International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:8, No:11, 2014

EUDIS-An Encryption Scheme for User-Data Security in Public Networks

S. Balaji, M. Rajaram

***Abstract*—**The method of introducing the proxy interpretation for sending and receiving requests increase the capability of the server and our approach UDIV (User-Data Identity Security) to solve the data and user authentication without extending size of the data makes better than hybrid IDS (Intrusion Detection System). And at the same time all the security stages we have framed have to pass through less through that minimize the response time of the request. Even though an anomaly detected, before rejecting it the proxy extracts its identity to prevent it to enter into system. In case of false anomalies, the request will be reshaped and transformed into legitimate request for further response. Finally we are holding the normal and abnormal requests in two different queues with own priorities.

***Keywords*—**IDS, Data & User authentication, UDIS.

I. Introduction

AUTHENTICATION is a big issue for all the computer networks in current enterprise station. Attackers have made several possible attempts to bring down higher end networks [4]. Many methods have been developed to secure the network infrastructure and communication over the shared System, along the use of firing Filter Systems, encryption and decryption techniques and individual private networks. Attacker's detection system is a relatively new addition to enhance the techniques [2], [6]. Using Attacker's detection methods, we can gather and use data from known types of attacks and find out if someone is trying to attack our network or particular hosts. The shared data is collected; which can be used to harden our network security, as well as for permissible purposes [14]. Both commercial and open source tools are available for these propose. Many exposure valuation tools are also available in the current system that can be used to assess various authentication holes present in our shared network. Attacker's Detection Systems help information systems developed for, and justify with attacks [1].

They accomplish this by collecting information from a variety of systems and information, and then checking the internet link information for possible security issues.

Attacker's detection will be responsible for

- Resultant analysis of user activity and system activity
- Multi Group Checking of system configurations and vulnerabilities
- Evaluating the reliability of serious system and data files
- Algebraic analysis of activity patterns based on the matching to known attacks
- abnormal activity analysis
- Operating system audit

There are components to the Attacker's detection system.

Network Intrusion Detection system (NIDS) performs an analysis for a passing traffic on the entire subnet [3]. Works in a promiscuous mode, and matches the traffic that is passed on the subnets to the library of knows attacks [5]. Once the attack is identified, or abnormal behavior is sensed, the alert can be send to the administrator [9].

Example of the NIDS would be installing it on the subnet where you firewalls are located in order to see if someone is trying to break into your firewall Network Node Intrusion detection system (NNIDS) – performs the analysis of the traffic that is passed from the network to a specific functioning host system [8]. The deviation between NIDS and NNIDS is that the traffic is monitored on the single host only and not for the entire subnet [10]. The example of the NNIDS could be, implementing it on a Generic device, to examine the network traffic was decrypted. This way we can see if some Attacker is trying to break into your Generic device [12]. Host Intrusion Detection System (HIDS) – takes a snap shot of our confidential system files and merges it to the previous snap shot [11]. If the acute system files were changed or removed, the alert is sent to the deployment of network administrator to monitor [7]. The example of the HIDS can be seen on the mission critical machines that are not expected to change their configuration.

Intrusion detection can be implemented either on the hosts that need to be protected or on a network device that can sniff the traffic for all the hosts on the network [13]. Based on the resultant locations, there are two issues of IDS, viz., i) host-based IDS, and ii) network-based IDS [15].

II. Existing System

A. Architecture of Hybrid IDS

In Hybrid IDS consists of six components viz., i) Data acquisition module, ii) Signature database, iii) Analyzer, iv) Anomaly detector, v) Signature generator, and vi) Counter-measure module (see Fig. 1).

Data acquisition module has multiple trackers. Trackers are placed either on virtual host or in defined network section. Trackers that are placed on individual hosts observe packets as they enter and leave that host. Trackers that are placed on a particular network segment read packets as they pass into and out of each segment. The trackers need to be selected in various tracks, where they will be able to capture all of the packets entering and leaving a host or network segment.

S.Balaji is Research Scholar with the Information and Communication Engineering, Anna University, Chennai, India (e-mail: sbalajiphd@gmail.com).

Prof.Dr. M. Rajaram is the Vice-chancellor of Anna University, Chennai, India (e-mail: rajaramgct@rediffmail.co.in).

Trackers that are placed on network segments do not always have the capacity, if the traffic level becomes large, to seizure every single packet. Reordering the trackers on each network host will improve accuracy even though the effort of installing them can be significant. The common thing is to be able to seizure all packets so that none can potentially circumvent the IDS. For our purposes we use Snort on the Windows operating system using WinPcap.

The Signature database records enable the IDS to have a set of signature, criteria or rules against which they can compare packets as they pass through the sensor. The database of signatures needs to be installed along with the IDS software and middleware. After the authenticated database is in place, trackers of the Data acquisition module gather data by reading packets from the network and reassemble them. As packets from network can receive out of queue order information, or can be reproduced. Moreover, packets arrive at a high speed therefore the Data storage is required to store the packets.

