IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

Opensec: Policy Based Security using Internal Software Defined Network

Narayini.S[1], Gokulapriya.M[2], Raashmi.V[3], Senthil Madasamy.N[4]

UG Student, Department of Information Technology, Kamaraj College of Engineering and Technology, Virudhunagar, Tamil Nadu, India[1, 2, 3]

Assistant Professor, Department of Information Technology, Kamaraj College of Engineering and Technology, Virudhunagar, TamilNadu, India[4]

Abstract: The software-defined network (SDN) and OpenFlow development made easierto manage the policy-driven network. Before this policy management are handled through manual configurations. The device has made the things complexity and takes more time to transfer messages. They use low-level language for configuration. Manual configuration is handled by network controller is being replaced by SDN. We proposed opensec, an OpenFlow security framework that allows network security operator to implement security policies written in the human readable language. Using opensec, openflow controller converts security policies into series of messages. Opensec automatically detects the malicious traffic. Opensec automatically reacts to the security alerts based on pre-defined network policies. Our framework improves Vulnerabilities detection and prevention and Transmission time.

Keywords: Openflow, Software Defined Network, Opensec, Policy-Driven Network, Security Policies.

I. INTRODUCTION

With the use of software defined network, the network operations are automated and simplified[7]. In SDN, complexity is shifted towards network controller and provides simplicity to the network user. We get closer to the automated implementation of network policies at each device [14]. SDN decouples control plane and data plane and migrates to the network controller.
Our goal is to leverage SDN to allow the network operator to write a high-level policy to achieve automatic configuration for all the network devices. The framework should automatically react to the security alerts to reduce the human intervention when suspicious traffic is detected[1]. An openflow security framework involves security against Bruteforce attack, SQL injection attack, and Wrapping attack. Data security is provided using RC4 algorithm during cloud storage.

Openflow is a protocol that standardizes how SDN controller communicates with the network devices [1]. Using openflow, an application running on network controller and control all the incoming packets. We proposeopensec, an openflow security based framework that allows the network operator to implement security policies across the network[5]. The network operators focus on simple and human readable security policies instead of configuring all the devices to achieve desired security. There are multiple external devices such as Intruder Detection System (IDS),encryption, deep packet checking and others perform their function and report the result to the network controller. The main goal of opensec is to allow the network controller to describe security policies for some security actions such as blocking, forwarding, and others. The network policy includes thelist of security services and describes how to react when malicious traffic is found [10].
While designing the openflow security framework, we have to consider three kinds of requirements. They are simplicity, traffic should be processed, react automatically to the security alerts[8].
Simplicity means that the policies are human readable. It is the main goal of our framework. Human readable describes the ways of encoding. Human readable is used to describe the names that are easier to understand and remember [13]. Human readable protocols reduce the cost of debugging.
Traffic should be processed by processing units such as network devices or hardware units that are responsible for network security. When controller becomes responsible for all the tasks it leads to bottleneck problem and the solution does not scale well [15]. In opensec, the network controller is responsible for only implementing security policies based on security alerts received from processing units.

The openflow policy based security network framework should react automatically to the security alerts to reduce the human intervention when suspicious traffic is found which is received from processing units. Usingopensec, the controller converts security policies into the series of openflow messages. The openflow messages are lined through the bridges. Once the defect found, the security alert clears the traffic automatically through the pre-defined network policies. It increases the transmission time when comparing to other existing systems[1].

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified

Vol. 6, Issue 3, March 2017

A brute force attack is a trial-and-error method used to obtain information such as the password or personal identification number(PIN).In brute force attack, automated software is used to generate a large number of consecutive guesses as to obtain the desired data. It is used by criminals to crack the encrypted data.
To defend brute force attack, Opensec enforces users to create the complex passwords, limiting the number of times the users can unsuccessfully attempt to log in and block the users who exceed the maximum number of failed login attempts.
When the user makes a request to the controller for uploading the file to the cloud from his account through the browser, the request is directed to the controller. In this controller, the SOAP message is generated. This message contains structural information. This message is exchanged between user and controller. The attacker duplicates the body of this message that is already signed by wrapper block and sent to the controller as a legitimate user. Using opensec, we propose to increase the security during the message passing from user to controller. For this purpose, random signature value generator is used for authenticity check.
SQL injection attack allows attackers to spoof identity, tamper with existing data, destroy the data or make it unavailable and become administrator of the database server. To defend SQL injection attack, opensec use type-safe SQL parameters for data access, use an account that has restricted permission in the database and avoids disclosing database error information.

