Hindawi
Mobile Information Systems
Volume 2017, Article ID 8501976, 13 pages
https://doi.org/10.1155/2017/8501976

Research Article

An Enhancement of Optimized Detection Rule of Security Monitoring and Control for Detection of Cyberthreat in Location-Based Mobile System

Wonhyung Park[1,2] and Byeong Ho Kang[1,2]

[1] *Department of Industrial Security, Far East University, Gamgok-myeon, Eumseong-gun, Chungcheongbuk-do 369-700, Republic of Korea*
[2] *School of Engineering and ICT, University of Tasmania, Private Bag 87, Hobart, TAS 7001, Australia*

Correspondence should be addressed to Byeong Ho Kang; bhkang@utas.edu.au

Received 9 December 2016; Revised 27 June 2017; Accepted 6 July 2017; Published 19 September 2017

Academic Editor: Floriano Scioscia

A lot of mobile applications which provided location information by using a location-based service are being developed recently. For instance, a smart phone would find my location and destination by running a program using a GPS chip in a device. However, the information leakage and the crime that misused the leaked information caused by the cyberattack of mobile information system occurred. So the interest and importance of information security are increasing. Also the number of users who has used mobile devices in Korea is increasing, and the security of mobile devices is becoming more important. Snort detection system has been used to detect and handle cyberattacks but the policy of Snort detection system is applied differently for each of the different kinds of equipment. It is expected that the security of mobile information system would be improved and information leakage would be blocked by selecting options through optimization of Snort detection policy to protect users who are using location-based service in mobile information system environment in this paper.

1. Introduction

The importance of location-based services (LBS), which is a wired and wireless Internet service utilizing current and past location information of users with terminal which can track location, is emphasized due to the development of mobile communication technology and the rapid spread of mobile terminals [1].

The location-based service is a service that identifies the user's location using Location Detection Technology and adds related applications. I think it can be used for various purposes, creating added value using application and location information of wired and wireless Internet.

Also, due to the recent development of cyberattack technology, information leakage as hacking and personal information exposure has become a problem. There is a high concern about exposure of personal information to the current location due to the nature of location information services. In the member information exposed through the online site, personal information such as a name, a resident registration number (ID), an address, and a resident registration number may be used for other purposes through theft. Further, the location information of the customer and the identification of the movement trajectory through the location information may already act as direct privacy violation factors. For this reason, concerns about privacy breaches caused by leakage of location-based services are more serious in Korea [2–4].

The key to security monitoring is rapid detection of cyberattacks. Among the various security monitoring systems, a network-based intrusion detection system (IDS) is the only system that can detect application attacks such as

Table 1: Location-based service utilization [5].

Sort	Field of application	Benefit
(1)	Tracking the location of a young or demented elderly	Missing child prevention, accident prevention
(2)	Tracking your pet location	Lost, accident prevention
(3)	Vehicle navigation	Identifying the route of the vehicle
(4)	Location of field rep	Effective management of field rep
(5)	Providing information about the current location	Nearby information services such as theaters, gas stations, restaurants
(6)	Police, security, military vehicle management	Crime prevention
(7)	Providing location information of courier and cargo	Reducing oil, transportation, and communication costs

web hacking most efficiently by installing them between the control network entrances. The function of the intrusion detection system (IDS) is to use a pattern matching method that detects an attack and generates an alarm when a header or Payload information communicating through the network is detected as an attack.

However, if an attacker encrypts communication signals due to attack packets or malicious code infections, the intrusion detection system (IDS) only checks the encrypted packets. Even if the attack packet is an actual attack packet, it cannot be detected and waypoint also cannot be detected. In order to detect such an attack, it is necessary to develop a behavior-based detection system that can detect and alert an attack using an unknown attack technique instead of a pattern matching methods [6–8].

Currently, security monitoring technology analyzes cyberattack techniques and malicious codes, extracts patterns such as certain strings, and then uses this pattern to develop detection patterns (signature) and apply them to intrusion detection systems. After that, if the cyberattack information matches the detection pattern, it is detected as an accident. If the attack technique is changed, the detection pattern should be corrected in a timely manner so as to maintain the optimized state. However, it is not easy to detect new attacks or malicious codes.

In addition, recent cyberattack techniques such as hacking and distribution of malicious code are developing rapidly and utilizing advanced and intelligent techniques such as double encryption technique to prevent detection by security monitoring or vaccine. So it is not enough to completely detect and block new cyberattacks.