The Analyzer module compares the packets it observes with the signatures or rules of normal patterns of behavior stored in Signature database using pattern-matching algorithm. We use the well-known Aho-Corasick algorithm for performing pattern matching. If analyzer finds any match then sends appropriate alert message for known attack to the Counter-measure module. Also it enters entry in log file about the event that due to the alert. If sensor does not find any match then sends segmented data to abnormal data conversion may be found. If Anomaly detector finds any anomaly then send appropriate message to Signature generator. The Signature generator creates rule or signature and makes new entry in Signature database.

When Counter-measure module receives the alert message of known attack from Analyzer, it notifies the administrator in one of several ways that the administrator has specified information in annexing systems. The module might display a pop-up window or sends an e-mail message to the designated individual, for example. Besides the automated response sent to the network observer, this segment can be configured to make specific function at the same time that an alert message is received. Typical actions are: i) Alarm, in which an alarm is sent to the observer, ii) Commit, in which the packet is committed without an error message being sent to the originating computer; and iii) Rollback, which controls the IDS to stop and restart network traffic and thus stop especially serious vulnerable attacks. This segment is also used by network observer to evaluate the alert message and to take proper actions such as dropping a packet or closing a connection. The observer can expect having to fine-tune the signature database to account for situations that seem to the IDS to be intrusions but that are actually made a legal traffic. For example, an correction might be made to enable traffic that might otherwise be seen by the firewall as apprehensive, such as a susceptibility scan performed by a scanning device located at a particular layout IP information. The IDS information would be specified to add a rule that changes the action performed by the IDS in response to traffic from that IP address from Alarm to Drop.

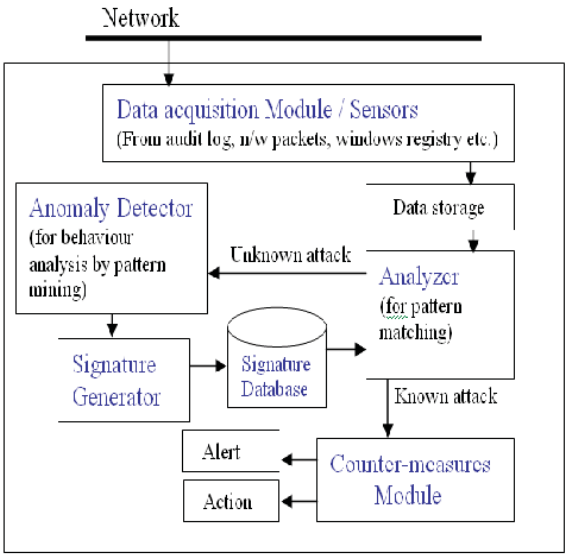

Fig. 1 Structure of Hybrid IDS

B. Characteristics of Hybrid IDS

The following are the list of unique characteristics of our hybrid IDS.

a) It is easy to install and configure.
b) It is adaptive in nature and adapts the changes in user and system behavior.
c) It is constantly with minimal human observation. It will construct signatures of new attacks.
d) Design of hybrid IDS makes it trouble shoot, so that it will be able to improve from collisions. It is able to get its prior state and resume its operation without any adverse effect.
e) It is able to monitor itself and detect attacks on it.
f) It consumes less memory to function. It can be run with less overhead on the systems where it is installed.
g) It is accurate and thereby there will be less number of false positives and false negatives.

III. Proposed System

In our system, instead of focusing on the intrusions we try to identify the intruders and before entering into the system by verifying data and user validity and increase the server capability.

A. System Architecture

In UDIS architecture, N number of clients can make the request. The critical part is how to respond to all the requests before expiration and detect the invalidate requests. Instead of directly accessing the web server, the connections are interrupted by the proxy who holds the requests for verification and validation. After the verification process the requests are classified into valid and invalid requests by passing through the session supervisor which is used to detect duplicate requests. The checking process is divided into user check and data check. The user check is taken over by the handler using the ID against the database the server holds. The

data check is taken over by the training set which holds the past history of that data if it is already there. If it's new, the data request will be made into the entry. The regular requests are placed in a regular queue. The invalid requests are not going to be rejected, it have to be reshaped against our syntactic architecture and placed in abnormal queue. The requests in both queues are passing to the web server.