Next, we focused our evaluation on three metrics. First, we measured the time to implement policies. Second, we measured the time to delay needed by the framework to react to the security alerts. Third, we show the benefits of automated blocking. Finally, we compare the opensec with the existing systems.

II. RELATED WORK

A. Cloud Computing – Issues, Research and Implementations

Mladen A. Vouk provided an overview of cloud computing. Inparticular, they built on decades of research in virtualization, distributed computing, networking, web, and software services [9]. It impliedon service oriented architecture and reduced information technology overhead for the end-user. It provided greater flexibility and reduced total cost of ownership and on-demand services. This paper discussed the concept of “cloud” computing issues. It tries to address related research topics, and a “cloud” implementation available today.

B. Security and Privacy Requirements Analysis within a Social Setting

Lin Liu, Eric Yu, John Mylopoulos et al. provided security issues for software systems that ultimately concerns about the relationships among social actors such as stakeholders, system users, potential attackers[12]. This paper proposed a methodological framework for dealing with security and privacy requirements based on agent-oriented requirements modellinglanguage. The framework supported a set of analysis techniques. In particular, attacker analysis helps to identify the system intruders and malicious traffic. Dependency and vulnerability analysis helps to detect vulnerabilities in terms of organizational relationships. Countermeasure analysis supported the dynamic decision-making process in addressing vulnerabilities. Finally, access control analysis bridges the gap between requirement model and implementation model. This framework concerned in the design of agent-based health information systems. In addition, they discuss model evaluation techniques based on model checking.

C. Policy-based management without SDN

Agarwal et al. provide an overview of how policy-based management can be applied for network administration and management [3] [8]. It supports extensive and flexible policy languages. The Automated manager is responsible for defining the policies and implements them. This paper also included the method to refine policies in policy management. Policy refinement allows deriving low-level policies from high-level policies. When a large numbers of policies are available in the system, two or more policies produce different output for the same input.

III. MOTIVATION

The main goal of opensec is to be human-friendly, dynamic and automated security framework. There are two design frameworks in our system. They are reacting automatically to the security events and creating the simple policies [14].

A. Reacting automatically to the security events

When a processing unit detects suspicious traffic, it issues a security alert. These alerts are received by the network operator. Network operator decides how to react to them. Our framework either blocks or simply alerts the network operator. Here, human participation is minimized.

B. Creating simple policies

Opensecallows the operator to redirect the traffic to the processing units and enable automated reaction.Opensec does not process with the end user. They are automatically reacted. This allows us to describe simpler syntax to describe the network flow.Opensecidentifies one or more services and specifies how to react when malicious traffic is detected.

IV. METHODOLOGY

A. RC4 Algorithm

RC4 is a stream cipher and symmetric key algorithm. The same algorithm is used for both encryption and decryption. Here, the data stream is simply XORed with the generated key sequence. The keystream is independent of the plaintext. It uses a variable length key from 1 to 256 bit.

This key is used to initialize a 256-bit state table. The state table is used for subsequent generation of pseudo-random bits and then the pseudo-random stream is XORed with the plaintext to give the ciphertext.

B. Ciphertext – Policy Attribute Based Encryption
This scheme consists of five functions. They are Setup, Encryption, Key generation, Decryption, and delegation [12].
1. Setup:
This takes implicit security parameters as input. The output is public parameter PK and a master key MK.
2. Encryption (PK, M, A):
The encryption takes the public parameters PK, a message M, and an access structure A as input. This will encrypt M and produce a ciphertext C such that only a user have a set of attributes that satisfies the access structure which will be able to decrypt M.
3. Key Generation (MK, S):
The key generation algorithm takes the master key MK and a set of attributes S that describe the key as input. The output is a private key SK.
4. Decryption (PK, CT, SK):
The decryption algorithm takes the public parameters PK, a ciphertext CT, which contains an access policy A, and a private key SK as input. If the set S satisfies the access structure A then the algorithm will decrypt the ciphertext and return a message M.
5. Delegation (SK, S˜):
The delegate algorithm takes a secret key SK for the set of attributes S and a set S˜ ⊆ S as input. It produces a secret key SK for the set of ˜ attributes S˜ as output.

C. Proxy Re-Encryption
A re-encryption scheme is one in which the proxy possesses both parties' keys. One key decrypts a plaintext, while the other encrypts it. The goal of proxy re-encryption is to avoid revealing the keys and the underlying plaintext to the proxy [5].