Therefore, in order to efficiently detect and respond to cyberattacks in systems that utilize location-based services in mobile information systems, it is necessary to optimize security monitoring detection techniques to share information among security monitoring centers or to standardize detection patterns according to heterogeneous equipment.

2. Related Work

2.1. Location-Based Service. LBS is an acronym for location-based service. It is generally defined as an application system and service that accurately grasps the location of a person or object based on the mobile communication network and utilizes it. Accordingly, the LBS is a system that grasps the location information of an individual or a vehicle through a mobile communication base station and a GPS (Global Positioning System) and provides various advanced services based on the information [1].

LBS provides various application services based on location information. These include emergency assistance, location information services, traffic congestion and navigation information, and location-based billing. Other applications include Intelligent Transport Systems (ITS), assistive devices for people with disabilities, L-Commerce based on location information, and cell ID-based friends using cell phones (see Table 1).

The current location information acquisition technology of the wireless communication network enables collecting more precise location information by combining the GPS and other location positioning technology and wireless communication network, and it is possible to provide more various application services. As the location information is connected with the mobile communication network, it is possible to provide a general service in the future, and the application service structure provided in the network is changing from a wired/wireless communication network structure with an independent vertical structure to a horizontal structure for wired/wireless integration. Also, all network entities will evolve into an open converged network that provides services based on an equalized All-IP network. Through the development of position location system such as A-GPS (Assistance-GPS) and the paradigm change of ubiquitous and pervasive computing environment, MT (Mobile Terminal) will become a subject of information provision independently and will develop its form to deliver its location information to LBS SP (Service Provider). With these developments, it is necessary to provide the components of location-based services with safety and reliability beyond the conventional wired and wireless network level.

2.2. Intrusion Detection System. Intrusion detection system (IDS) was introduced in 1980 by James Enderson of the United States in a paper called “Computer Security Threat Monitoring and Surveillance.” In 1986, Dorothy Denning published an article entitled “An Intrusion Detection Model” and was influenced by IDS.

Intrusion detection systems can reduce the misuse detection and improve the performance of the system by designing efficient and complete detection rules for cyberthreats. Rules should be as simple and flexible as possible and handle large amounts of network traffic without packet loss. This requires testing procedures to assess the appropriateness before applying the developed rules and periodic optimization to speed up the rules.

For exact detection rules, you must test them before applying them in the intrusion detection system (IDS). Inaccurate rules cause too many false positives and false negatives. A large number of false detection events may cause unnecessary analysis time, prevent detection of normal attack events, or cause the network sensor of the IDS to go down. In order to reduce false detection events, test procedures are required before the system is applied. When testing, efficiency, usability, accuracy, and uniqueness should be considered.

In addition, false positives should be reduced. False positive events occur when you configure detection rules extensively or when you activate unnecessary rules. In order to reduce this, we need to rigorously apply detection rules through precise analysis of the exploit. In addition, it disables the detection rules of the simple information providing format such as “ICMP UNREACH” to reduce the load of the cyberthreat attack event. Inaccurate rules flood false positives and generate false negatives. A large number of false detection events may cause unnecessary analysis time and may prevent detection of cyberthreat attack events.

Until now, the term “security monitoring” has not been defined as a legal rule. In recent years, it has been a step in the process of conceptualization in the academic sense. The term “security control” is used in English as “Security Monitoring” or “Security Monitoring & Control.” The dictionary meaning of “Monitoring” is to protect against various errors that may occur during computer program execution. And the Korean dictionary of the Korean language states that “control” means “to control and control by necessity at a country or an airport” [9] (see Figure 1).

2.3. Intrusion Detection System. The Snort intrusion detection system is one of the most widely used systems among intrusion detection systems (IDS) and is an open source network-based intrusion detection system (open source NIDS) [12–14].

The rule is divided into Header and Option. As shown in (Figure 2), detailed rules can be distinguished as conditions to be detected in the detection operation, protocol type, source address, source port, traffic transmission direction, destination IP address, and destination port. The elements used in these detailed rules are summarized as shown in Figure 2.

The Rule Header of Snort is an integral part of the detection rule that includes five elements: Rule Action, Protocol, Source, Destination IP, Source, Destination Port, and Traffic Direction. Rule Action specifies what the rule should do if the packet matches the rule. Snort has rule actions such as “pass, log, alert,” but in most cases it uses the alert Rule Action [15–18] (see Table 2).

