

**IMPROVING PERFORMANCE OF INTRUSION DETECTION SYSTEM  
USING GENERAL-PURPOSE COMPUTING ON GRAPHIC PROCESSING  
UNIT (GPGPU)**

By

Ahmad Rinaldi Widiyanto

12111019

BACHELOR'S DEGREE

in

INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

SWISS GERMAN UNIVERSITY  
**SQU**<sup>®</sup>

SWISS GERMAN UNIVERSITY  
EduTown BSDCity  
Tangerang 15339  
Indonesia

August 2015

**IMPROVING PERFORMANCE OF INTRUSION DETECTION SYSTEM  
USING GENERAL-PURPOSE COMPUTING ON GRAPHIC PROCESSING  
UNIT (GPGPU)**

By

Ahmad Rinaldi Widiyanto

12111019

BACHELOR'S DEGREE

in

INFORMATION TECHNOLOGY

FACULTY OF ENGINEERING AND INFORMATION TECHNOLOGY

SWISS GERMAN UNIVERSITY

The logo for Swiss German University (SGU) is displayed in a dark grey, bold, sans-serif font. It consists of the letters 'S', 'G', and 'U' in a stylized, interconnected manner. A registered trademark symbol (®) is located to the upper right of the 'U'.

SWISS GERMAN UNIVERSITY  
EduTown BSDCity  
Tangerang 15339  
Indonesia

August 2015

**Revision after the Thesis Defense on 11 August 2015**

## STATEMENT BY THE AUTHOR

I hereby declare that this submission is my own work and to the best of my knowledge, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at any educational institution, except where due acknowledgement is made in this thesis.

Ahmad Rinaldi Widianto

Student

Date

Approved by:

Charles Lim, MSc., ECSA, ECSP, ECIH, CEH, CEI

Thesis Advisor

Date

Dipl.-Inf. Kho I Eng

Thesis Co-Advisor

Date

Dr. Ir. Gembong Baskoro, M.Sc

Dean

Date

Ahmad Rinaldi Widianto

## ABSTRACT

### IMPROVING PERFORMANCE OF INTRUSION DETECTION SYSTEM USING GENERAL-PURPOSE COMPUTING ON GRAPHIC PROCESSING UNIT (GPGPU)

By

Ahmad Rinaldi Widiyanto

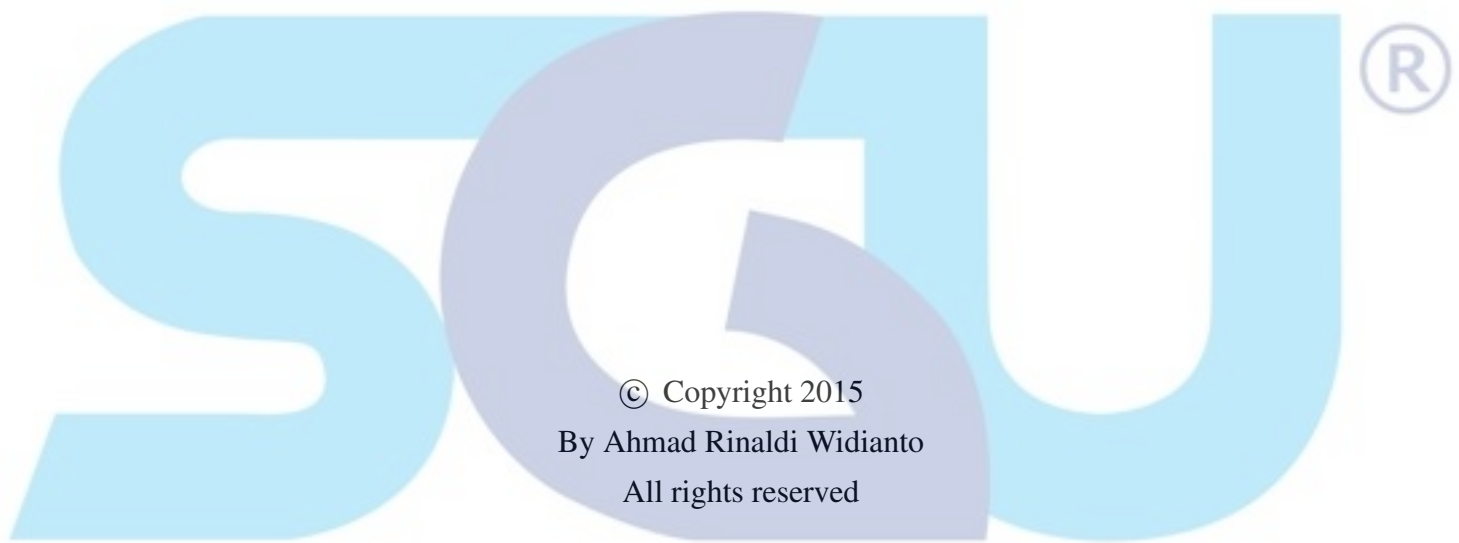
Charles Lim, MSc., ECSA, ECSP, ECIH, CEH, CEI, Advisor

