ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

An Intelligent Policy Based Security using Software Defined Network Approach

Abhilash Kamtam[1], Anshuman Kumar[2], Prof. U. C. Patkar[3](Guide)

B.E. Student, Dept. of Computer Engineering, BVCOEL, Pune, India[1]

B.E. Student, Dept. of Computer Engineering, BVCOEL, Pune, India[2]

HOD, Department of Computer Engineering, BVCOEL, Pune, India[3]

ABSTRACT- With the evolution of threat on network and with frequent attacks on the network it had become very important to work and improve the network security. So,to enhance the network security Software Defined Networks (SDN) and OpenFlow has received more attention. Devices in a network are automatically configured instead of manually configuring multiple devices. A software based network controller configures devices in the network. Other techniques like Intrusion Detection, Spam Detection, Firewall, Authorizations and many more techniques also provide network security. Most of the techniques or systems for "Policy Based Security" are having some performance issue that can lead to lower accuracy of the technique. So, this paper proposes a novel idea of "Policy Based Security". We propose OpenSec, an OpenFlow based security framework that provides network operator to create and implement security policies in human readable languages. Software Defined Network and OpenSec policies consist of routing protocols which routes different packets to different systems that are responsible for their operations.

KEYWORDS: Software Defined Network, OpenFlow, OpenSec Framework, Decision Making.

I. INTRODUCTION

The authorizing and implementation of security and privacy policies is usually a distributed process. In a medium size organization, there might be global privacy and security policy measures that are applied across the organization while individuals or local departments may have their own security policies. Each department implementing their own policies might also be responsible for the local implementation of the organization-wide policies for enhanced privacy and security measures [2]. Hence, with the advent increase of software-defined networks, efforts to automate and simplify the network operations have become a challenging task in this current network security scenario [3], [4]. In SDN, the complexity of the network is shifted towards the controller and brings in the simplicity and abstraction to the network operators. As we are moving away from manual configuration of network systems, we get closer to automated implementation of network policies and rules [1]. SDN decouples the control plane from the data plane and migrates the former network traffic to a logically centralized software-based network controller. On the basis of above description, more complex network-control applications can thus be implemented at the controller and exploit the fact that they are network wide aware due to the centralized nature of the control plane.

OpenFlow [6] is a network protocol that was introduced to standardize the communication process between a software-based controller and the network devices in an SDN architecture [1]. It provides an open protocol to program the flow-table in different switches and routers. A network administrator can partition the traffic into production and research flows. The authors have identified that it is difficult for different networking research communities to test their new ideas on the existing hardware. This is because, the source code of the software running on the switches or routers cannot be modified, thus the network infrastructure has become inflexible, as new network ideas were unable to be tested in real-time traffic settings. Thereby identifying all common features in the flow tables of Ethernet switches, the authors have provided a standardized protocol to control the flow table of a switch through software based application. OpenFlow provides an ability to control a switch without requiring the vendors to expose the code of their registered devices [5].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

OpenFlow was initially established in academic campus networks [6]. According to current statistics, at least nine universities in the United States have deployed this technology [7]. The main aim of the OpenFlow was to provide researchers a reliable and efficient platform in order to perform their experiments in various production networks. However, the networking industry has also embraced SDN and OpenFlow protocol as a strategy to increase the functionalities of the network thereby reducing the hardware cost and their complexities. Table 1 shows the list of vendors providing the OpenFlow compliant switches in the market. The Open Networking Foundation (ONF) [8] was founded in 2011 by Deutsche Telekom, Facebook, Google, Microsoft, Verizon, and Yahoo to promote the implementation of SDN and OpenFlow protocol and related technologies. Currently, ONF has more than 95 members, including several major vendors.

In this paper, we propose OpenSec, an intelligent OpenFlow- based network security framework that allows network operators to implement security policies across the network.To elaborate this work let us consider an example where a network operator needs to route the incoming data packets to an Intrusion Detection System (IDS), Encryption Decryption System and an e-mail traffic to a spyware detection device. Our aim is to route the respective data packets to the assigned system to perform various tasks regarding high-level policy for accessing the network, instead of manually configuring the system policies for each type of data packets. Furthermore, if the IDS detect malicious traffic, then the system will automatically collect related information of the sender, including his network hops and will be blacklisted for accessing the network and hence blocking the sender. Instead of configuring the edge router manually, we are more interested in blocking the sender automatically.