Snort’s rule options are divided into General, Payload Detection, and Nonpayload Detection rule options as shown in Table 3.

3. Optimization of Selected Snort-Based Detection Rule

3.1. Header Detection Rule Optimization. In Rule Action, “alert” generates a warning, “log” leaves a log, “pass” ignores the packet, “activate” sends a warning and activates the specified dynamic rule, and “drop” throws away the packet and leaves a log. Also “reject” leaves the connection and log, and “sdrop” discards the packet and leaves no log. Of these, 6 items including “log,” “pass,” “activate,” “dynamic,” “reject,” and “sdrop” are excluded. For this reason, “log” and “pass” are options for packet logging or packet ignoring. “Activate” and “dynamic” are used mainly for additional logging after detection of attack. They are not suitable for the purpose of notifying the occurrence of attack. “Reject” and “sdrop” are excluded because they are additional actions after interception.

In the protocol, “tcp,” “udp,” “icmp,” and “ip” support the TCP, UDP, ICMP, and IP protocols, respectively. In the protocol, “tcp” supports the TCP protocol, “udp” supports the UDP protocol, “icmp” supports the ICMP protocol, and “ip” supports the IP protocol (see Table 4).

In IP, “any” represents All-IP address targets, “numeric IP” represents a specific IP address target, “numeric IP list” supports up to 10, including CIDR among multiple IP addresses, “CIDR” represents the length object of a specific network address, and “negation(!)” represents All-IP address destinations except the specified IP address. In port, “any” represents all port number targets, “static port” represents fixed port number targets, “ranges(:)” represents port range targets, and “negation(!)” represents all port destinations except for specified ports. In Direction, “-> option” indicates the direction of the destination host from the source host, and “<> option” indicates the direction of both the source host and the destination host. The <- option lowers the detection efficiency by generating a lot of intrusion detection sensor load. Also, “<-” is to remove the mandatory option because it is necessary to use the -> option by changing the source IP and destination IP (see Table 5).

3.2. General Rule Optimization. In General, “msg” is used as an option to indicate a message to be recorded when detecting security control events. “Reference” is a reference to additional information, “gid” is the ID of the alert generation module, sid is used to identify the Snort detection rule, “<100” is the number reserved for future use, “100–1,000,000” indicates the number assigned by Snort, and “>1,000,000” represents a user-defined rule assignment number. “Rev” keyword indicates information about the revision of the sid, “classtype” identifies information that can classify the attack, and “priority” indicates the importance of the rule. In General, all options excluding “msg” are excluded. The “reference” case is excluded as an additional option for reference of detection rule information. “Gid” and “sid” are excluded

Table 2: Definition of Snort Header detection rules [4].

Snort instruction format	Definition
Header	
Rule Action	
alert	Generate Alert
log	Leave log
pass	Ignore pat
activate	Send alerts and activate dynamic rules
dynamic	It is activated by the activate rule and the Log option
drop	Drop a packet and leave a log
reject	Connection terminated and logged
drop	Discard packets and leave no logs
Protocol	
tcp	TCP protocol support
udp	UDP protocol support
icmp	ICMP protocol support
ip	IP protocol support
IP	
any	All IP address
numeric IP	Specific IP addresses
numeric IP list	Multiple IP addresses
CIDR	Specific network class destination (i) Class A Network (8 bits) (ii) Class B Network (16 bits) (iii) Class C Network (24 bits)
negation(!)	All IP addresses except the specified IP address
Port	
any	All port numbers
static port	Fixed Port Number
ranges(:)	Port range destination
negation(!)	All ports except the specified port
Direction	
->	From the origin host to the destination host
<-	Change the source and destination information and specify to "->"
bidirectional(<>)	Bidirectional detection support