Dipl.-Inf. Kho I Eng, Co-Advisor

SWISS GERMAN UNIVERSITY

The development of computer network and the Internet enables more machines to become connected. Together with mobile computing boom increases the number of data passed via network. More and more data stored in the cloud to meets mobile demands. This leads to even higher security threats. On the other hand, intrusion detection technology are mostly single threaded application. This solution works in the past where most computers are single threaded machines. With current changes in computing paradigm and the increase in network traffic this method becomes obsolete. The focus of this research is to develop a parallel intrusion detection by exploiting GPGPU. In parallel computing packets are processed concurrently, thus increasing system capacity in high throughput environment. GPGPU further exploit parallelism inside modern GPU, which is essentially SIMD engines. OpenCL will be used as programming library during this research. The result are in favour of GPU with original design improving performance by average of 3.51x against single-thread CPU implementation, and by average of 2.54x against multi-thread implementation. After some optimization the overall performance improved on the average by a factor of 11.21 compared to the older version. The performance advantage against CPU improve by a factor of 23.71 vs a factor of 12.28 faster than multi-thread implementation.

*Keywords:* Intrusion Detection, Graphic Processing Unit, GPGPU, Parallel Computing, OpenCL



**SWISS GERMAN UNIVERSITY**

## DEDICATION

I dedicate this work to my beloved mother who always taking care of me and push me to achieve new heights.

