Improving Intrusion Detection on Snort Rules for Botnet Detection

Saiyan Saiyod^{1,*}, Youksamay Chanthakoummane¹,
Nunnapus Benjamas², Nattawat Khamphakdee²
and Jirayus Chaichawanani¹

¹*Hardware-Human Interface and Communications (H2I-Comm) Laboratory,
Department of Computer Science, Faculty of Science, Khon Kaen University,
Muang, Khon Kaen, Thailand*

²*Advanced Smart Computing (ASC) Laboratory, Department of Computer Science,
Faculty of Science, Khon Kaen University, Muang, Khon Kaen, Thailand
E-mail: {saiyan; nunnapus}@kku.ac.th; {youksamay-c; jirayus.chaichawanani;
k.nattawat}@kkumail.com*

**Corresponding Author*

Received 10 April 2016; Accepted 2 May 2016;
Publication 29 May 2016

Abstract

The Botnets has become a serious problem in network security. An organization should find the solutions to protect the data and network system to reduce the risk of the Botnets. The Snort Intrusion Detection System (Snort-IDS) is the popular usage software protection of the network security in the world. The Snort-IDS utilizes the rules to match the data packets traffic. There are some existing rules which can detect Botnets. This paper, improve the Snort-IDS rules for Botnets detection and we analyze Botnets behaviors in three rules packet such as Botnets_attack_1.rules, Botnets_attack_2.rules, and Botnets_attack_3.rules. Moreover, we utilize the MCFP dataset, which includes five files such as CTU-Malware-Capture-Botnet-42, CTU-Malware-Capture-Botnet-43, CTU-Malware-Capture-Botnet-47, CTU-Malware-Capture-Botnet-49, and CTU-Malware-Capture-Botnet-50 with three rule files of the Snort-IDS rules. The paper has particularly focused on three rule files for performance evaluation

Journal of Software Networking, 191–212.

doi: 10.13052/jsn2445-9739.2016.011

© 2016 River Publishers. All rights reserved.

of efficiency of detection and the performance evaluation of fallibility for Botnets Detection. The performance of each rule is evaluated by detecting each packet. The experimental results shown that, the case of Botnets_attack_1.rules file can maximally detect Botnets detection for 809075 alerts, the efficiency of detection and fallibility for Botnets detection are 94.81% and 5.17%, respectively. Moreover, in the case of Botnets_attack_2.rules file, it can detect Botnets up to 836191 alerts, having efficiency of detection and fallibility for Botnets detection are 97.81% and 2.90%, respectively. The last case Botnets_attack_3.rules file can detect Botnets 822711 alerts, it can 93.72% of efficiency of detection and the value of fallibility is 6.27%. The Botnets_attack_2.rules file is most proficient rule for Botnets detection, because it has a high efficiency of detection for detection and a less of fallibility.

Keywords: Botnets detection, intrusion detection system, MCFP datasets, Snort-IDS.

1 Introduction

Currently, a transaction has mainly exchanged the information through the Internet, such as the purchase, sale and product exchange. However, the data security has become the most significant tool for personal computers, home, school, business, and offices. The events have been serious impact for business operations. If information is stolen, it will damage and the system needs to stop immediately. The current threat of attacks of security forces have increased. Moreover, the threats of the Botnets in different ways have been developed. Therefore, the implementations of business organizations via the Internet have increased the network security to protect the information efficiently. Thus, the attackers produce the information deviation and make network system damage [1].

Botnets or Bot is the robot software that is usually installed on the client computer and run the command preset automatically. For example, IRC (Internet Relay Chat) is Bot of chat rooms and Bot of games online. The Botnets is controlled remotely by hacker. Bot is a small type of computer virus, so users cannot notice that the Bot in computer. These computers are often controlled by hackers via the Internet and send orders through the system on the chat room or IRC [2]. The Botnets are malicious attacking and infecting, sending SPAM, DDoS, Blacklist, Neris, etc. Furthermore, the Bot is remotely managed and synchronization. The synchronization is not only between different Bot, but it is also within the same Bot. For example, the

Bot module sends the SPAM and the module maintains the C&C channel and stops sending packets at the same time when the Bot receives the order of not sending any more mails [3].

Today, an intrusion detection system (IDS) is able to maintain various kinds of techniques in network security. The main task of IDS is to monitor the network traffic data on the network system and analyze it. If any suspicious files alert is when a malicious attacks the system. Besides, the IDS can be presence of signature-base and anomaly-base. Another limit is easy to maintain the network security based on accuracy [4].

This paper, we propose the improvement of the Snort-IDS rules for Botnets detection and utilizing MCFP datasets with rules techniques. The rule techniques analyze the network traffic data pattern, which Snort-IDS are generated according to the network's traffic data behavior.