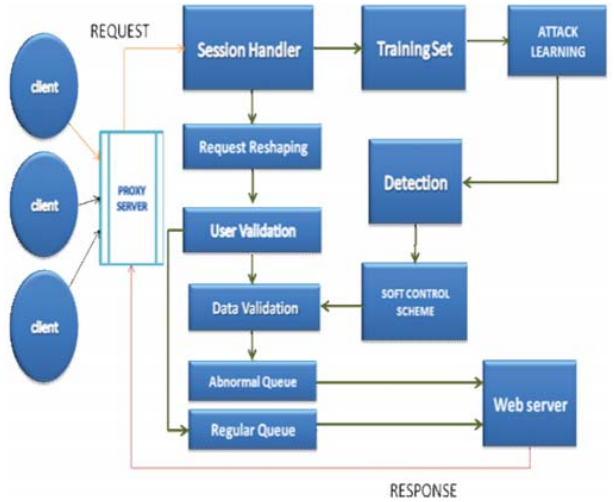

Fig. 2 System Architecture of UDIS

B. Session Handler

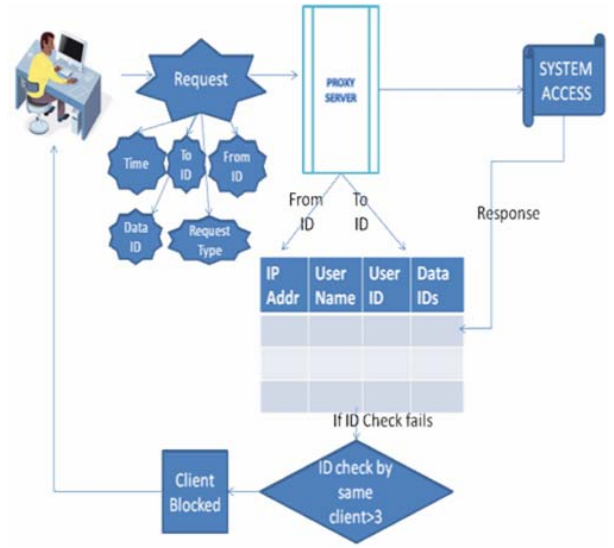

Fig. 3 Flow of Session Handler

Normally the user has to give the request either to upload or download the data from the server. The first of this process is request reshaper. When the user makes the request, the request is split into five parts like Request time, Sender ID, Destination ID, Data ID and Request type which can be either upload or download. After retrieving this information, the proxy server interrupts this data and using sender and destination ID it cross checks the base to conform the user authenticity. If the user is authenticated, the data ID requested check for availability if it is a download type and the system access will be granted. In case the authenticity fails, it checks the number of times the ID fails and if it exceeds 3, that particular client will be preempted from the network.

C. User Validation

Once the user is entered into the system, the attributes have to be derived from the user to determine the characteristics of profile creation. The uniformity of the user determines the time at which the user entered the system. The no of exceptions denotes the violations that take with the approval of administrator. If any interruptions take place into account that rises to the termination of session have taken into account. The violations without the approval of administrator have taken into account with its status light, normal and severe. No of hours spent by the user till the end of the formal session have taken into account as mentioned in Fig 4. Based on the all the value of the attributes, a profile is created for the user, where some of the attributes will be updated in a daily, weekly or monthly basis. The output of the profile creator determines whether the entered user is legitimate or intruder.

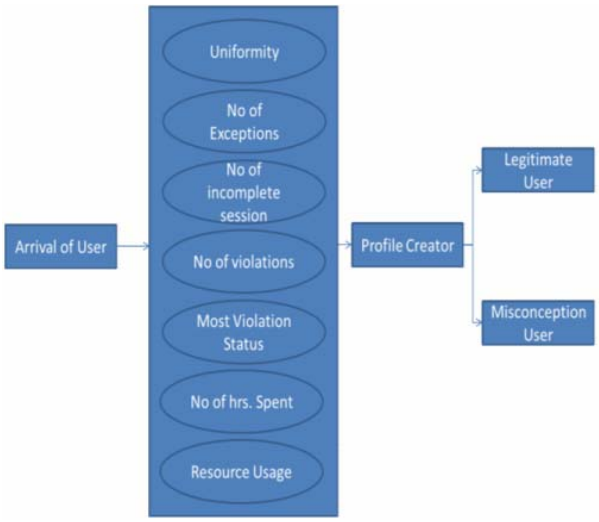

Fig. 4 Parameter Evaluation for Users

D. Data Validation

After user validation, the data to be uploaded have to given an ID that can be generated using the attributes of the data like data type, user info, Time of Entering System (TOES) and size of the data. Before that the data have to be shaped based on two parameters like size and format to increase the optimal performance which reduces the storage of unwanted bulky data. At the end of this process, two IDs are generated for that data among that one will be returned to the user as response for future reference and data authenticity for download. In case of download, the system first checks the data availability and then it cross refer the ID given by the asked user against the user who uploaded that data. If it matches, then it needs access permission from the owner unless it is public. After all this process results are authorized, data will be provided to the determined user.