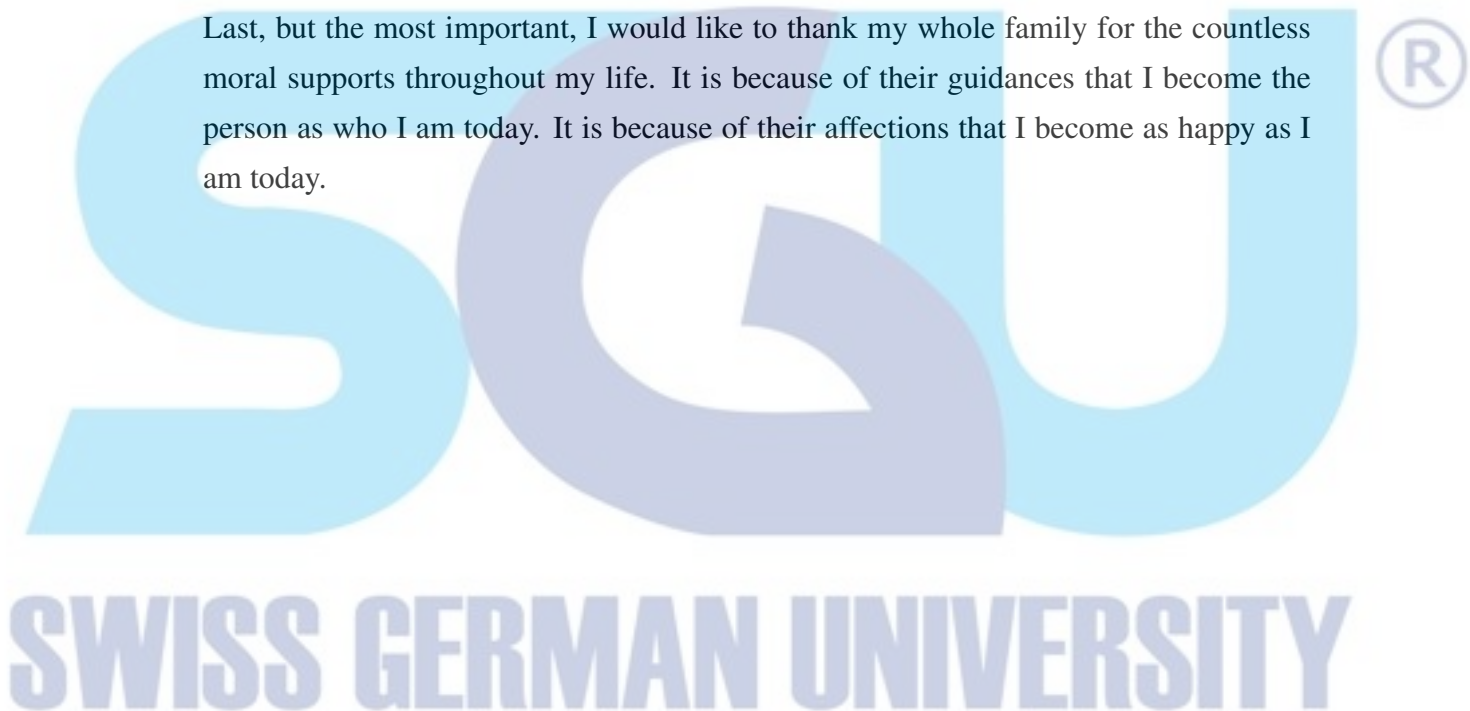


## ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to Mr. Charles Lim and Mr. Kho I Eng for the time, support, advice, and guidance given throughout this research project and the completion of this thesis report. It is because of their priceless contributions that this thesis report and the whole research project can arrive at this point.

I would like to thank all of my friends for their companionship, and to the countless number of people who have helped me throughout this research project, either directly or indirectly.

Last, but the most important, I would like to thank my whole family for the countless moral supports throughout my life. It is because of their guidances that I become the person as who I am today. It is because of their affections that I become as happy as I am today.



## Contents

<b>STATEMENT BY THE AUTHOR</b>	<b>2</b>
<b>ABSTRACT</b>	<b>3</b>
<b>DEDICATION</b>	<b>5</b>
<b>ACKNOWLEDGEMENTS</b>	<b>6</b>
<b>TABLE OF CONTENTS</b>	<b>9</b>
<b>LIST OF FIGURES</b>	<b>13</b>
<b>LIST OF TABLES</b>	<b>14</b>
<b>1 INTRODUCTION</b>	<b>15</b>
1.1 Research Background . . . . .	15
1.2 Research Questions . . . . .	16
1.3 Research Purpose . . . . .	16
1.4 Hypothesis . . . . .	16
1.5 Research Limitation . . . . .	17
1.6 Document Structure . . . . .	17
<b>2 LITERATURE REVIEW</b>	<b>19</b>
2.1 Graphic Processing Unit . . . . .	19
2.1.1 General Purpose Computing on Graphic Processing Units . . . . .	19
2.1.2 OpenCL . . . . .	19
2.1.3 CUDA . . . . .	20
2.1.4 AMD Graphic Core Next Architecture . . . . .	20
2.1.5 AMD APP SDK . . . . .	22
2.1.6 Research Consideration for GPGPU Implementation . . . . .	22
2.2 Intrusion Detection System . . . . .	22
2.2.1 Bro . . . . .	22
2.2.2 Snort . . . . .	22
2.2.3 Brief Comparison of Bro and Snort . . . . .	24
2.2.4 Research Consideration for IDS Implementation . . . . .	25
2.3 Reporting System . . . . .	25



