

FIREWALL FRAMEWORK FOR MULTI-CLOUD

Kuruppu Arachchilage Ishanka Gayan

209326E

Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

June 2024

FIREWALL FRAMEWORK FOR MULTI-CLOUD

Kuruppu Arachchilage Ishanka Gayan

209326E

Thesis submitted in partial fulfilment of the requirements for the degree

Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

June 2024

DECLARATION OF THE CANDIDATE & SUPERVISOR

I state that this is my own work and this dissertation does not include without acknowledgement anything that was previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and opinion it does not include any material previously published or written by another individual with the exception where the acknowledgement is made in the text.

Also, I hereby give the University of Moratuwa the non-exclusive right to reproduce and disseminate my thesis/dissertation, in whole or in part in print, electronic or, another form. I maintain the right to use this information in whole or part in future works (such as articles or books).

Signature:

Date: 2024-06-26

The above candidate has researched for the Master's thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the supervisor: **Prof. Shantha Fernando**

Signature of the supervisor:

Date: 2024-06-27

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to Prof. Shantha Fernando for the continuous support given throughout the degree program in imparting knowledge and guiding me throughout the dissertation.

Further, I would like to express my sincere gratitude to all the academic staff of the University of Moratuwa for providing necessary academic support throughout the degree program and sharing their vital knowledge and experience during the lectures. Moreover, I take this opportunity to express my gratitude to all the non-academic staff employees for their support in facilitating a good learning platform throughout my tenure at the university.

Last but not least, I would like to express my heartfelt gratitude to all my family members for providing the necessary support throughout my degree in numerous ways.

ABSTRACT

Multi-cloud systems' dispersed design offers issues in guaranteeing security for multi-cloud application components deployed across various clouds. As security concerns develop, cloud users are more interested in understanding the security state of their apps. The existing environment hinders the opportunity for enterprises to shift to the cloud, which may have provided an appealing alternative for corporations. As a result of this, it is vital to offer proper transparency and security attention in multi-cloud setups. A firewall is the main component that acts as the main shield for cloud assets. Managing a few cloud firewalls is becoming more and more complex when it comes to new cyber threats and concerns. This thesis explores the existing challenges in multi-cloud firewall management like lack of standardization, complexity and difficulty in enforcing security policies. This has been a significant problem due to operational overhead and security risks. The objective is to improve the dependability and trustworthiness of cloud services for organizations by applying appropriate security measures to solve current concerns in multi-cloud computing. This framework mainly includes the components like platform adaptor which converts the firewall rules into cloud-specific API calls and the orchestration engine which works as the heart of the framework. The proof-of-concept demonstrates the framework's ability to enforce context-aware policies, such as global restrictions or tag-driven access controls. This is a policy-driven framework for multi-cloud firewall management which is utilizing Open Policy Agent (OPA) and Terraform for infrastructure provisioning. The framework's effectiveness is demonstrated through a proof-of-concept implementation OPA based policy enforcement and configuration management using NOSQL database. The major goal of this research is to create a standard framework to administer multi-cloud firewalls and also this aims to reduce configuration errors, enhance security posture, and simplify auditability of firewall rules in multi-cloud environments.

TABLE OF CONTENT

Declaration of the Candidate & Supervisor.....	i
Acknowledgement.....	ii
Abstract.....	iii
Table of Content.....	iv
List of Figures.....	vii
List of Tables.....	viii
List of Abbreviations.....	ix
List of Appendices.....	x
Chapter 1.....	1
Introduction.....	1
1.1 Introduction.....	1
1.1.1 Multi-cloud approach.....	3
1.1.2 Transitioning to a multi-cloud datacentre.....	3
1.1.3 Multi-cloud security issues.....	4
1.1.4 Multi-cloud firewall management issues.....	5
1.2 Research problem.....	5
1.3 Research objectives.....	6
1.4 Significance of the study.....	6
1.5 Research question.....	7
Chapter 2.....	8
Literature Review.....	8
2.1 Introduction to multi-cloud security.....	8
2.1.1 Firewall management in multi-cloud environments.....	9
2.2 Interoperability in multi-cloud.....	9
2.3 Configuration management in cloud security.....	11
2.4 Policy driven security in cloud computing.....	12
2.4.1 Policy as Code (PAC) in multi-cloud security.....	13
2.4.2 Policy as Code tools and frameworks.....	14
2.4.3 Declarative security policies.....	15
2.4.4 Challenges in multi-cloud policy enforcement.....	15
2.4.5 Policy orchestration.....	16
2.5 Firewall automation for implementing security.....	17

