Utilizing Instant Messaging for Real-Time Notification and Information Retrieval of Snort Intrusion Detection System

Hargyo Tri Nugroho*, Bagas Adi Wicaksono**
* Department of Computer Engineering, Faculty of ICT, Universitas Multimedia Nusantara
** PT. Lyto Datarindo Fortuna, Indonesia

Keywords:

Intrusion Detection System
Snort
Real time
Notification
Instant Messaging
Command Line Interface

ABSTRACT

Snort is widely used Intrusion Detection System (IDS) software for detecting security incidents on the network. The Snort alerts are stored in a database that can be accessed by additional interfaces such as BASE web application. That architecture should be checked periodically to avoid missing an attack. However it is possible, an attack known sometime after the event so that the response may be too late to do. This research aims to build a cheap and reliable solution for Snort reporting system that provides notification of Snort alerts in a real time manner which can be accessed mobile. We utilize an instant messaging application to alert the user and as a command line interface (CLI) that enables user to obtain detail information of each alert sent by the server. Experiment results show that the system is able to send notifications to the user within an acceptable delay interval of 0.87 seconds, on average.

Corresponding Author:

Hargyo Tri Nugroho,
Department of Computer Engineering,
Faculty of Information & Communication Technology,
Universitas Multimedia Nusantara,
Scientia Garden, Jalan Boulevard Gading Serpong, Tangerang, Indonesia
Email: hargyo@umn.ac.id

1. INTRODUCTION

In the last five years, cybercrime is increasing [1]; which includes identity theft, viruses, and system intrusions. Therefore, Intrusion Detection System (IDS) took an important role in detecting intrusions so that can be addressed immediately. Snort [2] is one of the leading Network-Based IDS (Intrusion Detection Software) software with nearly four hundred thousand users. However, Snort does not provide a sufficient GUI (Graphical User Interface) so that the user has to install another application separately such as BASE [3,4] to get a better GUI.

Alert system is used by IDS to notify the system administrator that there was an attack on the system [5]. However, using web based interface like BASE [3], it is possible that users may miss some attacks so that the response become too late to do. Therefore a reporting system with real-time notification is highly needed.

On the other hand, there is a growing use of Instant Messenger. Not only used in desktop, Instant Messenger has been commonly used in mobile devices. Even Instant Messenger is projected to replace Short Message Service for texting [7]. Today, Instant Messenger is also used in many platforms as real time notification [8] as it provides query facility that cannot be easily applied in other communication media [8]. With a query on Instant Messenger, system administrators are able to interact with the system to get the status and condition of the system [9].

Unlike SMS gateway, Instant Messenger only needs an internet connection to connect to the server. Meanwhile, SMS gateway requires cost of each sent message so applying SMS gateway in a system notification need no small cost. So the costs incurred when using Instant Messenger is much less [8].

Yahoo! Messenger is widely used instant messaging protocol with about 22 million users [10] that runs on many mobile operating systems such as BBOS [11], iOS [12] and Android [13]. Based on the facts above, we utilize Yahoo! Messenger as an interactive interface for Snort IDS with real-time notification so that users can obtain intrusion alerts and their detail information in real-time manner.

2. SNORT INTRUSION DETECTION SYSTEM

Snort [2] is the de facto standard of Intrusion Detection System [14]. It is using a *libpcap* based packet sniffer to capture packets on the network [15]. Snort uses pattern matching technique to detect attacks such as buffer overflows, stealth port scans, CGI attacks, SMB probes, and so on. Snort has the capability to provide alerts in various ways such as recording alerts to syslog, Server Message Block "WinPopup" messages, or storing alerts in a separate file.