V. EXISTING SYSTEM

A. Cloud Watcher
Using this, a network operator can describe the flow of network and security services must be applied to detect the malicious defect. The author focuses on how to find the optimal route to clear traffic for the processing units[2].

B. Procera
It uses high-level policy to handle the network events. It mainly focused on enforcing the network policies [8].

C. Fresco
Fresco is the another type of openflow-based security framework. It exposes security modules to the users. An operator must define the type, input, output, the parameter, the action, and the event in the modules. The user has the ability tomanipulate the network events through predefined modules [4].

D. Disadvantages
Procera has complicated the things to use policies whereOpensec used the simpler language.
Cloud watcher takes more time to convert the policies into openflow messages since it uses more than fifty algorithms to evaluate the problem.

VI. PROPOSED SYSTEM

A. Introduction
We propose Opensec, an Open Flow-based network Security Framework that allows network operators to implement Security Policies across the network. Our goal is to leverage SDN to allow the network operator to write a high-level policy to achieve automatic configuration for all network devices. The framework should react to security alerts automatically to reduce human intervention when suspicious traffic is detected by processing units.Our proposed system involves security against Brute force, SQL injection and wrapping attacks. Here, no need to buy or install external software to the network.Data security is provided using RC4 algorithm during cloud storage.

B. Advantages
The benefit of SDN is they use the high-level language. The manual configuration has been converted into automatic configuration through SDN.
It gives security alert to the user if the malicious defect has been found. It provides security against brute force, SQL Injection and wrapping attack. The transfer time is faster compared to other existing systems.

C. System Architecture
The system architecture for opensec is shown in the figure 1.

D. Modules Description
1. Identity Management:
The identity management is divided into two roles. They are users and the security manager.
Users can encrypt each received key and his own key. They can split files into blocks and encrypts them with the key, followed by signing.

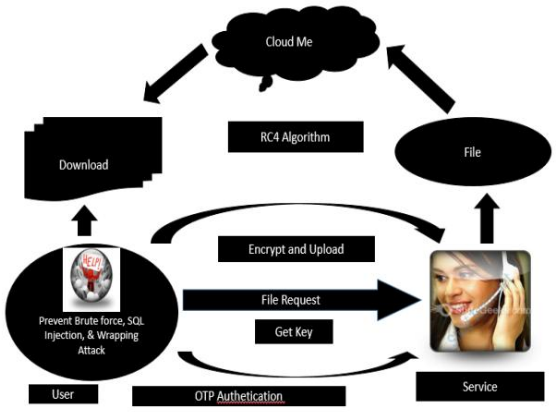

Fig 1: Opensec System Architecture

The resulting encrypted blocks are creating the storage request. For each file, this key will be used to decrypt and the original file during the retrieval phase. The user also uses single sign-on to access each block with a compact signature scheme.

Three roles are offered by the security manager. First, it can provide authentication for users during the storage/retrieval phase. Second, it can provide access control. Third, it can encrypt/decrypt data between users and their cloud
OTP authentication product generates highly secure one-time passwords ensuring that only properly authenticated users are authorized to access to critical application and data. OTP authentications are available in both time-based and event-based versions.
OTP is a password that is valid for only one login session. They are not vulnerable to replay attacks.
OTP generation algorithms make use ofpseudo-randomness or randomness. It makes the prediction of successor OTPs by an attacker to difficult. This algorithm also uses hash functions, which can be used to derive a value but are hard to reverse. Therefore, it is difficult for an attacker to obtain the data that was used for the hash. This is necessary because otherwise, it would be easy to predict future OTPs by observing previous ones.

2. Intrusion Detection and Prevention:
In this module, automatic Intrusion detection system (IDS), encryption, and deep packet inspection (DPI) take place. The result is reported to the controller. The main goal of Opensec is to allow network operators to describe security policies. The policies include a description of the network flow, a list of security services that apply to the flow and how to react when malicious traffic is found. The reaction can be to alert only or to quarantine traffic or even block all packets from a specific source. Hence we have considered automatic intrusion detection and alerting network operator automatically when an intruder tries to brute force, SQL injection and wrapping attack.

3. Encryption:
In our system, we proposed ciphertext-policy attribute-based encryption (CP-ABE). In our system, a user's private key will be associated with an arbitrary number of attributes expressed as strings. On the other hand, when a person encrypts a message, they specify an associated access structure. A user will only be able to decrypt a ciphertext if that user's attributes pass through the ciphertext's access structure.

