ISSN:0975-9646

Rapid Decentralized Network Intrusion Defense System on Multiple Virtual Machines

M.Judith Lucia[1] and T.Thirunavukarasu [2]

[1] PG scholar & [2] Assistant Professor
Department of Information Technology,
SNS College of Technology, Coimbatore, India

***Abstract*— Data breaches and cloud service abuse are the greatest cloud security threats according to cloud security alliance. Particularly, attackers can explore vulnerabilities of a cloud system and compromise virtual machines to deploy further large-scale Distributed Denial-of-Service (DDoS). DDoS attacks usually takes place at early stage actions such as exploitation in multiple number of steps, vulnerability scanning at a low frequency, and identifying vulnerable virtual machines as zombies, and finally DDoS attacks by the compromised zombies. Within the cloud system, in particular the Infrastructure-as-a-Service (IaaS) clouds, the recognition of zombie exploration attacks is tremendously complicated. This is for the reason that cloud users may install vulnerable applications on their virtual machines.**

To prevent vulnerable virtual machines from being compromised in the cloud, a distributed vulnerability detection and countermeasure selection mechanism called NICE is proposed, which is built on attack graph-based analytical models and reconfigurable virtual network-based countermeasures. In order to improve the detection accuracy, modified approach of NICE called RAPID is introduced. The system and security evaluations demonstrate the efficiency and effectiveness of the proposed solution.

***Index Terms*—Cyber Warfare, Network centric warfare, attack graph, distributed firewalls, rule anomalies, zombie explorative attacks.**

1. Introduction

Modern studies have shown that users migrating to the cloud regard security as the most important factor. A current Cloud Security Alliance (CSA) survey shows that amongst all security issues, abuse and nefarious utilize of cloud computing is considered as the top security threat, in which attackers can exploit vulnerabilities in cloud and utilize cloud system resources to deploy attacks. In traditional data centres, where system administrators have full control over the host machines, vulnerabilities can be identified and got hold by the system administrator in a centralized manner. However, patching known security holes in cloud data centres, where cloud users usually have the benefit to be in command of software installed on their managed VMs, may not work effectively and can breach the Service Level Agreement (SLA).

Furthermore, cloud users can install vulnerable software on their own VMs, which fundamentally contributes to loopholes in cloud security. The challenge is to begin an effective vulnerability/attack detection and response system for accurately identifying attacks and minimizing the impact of security violation to cloud users.

Such attacks are more effective in the cloud environment because cloud users usually share computing resources, e.g., being connected through the same switch, sharing with the same data storage and file systems, even with potential attackers . The analogous setup for VMs in the cloud, e.g., virtualization techniques, VM OS, installed vulnerable software, networking and so on, attracts attackers to compromise numerous VMs. Network Intrusion Detection and Countermeasures Election in virtual network systems (NICE) to establish a defense-in-depth intrusion detection framework was proposed. For better attack detection, RAPID incorporates attack graph analytical procedures into the intrusion detection processes. It must be noted that the design of RAPID does not intend to improve any of the existing intrusion detection algorithms; indeed, RAPID employs a reconfigurable virtual networking approach to detect and counter the attempts to compromise VMs, thus preventing zombie VMs.

In general, RAPID includes two main phases: deploy a lightweight mirroring-based network intrusion detection agent (NICE-A) on each cloud server to capture and analyze cloud traffic. A NICE-A periodically scans the virtual system vulnerabilities within a cloud server to establish Scenario Attack Graph (SAGs), and then based on the severity of identified vulnerability toward the collaborative attack goals, RAPID will decide whether or not to put a VM in network inspection state. Once a VM enters inspection state, Deep Packet Inspection (DPI) is implemented, and/or virtual network reconfigurations can be deployed to the inspecting VM to make the impending attack behaviors outstanding. RAPID significantly advances the current network IDS/ IPS solutions by employing programmable virtual networking approach that allows the system to construct a dynamic reconfigurable IDS system.