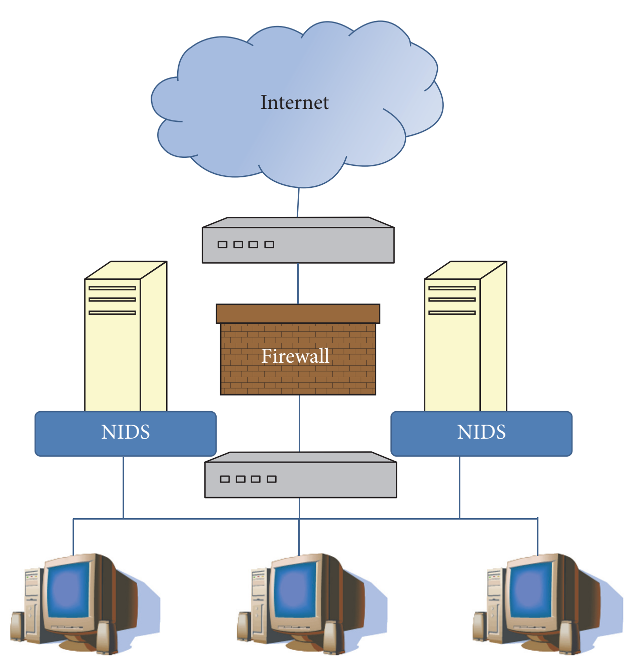

Figure 1: NIDS network [9].

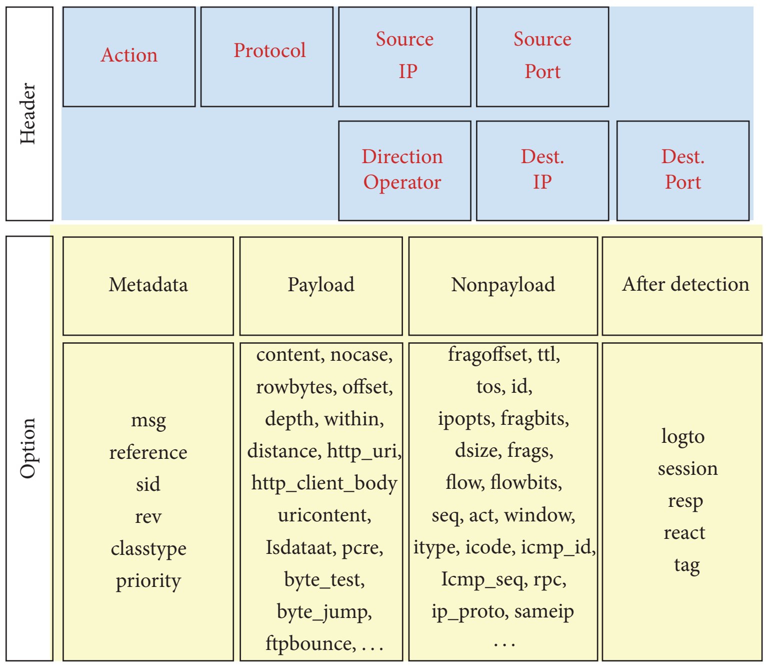

Figure 2: Snort basic rule set [10, 11].

from Snort configuration module as indicating module ID and detection rule ID that generated warning. Also, "rev" is excluded as an option for version control of detection rules, and "classtype" and "priority" are excluded due to lack of usability as an option for sorting and prioritizing detection rules (see Table 6).

3.3. Payload Detection and Nonpayload Detection Rule Optimization. In Payload Detection (Content, Content Modifier), "content" indicates the specific content to be found in the Payload of the packet, and "nocase" means not case sensitive. "Rawbytes" ignores the decoding process and indicates raw packet data inspection, offset indicates the pattern search start position, depth indicates the pattern search range, distance indicates a new pattern search start position after the previous pattern matching, and within indicates the pattern search range. The http_client_body searches in the body part of the HTTP request. The http_cookie searches in the cookie part of the HTTP header. The http_header searches in the HTTP header part. The http_method searches in the HTTP method part. The http_uri part searches the HTTP URI part in the fast_pattern Eye. This is the command to designate the pattern to search first. However, HTTP related commands can be specified with the content option and can be excluded. Fast_pattern excludes string matching as a priority (see Table 7).

In Payload Detection, "uricontent" searches patterns from URI information of HTTP, "urilen" checks HTTP URI length, and "isdataat" searches whether Payload has a certain number of bytes. "Pcre" searches for a regular expression, byte_test compares it to a specific value after a certain byte operation, and "byte_jump" jumps to a result value after a certain byte operation. "Ftpbounce" detects an FTP bounce attack, "asn1" detects a malicious encoding, and "cvs" detects an invalid entry string in CVS. Also, "dce_iface," "dce_opnum," and "dce_stup_data" detect the DCE/RPC request traffic pattern. Of these, "urilen" is excluded because it can be specified using mandatory options, and "ftpbounce," "asn1," "cvs," "dce-iface," "dce-opnum," and "dce_stup_data" commands should be excluded because these are the options for detecting specific attacks on specific services (see Table 8).