All the information has been clarified as following details. The Section 2 present related work. The Section 3 presents the background. The Section 4 present system architecture. The Section 5 present evaluation performance. The Section 6 present conclusion and future work.

2 Related Work

Botnets particularly, pose a significant threat to the security on the Internet. As a result, there are many interests in the research community to improve sufficient solutions. In paper [5], the IDS-base and multi-phase IRC-Bot and Botnets behavior detection model are based on C&C reopens message and malicious behaviors of the IRC-Bot inside the network environment. Moreover, a detection method for detecting Botnets is based on behavior features. It's capable of detecting in both known and unknown Botnets and it is less updates in the research fields of Botnet detection [6]. The intrusion detection system was implemented with Snort and configured with WinPcap, within Windows-based environment. It was possible to configure it as a firewall on the Windows. The Snort-IDS rules, however, were not improved [7].

The intrusion detection on the network Botnets attacks were improved through the utilization of the MCFP (Malware Capture Facility project) datasets [3]. The authors proposes three new Botnets detection method and the new model of Botnets behavior, which are based on a deep understanding of the Botnets behavior in the network such as the SimDetec, the BClus method, and the CCDetect. These algorithms can access a better datasets to start showing the particular result. The shift of the detection techniques for behavior base

models has proved to be a better approach to the analyze Botnets pattern. However, the current knowledge of the Botnets detection and the pattern does not have an obvious analysis.

Additional, utilizing of Snort-IDS monitor Web content in a certain time by identifying the abnormal behavior patterns within a campus network security monitoring system [8]. The performance evaluation results generate alert analysis of the Snort-IDS in high-speed network which means that Snort has detected 12 signatures among which detection ICMP PING attacks [9]. In model Snort-IDS structure analyzes the pattern synchronize the protocol or improve speed and accuracy of intrusion detection system in campus network. ACID stand for Analysis Center Intrusion Detection which chosen to shows alarm information [10]. Nevertheless, all research that mention can detect the Botnets, but those research cannot improve the Snort-IDS rules.

3 The Background

We discuss the background of Snort-IDS, the MCFP datasets, and the background of Botnet.

3.1 The Background of Snort-IDS

Snort is useful software for security network. In 1998 Martin Roesch developed the Snort Intrusion Detection System and Intrusion Prevention System (Snort-IDS/IPS) by using C language as an open-source software and lightweight software application. Snort can be installed on numerous platforms of operating systems such as Windows, Linux, etc. Snort has a real time alerting the traffic data network and analyzes capability. The alert will be sent to syslog or a separated 'alert' files, or to popup windows. Snort is logically divided into multiple components. These components working step by step process of detecting particular attacks and to generate output in required format from the detection system. The components of Snort are packet decoder, preprocessors, detection engine, logging and alerting system, and output modules [11]. Snort utilize with rule to alert the network traffic data. The Snort-IDS rules have two logical parts such as the rule header and the rule option [12] as shown in Figure 1.

Figure 1 Structure of Snort rules.

The rule header. The rule header describes attributes of a packet and to command the Snort what to do when it founding the packet that matches the rule as shown in Figure 2.

The rule options. The rule options will follow the rule header and can alert message, information on which part of packet.

The example rule as shown in Figure 3. The network traffic data of TCP protocol for example source address is any, source port is 21, destination address is 10.199.12.8, destination port is any, an generate that outputs the message “TCP Packet is detected” with signature id:1000010.

3.2 The Background of MCFP Datasets

The Malware Capture Facility Project (MCFP) datasets [3, 13]. The MCFP were capture in the CTU University in Czech Republic. The datasets have large size, so they are stored in the server in the university. The goals of datasets were to have a large capture of Botnets traffic mixed with normal traffic and background traffic. The totals of MCFP datasets are 13, but for this research we use 5 datasets.

- CTU-Malware-Capture-Botnet-42 is dataset corresponds to an IRC-based Botnets to send spam for almost six and a half hours and the completed Pcap size is 56 MB, total of Botnets in datasets are 323154.

Action	Protocol	Source Address	Source Port	Direction	Destination Address	Destination Port
--------	----------	----------------	-------------	-----------	---------------------	------------------

Figure 2 Structure of Snort rule header.

Figure 3 Snort rule example.

