

# Tree Rule Firewall

A Thesis Submitted for the Degree of  
Doctor of Philosophy

By

Thawatchai Chomsiri

in

Faculty of Engineering and Information Technology  
UNIVERSITY OF TECHNOLOGY, SYDNEY

17<sup>th</sup> November 2016

Copyright by Thawatchai Chomsiri, 2016

# CERTIFICATE

Date: 17<sup>th</sup> November 2016

Author: Thawatchai Chomsiri

Title: Tree Rule Firewall

Degree: Ph.D.

I certify that this thesis has not already been submitted for any degree and is not being submitted as part of candidature for any other degree.

I also certify that the thesis has been written by me and that any help that I have received in preparing this thesis, and all sources used, have been acknowledged in this thesis.

-----  
Signature of Author

# Acknowledgements

I am greatly indebted to my supervisor, Professor Xiangjian (Sean) He, for his scholarly guidance, inexhaustible patience, constant encouragement, and invaluable suggestions. He is a nice, generous, helpful and hospitable person. I feel happy and comfortable with him during my PhD study. Many thanks are also expressed to my co-supervisors, Dr. Priyadarsi Nanda, for his valued suggestions and constantly technical supports. I appreciatively acknowledge the useful suggestion from Dr. Zhiyuan (Thomas) Tan. His suggestions help me to improve many paper publications of my research. I also gratefully acknowledge useful discussions with my team mainly consisting of Dr. Mian Ahmad Jan, Dr. Mohammed Ambusaidi and Dr. Aruna Jamdagni. I appreciate the supports, which I received from the Faculty of Engineering and Information Technology, and its staff. I appreciate the financial assistance, International Research Scholarship (IRS), which I have received from Graduate Research School - University of Technology Sydney. Lastly, I would like to thank my father (Mr. Weera Chomsiri), my mother (Ms. Darawadee Chomsiri) and members of my family for everything.

# Table of Contents

<b>List of Tables.....</b>	<b>vii</b>
<b>List of Figures .....</b>	<b>viii</b>
<b>Abbreviation .....</b>	<b>x</b>
<b>Abstract .....</b>	<b>1</b>
<b>Chapter 1 Introduction .....</b>	<b>4</b>
1.1 Background and Motivation .....	5
1.1.1 Firewall in General Terms .....	5
1.1.2 Packet Filtering Firewall.....	6
1.1.3 Stateful Firewall.....	8
1.1.4 Anomaly Detection .....	9
1.1.5 Motivation.....	11
1.2 Objectives .....	12
1.3 Contributions and Innovation .....	14
1.4 Structure of the Thesis .....	15
<b>Chapter 2 Related Works .....</b>	<b>16</b>
2.1 Rule Conflict Identifications and Firewall Tools .....	16
2.2 Hierarchical and Tree Models in Previous Works.....	18
2.3 Stateful mechanism inside IPTABLES .....	20
2.4 Summary.....	23
<b>Chapter 3 Limitations of Listed-Rule Firewalls .....</b>	<b>25</b>
3.1 Background and Related Works .....	25
3.2 Limitations of Listed-Rule Firewall .....	26
3.2.1 Limitations on the shadowed rule.....	29
3.2.2 Limitation about swapping position between rules .....	31
3.2.3 Limitation about redundant rules .....	32

3.2.4	Limitation of rule design.....	34
3.2.5	Limitation from sequential computation.....	36
3.3	Conclusion.....	37
<b>Chapter 4</b>	<b>Tree-Rule Firewall .....</b>	<b>38</b>
4.1	Background and Related Works .....	38
4.1.1	Firewall background .....	39
4.1.2	Firewall on cloud environment.....	40
4.1.3	Chapter organization.....	42
4.2	Design and implementation of the Tree-Rule firewall .....	42
4.2.1	Basic design .....	42
4.2.1.1	Time complexity of the basic design .....	45
4.2.1.2	Additional benefits of the basic design.....	46
4.2.2	Improvement of basic design .....	47
4.2.2.1	Time complexity of improved design .....	48
4.2.3	Implementation .....	50
4.3	Experimental results .....	52
4.3.1	Testing in LAN .....	52
4.3.2	Testing on cloud environment .....	54
4.3.2.1	Testing on ESXi.....	55
4.3.2.2	Testing on Hyper-V .....	56
4.3.2.3	Testing analysis.....	56
4.4	Conclusion .....	60
<b>Chapter 5</b>	<b>A Stateful Mechanism for the Tree-Rule Firewall.....</b>	<b>61</b>
5.1	Background and Previous Works .....	63
5.1.1	Traditional firewall (Listed-rule firewall).....	63
5.1.2	The Tree-Rule firewall.....	65
5.1.3	Packet filtering and Stateful firewalls.....	66
5.2	Design and Analysis .....	68
5.2.1	Using one node per connection.....	68