The programmable virtual networking architecture of RAPID enables the cloud to establish inspection and quarantine modes for mistrustful VMs according to their current vulnerability state in the existing SAG. Based on the collective behavior of VMs in the SAG Using this approach, RAPID does not need to block traffic flows of a suspicious VM in its premature attack stage. The contributions of RAPID are presented as follows: RAPID, a new multiphase distributed network intrusion detection and prevention framework in a virtual networking environment that captures and inspects suspicious cloud traffic without interrupting users' applications and cloud services is introduced. RAPID incorporates a software switching solution to quarantine and inspect suspicious VMs for further investigation and protection. Through programmable network approaches, RAPID can improve the attack detection probability and improve the resiliency to VM exploitation attack without interrupting existing normal cloud services.

RAPID employs a novel attack graph approach for attack detection and prevention by correlating attack behaviour and also suggests effective countermeasures.

RAPID optimizes the implementation on cloud servers to minimize resource consumption. Our study shows that RAPID consumes less computational overhead compared to proxy-based network intrusion detection solutions. In the last century, computer turned out to be an inseparable part of daily human life. For recent years and with the invent of the Internet, it has been deployed for communication and accessing data. However, currently people rely on the Internet to satisfy their demands utilizing its services, which can be defined as some computing function, rather than accessing the mass data from the Internet. Along with the proposal of the cloud computing concepts, a new paradigm of software development and deployment of resources has emerged. It is possible to get rid of the great amount of the spending for fixed assets, such as expensive network servers and software.

In classical enterprise settings, an IDS is normally deployed on dedicated hardware at the edge of the defended networking infrastructure or run on individual hosts on the network, in order to protect respective network or host from external attacks. Today small and medium companies are increasingly realizing that simply by tapping into the Cloud they can gain fast access to best business applications, without training new personnel, or licensing new software. IDS is not an exception to this tread and the interests for embedding IDS to a Cloud environment is undeniable.

At present the safety of commonly used technologies such as message encryption, firewalls protect the network and can be used as a first line of defense, but only these technologies is not enough. Intrusion Detection Systems (IDS) has been proposed for years as an efficient security measure and is nowadays widely deployed for securing critical IT-Infrastructures. Many commercial and open source implementations have emerged and been widely used in practice for identifying malicious behaviors against protected hosts or network environments. They can propose security measures by scanning configurations, logs, network traffic, and user actions to identify typical attack behavior.

2. RELATED WORK

In this section, literatures of several highly related research areas to RAPID, including: Zombie detection and prevention, attack graph construction and security analysis, and software definite networks for attack countermeasures are proposed. The area of detecting malicious behaviour of users has been well explored.

It is hopeful to help researchers who want to engage in the field and want to propose some solution to those problems. With the protocol it is identified that 661 publications about the subject, where the Security Domains involved can be analyzed. Various types of solutions proposed by the authors [1], and identified that some of those publication were concerned with the compliance of some standard are presented.

Also those compliances and reference the respective publications to ease the work of the researcher that wants to explore a specific compliance. Threat #7 is the most explored in literature and, in consequence, the Domains of Risk Analysis and Management and Trust Model and Management have expressive results have been identified. Also many combinations of Domains related to Access Control, Applied Cryptography, Data or Database Protection and Privacy are identified.

Firewall security, like any other expertise, requires proper management in order to offer proper security services. By having firewalls on the boundaries of the network or between sub-domains may not necessarily make the network secure. One reason of this is the complexity of managing firewall rules and the resulting network vulnerability due to rule anomalies. The Firewall Policy Advisor [4] presented provides a number of techniques for purifying and protecting the firewall policy from rule anomalies. The administrator may use the firewall policy advisor to manage legacy firewall policies without prior analysis of filtering rules. It is formally defined a number of firewall policy anomalies in both centralized and distributed firewalls and these are the only conflicts that could exist in firewall policies have been proved. And presented a set of algorithms to detect rule anomalies within a single firewall (intra-firewall anomalies), and between inter-connected firewalls (inter-firewall anomalies) in the network. When an anomaly is detected, users are prompted with proper corrective actions. A tool is intentionally made not to automatically correct the discovered anomaly but rather alarm the user because it is believed that the administrator should have the final call on policy changes. Finally, a user-friendly Java-based implementation of Firewall Policy Advisor is presented. Using Firewall Policy Advisor was shown to be very effective for firewalls in real-life networks. In regards to usability, the tool was able to discover filtering anomalies in rules written by expert network administrators. In regards to performance, although the policy analysis algorithms are parabolically dependant on the number of rules in the firewall policy, our experiments show that the average processing time in intra- and inter-firewall anomaly discovery is very reasonable for practical applications