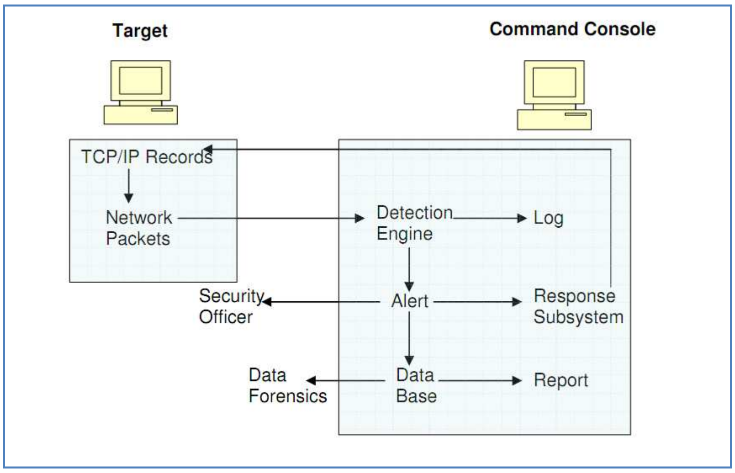

Figure 1. Scheme of Intrusion Detection System [6]

The scheme, as shown in Figure 1, illustrates that the IDS console read each network packet from the target. The packet is then inspected whether the packet is a malicious packet. Each detection result will be stored into the log. If there is finding of a malicious packet, the console will send an alert signal to be sent to the administrator and save this finding into a database that will be used to create reports.

3. YAHOO MESSENGER API

Yahoo! Messenger provides an API (Application Program Interface) to allow developers to build applications integrated to Yahoo! Messenger [16]. To communicate with Yahoo! Messenger, users are required to perform the authorization to perform activities with Yahoo! Messenger. After that, the user will be given a token that has a time limit as identification.

As shown in Figure 2, communication process with Yahoo! Messenger begins with a *request token* requested by the application. When requesting a *request token*, the application also includes *OAuth* key that was obtained in registration. After obtaining a *request token*, the application have to send a response that contains the *request token*, *OAuth* key, and signature key which is a modification of the secret key. After that, the server will give an answer in the form of an *access token* that will be used in the later stages. New session is made by sending the *access token* and other information needed for authentication. Session is used for sending / receiving message.

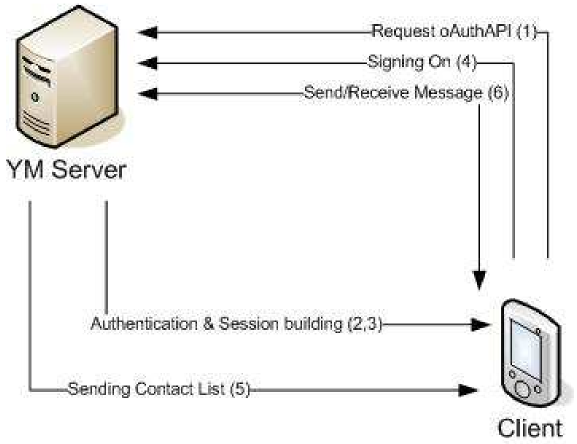

Figure 2. Process flow on Yahoo! Messenger API

4. PROPOSED SYSTEM ARCHITECTURE

As shown in Figure 3, in this architecture, the user may interact with the system using a computer, cell phone, or other device using a Yahoo! Messenger client application. Users will get a notification if there is an attack through Yahoo! Messenger using user ID that has been registered in the system.

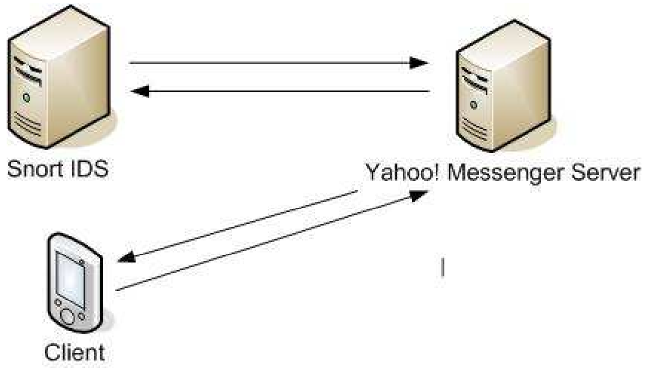

Figure 3. The proposed system architecture

Broadly speaking, this system consists of two engines, namely the notification engine and the interaction/information retrieval engine. The notification engine is a process that handles how the system sends alerts to the user whenever there is an incident detected by Snort. While the interaction engine manages how the users may interact with the system to ask for the details of the attacks which are stored in the Snort database.