OpenSec itself provides an abstraction of the network. Here, the operators can focus on specifying simple and human-readable security policies, instead of configuring all the devices to achieve the desired security. OpenSec consists of a software layer running on top of the network controller, as well as multiple devices that perform security services in background (such as Intrusion Detection System (IDS), encryption, spam detection, firewall, deep packet inspection (DPI) and many more) and report the corresponding result to the controller [1]. The policies defined includes a description of the flow of traffic, a list of security services that apply to the flow and how to react in case malicious content is detected. The response to this can be in the form alert only, or to quarantine traffic or even block all the source packets containing malicious data.

Supporting this paper, we have built OpenSec taking three design requirements into consideration. First, security policies should be human-readable. As, simplicity is one of the main goals of our framework and although current work is being focused on creating human-readable policies [10], [9], [12]. Second, data plane traffic should be processed by the processing units (such as network devices, middleboxes or any other hardware that provides security services to the network). This is because, when the controller is responsible for all the tasks, their increases the probability of occurring bottleneck condition which may impact the solutions generated and may not scale well. In OpenSec,the controller is subjected to have a low workload and is responsiblefor implementing policies and modifying forwarding rulesbased on the security alerts received from the processingunits. Third, the framework should react to security alertsautomatically to reduce human intervention when suspicious traffic is detected.
The goal of OpenSec is straightforward, that is to facilitate a simple, human readable language to automatically implement network security policies. In this paper, we have evaluated the scalability of OpenSec using a real dataset with more than 10 million data flows and have added more evaluation circumstances.

To elaborate our proposed framework, we have considered a normal network scenario. In this, we evaluate OpenSec at a realistic level, we use an existing dataset available at the University of Pune that is comprised of traffic directed to a honeypot. Next, we focused our evaluation on four measures.First, we measured the time complexity of an OpenSec in order to implementsecurity policies based on the number of switches, processingunits and existing policies in the system. Second, we measuredthe delay needed by the framework to react to alerts raised byprocessing units. Third, we show the benefits of automatedblocking. Fourth, we discuss the trade-off of moving middleboxesaway from the datapath and also mirroring traffic as opposed to doing in-line processing.

Our results prove that OpenSec measures well because the delay observed in reacting to alerts remains constant even when the traffic rate increases.
In this paper, Section II represents related work and Section III elaborates proposed technique in detail. The performance of the system is analyzed in Section IV and finally this paper is concluded with future extension traces in Section V.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

II. LITERATURE SURVEY

An enormous amount of work has been focused on policy based security check using framework [1], policy specification [17], policy refinement [18], [19], conflict detection [14], [15] and policy analysis [20] in the networks.

Adrian Lara et al. [1] elaborates OpenSec: Policy-based Security using Software Defined Networking where various policies in human-readable language have been used for providing the access rights to an individual over the network. Furthermore, the working of an OpenSec framework and making of policies with their algorithm has been explained in detail in the paper. The author has explained the work using different experimental scenarios such as multi-switch and multi-unit network, incoming traffic in a campus network, deploying a science DMZ and large number of policies. The system developed by the authors is less efficient because the whole process has to be performed on the incoming packets in order to detect the malicious data. Hence, it requires more time and computation for identifying the threats.

The authors of [5] describes the use of programmable networks prior to SDN and OpenFlow. Their aim was to use SOFTNET and Active Networks(ANs). SOFTNET introduced the idea of adding commands to the contents of each packet [5]. It helps in modifying a network node during operation time, using the commands written in SOFTNET language. Author of [5] also describes the main idea behind using Active Networks i.e. to allow packets to contain programs that could be executed by the network devices that they traversed. The major drawback of using SOFTNET and ANs is that they do not use software components to control network devices.

The authors of [11] explain how the quality of service can be maintained. They used a QoS management framework using the X.500 directory where policies are represented as if-then-else rules. However, the network level policies cannot be directly mapped to devices. This mapping functionality has to be done by relay nodes. Here the interaction between application and network policies had been ignored which may be a drawback for the following method.