Firewall security, like any other technology, requires proper management in order to provide proper security services. Thus, just having firewalls on the network boundaries or between sub-domains may not necessarily make the network any secure. One reason of this is the complexity of managing firewall rules and the resulting network vulnerability due to rule anomalies. The Firewall

Policy Advisor [4] presented provides a number of techniques for purifying and protecting the firewall policy from rule anomalies. The administrator may use the firewall policy advisor to manage legacy firewall policies without prior analysis of filtering rules. It is formally defined a number of firewall policy anomalies in both centralized and distributed firewalls and these are the only conflicts that could exist in firewall policies have been proved. And presented a set of algorithms to detect rule anomalies within a single firewall (intra-firewall anomalies), and between inter-connected firewalls (inter-firewall anomalies) in the network. When an anomaly is detected, users are prompted with proper corrective actions. A tool is intentionally made not to automatically correct the discovered anomaly but rather alarm the user because it is believed that the administrator should have the final call on policy changes. Finally, a user-friendly Java-based implementation of Firewall Policy Advisor is presented. Using Firewall Policy Advisor was shown to be very effective for firewalls in real-life networks. In regards to usability, the tool was able to discover filtering anomalies in rules written by expert network administrators. In regards to performance, although the policy analysis algorithms are parabolically dependant on the number of rules in the firewall policy, our experiments show that the average processing time in intra- and inter-firewall anomaly discovery is very reasonable for practical applications.

The technical contributions [5] are three-fold. First, formally a firewall tree is specified. Second, two classes of properties, namely accept and discard properties, of firewall trees are identified. Third, two algorithms that can be used to verify whether any given firewall tree satisfies a given accept or discard property of that tree.

Firewalls are core elements in network security. However, managing firewall rules, particularly in multi-firewall enterprise networks [6] has become a complex and error-prone activity. Firewall filtering rules need to be written, ordered and distributed carefully in order to avoid firewall policy anomalies that might cause vulnerability in the network. Hence, inserting or modifying filtering rules in any firewall requires systematic intra- and inter-firewall analysis to establish the proper rule placement and ordering in the firewalls. All anomalies that could exist in a single- or multi-firewall environment are recognized and a set of techniques and algorithms to routinely determine policy anomalies in centralized and distributed firewalls. These techniques are implemented in a software tool called the “Firewall Policy Advisor” that simplifies the management of filtering rules and maintains the security of next-generation firewalls.

3. System Design

The server/user can be authenticated using the username and password. If the user is invalid the user is prevented from entering the system. The authenticated user enters IDEA (Intrusion Detection and Elimination Architecture). This module is secure and the user would be able to choose the system to which the data needs to be transmitted. The VM profiling module collects details including IP address, time and maximum service capacity of the virtual machine in the network. This defines authority to access resources in the cloud.

Alert Correlation algorithm is used for constructing the attack graph which in turn is used to define the possible vulnerable paths in the network. If a particular alert is new, a specific edge in the attack graph is created which leads to new set of alerts. The output of Alert Correlation Graph is the set of attack paths. This module uses Active Alert graph verification algorithm.

The Entropy calculation is used for calculating the amount of traffic. Sometimes the intruder may impose DDOS attack/flooding attack. So the system which is implementing more traffic needs to be tracked.

The monitoring process can be done to calculate the behavioral distance. Abnormal behaviour of the client is frequently reported to the administrator.

Also the countermeasures are selected to perform which include traffic redirection, port blocking, network reconfiguration, updates the filtering rules, Deep packet inspection, Virtual Machine isolation.

4. System Components

4.1 Cloud Server With Authentication

The module contains both the administrator and user authentications. The admin would have the privilege to view the whole process processed by the user. Once the user registers, user would be able to view only the authenticated page. The personal information and the data which are transferred by the user can be viewed by the user. The login in the secure module is non-dynamic and secure. Once logged in the server would be able to receive the data packets.

The network is classified by workgroups. The active and the connected systems over the network are obtained with the use of this module. Once logged in to the process, the module obtains the active systems and displays to the user. The user would be able to choose the system to which the data needs to be transmitted by file transfer.