4. Key Seed Mechanism:
Time-based Group Key Management algorithm is used for cloud storage. This algorithm uses the proxy re-encryption algorithm to transfer major computing task of the key management to the cloud server. So, the TGKM scheme greatly reduces the user's computation and storage overhead. TGKM makes full use of cloud server to achieve an efficient group key management for the cloud storage. Moreover, we use a key seed mechanism to generate a time-based dynamic group key. These keys are effectively strengthening the cloud data security. Our security analysis and performance evaluations both show that the TGKM scheme is secure. The efficient group key management protocol is used for the cloud storage produces low overheads of computation and communication.

5. Split and Merge:
Cloud Storage usually contains business-critical data and functions. High security is needed between the cloud users and cloud service providers to protect these critical data and functions. To overcome the security threats, this paper proposes multiple cloud storage. Thus the common forms of data storage such as files and databases of a specific user are split and stored in the various cloud storages (e.g. Cloud 1 and Cloud 2).
A database consists of tables, rows, and columns. Databases are easy to store in multiple cloud storages. Our application will act as a combiner. It stores different parts of the table such as rows and columns in multiple clouds using fragmentation either vertical fragmentation or horizontal fragmentation. These rows and columns will be encrypted using the RC4 algorithm. During the response, our application combines the data and sends to the verifier.
Files are stored in multiple clouds using data splitting. The file is split into fragments. Each fragmentisstored in distinct cloud servers with the encrypted key. Thus once the authorized token for the specific file is requested, search engine allows keyword search on encrypted data and combines the fragments. This is sent to the verifier.

6. Cloud Storage:
Multiple cloud system includes more than one cloud. We are storing the files in multiple clouds for the purpose of secure storage. The main advantage of storing in multiple clouds is used for security and ease access. The user files are split and stored in multiple cloud servers namely Cloud A and Cloud B. The user can access the respective file only by giving the appropriate key.
The cloud storage providers provide data availability and accessibility. Cloud storage is accessible by the web service applicationprogramming interface. Cloud storage is based on highly virtualized infrastructure. We use dropbox as cloud storage. Dropbox simplifies the way you create, share and collaborate. Dropbox is a cloud-based service. It allows users to upload documents. Dropbox allows users to use as a backup for their own system such as the laptop, smartphone, etc. Everything you kept in dropboxis synced automatically. It's easy to share large files with anyone.

VII. EXPERIMENAL RESULTS

A. Malicious Traffic Detection
We first need to measurethe time needed for the processing units which is used to detect the malicious traffic. While comparing external SDN with internal SDN,

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 6, Issue 3, March 2017

internal SDN detects the malicious traffic in thefaster manner which is shown in fig 2 and table 1 and 2. Hence internal SDN is more secure than the external SDN.

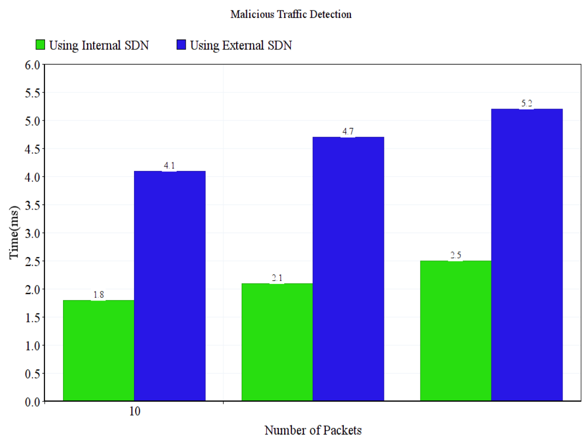

Fig 2: Simple graph that represents Malicious Traffic Detection

Table 1: Malicious Traffic Detection using Internal SDN

Number of Packets	10	20	30
Time to detect Malicious Traffic(ms)	1.8	2.1	2.5

Table 2: Malicious Traffic Detection using External SDN

Number of Packets	10	20	30
Time to detect Malicious Traffic(ms)	4.1	4.7	5.2

B. Performance of policy implementation
We define policy to detect the malicious traffic. When processing unit identifies the malicious traffic, it alerts the network controller. By using these policies, our framework blocks the malicious data. So we calculate the reaction time for a policy which is shown in fig 3 and Table 2.