Authors of [12] elaborates the use of SDN and OpenFlow with the use of Campus Networks. Campus Networks are dynamic environments with multiple events occurring simultaneously across the network. Network Policies for campus and enterprise are very complex and thus error-prone [12]. So, Authentication od devices is done and after successful authentication devices are scanned for vulnerabilities. If nothing is found they are granted with access. The major problem with this techniques is that it involves complex mechanism that requires input from external tools.

Authors of [13] have explained the about Policy Description Language (PDL). PDL is a domain independent Policy description language. In PDL, a Policy P is represented as a collection of expression.Author of [13] have also described the algorithm for evaluating policies specified using PDL. The algorithm is implemented as the policy engine of a policy server embedded in the “softswitch”, a next generation switch for circuit and packet telephony networks.
Charalambides et al. [14] address conflict resolution in PBM, a crucial aspect when managing a system using policies. In this paper, as the authors point out, when several policies coexist it is likely to encounter that two or more policies give a different output for the same input. Hence, this study addresses the problem of conflict resolution when using policies to provide Quality of Service (QoS) in the networking domain.

Charalambides et al. [15] explains the area of dynamic conflict detection and resolution in the domain of QoS management of IP Differentiated Services (DiffServ) Networks. In order to identify the policies and conflicts involved in the DiffServ QoS management, authors have used the framework developed in the context of the EU IST TEQUILA project. Authors have specifically focused on conflicts that may arise from policies driving the Dynamic Resource Management (DRsM) module of the TEQUILA framework. This paper uses high level calculations for generating the policies, to be specific, the paper is based on the work using the Event Calculus (EC) which uses first-order logic for formalizing policy specification and the mapping to and from the Ponder policy language. The major drawback of this paper is the time and space complexity involved in this project is quite larger than the minimal requirement and these policies generated may not be in a human readable language thus requires more analyzing power for providing the access to the network.

Seugwon Shin et al. [16] presents FRESCO, an application development framework specifically designed to address various network policy problems. The authors have introduced the FRESCO architecture and its integration with the NOX OpenFlow controller, and present several illustrative security applications written in the FRESCO scripting language. The framework developed in this project requires additional knowledge of the scripting language for

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

development of different policies for network security. Policies generated requires more time to process using the developed framework and thus more time is required to process them. As there is the use of different scripting language the paper does not provide feasibility and flexibility for users to use the framework and thus this leads to major drawback for the framework being developed.

III. PROPOSED METHODOLOGY

The proposed methodology for adaptive policy based security for software defined networks can be shown in the figure 1. The methods that involve in the development cycle can be elaborate as folows.

A. This is ths intial step of the system where clients are sending differnet kinds of request from different networks for different tasks, These tasks and requests are been collected in a vector for further process.
B. Here in this step all the requests that are collected in a vector are put to preprocess, where essential required attributes are selected and stored in a new vector to yield preprocessed vector.
C. Here preprocessed data vector is used to create different clusters based on the request information. This clustering need to be done based on the logical aspect as these clusters are indeed plays important role in policy making.