- CTU-Malware-Capture-Botnet-43 corresponds to an IRC-based Botnets to send spam for 4.21 hours and complete Pcap size as 30 MB, total of Botnets in dataset are 176064.
- CTU-Malware-Capture-Botnet-47, the Botnets in this scenario scanned SMTP (Simple Mail Transfer Protocol) servers for two hours and connected to several RDP (Remote Desktop Protocol) services. However, it does not send any SPAM and attack. The C&C server uses proprietary protocols that connects every 33 seconds and send an average of 5,500 bytes on each connection. It has Pcap size as 5.0 MB and the total of Botnets in dataset are 24764.
- CTU-Malware-Capture-Botnet-49, the Botnets contact a lot of different C&C hosts with Chinese-based IP addresses and the Blacklist DNS. It receives the large amounts of encrypted data and Pcap size as 20 MB and the total of Botnets in dataset are 85735.
- CTU-Malware-Capture-Botnet-50, ten hosts were infected using the same Neris botnet as in scenario 1 and 2. For five hours more than 600 SPAM mails can be successfully sent. It has completes Pcap size as 1.0 GB and the total of Botnets I dataset are 259949.

3.3 The Background of Botnets

Botnets are the technological backbone supporting myriad of attack, including identity stealing, organizational spying, DDoS, SPAM, and government-sponsored attacks. Botnet is a network interface machine which aims to disseminate malicious code over the Internet without user intrusiveness. There are many types of Botnets that shows some serious attacks follow as;

- The Command and Control channel (C&C channels) is the architecture of the C&C Botnets which is a serious attack when the attack would not reveal the name of the attacker. In addition, the infected machines (Bots) receive instructions from C&C and respond it depend on those in striations. The instructions/commands rang from initiating a worm or spam attack over the Internet to disrupt a legitimate user request [14].
- IRC Bot. Many of these IRC Bot is passed by undetected until they become a significant problem. There are several reasons for this. For example they do not follow the same pattern of contagion, some state full firewall both hardware and the application might not alert this traffic, until it is initiated at the client side once the compromise has taken the place. IRC has several forms including Neris, Rbot, SPAM, Virut, NSIS, Menti, Sogou Murlo etc [14].

4 System Architecture

The system architecture, we proposed system consist of the following details; improving of Snort-IDS rules procedure, analysis Botnets, improved Snort-IDS rules, and proposed Snort-IDS rules.

4.1 Improving of Snort-IDS Rules Procedure

The Snort rules evaluation procedure, the MCFP datasets are utilized to test and evaluate detection performance. The datasets were recorded with various amounts of Botnets. In this paper, we utilize .Pcap file type, which contains the traffic data such as source address and destination address, source port and destination port, time to live (TTL), Ack, flags and so on. All attributes that mention are very important parameters for analyzing the attacking type and improving Snort-IDS rules. It would need to increase the perfection of the detection rules and decrease false alert [14] shown as Figure 4.

The Snort-IDS sensor was installed by using the Snort version 2.9.2.2, which runs on the Linux CentOS 64 bit version 6.4 operating system. The

Figure 4 The overall architecture of the improved Snort-IDS rules procedure.

architecture configurations consist of Mysql database, Intel(R) Core(TM)2 Quad 2.66GHz CPU, 4GB DDR-RAM, 250GB HDD and Marvell 88E8071 GB Ethernet 10/100/1000 BaseT [15].

4.2 Botnets Analysis

Botnets analysis is a significant part of the system because it acts as a means to analyze and convert the traffic data of a network. Tokens Matching Algorithm is proposed to analyze the data format to be used for creating the new rules.

- Creating Tokens will reform the structure of the interested in traffic data to be the Tokens form.
- Comparing with Data Dictionary is to compare the created Tokens with Data Dictionary. If the created Tokens is matched by the Data Dictionary, the new will be created by considering the corresponding contents.

Figure 5 Flow chart for Botnets analysis.

Table 1 The set of letters, number, and symbols for creating the tokens

Tokens	Symbols
<c>	a b c d e f g h i j k l m n o p q r s t u v w x y z
<C>	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
<N>	0 1 2 3 4 5 6 7 8 9
<a>	. , - :

Note: Table 1. shows the set of letters, numbers, and symbols that can be compared with contents in traffic data.

```

tokenizer (char) return (token)
    if is_small (char)
        return 'c'
    else if is_big (char)
        return 'C'
    else if is_num (char)
        return 'N'
    else
        return 'a'
    end
end

```

Figure 6 Function of creating tokens.

4.3 Improved Snort-IDS Rules

We compare the analytical information of data packets between each packet connection to dataset with attacking event example type of Botnet in dataset, time for capture, and number of Botnets capture in datasets, etc. [13]

4.4 Proposed Snort-IDS Rules

We explain more detail of Snort-IDS rules which are utilized for Botnet detection.

5 Evaluation of the Performance

In this section, we propose the experimental evaluation of the Snort-IDS rules to compare the detection performance. Evaluation of the performance consist of two procedure are the evaluation of the Snort-IDS rules procedure and detection accuracy comparison of the Snort-IDS rules.