Among the Nonpayload Detection options, the commands related to IP such as fragoffset, fragbits, tos, id, ipopts, and TCP related commands seq, ack, and windows are excluded because they are not useful in creating detection rules (see Table 9).

In Nonpayload Detection, "dsize" checks packet payload size to detect packets of abnormal size, and "flow" defines packet direction in relation to client-server communication stream. "Flowbits" is an option to support session-based detection, and "Rpc" acts to identify the rpc service but it is excluded because it can be specified using mandatory options. The "sameip" checks whether the source and destination IPs are the same, and the "stream size" checks the size of the session according to the TCP sequence number, but it is excluded because it can be specified through the "dsize" option. In Rule Thresholds, "Limit" indicates the first occurrence of a warning when a number of identical events occur within a certain time, and "Threshold" indicates a warning when the number of the same events occurring

Table 3: Definition of Snort option detection rules [4].

Snort instruction format	Definition
Option	
General	
msg	Message to record when Alert or logging
reference	References to additional information
gid	Alert generation module id
sid	Use to distinguish snort detection rules
rev	Display information about revision of rule with sid
classtype	Information that can classify an attack
priority	Show the importance (priority) of detection rules
Payload Detection	
content	Specific content looking for in the payload of a packet
content modifier	
nocase	Not classifying capital and small letter
rawbytes	Ignore the decoding process and check the raw packet data
offset	Specify whether to start pattern search after the first few bytes of the packet
depth	Specify how to compare pattern search from offset to how many bytes
distance	Specify whether to start pattern after how many bytes from previous pattern matching.
within	Specify how to compare pattern searches from distance to how many bytes
http_client_body	Search in body part of HTTP request
http_cookie	Search in the cookie portion of the HTTP header
http_header	Search in the HTTP header section
http_method	Search in the HTTP methods section
http_uri	Search in the HTTP URI section
fast_pattern	Specify the pattern to search first
uricontent	Retrieve patterns from URI information in HTTP
urilen	Check HTTP URI length
isdataat	Checks if the payload has a certain number of bytes
pcre	Search by regular expression
byte_test	Compare with specific value after specific byte operation
byte_jump	Jump as much as the operation result value after a certain byte operation
ftpbounce	FTP bounce attack detection
asn1	Detect malicious encoding
cvs	Detect invalid Entry string in CVS
dce_iface dce_opnum dce_stup_data	Detect traffic pattern requesting DCE/RPC
Non-Payload Detection	
IP	
fragoffset	IP fragment offset field check
fragbits	IP fragment offset field check
tos	IP Service type field check
id	IP identification field check
ttl	IP Time To Live field check
ip_proto	IP protocol inspection
ipopts	IP Options field check
TCP	
seq	TCP sequence number check
ack	TCP acknowledge number check
flags	TCP flag bit field check
window	TCP window size check

Table 3: Continued.

Snort instruction format	Definition
ICMP	
itype	ICMP type check
icode	ICMP code check
icmp_id	ICMP identification check
icmp_seq	ICMP sequence number check
dsize	Detect the payload size of packets to detect abnormal size packets
flow	Defines the direction of the packet in relation to the client-server communication stream
flowbits	Options to support session-based detection
rpc	rpc service identification
sameip	Check if origin and destination IP are the same
stream_size	Check the size of the session according to the TCP sequence number
Thresholding	
limit	Only the first warning occurs when multiple identical events occur within a certain time
threshold	Alert when the number of the same events that occur within a certain time is exceeded

Table 4: Optimization of Header Rules: Rule Action, Protocol.

Command format		Selection of detection rule standardization	
Rule Action	alert	Generate a warning	
	drop	Drop the packet and leave a log	
Protocol	tcp	TCP protocol support	
	udp	UDP protocol support	
	icmp	ICMP protocol support	
	ip	IP protocol support	
Command format		Excluded detection rules standardized/excluded reasons	
Rule Action	log	Logged	It is an option for packet logging or packet override, which is mainly used for logging after attack detection, but it is for the purpose of notifying the occurrence of an attack
	pass	Ignore packets	
	activate	Send an alert and activate the specified dynamic rule	This option is used for additional logging after detection of an attack, but it is consistent with the purpose of notifying the occurrence of the attack
	dynamic	It is activated by the activate rule and acts like the log option	
	reject	Connection terminated and logged	Added after Intrusion rrevention and exclude as action
	sdrop	Discards packets and leaves no logs	

Table 5: Optimization of Header Rules: IP, Port, Direction.