4.2 VM Profiling And Rule Generation

The VM profile module phase defines the initialization and description of virtual machines in the cloud environment. This also defines the authority to access the resources in the cloud. Only the authenticated persons can upload and download the files in virtual machines. For this process VM should enter with all basic information. Virtual machines in the cloud can be profiled to get precise information about their state, services running, open ports. One major factor that counts toward a VM profile is its connectivity with other VMs. Admin should define with the IP and proper details to initiate the VM.

4.3 Flow Monitoring

Request processing and packet inspecting module will get the detail about the normal data transfer between the source and destination. Based on the data traffic level it analyses the node link ability and scalability. This process of considering traffic level will improves the data non repudiation as well as data integrity. This process based on entropy variations and helps to make the trace back effective.

4.4 Entropy Calculation

This module takes the peak and off peak traffic time value of the router and breakdown to seconds for easy calculation of the data traffic. The normal data packet will be reached destination as the way but the DDos attacked packet will get differ from the other, this might happened y the zombies. They can consider single attack part other than the overlap packet attacks.

4.5Alert Correlation And Behaviour Monitoring

The system to which the data has to be transferred and the file which has to be transferred are selected by the user. In order to securely transfer the file would be encrypted. The key would be enabled and decrypted on its own once the data is received at the destination. To measure the behavioral distance or evolutionary distances the monitoring protocol would be initiated automatically when the process is started by the user.

To review the network security risk condition for the current network configuration, security metrics are desired in the attack graph to measure risk probability. After an attack graph is constructed, vulnerability information is incorporated in the graph. So this module uses Active Alert graph Verification algorithm.

4.6 Attacking Model With Monitoring

The client data which are transferred to the destination need to be monitored. Reports are of no use after the data has been affected by the intruder. Each data information path is traced back from one end to the other. When the data gets deviated from the desired path, the monitoring stub would report to the client.

4.7 Reports

Administrator would be able to view all the data transactions. The administrator also manages the network path. The transactions and intruder information are forwarded to the administrator. The administrator manages all the previous history of data transactions in order to induce denial of service from the reports module.

5. Methodology

RAPID optimizes the implementation on cloud servers to minimize resource consumption. The study shows that RAPID consumes less computational overhead compared to proxy-based network intrusion detection solutions. Intrusion Detection System (IDS) and firewall are widely used to monitor and detect suspicious events in the network

To identify the source or target of the attacker in the network, especially to detect multistep attack, the alert correction is a compulsory tool. The primary goal of alert correlation is to afford system support for a global and condensed view of network attacks by analyzing raw alerts. Recently many attack graph-based alert correlation techniques have been proposed. A memory structure, called queue graph (QG), to sketch alerts matching each exploit in the attack graph is devised. However, the implicit correlations in this design make it difficult to use the correlated alerts in the graph for analysis of similar attack scenarios.

The nodes in the attack graph with manifold mapping functions, an alert dependencies graph (DG) is introduced to integrate related alerts with multiple correlation criteria. Each path in DG represents a division of alerts that might be part of an attack situation. However, their algorithm involved all pairs shortest path searching and sorting in DG, which consumes significant computing energy.

Algorithm 1 : Alert Correlation

Here is a method for utilizing SAG and ACG together so as to forecast an attacker's behavior. Alert Correlation algorithm is implemented for every alert detected and returns one or more paths S_i. For every alert a_c that is received from the IDS, it is added to ACG if it does not stay alive. For this new alert a_c, the equivalent vertex in the SAG is established by using function map (a_c) (line 3). For this vertex in SAG, alert equivalent to its parent vertex of type N_c is then correlated with the present alert ac (line 5). This creates a new set of alerts that fit in to a path Si in ACG (line 8) or creates out a new path Siþ1 from Si with subset of Si before the alert a and appends ac to Siþ1 (line 10). In the end of this algorithm, the ID of ac will be added to alert feature of the vertex in SAG.

```
Require: alert ac, SAG, ACG
 1: if (ac is a new alert) then
 2: create node ac in ACG
 3:  $n_1 \leftarrow v_c$ € map ($a_c$)
 4:  for all n2 € parent (n1) do
 5:  create edge (n2.alert, ac)
 6:  for all Si containing a do
 7:  if a is the last element in Si then
 8:  append ac to Si
 9:  else
10: create path $S_{i+1} = \{subset(S_{I},a), a_{c}\}$
11: end if
12: end for
13: add $a_c$ to n1.alert
14: end for
15: end if
16: return S
```