No.	Name	Contents and Tokens									
1	IRC_SPAM	Con =			1	2	1	2	1	2	
		Tok =	c	c							
2	C&C_Botnets	Con =				.	m	s	-	P	
		Tok =	c	c	c	a			a		
3	Possible_NBSS Neris.exe	Con =						.	v	e	
		Tok =	c	c	c	c	c	a			
4	Win32.Domsingx. A contact to C&C server attempt	Con =							-		
		Tok =	c	c	c	c	c	c	a	C	
5	TT-bot botnet contact to C&C server attempt	Con =			-						
		Tok =	c	c	a	C	c	c	c	c	
6	Paleo bot DNS request for C&C attempt	Con =	e	n	g	t	h	:			
		Tok =						a	a	N	
7	Botnets attack Chiness_C&C_ho sts on http	Con =				-					
		Tok =	c	c	c	a	C	c	c	c	
8	Neris_IRC sent to HTTP Attacks	Con =	.	.	.						
		Tok =	a	a	a	N	N	N	N	N	
9	BLACKLIST DNS request for known malware domain www...	Con =								.	
		Tok =	c	c	c	c	c	c	c	a	
10	Trojan.Win32.Shy lock.A contact to C&C server attempt..	Con =					1	2	1	2	
		Tok =	c	c	c	c	a	a	C	c	

Everything is possible or it is anything

Like the contents

Table 2 Data dictionary

```

lookup_datadic (content, token) return (botnet_name_list)
  for i = 1 → botnet_number
    if bn_c [i] == content & bn_t [i] == token
  then
    botnet_name_list += bn_n [i]
  end
end
end
return botnet_name_list

```

Figure 7 Comparing with data dictionary function.

5.1 The Evaluation of the Sort-IDS Rules Procedure

MCFP dataset that has been tested to evaluate performance the intrusion detecting of S. García [3], which has been stored in various file formats. In the datasets will contain Background files, Normal files and Botnet files. In this paper, we choose .Pcap which files have capacity the number of Botnet attacks. The Snort system is installed by utilizing Snort version 2.9.2.2, which runs on the Linux CentOS system [14]. The experiment results of the Snort rules need to modified *snort.conf* file to make it conform to the dataset. We also utilize database mysql for storage the alert data.

In Figure 8, shows the Snort testing procedure. We take command of *Snort -N -r botnet-capture-20110810-neris.pcap -c /etc/snort/snort.conf* to direct Snort-IDS to process dataset file [14]. We explain some command lines, for *-N* is option to make sure that the Snort does not log each packet to the terminal, for *-r* is option which .Pcap must be loaded, and the last one for *-c* option to specify where the config file is located. When the Snort-IDS detected traffic packet which match or synchronize to the Snort rule. The system will generate alert and to bring the recode alert into the database.

Performance evaluation of the Botnets_attack_1.rules, Botnets_attack_2.rules, and Botnets_attack_3.rules files can detect many Botnets such as Neris-IRC, IRC to send spam, Blacklist malware, generic IRC Botnets connection attempt, and malicious URI etc. The numbers of alert are shown in Table 3, Table 4 and Table 5.

Figure 8 Evaluation of the Snort-IDS rules procedure.

Table 3 Number of Snort-IDS rules

No.	Type of Botnets Rules Files	Number of Rules
1	Botnets_attack_1.rules	389
2	Botnets_attack_2.rules	286
3	Botnets_attack_3.rules	273

Note: Improving the Snort-IDS rules already, all rules are saving or recorded in text file format. Nevertheless, there are numerous ways of the Botnets detection. Thus, we categorize the Botnet detection into 3 groups as shown in Table 3.

Table 4 Some example proposed Snort-IDS rules

No.	Type of Botnets Rules
1	<p>Rule 1: alert tcp any any -> any any (msg:"Possible_NBSS Neris.exe SENT TO BROWSER"; flow:to_server,established; content:"ACACACAC"; ttl:128; sid:2000341; rev:1)</p> <p>Rules 2: alert tcp any any -> any any (msg:"Neris_IRC sent to HTTP"; flow:to_server,established; content:"7&_salt="; ttl:128; sid:2000366; rev:1)</p>
2	<p>Rules 1: alert tcp any any -> any any (msg:"TROJAN Possible IRCBot.DDOSCommon Commands"; flow:to_server,established; content:"AAC6F603"; ttl:128; sid:1000255; rev:7)</p> <p>Rules 2: alert udp any any -> any any (msg:"IRC-bot sent spam BROWSER"; flow:to_server,established; content:"FENEBOEC"; ttl:128; sid:1000199; rev:1)</p>
3	<p>Rules 1: alert tcp any any -> any any (msg:"Win32.Domsingx.A contact to C&C server_attempt"; flow:to_server,established; content:"ent-Type"; ttl:128; sid:4000309; rev:2)</p>