Command format		Selection of detection rule standardization	
IP	any	All IP address	
	numeric IP	Specific IP addresses	
	numeric IP list	Multiple IP address up to 10 including CIDR	
	CIDR	The length of a specific network address.	
Port	any	all port numbers	
	static port	Fixed Port Number	
	ranges(;)	Port range destination	
Direction	->	Direction from the origin host to the destination host	
	<>	Origin host and destination host bidirectional	
Command format		Excluded detection rules standardized/excluded reasons	
Direction	<-	Source Host and Destination Host Reverse	It is excluded because it can be made by changing source IP and destination IP and generate load

Table 6: Optimization of General Rules.

Command format	Selection of detection rule standardization		
General	msg	Message to record when detecting	
Command format	Excluded detection rules standardized/excluded reasons		
General	reference	References to additional information	Excluded as an additional option for reference of detection rule information
	gid	Alert generation module id	Except for the module ID of the configuration module and the ID of the detection rule (Snort-specific function)
	sid	Use to distinguish Snort detection rules <100 reserved number for future use 100–1,000,000 number assigned by Snort >1,000,000 custom rule assignment numbers	
	rev	Information on revision of rules with sid	Excluded as an option for versioning of detection rules
	classtype	Information that can classify an attack	Excluded as an option for risk display and classification of detection
	priority	Significance of detection rules (top/middle/bottom)	Exclude as an option for indicating the importance of detection rules

Table 7: Optimization of Payload Detection (Content, Content Modifier) Rules.

Command format	Selection of detection rule standardization		
Payload Detection	content	Specific content to look for in the payload of a packet	
	nocase	Case insensitive	
	rawbytes	Ignore the decoding process and check raw packet data	
	offset	Pattern search start position (after the first few bytes of the packet)	
	depth	Pattern search range (compare pattern search from offset to several bytes)	
	distance	New pattern search start position after a previous pattern match (after a few bytes)	
	within	Pattern search range (compare pattern search from distance to several bytes)	
Command format	Excluded detection rules standardized/excluded reasons		
Payload Detection	http_client_body	Search in body part of HTTP request	Except for the content option
	http_cookie	Search in the cookie portion of the HTTP header	
	http_header	Search in the HTTP header section	
	http_method	Search in the HTTP methods section	
	http_uri	Search in the HTTP URI section	
	fast_pattern	Specify the pattern to search first	Excluded as string matching from specified priority

Table 8: Optimization of Payload Detection Rules.

Command format		Selection of detection rule standardization	
Payload Detection	isdataat	Check if the payload has a certain number of bytes	
	pcre	Search by regular expression	
	byte_test	Compare with specific value after specific byte operation	
	uricontent	Search patterns from URI information in HTTP	
Command format		Excluded detection rules standardized/excluded Reasons	
Payload Detection	urilen	Check HTTP URI length	Excluded as assignable opting using mandatory option
	ftpbounce	FTP bounce attack detection	Excluded as assignable opting using mandatory option
	asn1	Detect malicious encoding	
	cvs	Detect invalid Entry String in CVS	
	dce_iface	DCE/RPC request traffic pattern detection	
	dce_opnum		
	dce_stup_data		

Table 9: Optimization of Nonpayload Detection Rules 1.

Command format		Selection of detection rule standardization	
Nonpayload Detection (IP)	ttl	Inspect IP Time-To-Live field	
	ip_proto	Inspect IP protocol field	
Nonpayload Detection (TCP)	flags	Inspect TCP flag bit field	
Nonpayload Detection (ICMP)	itype	Inspect ICMP type	
	icode	Inspect ICMP code	
	icmp_id	Inspect ICMP identification field	
	icmp_seq	Inspect ICMP sequence number	
Command format		Excluded detection rules standardized/excluded reasons	
Nonpayload Detection (IP)	fragoffset	Inspect IP fragment Offset field	It is excluded through consultation with related companies, Because it is not useful in creating detection rule
	fragbits	Check whether IP fragmentation and reserved bits are set	
	tos	Inspect IP Service type field	
	id	Inspect IP identification field	
	ipopts	Inspect IP Options field	
Nonpayload Detection (TCP)	seq	Inspect TCP Sequence number	
	ack	Inspect TCP acknowledge number	
	window	Inspect TCP window size	

Table 10: Optimization of Nonpayload Detection Rules 2.