4.1. Real-time Notification

In the notification engine, there are two sub-processes. The first sub-process is a trigger that is mounted on table alert in the Snort database. When there is new attack data inserted to the Snort database, the trigger will be run automatically. This trigger will activate the sub-process that served to send real-time notification using incident ID and attack details as the arguments. The message is encrypted using *base64* encryption and then sent to the user by using Yahoo! Messenger API.

4.2. Command Line Interface for Information Retrieval

Compared to the Graphical User Interface, Command Line Interface has some advantages. These advantages include fast and powerful when used by an experienced person, interactions are controlled by the user, minimal interaction with the computer (without the mouse), and can be combined with other interface styles [17].

Command line interface (CLI) in this system is built using simple string processing as only few number of commands are provided. The CLI system employs a listener process to grab the message sent by the user. Message from user is parsed into command and arguments then executed using associated functions.

There are three main commands that can be sent by the user is "showAlert", "setAlert", and "statusAlert". In addition, there is also the command "help" to display help on commands..

The "showAlert" command is used to display n number of alerts that are stored in the database by the latest, day, month, or year. The "setAlert" command is used to enable or disable the real-time notification function. The "statusAlert" command is used to view the status of the notification function whether it is turned on or not.

4. EXPERIMENT RESULTS

To determine the effectiveness of this system, we calculate the amount of delay between the occurrence of the incident until when the notification is received by the user. Testing was conducted using 100 similar attacks from single source of attack and using single client as the recipient of the notification.

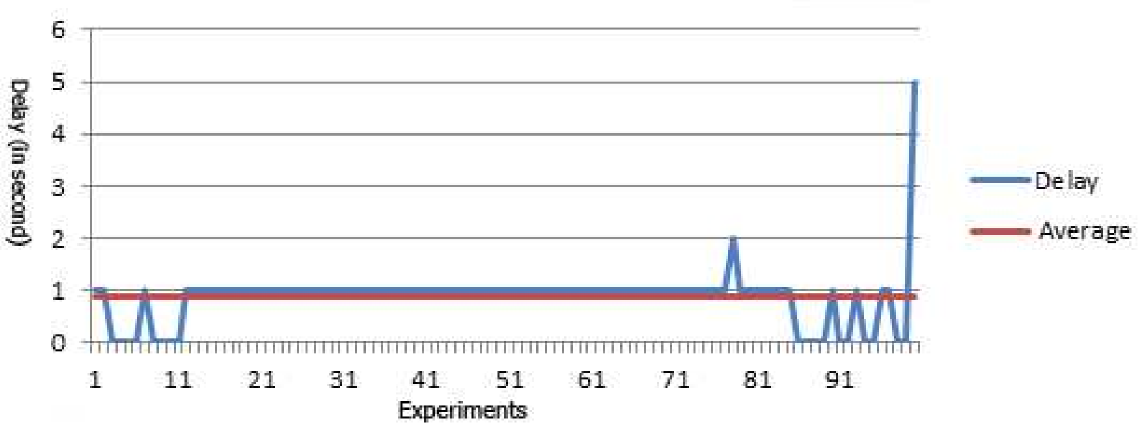

Figure 4. Notification delay

As shown in Figure 4, the delay is 0.87 seconds on average. Although there is no basic standard, the delay can be said to be still in an acceptable interval for system administrators to respond to attacks. However, there are non-uniform delays occurred during the experiment. This phenomenon may occur for several reasons. First, the number of processes running on the client and Snort server may affect the processing time on the client and the Snort server. Second, Yahoo! Messenger servers which manage many requests from users around the world are resulting in fluctuations in processing time on the server. In addition, the Quality of Service of Internet connection affects the transmission delay from the Snort server to Yahoo! Messenger servers and Yahoo! Messenger servers to the client.

5. CONCLUSION

This research produced a reporting system that has real-time notification feature by using Yahoo! Instant Messenger by applying trigger which executes the command to send notifications to users. The experimental results show that this system has a delay at an acceptable interval with an average of 0.87 seconds. This result indicates that instant messaging may be used as a low cost alternative solution for real-time notification and command line interface. However, the use of a proprietary instant messaging application is highly recommended because of some instant messaging applications available today can violate security policies [18,19].