2.6	Multi-cloud orchestration.....	18
2.6.1	Infrastructure as a Code (IAC).....	18
2.7	IAAS cloud firewall.....	19
2.7.1	IAAS firewall security threats and limitations.....	20
2.7.2	IAAS approach in firewalls.....	21
2.8	Distributed firewall concept.....	22
2.9	Linux net filter firewall.....	22
2.10	Topology and Orchestration Specification for Cloud Applications (TOSCA).....	24
2.10.1	TOSCA objectives.....	24
2.10.2	Review of previous research with TOSCA.....	24
2.11	State-of-the-art in multi-cloud firewall solutions.....	26
2.11.1	Barracuda Cloudgen firewall.....	26
2.11.2	Check point Cloudguard.....	27
2.11.3	Tufin orchestration suite.....	28
2.11.4	Cloud custodian.....	30
2.11.5	Palo Alto Networks' Panorama.....	31
2.12	Conclusive remarks.....	32
	Chapter 3.....	34
	Methodology.....	34
3.1	Research design.....	34
3.1.1	Overview.....	34
3.1.2	Framework design.....	35
3.1.3	Tools selection justification.....	43
3.2	Experiment setup.....	47
3.2.1	Environment configuration.....	47
3.2.2	Scenario definition.....	48
3.2.3	Data collection.....	49
3.2.4	Cloud based real-world testing.....	49
3.2.5	Tools and technologies.....	50
3.3	Ethical considerations.....	50
3.4	Limitations.....	51
	Chapter 4.....	54
	Implementation And Testing.....	54
4.1	Cloud environment setup.....	54

4.2	Framework configuration.....	55
4.3	Integration with Infrastructure as a Code.....	56
4.4	Policy implementation.....	58
4.5	Orchestration engine execution.....	60
4.6	Configuration management.....	62
4.7	Testing scenarios.....	64
4.8	Data collection and analysis.....	69
4.9	Error handling and security.....	70
4.10	Challenges encountered.....	71
Chapter 5.....		73
Results And Discussion.....		73
5.1	Implementation results.....	73
5.2	Testing and validation.....	75
5.3	Framework evaluation against alternative approaches.....	77
5.4	Discussion of results and lessons learned.....	78
5.5	Limitations and future work.....	80
Chapter 6.....		83
Conclusion.....		83
References.....		85
Appendices.....		92

LIST OF FIGURES

Figure	Description	Page
Fig. 1.1	Cloud computing model	2
Fig. 1.2	Cloud delivery models	3
Fig. 1.3	Transition from on-premise to cloud	4
Fig. 2.1	Multi-Cloud Architecture Overview	8
Fig. 2.2	HBase Architecture Overview	11
Fig. 2.3	Conventional process for a policy-as-code solution	13
Fig. 2.4	The rules and processes with Netfilter packet in chain matches	23
Fig. 3.1	Diagram with module interactions	40
Fig. 3.2	Architecture diagram with workflow	41
Fig. 4.1	Architecture Diagram with Cloud Provider Interactions	55
Fig. 4.2	REGO Policies in code repository for validating the rules	58
Fig. 4.3	OPA Policy validation workflow	59
Fig. 4.4	UI representation of policy violation	60
Fig. 4.5	UI representation of a valid policy	60
Fig. 4.6	DB schema data in Mongo db document	62
Fig. 4.7	DB connections to the Python application	63
Fig. 4.8	Global SSH rule adding for AWS Provider	64
Fig. 4.9	UI representation of Global SSH policy violation	65
Fig. 4.10	Adding acceptable Source IP	65
Fig. 4.11	Successful validation message and related log in UI	65
Fig. 4.12	Backend terraform plan output for the deployment	66
Fig. 4.13	Successful deployment UI representations	66
Fig. 4.14	Provisioned Resource in AWS Cloud provider End	66
Fig. 4.15	Security Group Applied With Wrong Tags	67
Fig. 4.16	Mandatory Tags Policy Violation	67
Fig. 4.17	Applying Correct Tags	68
Fig. 4.18	Resource Deployments with Correct Tags	68
Fig. 4.19	Deployed Resources with Correct Tags in AWS Provider	69
Fig. 4.20	Encrypted sensitive config data in Mongo DB	71

LIST OF TABLES

Table	Description	Page
Table 3.1	System specifications for framework deployment	47
Table 4.1	System specifications for framework deployment	56
Table 5.1	Performance Comparison of the Rule Provisioning	74
Table 5.2	Performance Comparison of the Provisioning from Direct Cloud API	75

LIST OF ABBREVIATIONS

Abbreviation	Description
AWS	Amazon Web Service
CSP	Cloud Service Provider
HLVQ	Hierarchical Learning Vector Quantization
IAAS	Infrastructure as a service
KVM	Kernel Virtual Machine
NFV	Network virtual function
NIST	National Institute of Standards and Technology
PAAS	Platform as a service
SAAS	Software as a service
SDN	Software-defined Networking
VLAN	Virtual Local Area Network
VM	Virtual Machine
VNF	Virtual Network Function
WAN	Wide Area Network

LIST OF APPENDICES

Appendix	Description	Page
Appendix A	Symmetric encryption and decryption services using Fernet library	92
Appendix B	Command executor to run CLI commands in given location	92
Appendix C	AWS tag validation REGO policy	92
Appendix D	DB connection establishment	93
Appendix E	Direct API rule provisioning python script for AWS	93