Rules 2: alert tcp any any -> any any (msg:"BLACKLIST DNS request for known malware/domain/sxzyong.com";flow:to_server,established;content:"mail.com";fast_pattern:only;/metadata:impact_flagred,service_dns;reference:url,www.virustotal.com/filescan/report.html?id=a7f97ed5c064b038279dbd02554c7e555d97f67b601b94bfc556a50a41dae137-1304614426; classtype:trojan-activity; ttl:128;sid:3000230; rev:2)

Note: Some of examples the Snort-IDS Rules are shown in Table 4. In No. 1, we improved which called "Neris.exe attacks". In this rule allows the Snort-IDS to attack in the Internet network connection. The Snort-IDS will to alert when the attacker scan the computer on the web, which open to any port. In Rules 2, we improve "Neris-IRC" which rule allow to the internet connection when the attacker send to website.

Some of example No. 2, in Rules 1, we improve which call "TROJAN-IRC" which rules allow to Snort-IDS rule when the attacker to send command on the computer. Rules 2, we improved which called "IRCBot". This is rule allow the Snort-IDS network. The Snort-IDS will to alert when the attacker send spam to browser by UTP protocol packets which open any port and it have time to live (TTL) 128.

Some of example No. 3, in Rule 1, we improve alert by TCP protocol packets when the attacker to send "Win32" to C&C server, which open any port and any source address and destination port. In Rule 2, we improve the attacking detection rules for "Botnet BLACKLIST Malware" which is newly added to the dataset. The attacker utilize BLACKLIST DNS request for known malware domain on this website: sxzyong.com.

Table 5 The performance evaluation Botnets_attack_1.rules file

No.	Rules Type	Botnets	Number of Alert	Total Detected Number
1	CTU-Malware-Capture-Botnet-42	IRC_SPAM	192202	296714
		C&C_Botnet	2279	
		Possible_NBSS Neris.exe	9324	
		Neris_IRC sent to HTTP Attacks	92162	
		BLACKLIST DNS	747	
2	CTU-Malware-Capture-Botnet-43	Possible_NBSS Neris.exe	112679	171373
		C&C_Botnet	1403	
		IRC_SPAM	5817	
		Neris_IRC sent to HTTP	51402	
		BLACKLIST DNS	72	
3	CTU-Malware-Capture-Botnet-47	BLACKLIST DNS	23430	24108
		Spyeye bot contact to C&C server	44	
		possible_SPAM	310	
		Neris	98	
		C&C_Botnet	226	
4	CTU-Malware-Capture-Botnet-49	C&C_Botnet	79006	83235
		Virut DNS request for C&C attempt	4034	
		Neris_IRC sent to HTTP	188	

(Continued)

Table 5 Continued

No.	Rules Type	Botnets	Number of Alert	Total Detected Number
5	CTU-Malware-Capture-Botnet-50	IRC_SPAM	102081	
		Neris	131273	
		C&C_Botnet	291	233645
Total				809075

Note: The performance evaluation shows in Table 5 are very efficiency for Botnets detection. The Botnets_attack_1.rules file consist the event of Botnet types which are IRC-bot, BLACKLIST, SPAM, and NBSS Neris etc. The number of detection may not be enough, however the rule file can help administrators know how attacks from Botnets.

5.2 Detection Accuracy Comparison of the Snort-IDS Rules

The procedure of comparing the accuracy of Snort-IDS rule for Botnets detection. The information alert from database will be compared with events of actual invasion. The information is available in the Detection Scoring Truth, Which we used number of Botnet in the MCFP datasets and the Botnets have been alert in our three rule files (Botnets_attack_1.rules, Botnets_attack_2.rules, Botnets_attack_3.rules). The information are compared with experimental information and having the most accurate. We does not include, resolve in this section, more detail shows as Figure 9.

Figure 9 The procedure of comparison from alert.