6. FUTURE WORK

This study will be continued by converting the simple string processing to Natural Language Programming to handle more complex commands from command line interface.

REFERENCES

[1] HTCIA, Inc., “2010 Report on Cyber Crime Investigation,” 2010.
[2] “Snort :: About Snort.” [Online]. Available: http://www.snort.org/snort. [Accessed: 05-Nov-2011].

[3] J. Lay, “Comparison of Popular Snort GUIs.” [Online]. Available: http://www.snort.org/assets/187/Snort_Frontend_Compare.pdf. [Accessed: 05-Nov-2011].

[4] “Basic Analysis and Security Engine (BASE) -- Homepage.” [Online]. Available: http://base.secureideas.net/about.php. [Accessed: 05-Nov-2011].

[5] N. Stakhanova, “A Taxonomy of Intrusion Response System,” *Int. J. Information and Computer Security*, vol. 1, 2007.

[6] Meera Gandhi, “Detecting and preventing attacks using network intrusion detection systems”. *Computer Science Journals*.

[7] T. Davey, “Instant Messaging: Functions of a New Communicative Tool.”

[8] Chi-Huang Chiu, Ruey-Shyang Wu, Chi-Io Tut, Hsien-Tang Lin, and Shyan-Ming Yuan, “Next Generation Notification System Integrating Instant Messengers and Web Service,” in *International Conference on Convergence Information Technology, 2007*, 2007, pp. 1781–1786.

[9] Siti Rahayu Abdul Aziz, Adlan Al-Farooq Razalan, Noorhayati Mohamad Noor, and Mohd Suhaimi Sauti, “Proactive notification system using instant messaging bot (IM bot),” in *2010 International Conference on Science and Social Research (CSSR)*, 2010, pp. 695–698.

[10] T. van Lokven, “Review and Comparison of Instant Messaging Protocols,” Radboud University Nijmegen, Nijmegen, 2011.

[11] “BlackBerry - Official BlackBerry - Tablets - Smartphones - Cell Phones - Mobile Phones - Apps at BlackBerry US.” [Online]. Available: http://us.blackberry.com/. [Accessed: 17-May-2012].

[12] “Apple - iOS 5 - 200+ new features for iPad, iPhone, and iPod touch.” [Online]. Available: http://www.apple.com/ios/. [Accessed: 17-May-2012].

[13] “Android.” [Online]. Available: http://www.android.com/. [Accessed: 17-May-2012].

[14] H. T.Nugroho, “Performance Improvement of Intrusion Detection System Using Locality Aware Buffer,” *IEEE Intl Conference on Software Modelling ICSSM 2010*, 2010.

[15] M. Roesch, “Snort - Lightweight Intrusion Detection for Networks.” [Online]. Available: http://assets.sourcefire.com/snort/developmentpapers/Lisapaper.txt. [Accessed: 05-Nov-2011].

[16] “Yahoo! Messenger IM SDK *User* Guide - YDN.” [Online]. Available: http://developer.yahoo.com/messenger/guide/index.html. [Accessed: 05-Nov-2011].

[17] A. Dix, *Human-computer interaction*. Harlow, England; New York: Pearson/Prentice-Hall, 2003.

[18] “Snort SID 3130”. [Online]. Available: http://www.snort.org/search/sid/1-3130. [Accessed: 3-Sept-2013].

[19] “Snort SID 15560”. [Online]. Available: http://www.snort.org/search/sid/15560. [Accessed: 3-Sept-2013].

BIBLIOGRAPHY OF AUTHORS

	Hargyo Tri Nugroho is a lecturer at Department of Computer Engineering, Universitas Multimedia Nusantara, Indonesia. He is actively involved in various open source projects with research interests in information security, semantic web, and embedded systems.
	Bagas Adi Wicaksono holds bachelor degree in Informatics from Universitas Multimedia Nusantara. Currently working as a mobile application developer on PT. Lyto Datarindo Fortuna, Indonesia. He has an interest in bringing information technology to all area of human life.