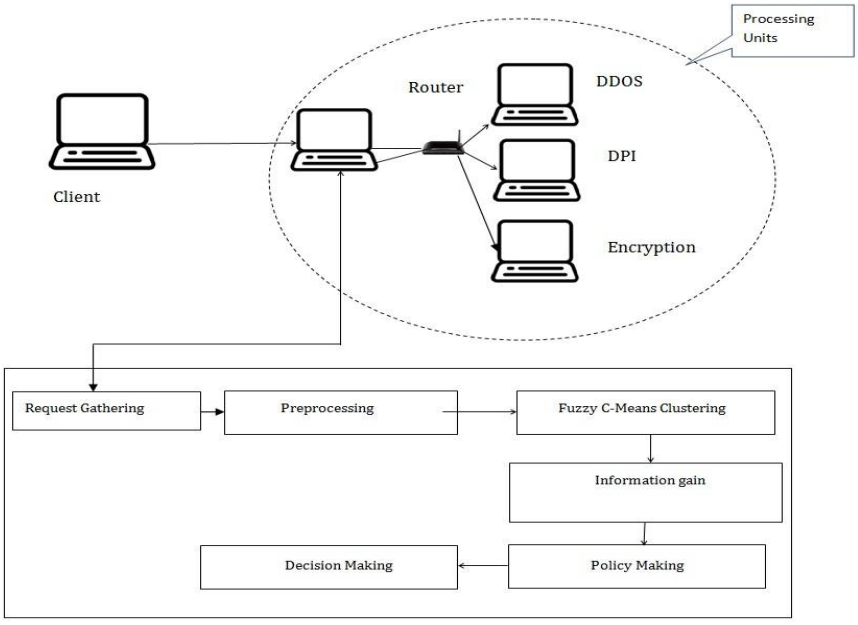


Figure 1: System Overview

Improved Fuzzy C-means clustering technique is used to cluster the data, whose flow can be depicted in figure 2.

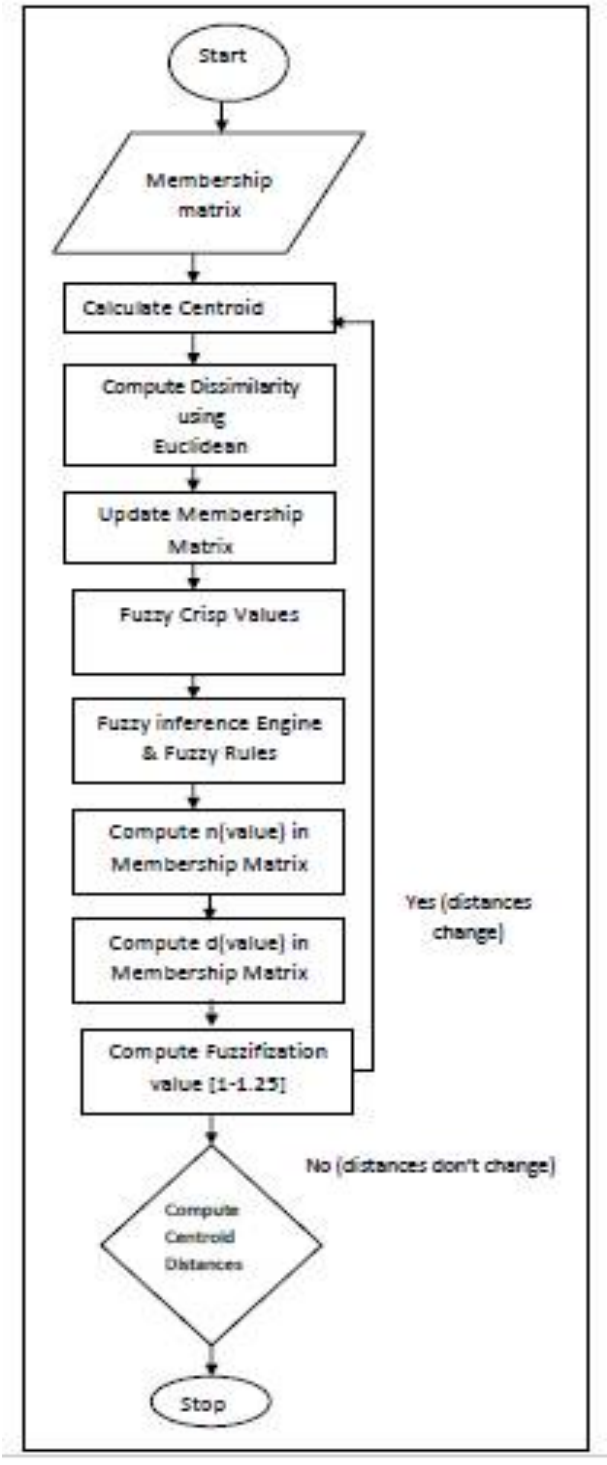
ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017



Fiure 2: Improved Fuzzy C means Clustering

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

D. In order to select the best attribute for creation of policy system uses information gain theory. Here Weight based selction process is done based on the shannon information gain theory, Which is mentioned in equation 1.

$$IGR(C) = -\Sigma (|C_i| / |C|) \log (|C_i| / |C|) \text{ --------(1)}$$

Where C_i is the frequency of the requests in Cluster C.