2.3.1	Minami . . . . .	25
2.4	Brief Review on Improving IDS . . . . .	26
2.5	Related Works . . . . .	27
<b>3</b>	<b>RESEARCH METHODOLOGY</b>	<b>29</b>
3.1	Methodology Overview . . . . .	29
3.2	Systems Development Life Cycle . . . . .	30
3.3	Development and Testing Overview . . . . .	31
3.4	System Architecture Overview . . . . .	32
3.4.1	GPU vs CPU . . . . .	34
3.5	Benchmarking Methodology . . . . .	35
3.5.1	Benchmark Metrics Detail . . . . .	36
3.5.1.1	Computation Time . . . . .	36
3.5.1.2	Rules per Second . . . . .	36
3.5.1.3	Request per Second . . . . .	36
3.5.1.4	Total CPU Utilisation . . . . .	36
3.5.1.5	Per Core CPU Utilisation . . . . .	36
3.5.1.6	GPU Utilisation . . . . .	37
3.5.1.7	Power Consumption . . . . .	37
3.5.2	Synthetic Benchmark . . . . .	37
3.5.3	Simulated Benchmark . . . . .	37
<b>4</b>	<b>IMPLEMENTATION AND BENCHMARKING RESULT</b>	<b>39</b>
4.1	Implementation . . . . .	39
4.1.1	Detailed Architecture . . . . .	39
4.1.1.1	Core Process . . . . .	39
4.1.1.2	Bro Process . . . . .	40
4.1.1.3	OpenCL Process . . . . .	40
4.1.1.4	Reporting Process . . . . .	41
4.1.1.5	CPU vs GPU Implementation . . . . .	42
4.2	Benchmarking Set up . . . . .	44
4.2.1	Computer Specification . . . . .	44
4.2.2	Network Architecture . . . . .	44
4.3	Initial Testing . . . . .	45
4.4	Synthetic Benchmark Result and Analysis . . . . .	46
4.4.1	Power Consumption Analysis . . . . .	48
4.5	GPU Engine Improvement . . . . .	49
4.6	Simulated Benchmark . . . . .	51

4.6.1	Result and Analysis . . . . .	52
4.7	Result Summary . . . . .	55
<b>5</b>	<b>CONCLUSION AND FUTURE WORK</b>	<b>56</b>
5.1	Conclusions . . . . .	56
5.2	Future Work . . . . .	56
	<b>GLOSSARY</b>	<b>58</b>
	<b>REFERENCES</b>	<b>61</b>
	<b>APPENDICES</b>	<b>61</b>
<b>A</b>	<b>COMPUTER SPECIFICATION SCREENSHOTS</b>	<b>62</b>
<b>B</b>	<b>DETAILED BENCHMARK RESULT</b>	<b>66</b>
B.1	100 Rules . . . . .	66
B.2	200 Rules . . . . .	67
B.3	400 Rules . . . . .	69
B.4	800 Rules . . . . .	70
B.5	1000 Rules . . . . .	72
B.6	2000 Rules . . . . .	73
B.7	3000 Rules . . . . .	75
B.8	4000 Rules . . . . .	76
B.9	5000 Rules . . . . .	78
B.10	10000 Rules . . . . .	79
B.10.1	Threaded . . . . .	81
B.11	20000 Rules . . . . .	82
B.11.1	Threaded . . . . .	83
B.12	30000 Rules . . . . .	84
B.12.1	Threaded . . . . .	86
B.13	40000 Rules . . . . .	87
B.13.1	Threaded . . . . .	88
B.14	50000 Rules . . . . .	89
B.14.1	Threaded . . . . .	91
B.15	100000 Rules . . . . .	92
	<b>CURRICULUM VITAE</b>	<b>94</b>