5.2.2 Expanding Hashing Table size vertically .....	73
5.2.3 Using one node per bucket.....	75
5.2.4 Verifying non-first packets using Tree Rule before Hashing Table .....	76
5.2.5 Use of Static Node and Label to identify free nodes .....	78
5.2.6 Using the Label for Time Out instead of Timer Object.....	80
5.3 Firewall Implementation and Experimental Analysis .....	81
5.4 Conclusion .....	85
<b>Chapter 6 Hybrid Tree-rule Firewall for High Speed Data Transmission .....</b>	<b>86</b>
6.1 Background and Related Works .....	89
6.1.1 Enhancing processing speed via rule conflict elimination.....	90
6.1.2 Enhancing processing speed via hardware implementation .....	92
6.1.3 Enhancing processing speed via advanced filtering mechanisms.....	93
6.1.4 Background of Tree-Rule firewall .....	95
6.2 Our Approach .....	96
6.2.1 Methodology.....	97
6.2.2 Discussion on efficiency .....	101
6.2.3 A mathematical model for measuring time consumption .....	105
6.2.4 Determining time interval $w$ .....	108
6.3 Implementation and Experimentation.....	110
6.3.1 Experimental setup and environment .....	110
6.3.2 Experiments .....	112
6.4 Conclusion .....	120
<b>Chapter 7 Conclusion .....</b>	<b>122</b>
7.1 Summary.....	123
7.2 Future Work.....	126
<b>Bibliography .....</b>	<b>128</b>

# List of Tables

3.1 An example of rules on the Listed-Rule Firewall .....	26
3.2 An example of rules on a medium size network .....	35
4.1 An example of rules on the Listed-Rule Firewall .....	39
4.2 An example of rules on a medium size network .....	45
4.3 Throughput comparison between Tree-Rule firewall and IPTABLES on ESXi and Hyper-V .....	55
4.4 Capability comparison among the firewalls on cloud environment .....	59
5.1 An example of rule in Listed-rule firewall.....	64
5.2 Time consumption of packet decision on Tree-Rule firewall and Netfilter / IPTABLES .....	83
6.1 A set of listed rules created for an example network in Figure 6.1 .....	91
6.2 Example of a listed rule transformed from a rule path .....	99
6.3 Examples of two listed rules transformed from a rule path .....	99
6.4 The listed rules transformed from the tree rule in Figure 6.2 .....	100
6.5 Time consumption for transferring files from servers to clients (minutes) .....	115
6.6 Time saves in percentage .....	116
6.7 Speed achieved through IPTABLES .....	117
6.8 Speed of proposed firewall without 'Automatic rule sorting' .....	118
6.9 Speed of proposed firewall with 'Automatic rule sorting' .....	119

# List of Figures

1.1 A stateful mechanism in IPTABLES firewall .....	9
2.1 A stateful mechanism in IPTABLES firewall .....	22
3.1 The 2D-Box Model (left) and the rules of a Listed-Rule firewall (right) .....	27
3.2 A medium size network with DMZ .....	36
4.1 Firewall models in cloud environment .....	41
4.2 A basic Tree-Rule firewall structure .....	43
4.3 Improved design of a Tree-Rule firewall using IP address ranges and port ranges .....	48
4.4 Slightly increased $t$ when a range of number is applied .....	49
4.5 Implementation of Tree-Rule firewall .....	50
4.6 CPU load of firewalls .....	53
4.7 Throughput of firewalls .....	53
4.8 Throughput of firewalls experimented on ESXi (left) and Hyper-V (right).....	55
5.1 The design of Tree-Rule firewall .....	65
5.2 Hashing Tables of IPTABLES and the Tree-Rule firewall .....	69
5.3 Steps of Netfilter's connection tracking .....	72
5.4 Expanding hashing tables in vertical direction .....	74
5.5 Verifying non-first packets using Tree Rule before Hashing Table .....	77
5.6 The four cases which was evaluated on speed issue .....	81



5.7 Representation of time consumption on Tree-Rule firewall and Netfilter/IPTABLES .....	83
6.1 An example network .....	90
6.2 A Tree rule structure created for an example network in Figure 6.1 .....	96
6.3 Four steps of proposed scheme.....	97
6.4 Transmission speed versus transmission time .....	102
6.5 Time ( $T$ ) used for data transmission and the five main factors ( $w$ , $F$ , $S$ , $e$ and $g$ ) .....	104
6.6 Relation between Time use ( $T$ ) and Time Interval ( $w$ ) .....	107
6.7 Experiment with ESXi .....	111
6.8 Five Linux Web Servers in an ESXi Hypervisor .....	112
6.9 Three cases of 'non automatic rule sorting' and a case of 'automatic rule sorting' .....	114
6.10 Sequences of rules in 'automatic rule sorting' .....	114
6.11 Speed of IPTABLES (represented in graph) .....	118
6.12 Speed of Proposed Firewall without 'Automatic rule sorting' (represented in graph) .....	118
6.13 Comparison of Firewalls' speeds .....	119