Table 6 The performance evaluation of Botnets_attack_2.rules file

No.	Rules Type	Botnets	Number of Alert	Total Detected Number
1	CTU-Malware-Capture-Botnet-42	Ozdok botnet communication C&C server	170	311384
		IRC_Bot	84	
		BLACKLIST DNS	29	
		Possible IRC-bot-Sent Spam	145427	
		Neris	165674	
2	CTU-Malware-Capture-Botnet-43	Neris Botnet	74710	169526
		IRC-bot sent spam	94693	
		BROWSER	123	
3	CTU-Malware-Capture-Botnet-47	Botnets attack Chinese_C&C_hosts on http	1522	24522
		BLACKLIST	1522	
		USER-AGENT	9266	
		Spyeye bot contact to C&C server attempt	13589	
		C&C_Botnets	145	
4	CTU-Malware-Capture-Botnet-49	IRC sent_spam	21260	85140
		Gozi Trojan connection to C&C	650	
		IRC_Botnet	62512	
		Virut DNS request for C&C attempt	718	
		Neris	423	
5	CTU-Malware-Capture-Botnet-50	Spyeye bot contact to C&C server attempt	245079	245619
		Neris	36	
		BLACKLIST DNS	81	
		IRC-bot sent spam		
		BROWSER		
Total				836191

Note: The performance evaluation shows in Table 6, this rule has a high accuracy form Botnet attacks. All viruses that detected are the most serious attack on the network system and developing rapidly the attacks. However, these rule file is the best tool to help administrators.

Performance evaluation of fallibility. The fallibility is abnormal traffic which it is incorrectly Botnets detection as normal traffic. The values is lower that indicates better performance, shown in equations [12].

Table 7 The performance evaluation of Botnets_attack_3.rules file

No.	Rules Type	Botnets	Number of Alert	Total Detected Number
1	CTU-Malware-Capture-Botnet-42	Neris_Botnet sent to HTTP	308087	318087
		C&C.Botnets	159	
		IRC sent spam	9841	
2	CTU-Malware-Capture-Botnet-43	Ozdok botnet communication with C&C server attempt	347	170334
		Neris	169987	
		Spyeye bot contact to C&C server attempt	4189	
3	CTU-Malware-Capture-Botnet-47	C&C.Botnets	13569	24189
		BLACKLIST	250	
		USER-AGENT	6181	
		Win32.Domsingx.A contact to C&C	5874	72220
4	CTU-Malware-Capture-Botnet-49	Koobface worm submission of collected data to C&C . . .	66169	
		Botnets attack Chinese_C&C.hosts on http	23	
		Neris	227767	237881
5	CTU-Malware-Capture-Botnet-50	Neris_Botnet sent to HTTP	35	
		BLACKLIST DNS	9875	
		IRC sent spam to HTTP	204	822711
		TT-bot botnet contact to C&C server		
Total				

Note: The performance evaluation shows in Table 7, this is last of the rules for Botnets detection. The Botnets_attack_3.rules is including type of Botnet attack and it has the effective detection. This rule contains the information knowledge of the attack form attacker, this Botnets_attack_3.rules file can answer about every problem at the intrusion.

Table 8 Summary of Botnets detection of three files of rule

No.	Botnets Rules	Number of Botnets Detection
1	Botnets_attack_1.rules	809075
2	Botnets_attack_2.rules	836191
3	Botnets_attack_3.rules	822711

Note: The performances of evaluation shows in Table 8, each rules can be detected more efficient. The Botnets_attack_2.rules files can Botnet detection more than other rules. However, every rule files have important for Botnets detection.

Table 9 Efficiency of detection and fallibility performance of Botnets_attack_1.rules

No.	Datasets	Total Botnets of Datasets	Detected Number	Fallibility (%)	Efficiency of Detection (%)
1	CTU-Malware-Capture-Botnet-42	323154	296714	7.56	92.43
2	CTU-Malware-Capture-Botnet-43	176064	171373	2.66	97.33
3	CTU-Malware-Capture-Botnet-47	24764	24108	2.64	97.35
4	CTU-Malware-Capture-Botnet-49	85735	83235	2.91	97.08
5	CTU-Malware-Capture-Botnet-50	259949	233645	10.11	89.88

Note: In Table 9, efficiency of detection and fallibility performance of each datasets which we utilize Botnets_attack_1.rules file. Each datasets have value of Botnets detection different such as CTU-Malware-Capture-Botnet-47 can detection 97.35%, which this dataset is high efficiency for Botnets detection. Moreover, it has 2.64% of fallibility. In the other hand, the CTU-Malware-Capture-Botnet-50 can detection 89.88%, which this dataset is less efficiency of rules.