Algorithm 2: Countermeasure Selection

Algorithm 2 presents how to select the optimal countermeasure for a given attack scenario. Input to the algorithm is an alert, attack graph G, and a pool of counter measures CM. The algorithm starts by selecting the node vAlert that corresponds to the alert generated by a RAPID-A. Before selecting the countermeasure, the distance of vAlert to the target node is counted. If the distance is greater than a threshold value, do not perform countermeasure selection but update the ACG to keep track of alerts in the system (line 3). For the source node vAlert, all the reachable nodes (including the source node) are collected into a set T (line 6). Because the alert is generated only after the attacker has performed the action, set the probability of vAlert to 1 and calculate the new probabilities for all of its child (downstream) nodes in the set T (lines 7 and 8. The change in probability of target node gives the benefit for the applied countermeasure using (7). In the next double for-loop, compute the Return of Investment (ROI) for each benefit of the applied countermeasure based on (8). The countermeasure which when applied on a node gives the least value of ROI, is regarded as the most favorable countermeasure. Finally, SAG and ACG are also reorganized before terminating the algorithm. The complexity of Algorithm 2 is O(|V| * |CM|), where |v| is the number of vulnerabilities.

```
Require: Alert, G(E,V), CM
1: Let $v_{alert}$= Source node of the Alert
2: if Distance to Target($_{valert}$) > threshold then
3:     Update_ACG
4:     return
5: end if
6: Let T = Descendant($v_{alert}$) U $v_{alert}$
7: Set $p_r(v_{alert})$=1
8: Calculate_Risk_Prob(T)
9: Let benefit[|T|,|CM|]=0
10: for each t € T do
11:     for each cm € CM do
12:         if cm.condition(t)  then
13:             $P_r(t) = P_r(t)$*(1-cm.effectiveness)
14:             Calculate_Risk_Prob(Descendant(t))
15:             benefit[t,cm] =$\Delta P_r$(target_node)
16:         end if
17:     end for
18: end for
19: Let ROI[|T|,|CM|]=0
20: for each t €T do
21:     for each cm € CM do
22:         ROI[t,cm]= benefit[t,cm]/cost.cm+intrusiveness.cm
23:     end for
24: end for
25: Update SAG and Update ACG
26: return Select Optimal CM(ROI)
```

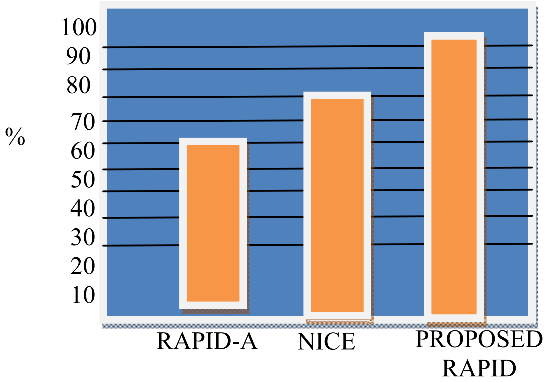

Fig 1: Accuracy and efficiency comparison of Proposed RAPID with existing approaches.

6. Conclusion And Future Work

RAPID utilizes the attack graph model to carry out attack detection and prediction. The proposed solution investigates how to use the programmability of software switches-based solutions to improve the detection accuracy and overcome victim exploitation phases of collaborative attacks. The system performance evaluation demonstrates the feasibility of RAPID and shows that the proposed solution can significantly diminish the risk of the cloud system from being exploited and abused by internal and external attackers. RAPID only investigates the network IDS approach to counter zombie explorative attacks. To improve the detection accuracy, host-based IDS solutions are required to be incorporated and to cover the whole spectrum of IDS in the cloud system.

Additionally, the scalability of the proposed RAPID solution by investigating the decentralized network control and attack analysis model based on current study will be investigated.

References

[1] Carlo Marcelo Revoredo da Silva, Jose Lutiano Costa da Silva, Ricardo Batista Rodrigues, Leandro Marques do Nascimento, Vinicius Cardaso Garcia, "Systematic Mapping Study on Security threats in Cloud Computing". (IJCSIS) International Journal of Computer Science and Information Security, Vol. 11, No. 3, March 2013.