Our policies are

Policy 1.1: Flow: VLAN=15; Service: IDS, DPI; react: block

Policy 1.2: Flow: EtherPort =1; Service: Firewall; React: alert.

Policy 1.3: Flow: TCP-dp=25; Service: Spyware; react: block.

Policy 1.4: Flow: VLAN=3; Service: 1.1, 1.2, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2, 6.1, 6.2, 7.1, 7.2; React: alert.

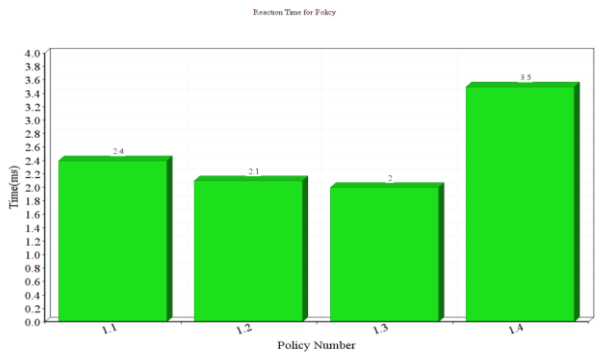

Fig 3: Simple graph that represents reaction time for policy

Table 3: Reaction Time for Policy

Policy Number	1.1	1.2	1.3	1.4
Time (ms)	2.4	2.1	2	3.5

From the above graph, we conclude that the policies 1.1, 1.2 and 1.3 correspond to realistic scenarios and the implementation time is in the order of a few milliseconds. We used policy 1.4 to evaluate if the number of switches and processing units affects the implementation time. Indeed, this policy shows the highest time, but it is still in the order of milliseconds.

C. Measure transmission time for packets
Then, we calculate the transmission time for each packet. Next, we compared the result with external SDN which is shown infig 4 and table 4 and 5. While comparison, we conclude that internal SDN transmits the packets in a faster manner.

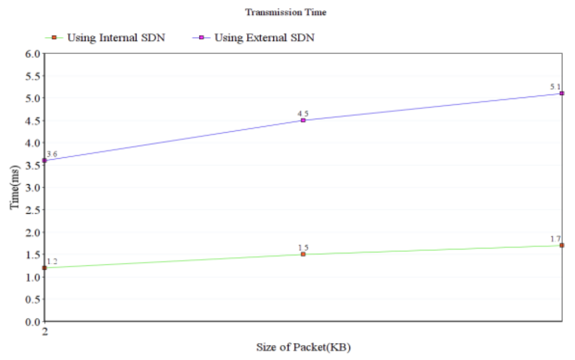

Fig 4: Simple graph that represents the packets transmission time

Table 4: Transmission Time for Packets using Internal SDN

Size of Packet(Kb)	2	5	10
Time(ms)	1.2	1.5	1.7

Table 5: Transmission Time for Packets using External SDN

Size of Packets(Kb)	2	5	10
Time(ms)	3.6	4.5	5.1

VIII. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we presented Opensec, an OpenFlow policy-based security framework that allows network operators to describe security policies using human-readable language. Opensec is used to implement them across the network. Opensec acts as a virtual layer between the user and the OpenFlow controller. It automatically converts security policies into a set of rules that are pushed into network devices. Opensec also allows network operators to specify how to automatically react when malicious traffic is detected. Our evaluation shows several advantages of Opensec. First, moving the traffic away from the controller and into the processing units makes our framework more scalable. Even when the load is high, the controller is not a bottleneck. Second, Opensecmoving the security controls away from the core of the networkThe transmission time is reduced one thrid than the Existing system. Vulnerabilities such as brute force, SQL injection and Wrapping attacks are detected and prevented in a secure and faster manner than the existing system.

Our experience with SDN is excellent and we are in the process of adding new functionalities and features that will make the framework even more suitable for cloud storage and file transfer.The next step of our paper is to develop a series of formalisms based on the current methodology, and tools supporting the various levels of security. Finally, we plan to investigate how to speed up the processing using programmable boards such as NetFPGA cards to achieve a tighter integration between the Opensec controller and the processing units.