Command format	Standardization of detection rules Candidates for selection		
Nonpayload Detection	dsize	Packet detection of abnormal size by checking the packet's payload size	
	flow	Defines the direction of the packet in relation to the client-server communication stream	
	flowbits	Options to support session-based detection	
Rule Thresholds	Limit	Alert for the first time when multiple identical events occur within a certain time	
	Threshold	Alert when the number of the same events that occur within a certain time is exceeded	
Command format	**Excluded detection rules standardized/excluded Reasons**		
Nonpayload Detection	rpc	Identify the rpc service	It identifies the rpc service, but it can be specified using mandatory options. It can be specified through the dsize option.
	sameip	Check if origin and destination IP are the same	
	stream size	Check the size of the session according to the TCP sequence number	

within a certain time exceeds the corresponding number. Threshold option was used before Snort 2.8.5 version; Snort 2.8.5.1 or later uses Detection Filter or Event Filters option (see Table 10).

4. Comparison Analysis of Existing Snort Detection Options

Optimizing the existing Snort detection grammar will allow the user to understand and analyze the wrong type of policy created without considering the performance and false positives of the detection sensor in the event of a vulnerability attack. Also this can elaborate detection rules. In order to normalize the detection rules; first, if short strings are applied, frequent detection of the intrusion detection system sensor occurs, thereby degrading the performance of the intrusion detection system sensor. Therefore, it is necessary to create a policy that detects a string of at least 4 bytes or more. Second, when a communication string is detected frequently, a large number of detection events are generated, which may cause a false alarm, and the performance of the detection sensor may be reduced, thereby limiting communication traffic in a typical Internet environment. Third, in the PCRE grammar, . (Dot), $*$ (Asterisk) is a special character that matches any string. Because this matching matches all strings in the packet Payload, the PCRE computation consumes a lot of system resources and leaks from the intrusion detection system sensor. Fourth, if the setting value exceeds the detection string length limit of the intrusion detection system sensor, it may cause a problem that it cannot be detected. In addition, long length PCRE matching causes performance load of the intrusion detection system sensor. Fifth, when searching a continuous pattern of the same character, a looping phenomenon may occur as repeated operations are performed, which causes a heavy load on the CPU usage. Therefore, there is a purpose to improve these five problems by optimizing Snort detection grammar (see Table 11).

5. Conclusion

The purpose of this paper is to find a detection rule optimization method for protecting users who use location-based services in mobile information systems and proving the compatibility of detection rules between different intrusion detection systems (IDS/IPS) introduced in each security control center (cybersafety center) based on IDS Snort in order to prepare for new cyberthreats and cyberattacks.

Recent hacking technologies understand cyberattack packet contents in order to detect new cyberthreats that are developing rapidly and present the best intrusion detection rules for network environment. Based on the Snort detection rules, we designed the models and options of the essential detection rules and suggested the most optimized detection rule production standards through understanding and analyzing the wrong policies such as the performance of the detection sensor and the policy that does not consider the false positives. In this paper, we propose an efficient detection and countermeasure of new cyberattacks through the Snort-based detection rule standard requirements. Also, constructing a standardized security management system of the heterogeneous intrusion detection system by maintaining the optimization state by correcting and revising the detection pattern according to the actual situation of each security control center is possible.

This standardization of integrated intrusion detection pattern is expected to establish an efficient operation system of each security control center (cybersafety center) performing security control.

Table 11: Comparison of Snort Detection Rules and Optimization Options.

Detection rule options	Snort	Detection rule optimization grammar selection
Header (24/17)		
Rule Actions (8/2)		
alert	O	O
log	O	X
pass	O	X
activate	O	X
dynamic	O	X
drop	O	O
reject	O	X
sdrop	O	X
Protocols (4/4)		
tcp	O	O
udp	O	O
icmp	O	O
ip	O	O
IP (5/5)		
any	O	O
numeric IP	O	O
numeric IP list	O	O
CIDR	O	O
negation(!)	O	O
Port (4/4)		
any	O	O
static port	O	O
ranges(:)	O	O
negation(!)	O	O
Direction (3/2)		
->	O	O
<-	O	X
bidirectional(<>)	O	O
Option (47/24)		
Meta Data (6/1)		
msg	O	O
reference	O	X
sid	O	X
rev	O	X
classtype	O	X
priority	O	X
Payload Detection (19/12)		
content	O	O
content modifier		
Nocase	O	O
Rawbytes	O	O
Depth	O	O
Offset	O	O
Distance	O	O
Within	O	O
http_client_body	O	X
http_uri	O	X

Table 11: Continued.