# Abbreviation

Abbreviations	Descriptions
2D-Box	Two Dimensions Box
ACL	Access Control List
Cent OS	Community ENTERprise Operating System
ConnTrack	Connection Tracking
CPU	Central Processing Unit
DEC	Digital Equipment Corporation
Dest IP	Destination IP Address
Dest Port	Destination Port
DIP	Destination IP Address
DMZ	Demilitarized Zone
DPT	Destination Port
DstIP	Destination IP Address
DstPT	Destination Port
ESXi	Elastic Sky X – integrated (ESXi is the primary component in the VMware Infrastructure software suite)
FDD	Firewall Decision Diagram
GHz	Gigahertz
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol

## Abbreviations

## Descriptions

Hyper-V	is formerly known as Windows Server Virtualization
IOS	Cisco's Internetworking Operating System
IP	Internet Protocol
IPSEC	Internet Protocol Security
LAN	Local Area Network
Max	Maximum
Mbps	Megabits per second
Min	Minimum
NICs	Network Interface Cards
OS	Operating System
PVFS	Procs Virtual File System
RAM	Random-Access Memory
SIP	Source IP Address
SrcIP	Source IP Address
SrcPT	Source Port
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VM	Virtual Machine

# Abstract

Firewall is a network component for deciding packets whether they will be accepted or denied. The packet decision results are dependent on rule policy pre-defined by firewall administrators. In traditional firewalls, the rule policy will be arranged in a list of rule line called 'listed rule'. The listed rule can cause three significant problems consisting of speed, security, and user friendly problems. The speed problems can occur because many packets will be matched with the rule positioned in bottom positions. Firewall may waste time to verify packets with many rules positioned above the matched rule. Moreover, the traditional firewalls also face to rule conflicts, e.g., shadowed rules. Many rules written to prevent attacking packets may be shadowed by some rules above them and cannot block any packet so that dangerous packets originated from outside can reach internal networks. Additionally, the traditional firewalls are involved with the lack of user-friendly features because administrators must have enough experience in order to create enough efficiency rules.

This research proposes a novel firewall by using a tree structure of rules to solve the above problems. In the proposed approach, firewall administrators are able to design rules in the tree format, and then a core processor of firewall will process packets according to this format. The tree structure can be seen in both users' view and firewall's view. Packets will be verified with the tree shape of rule called 'tree rule'. To decide packet, searching for a data in the tree rule can be done quickly in comparison to searching data in the listed rule of traditional firewalls. This is because searching data in the Tree is faster than sequential searching data in Arrays. Moreover, rule conflicts can be eradicated, since each packet will be verified with the corresponding 'rule path' in the tree rule. This can avoid rule conflicts and shadowed rules. Thus, security problems

caused by shadowed rules cannot be found in the tree rule firewall. Moreover, administrators can create rules easier with the GUI (Graphical User Interface) rule editor. They can design tree rule by creating nodes and links. There are ranges of IP addresses or ports inside each node. The GUI can sort the data inside nodes automatically and maintain consistency of the rule. Thus, the tree rule can be designed easily.

Therefore, the Tree-Rule firewall can provide faster functional speed, be more secure, and be easier to use compared to traditional Listed-Rule firewalls.

## Papers from the Thesis

1. **T. Chomsiri**, X. He, P. Nanda, “Limitation of listed-rule firewall and the design of tree-rule firewall”, in: Proceedings of the 5th International Conference on Internet and Distributed Computing Systems, China, 2012, pp. 275–287.
2. X. He, **T. Chomsiri**, P. Nanda, Z. Tan, “Improving cloud network security using the Tree-Rule firewall”, **Future Generation Computer Systems**, Elsevier, 30 (2014) 116-126. [ERA **Tier-A** Journal, **IF=2.430**, SCImago Journal Rank: **Q1**]
3. **T. Chomsiri**, X. He, P. Nanda, Z. Tan, “A Stateful Mechanism for the Tree-Rule Firewall”, IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom2014), 2014, pp. 122-129. [ERA **Tier-A** conference]
4. **T. Chomsiri**, X. He, P. Nanda, Z. Tan, “Hybrid Tree-rule Firewall for High Speed Data Transmission”, **IEEE Transactions on Cloud Computing**, 2016, no. 1, pp. 1, PrePrints, doi:10.1109/TCC.2016.2554548. [SCImago Journal Rank: **Q1**]
5. **T. Chomsiri**, X. He, P. Nanda, Z. Tan, “An Improvement of Tree-Rule Firewall for a Large Network: Supporting Large Rule Size and Low Delay”, IEEE 15th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom2016), 2016, pp. 178-184. [ERA **Tier-A** conference]