E. Once the Important requests are been identified then policies are been defined using decision tree technique by ecaluating the required entropy of the different perforamance servers.

IV. RESULTS AND DISCUSSIONS

The proposed methodlogy of adaptive policy making for software defined networks is deployed using java enabled 5 machines of windows platform. System deployed using Apchae tomcat webserver and netbeans as IDE.

For evaluation of the system we consider the Mean reciprocal ratio as analysing parameter (MRR). System is required to submit a ranked list of five opinions for set policy of software defined network. Each policy received a score equal to the inverse of the rank at which the first correct opinion was found.

That is called the Reciprocal Rank (RR), that the values of RR are 1,1/2,1/3,1/4,1/5,0. E.g., if a correct rank appears on the second rank, then it is one over two, so the score will be 0.5, etc. If none of the top five responses contained a correct rank, then the score was zero. The mean reciprocal rank (MRR) is the average score over all set policies.

$$MRR = \frac{\sum_{i=1}^{N} 1/Rank_i}{N}$$

Where, $Rank_i$ is the rank of the first correct occurrence in the top five ranks for policies i; N is the number of test policy asked for the policy opinion; If for a policy i, the correct rank is not in the top five responses then it is taken to be zero.

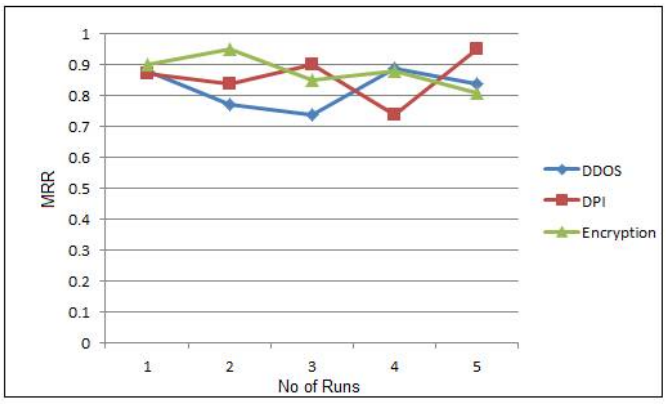


Figure 3: MRR for different set of policies

We performed an experiment to evaluate the rank retrieval using the MRR metric up to the top three responses, defined as follows. The result is shown in Table 1.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

No of Runs	DDOS	DPI	Encryption
1	0.88	0.87	0.9
2	0.77	0.84	0.95
3	0.74	0.9	0.85
4	0.89	0.74	0.88
5	0.84	0.95	0.81

Table 1. MRR for different Runs

In the Fig. 5, we observe that the tendency of average MRR is 0.854 for the mentioned policies in table 1. So this is actually a better performance in our very first attempt of policy making for software defined network.

V. CONCLUSION AND FUTUREWORK

Network Routers and switches based on the hardware are create a tough task of integration in heterogenous network. So software defined networks are playing a vital role to overcome this problem with fine possibilities of adapting network protocols.

So proposed metholodgy deploys adaptive policy making structure using decsion tree technique which is been powered with the concept of improved fuzzy C – means clustering process. This enables the system to take fine decision for policy setting in run time for the servers.

This system can be enhance to work in real time cloud computing scenario for its different padarigm like Paas,Saas and iaas as its future work.

