Information (to network administrator) | Receiving text sms (from network administrator) | Success to execute command | Failed execute command | Percentage of successfully execute |
|---|---|---|---|---|---|---|---|---|
| **Time Evaluation (Week)** | **1** | 1254 | 741 | 735 | 735 | 654 | 81 | 88.98% |
| | **2** | 879 | 615 | 582 | 537 | 529 | 8 | 98.51% |
| | **3** | 3023 | 2905 | 2859 | 2859 | 2742 | 117 | 95.91% |
| | **4** | 2766 | 2407 | 2407 | 2407 | 2132 | 275 | 88.57% |
| | **5** | 1940 | 1540 | 1304 | 1304 | 1194 | 110 | 91.56% |
| | **6** | 3431 | 2658 | 2569 | 2569 | 2125 | 444 | 82.72% |
| | **7** | 547 | 523 | 504 | 500 | 437 | 63 | 87.40% |
| | **8** | 2112 | 1743 | 1733 | 1714 | 1471 | 243 | 85.82% |
| | **9** | 427 | 113 | 113 | 113 | 113 | 0 | 100.00% |
| | **10** | 2241 | 1831 | 1723 | 1649 | 1363 | 286 | 82.66% |
| | **11** | 1143 | 893 | 887 | 887 | 859 | 28 | 96.84% |
| | **12** | 862 | 681 | 674 | 639 | 573 | 66 | 89.67% |

The evaluation results show that the automatic response system built has stability in the process of executing the instructions sent by the network administrator with success over than 80%. The success of executing instructions depends on the stability of the sms gateway tool used in receiving and sending sms text. A change in the number of alerts that are detected and sent to the network administrator depends on the density of the process being executed by the messaging manager part and the section converts the message text into the instruction command.

To anticipate the existence of bottle neck process, then in research using time lag process for 1 second when queue process that must be done to send and read sms message, including in its process change full format alert into sms to network administrator.

## 5 Conclusion

Based on the results of research conducted, it can be concluded that the development of automatic response model against IDS detection can be run with stability above 80% in testing time is 12 weeks. Factor failure execution instruction commands sending via sms caused by queue process that runs on automatic response system. The failure process, we assume that coming from the distuingished amount of information alerts sent to the network administrator, the difference between the alert information sent to the network administrator and the answer of the network administrator to be accepted by the automatic response system and the difference of the number of instructions that must executed based on the number of messages received by system

Further research for an automated response system can be developed taking into account the ongoing queue process from the process of sending alert information and the instruction process performed on the basis of received sms can be optimized by scheduling the process. The addition of features that can fix by identifiying unstructure message text format sending from the network administrator and processing as a command instruction by automatic system response model.

### Acknowledgments

The appreciation is delivered to the Department of Computer Engineering STMIK-STIKOM Bali and colleague that has given opportunity to perform the research, and also for all parties that always give support directly and indirectly in the finishing of this scientific journal especially we would like to thank you again.

## References

[1] E. G. Amoroso, Intrusion Detection: An Introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response. Intrusion.Net Books, 1999.

[2] R. U. Rehman, Introduction to Intrusion Detection and Snort. 2003.

[3] N. B. Anuar, S. M. Furnell, M. Papadaki, and N. L. Clarke, “Response Mechanisms for Intrusion Response Systems ( IRSs ),” Proc. SEIN 2009, pp. 3–14, 2009.

[4] S. Khamitkar, “Network Intrusion Detection using SNORT,” Int. J. Eng. Res. Appl., vol. vol.2, no. Issue 2, pp. 1288–1296, 2012.

[5] H. Alnabulsi, M. R. Islam, and Q. Mamun, “Detecting SQL injection attacks using SNORT IDS,” Asia-Pacific World Congr. Comput. Sci. Eng. APWC CSE 2014, no. November, 2014.

[6] B. Sergey, “Intrusion Detection System and Intrusion Prevention System with Snort provided by Security Onion .,” Bachelor’s Thesis Inf. Technol. MAMK Univ. Appl. Sci., no. May, 2016.

[7] N. Khamphakdee, N. Benjamas, and S. Saiyod, “Improving Intrusion Detection System Based on Snort Rules for Network Probe Attacks Detection with Association Rules Technique of Data Mining,” J. ICT Res. Appl., vol. 8, no. 3, pp. 234–250, 2015.

[8] “Snort IDS.” [Online]. Available: https://www.snort.org/.

[9] S. Anwar et al., “From intrusion detection to an intrusion response system: Fundamentals, requirements, and future directions,” Algorithms, vol. 10, no. 2, 2017.

[10] J. September, “SMS Based Information Systems,” 2011.

[11] T. S. Ueng, Z. D. Tsai, and J. C. Chang, “SMS alert system at NSRRC,” Proc. IEEE Part. Accel. Conf., pp. 401–403, 2007.
[12] A. A. A, A. Ademola, and A. A. A, “Development Of An SMS Based Alert Systemusing Object Oriented Design Concept,” vol. 3, no. 5, pp. 71–76, 2014.
[13] E. Kuantama, L. Setyawan, and J. Darma, “Early flood alerts using Short Message Service (SMS),” Proc. 2012 Int. Conf. Syst. Eng. Technol. ICSET 2012, no. September, 2012.
[14] O. Olaleye, A. Olaniyan, O. Eboda, and A. Awolere, “SMS-Based Event Notification System,” J. Inf. Eng. Appl., vol. 3, no. 10, pp. 55–62, 2013.