REFERENCES

[1] A. Lara and B. Ramamurthy, “Opensec: A framework for implementingsecurity policies using OpenFlow,” in Proc. IEEE Globecom Conf., Austin, TX, USA, Dec. 2014, pp. 781–786.
[2] A. Lara, A. Kolasani, and B. Ramamurthy, “Network innovation usingOpenFlow: A survey,” IEEE Commun. Surveys Tuts, vol. 16, no. 1,pp. 493–512, Feb. 2014.
[3] A. Lara, A. Kolasani, and B. Ramamurthy, “Simplifying network managementusing software defined networking and OpenFlow,” in Proc.IEEE Int. Conf. Adv. Netw.Telecommun. Syst. (ANTS), Bangalore, India,Dec. 2012, pp. 24–29.
[4] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, andS. Shenker, “Ethane: Taking control of the enterprise,” SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 1–12, Oct. 2007.
[5] H. Kim and N. Feamster, “Improving network management with softwaredefined networking,” IEEE Commun. Mag., vol. 51, no. 2, pp. 114–119, Feb. 2013.
[6] N. McKeownet al., “OpenFlow: Enabling innovation in campus networks,”SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Mar. 2008.
[7] A. Voellmy, H. Kim, and N. Feamster, “Procera: A language for high-levelreactive network control,” in Proc. Workshop Hot Topics Softw.Defined Netw. (HotSDN), Helsinki, Finland, Aug. 2012, pp. 43–48.
[8] S. Shin and G. Gu, “Cloud Watcher: Network security monitoring usingOpenFlow in dynamic cloud networks (or: How to provide security monitoringas a service in clouds?),” in Proc. 20th IEEE Int. Conf. Netw.Protocols (ICNP), Austin, TX, USA, Oct. 2012, pp. 1–6.
[9] R. Hofstede, L. Hendriks, A. Sperotto, and A. Pras, “SSH compromisedetection using NetFlow/IPFIX,” ACM SIGCOMM Comput. Commun.Rev., vol. 44, no. 5, pp. 20–26, 2014.
[10] S. Shin, P. Porras, V. Yegneswaran, M. Fong, G. GU, and M. Tyson,“FRESCO: Modular composable security services for software defined Networks,” in Proc. Netw. Distrib. Syst. Sec. Symp. (NDSS), San Diego,CA, USA, Feb. 2013, pp. 1–16.
[11] A. K. Bandara, E. C. Lupu, and A. Russo, “Using event calculus toformalise policy specification and analysis,” in Proc. IEEE Workshop Policies Distrib. Syst. Netw., Lake Como, Italy, Jun. 2003, pp. 26–39.
[12] A. K. Bandara, E. C. Lupu, J. Moffett, and A. Russo, “A goal-basedapproach to policy refinement,” in Proc. IEEE Workshop Policies Distrib.Syst. Netw., Yorktown Heights, NY, USA, Jun. 2004, pp. 229–239.
[13] A. K. Bandara, A. Kakas, E. C. Lupu, and A. Russo, “Using argumentationlogic for firewall policy specification and analysis,” in Large ScaleManagement of Distributed Systems. New York, NY, USA: Springer,2006, pp. 185–196.
[14] J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, G. Pavlou, andA. L. Lafuente, “Using linear temporal model checking for goal-orientedPolicy refinement frameworks,” in Proc. IEEE Workshop Policies Distrib.Syst. Netw., Stockholm, Sweden, Jun. 2005, pp. 181–190.
[15] J. Rubio-Loyola, J. Serrat, M. Charalambides, P. Flegkas, and G. Pavlou,“A methodological approach toward the refinement problem in policy basedmanagement systems,” IEEE Commun. Mag., vol. 44, no. 10,pp. 60–68, Oct. 2006.

BIOGRAPHIES

Narayini.S Pursuing Final year of Bachelor of Technology in Information Technology from Anna University Chennai in Kamaraj College of Engineering and Technology, Virudhunagar. Her research interest is networking.

Gokulapriya.M Pursuing Final year of Bachelor of Technology in Information Technology from Anna University Chennai in Kamaraj College of Engineering and Technology, Virudhunagar. Her research interest is networking.

Raashmi.V Pursuing Final year of Bachelor of Technology in Information Technology from Anna University Chennai in Kamaraj College of Engineering and Technology, Virudhunagar. Her research interest is networking.

N. Senthil Madasamy - Received Bachelor of Engineering in Computer Science and Engineering from Manonmaniam Sundaranar University in Government college of engineering, Tirunelveli in 1998 and Master of engineering in Computer Science and Engineering from Anna University, Chennai in 2007. Currently he is doing Ph.D. in Peer to Peer Network at Anna University, Chennai. He is working an Assistant Professor in Kamaraj College of Engineering and Technology, TamilNadu, India. His research interests include Networking, Parallel Computing and Image Processing.