Table 10 Efficiency of detection and fallibility performance of Botnets_attack_2.rules

No.	Datasets	Total Botnets of Datasets	Detected Number	Fallibility (%)	Efficiency of Detection (%)
1	CTU-Malware-Capture-Botnet-42	323154	311384	3.64	96.35
2	CTU-Malware-Capture-Botnet-43	176064	169526	3.71	96.28
3	CTU-Malware-Capture-Botnet-47	24764	24522	0.97	99.02
4	CTU-Malware-Capture-Botnet-49	85735	85140	0.69	99.30
5	CTU-Malware-Capture-Botnet-50	259949	245619	5.51	94.48

Note: In Table 10 shows efficiency of detection and fallibility performance of Botnets_attack_2.rules file, which each of datasets have value high and low. The CTU-Malware-Capture-Botnet-49 can Botnets detection higher than other datasets, which it has efficiency of detection 99.30%. In addition, Malware-Capture-Botnet-49 is less of fallibility.

$$\text{Fallibility} = \frac{\text{FN}}{\text{FN} + \text{TP}} \times 100 \quad (1)$$

Performance evaluation of efficiency of detection. The efficiency of detection is indicative of the ability system, which the information has been required from detecting all of data. If a higher value, the system is effective in

Table 11 Efficiency of detection and fallibility performance of Botnets_attack_3.rules

No.	Datasets	Total Botnets of Datasets	Detected Number	Fallibility (%)	Efficiency of Detection (%)
1	CTU-Malware-Capture-Botnet-42	323154	318087	1.56	98.43
2	CTU-Malware-Capture-Botnet-43	176064	170334	3.25	96.74
3	CTU-Malware-Capture-Botnet-47	24764	24189	2.32	97.71
4	CTU-Malware-Capture-Botnet-49	85735	72220	15.76	84.23
5	CTU-Malware-Capture-Botnet-50	259949	237881	8.48	91.51

Note: In Table 11 shows efficiency of detection and fallibility performance of Botnets_attack_3.rules file, which the numbers of rules are lower “273” than other rules. The response of Botnets_attack_3.rules to the datasets reasonably well and the CTU-Malware-Capture-Botnet-42 is well for recompense. It can Botnets detection 98.43% of efficiency of detection. However, this CTU-Malware-Capture-Botnet-42 has value of fallibility in 1.56%.

Table 12 Summary for efficiency of detection and fallibility performance comparison

No.	Rule Files	Total of Fallibility (%)	Total Efficiency of Detection (%)
1	Botnets_attack_1.rules	5.17	94.81
2	Botnets_attack_2.rules	2.90	97.81
3	Botnets_attack_3.rules	6.27	93.72

Note: Efficiency of detection and fallibility of the performance comparison in Table 12, we observe the efficiency of detection on three rule files have difference efficiency to Botnet detection. The Botnets_attack_2.rules file is height efficiency of detection for Botnet detection, it's having 97.81%. Not only, it's having 2.90% of fallibility. In other hand, the Botnets_attack_3.rules of less efficiency of detection for detection, it is having 93.72% and it has value of fallibility 6.27%. However, the values of three rule files in efficiency of detection and fallibility are difference for Botnets detection, but it has helping administrator from attackers.

detection. In the other hand, if a low value that the system is effective a low detecting, shown in equations [12].

$$\text{Efficiency of detection} = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \quad (2)$$

where (TP) True positive is correctly identified of Botnet detection,
 (TN) True Negative is incorrectly identified of Botnet detection,
 (FP) False Positive is correctly rejected of Botnet detection,
 (FN) False Negative is incorrectly rejected of botnet detection.

6 Conclusion and Future Work

The Snort-IDS are effective intrusion detection and it is a network security tool which can monitor the abnormal behavior. This paper, we improve the Snort-IDS rules for Botnets detection and we analyze Botnets behaviors in three rule files. Moreover, we utilize the MCFP dataset, which includes five files such as CTU-Malware-Capture-Botnet-42, CTU-Malware-Capture-Botnet-43, CTU-Malware-Capture-Botnet-47, CTU-Malware-Capture-Botnet-49, and CTU-Malware-Capture-Botnet-50. The performance evaluation of Botnet detection, the performance evaluation of efficiency of detection, and the performance evaluation of fallibility for three rule files were evaluated by detecting each packet. The experimental results shown that, the case of Botnets_attack_1.rules file can maximally detect Botnets detection for 809075 alerts, the efficiency of detection and fallibility for Botnets detection are 94.81% and 5.17%, respectively. Moreover, in the case of Botnets_attack_2.rules file can detect Botnets up to 836191 alerts, having efficiency of detection and fallibility for Botnets detection are 97.81% and 2.90%, respectively. The last case Botnets_attack_3.rules file can detect Botnets 822711 alerts, it can 93.72% of efficiency of detection and the value of fallibility is 6.27%. However, the Botnets_attack_2.rules file is most proficient for Botnets detection, because it has a high efficiency of detection for Botnets detection and it has less of fallibility. Moreover, this paper can support the administrator to secure the network quickly. In addition, they must also regularly update the Snort-IDS rules, because the attackers could find the way to attack the system any time. In the future work, we will find new techniques for intrusion detection system and using tool in data mining for more Botnets detection.