REFERENCES

1. Adrian Lara, Byrav Ramamurthy, “OpenSec: Policy-based Security Using Software-defined Networking”, DOI 10.1109/TNSM.2016.2517407, IEEE Transactions on Network and Service Management.
2. Elisa Bertino, Carolyn Brodie, Seraphin Calo, Lorrie Cranor, Clare-Marie Karat, John Karat, Ninghui Li, Dan Lin, Jorge Lobo, Qun Ni, Prathima Rao and Xiping Wang, “Analysis of Privacy and Security Policies”.
3. M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, “Ethane: taking control of the enterprise,” SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 1–12, October 2007.
4. H. Kim and N. Feamster, “Improving network management with software defined networking,” IEEE Communications Magazine, vol. 51, no. 2, pp. 114–119, February 2013.
5. Adrian Lara, Anisha Kolasani, and Byrav Ramamurthy, ”*Network Innovation using OpenFlow: A Survey*” IEEE 2013.
6. N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, 2008.
7. S. Shin and G. Gu, “Cloudwatcher: Network security monitoring using openflow in dynamic cloud networks (or: How to provide security monitoring as a service in clouds?),” in 2012 20th IEEE International Conference on Network Protocols (ICNP), Austin, Texas, October 2012, pp. 1–6.
8. A. Voellmy, H. Kim, and N. Feamster, “Procera: a language for high-level reactive network control,” in Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN), Helsinki, Finland, August 2012.
9. Md. Faizul Bari, Shihabur Rahman Chowdhury, Reaz Ahmed, and Raouf Boutaba, “PolicyCop: An Autonomic QoS Policy Enforcement Framework for Software Defined Networks”,
10. Hyojoon Kim and Nick Feamster, “Improving Network Management with Software Defined Networking”, IEEE February 2013.
11. Randeep Bhatia Jorge Lobo Madhur Kohli, “Policy Evaluation for Network Management”
12. M. Charalambides, P. Flegkas, G. Pavlou, A. Bandara, E. Lupu, A. Russo, N. Dulav, M. Sloman, and J. Rubio-Loyola, “Policy conflict analysis for quality of service management,” in IEEE International Workshop on policies for Distributed Systems and Networks, Stockholm, Sweden, June 2005.
13. M. Charalambides, P. Flegkas, G. Pavlou, J. Rubio-Loyola, A. K. Bandara, E. C. Lupu, A. Russo, M. Sloman, and N. Dulay, “Dynamic policy analysis and conflict resolution for diffserv quality of service management,” in IEEE/IFIP Network Operations and Management Symposium (NOMS), Vancouver, Canada, April 2006.
14. Seugwon Shin, Phillip Porras, Vinod Yegneswaran, Martin Fong, Guofei Gu, Mabry Tyson, “FRESCO: Modular Composable Security Services for Software-Defined Networks”, in the ISOC Network and Distributed System Security Symposium, February 2013.
15. A. K. Bandara, E. C. Lupu, and A. Russo, “Using event calculus to formalise policy specification and analysis,” in IEEE Workshop on Policies for Distributed Systems and Networks, Lake Como, Italy, June 2003.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 4, April 2017

16. A. K. Bandara, A. Kakas, E. C. Lupu, and A. Russo, "Using argumentation logic for firewall policy specification and analysis," in Large Scale Management of Distributed Systems. Springer, 2006, pp. 185–196.
17. J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, and G. Pavlou, "A methodological approach toward the refinement problem in policybased management systems," IEEE Communications Magazine, vol. 44, no. 10, pp. 60–68, 2006.
18. D. Agrawal, K.-W. Lee, and J. Lobo, "Policy-based management of networked computing systems," Communications Magazine, IEEE, vol. 43, no. 10, pp. 69–75, Oct 2005.

BIOGRAPHY

Abhilash R Kamtam is pursuing his B.E Degree from Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India. He is being affiliated to Savitribai Phule Pune Unviversity, Pune, India. He is currently pursuing his B.E Degree in Computer Science and Engineering. His current research interest includes Artificial Intelligence, Network Security, and Algorithm Design.

Anshuman Kumar is pursuing his B.E Degree from Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India. He is being affiliated to Savitribai Phule Pune Unviversity, Pune, India. He is currently pursuing his B.E Degree in Computer Science and Engineering. His current research interest includes Operating Systems, Network Security and Networking.

Prof. U. C. Patkar has completed his B.E. Degree in Computer Engineering from SSBT College of Engineering, Jalgaon, Maharshtra, India and pursued his degree from North Maharashtra University. He later completed his M. Tech Degree in the field if Information Technology from Bharati Vidyapeeth College of Engineering, Pune, Maharshtra, India and pursued his degree from Bharati Vidyapeeth Deemed University, Pune, India. He then completed his Post Graduation Diploma in Business Management from IBMR College, Pune and was affiliated to Pune University, India. He is currently working as Head of Department (HOD) for Computer Engineering Department in Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India.