Detection rule options	Snort	Detection rule optimization grammar selection
http_header	O	X
http_cookie	O	X
uricontent	O	O
isdataat	O	O
pcre	O	O
byte_test	O	O
byte_jump	O	O
ftpbounce	O	X
asn1	O	X
regex	O	X
Non Payload Detection (20/9)		
fragoffset	O	X
ttl	O	O
tos	O	X
id	O	X
ipopts	O	X
fragbits	O	X
dsize	O	O
flags	O	O
flow	O	O
flowbits	O	O
seq	O	X
ack	O	X
window	O	X
itype	O	O
icode	O	O
icmp_id	O	O
icmp_seq	O	O
rpc	O	X
ip_proto	O	X
sameip	O	X
Thresholding (2/2)		
limit	O	O
threshold	O	O

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] A. Dey, J. Hightower, E. De Lara, and N. Davies, "Location-based services," *IEEE Pervasive Computing*, vol. 9, no. 1, pp. 11-12, 2010.

[2] R. Bejtlich, "The practice of network security monitoring: understanding incident detection and response," *No Starch Press*, pp. 2–20, 2013.

[3] J. S. Hong, Y. H. Lim, W. H. Park, and K. H. Kook, "Improved Security Monitoring and Control Using Analysis of Cyber Attack in Small Businesses," *The Journal of Society for e-Business Studies*, vol. 19, no. 4, pp. 195–204, 2014.

[4] W. Park and S. Ahn, "Performance Comparison and Detection Analysis in Snort and Suricata Environment," *Wireless Personal Communications*, vol. 94, no. 2, pp. 241–252, 2016.

[5] T. Xu and Y. Cai, "Feeling-based location privacy protection for location-based services," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 348–357, ACM, Chicago, Ill, USA, November 2009.

[6] M. Roesch, *Snort: Lightweight Intrusion Detection for Networks*, vol. 229, Stanford Telecommunications Inc, Santa Clara, Calif, USA, 1999.

[7] Y.-H. Kim and W. H. Park, "A study on cyber threat prediction based on intrusion detection event for APT attack detection,"

Multimedia Tools and Applications, vol. 71, no. 2, pp. 685–698, 2014.

[8] M. Roesch, *Snort: Lightweight Intrusion Detection for Networks*, vol. 229, Santa Clara, Calif, USA, Stanford Telecommunications, Inc, 1999.

[9] Z. Zhou, Z. Chen, T. Zhou, and X. Guan, "The study on network intrusion detection system of snort," in *Proceedings of the 2nd International Conference on Networking and Digital Society, lCNDS 2010*, pp. 194–196, Wenzhou, China, May 2010.

[10] M. Norton and D. Roelker, *SNORT 2.0: Hi-Performance Multi-Rule Inspection Engine*, Sourcefire Network Security Inc, 2002.

[11] P. Garcia-Teodoro et al., *Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges*, computers & security 28.1, 2009.

[12] G. C. Tjhai, M. Papadaki, S. M. Furnell, and N. L. Clarke, "Investigating the problem of IDS false alarms: An experimental study using Snort," *IFIP International Federation for Information Processing*, vol. 278, pp. 253–267, 2008.

[13] J. D. Rance, *Structured Exception-Handling Methods, Apparatus, and Computer Program Products*, Los Gatos, Calif, USA.

[14] S. Chakrabarti, M. Chakraborty, and I. Mukhopadhyay, "Study of snort-based IDS," in *Proceedings of the International Conference and Workshop on Emerging Trends in Technology 2010, ICWET 2010*, pp. 43–47, ind, February 2010.

[15] B. Caswell, J. Beale, and A. Baker, *Snort IDS and IPS Toolkit*, Syngress, New York, NY, USA, 2007.

[16] D. Burks, *Security Onion: Peel Back the Layers of Your Network in Minutes*, Software Engineering Institute, January 2014.

[17] A. Deuble, *Detecting and Preventing Web Application Attacks with Security Onion*, SANS Institute 4.1, 2012.

[18] P. Wonhyung, *Requirements of Detection Rules in Intrusion Detection System based on SNORT*, Telecommunications Technology Association in South Korea, 2015.