References

- [1] Konhiatou, C. Y., Kittitornkun, S., Kikuchi, H., Sisaat, K., Terada, M., and Ishii, H. (2013). "Clustering Top-10 Malware/Bots based on Download Behavior," In *2013 International Conference On Information Technology and Electrical Engineering (ICITEE)*, Yogyakarta, 62–67.
- [2] Ramsbrock, D., and Wang, X. (2013). *The Botnet Problem*, Chap. 12. Available at: <http://www.sciencedirect.com/science/article/pii/B978012394397200012X>
- [3] García, S., Grill, M., Stiborek, J., and Zunino A. (2014). An Empirical Comparison of Botnet Detection Methods. *Comput Security*, 45, 100–123.

- [4] Sathish, V., and Khader, P. S. A. (2014). Deployment of proposed botnets monitoring platform using online malware analysis for distributed environment. *Indian J. Sci. Technol.* 7, 1087–1093.
- [5] Awadi, A. H. R. A. and Belaton, B. (2013). Multi-phase IRC botnet and botnet behavior detection model. *Int. J. Comput. Appl.* 66, 41–51.
- [6] Li, W. M., Xie, S. L., Luo, J., and Zhu, X. D. (2013). A detection method for botnet based on behavior features *Adv. Mater. Res.* 765–767.
- [7] Shah, S. N. and Singh, M. P. (2012). Signature-based network intrusion detection system using SNORT And WINPCAP. *Int. J. Eng. Res. Technol.* 1, 1–7.
- [8] Geng, X., Liu, B., and Huang, X. (2009). “Investigation on security system for snort-based campus network,” in *Proceedings of the 1st International Conference on Information Science and Engineering (ICISE)*, Nanjing University of Science and Technology, Nanjing, China, 1756–1758.
- [9] Rani, S., and Singh, V. (2012). SNORT: an open source network security tool for intrusion detection in campus network environment. *Int. J. Comput. Technol. Electron. Eng.* 2, 137–142.
- [10] Huang, C., Xiong, J., and Peng, Z. (2012). “Applied research on snort intrusion detection model in the campus network,” in *IEEE Symposium on Robotics and Applications (ISRA)*.
- [11] Roesch, M. (1999). “Snort – Lightweight Intrusion Detection for Networks,” in *Systems Administration Conference*, Washington, USA, 229–238.
- [12] Khamphakdee, N., Benjamas, N., and Saiyod, S. (2015). Improving Intrusion Detection System based on Snort Rules for Network Probe Attacks Detection With Association Rules Technique of Data Mining. *J. ICT Res. Appl.* 8, 234–250.
- [13] <http://mcfp.weebly.com/mcfp-dataset.html> [Accessed May 2015].
- [14] Khamphakdee, N., Benjamas, N., and Saiyod, S. (2014). “Improving Intrusion Detection System based on Snort Rules for Network Probe Attack Detection,” in *International Conference on Information and Communication Technology (Icoict)*, Bandung, 69–74.
- [15] Chanthakoummane, Y., Saiyod, S., and Khamphakdee N. (2015). “Evaluation Snort-IDS Rules for Botnets Detection,” In *National Conference on Infomation Technology*.

Biographies

S. Saiyod received the B.Sc. degree in Computer Science from Mahasarakham University in 2000 and M.Eng. degree in computer Engineering in 2005 and the D.Eng degree in 2011 form King Mongkut's Institute of Technology Ladkrabang, Thailand. His current interests are in the area of performance evaluation on communication networks, digital-signal-processing, and mobile communication.

Y. Chanthakoummane received his B.Eng. degree in Computer Engineering from National University of The Laos PDR in 2009. He is a master's student at the Department of Computer Science, Faculty of Science, Khon Kaen University, Thailand.

N. Benjamas received her B.Sc. degree in Computer Science at the Department of Computer Science, Faculty of Science, Khon Kaen University in 2000 and M.Sc. degree in Computer Science in 2005 and D.Eng. degree

in Computer Engineering in 2012 from Kasetsart University, Thailand. Her current interests are big data analytics, data mining and knowledge discovery, parallel computing, distributed computing, high performance computing (HPC), and cloud computing.

N. Khamphakdee received his B.Sc. Computer Science from Udon Thani Rajabhat University in 2005 and M.Sc. degree in Computer Science in 2015 from Khon Kaen University. He is Ph.D student at Computer Science at the Department of Computer Science, Faculty of Science, Khon Kaen University. His current interests are in the area of data mining, big data, and network security.

J. Chaichawanit received his B.Sc. degree in Computer Science from Khon Kaen University, Thailand in 2014. He is a master's student at the Department of Computer Science, Faculty of Science, Khon Kaen University, Thailand.