[2] Martin R. Stytz, Ph.D. Sheila B. Banks, Ph.D., "Cyber Warfare Simulation to prepare to control cyber space".Anirudh Ramachandran and Nick Feamster, College of Computing, Georgia Tech, "Understanding the Network-Level Behavior of Spammers".

[3] Fernando Sanchez, Zhenhai Duan, Yingfei Dong , "Understanding Forgery Properties Of Spam Delivery Paths".

[4] Jiri Matas, Jan Sochman, "Wald's Sequential Analysis For Time-constrained Vision Problems" , Springer US, Unifying Perspectives in Computational and Robot Vision Volume 8, 2008, pp 57-77.

[5] Mengjun Xie, Heng Yin, Haining Wang, Department of Computer Science, The Collegeof William and Mary, Williamsburg, VA 23187, "An Effective Defense Against Email Spam Laundering".

[6] Cloud Security Alliance, "Top Threats to Cloud Computing, Mar. 2010.

[7] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A.Konwinski, G. Lee, D.Patterson, A. Rabkin, I. Stoica, and M.Zaharia, "A View of Cloud Computing," ACM Comm., vol. 53,no. 4, pp. 50-58, Apr. 2010.

[8] B. Joshi, A. Vijayan, and B. Joshi, "Securing Cloud Computing Environment Against DDoS Attacks," Proc. IEEE Int'l Conf. Computer Comm. and Informatics (ICCCI '12), Jan. 2012.

[9] H. Takabi, J.B. Joshi, and G. Ahn, "Security and Privacy Challenges in Cloud Computing Environments," IEEE Security and Privacy, vol. 8, no. 6, pp. 24-31, Dec. 2010.

[10] "Open vSwitch Project," http://openvswitch.org, May 2012.

[11] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, Detecting Spam Zombies by Monitoring Outgoing Messages," IEEE Trans. Dependable and Secure Computing, vol. 9, no. 2, pp. 198-210, Apr. 2012.

[12] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "Bot Hunter: Detecting Malware Infection through IDS-driven Dialog Correlation," Proc. 16th USENIX Security Symp. (SS '07), pp. 12:1-12:16, Aug. 2007.

[13] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," Proc. 15th Ann. Network and Distributed Sytem Security Symp. (NDSS '08), Feb. 2008.

[14] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J.M. Wing, "Automated Generation and Analysis of Attack Graphs," Proc. IEEE Symp. Security and Privacy, pp. 273-284, 2002.

[15] "NuSMV: A New Symbolic Model Checker," http://afrodite.itc.it:1024/nusmv. Aug. 2012.

[16] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph based

network vulnerability analysis," Proc. 9th ACM Conf.Computer and Comm. Security (CCS '02), pp. 217-224, 2002.

[17] X. Ou, S. Govindavajhala, and A.W. Appel, "MulVAL: A Logic-Based Network Security Analyzer," Proc. 14th USENIX Security Symp., pp. 113-128, 2005.

[18] R.Sadoddin and A. Ghorbani, "Alert Correlation Survey: Framework and Techniques," Proc. ACM Int'l Conf. Privacy, Security and Trust: Bridge the Gap between PST Technologies and Business Services (PST '06), pp. 37:1-37:10, 2006.

[19] L.Wang, A.Liu, and S. Jajodia, "Using Attack Graphs for Correlating, Hypothesizing, and Predicting Intrusion Alerts," Computer Comm., vol. 29, no. 15, pp. 2917-2933, Sept. 2006.

[20] S. Roschke, F. Cheng, and C. Meinel, "A New Alert Correlation Algorithm Based on Attack Graph," Proc. Fourth Int'l Conf. Computational Intelligence in Security for Information Systems, pp. 58-67, 2011.

AUTHORS

M.Judith Lucia She received the B.E degree in the specialization of Computer Science Engineering from Velammal College of Engineering and Technology under Anna University. Now she is currently pursuing post graduation (M.E) in Department of Information Technology, SNS College of Technolgy,Coimbatore.

T.Thirunavukarasu He received the M.E degree in the specialization of computer Science Engineering from Sona College of Technology, Salem on 2010 under Anna University Coimbatore. Now he is Working as a Assistant Professor, Department of Information Technology in SNS College of Technology, Coimbatore