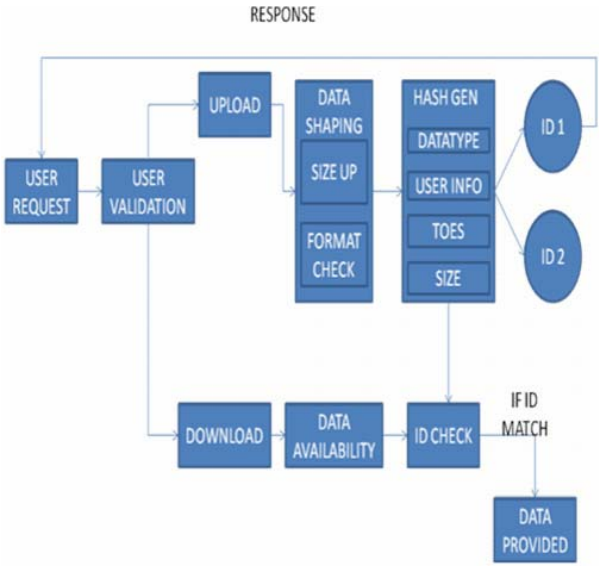

Fig. 5 Data Validation

E. Data Queuing

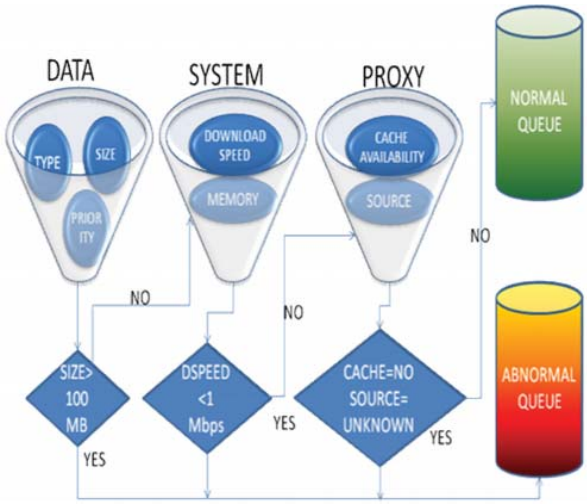

Fig. 6 Process of Data Queuing

All the above process tells about the efficient way of transferring data to and from the server. When the request exceeds the capability of server, there is a possibility of rejection of requests. In order to avoid this, two type of queue are formed in this system to hold the requests in two categories like normal and abnormal queue. The request is put into the one of the above queues by passing through 3 stages. First, it is based on the data whose attributes are type, size and priority and it checks whether the size exceeds 100 MB. Second, it is based on the system which takes the attributes like download speed and memory available and checks if DSpeed is less than 1 Mbps. Third, it is turn of proxy checks the cache availability and source for the data requested. If the request has passed all the three stages, then data will be placed in a normal queue otherwise it will be placed in abnormal queue.

F. Request Reshaping

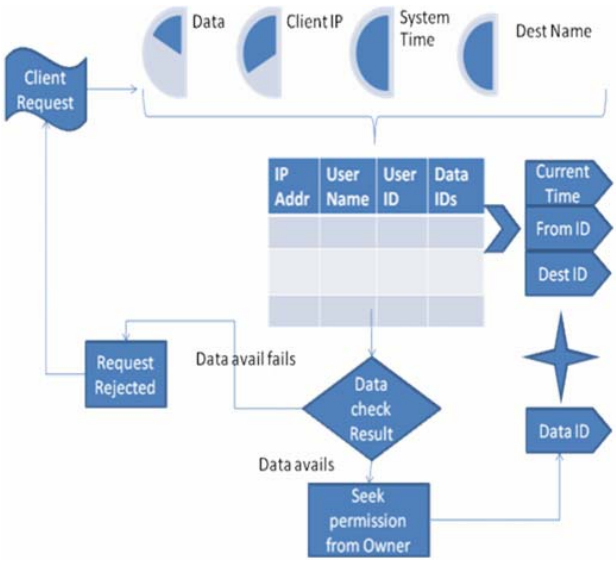

Fig. 7 Shaping of Request

Before processing the request, the request has to be reshaped in order to avoid the false positives. Here the client request is split into four parts like data, Client IP, System time and destination name. The proxy checks the splitted data against the base to check the data availability. If data avails, then it has to verify the access permissions and data ID is added if it is public otherwise it have to seek permissions from owner of the data or the requestor have to provide the original data ID. Now the client IP is replaced by the current time of created shaped request. Finally the shaped request holds the current time, form ID, to ID and data ID.

$$ST = UVT + DVT + DAT + DLT \quad (1)$$

$$MT = Res\ T - Req\ T \quad (2)$$

$$UDIS\ Time = ST + MT \quad (3)$$

IV. Performance Analysis

TABLE I
PERFORMANCE COMPARISON BETWEEN EXISTING AND UDIS

Data Size	Connections UDIS	Connections Exited	Request Time	Response Time	Mean Time	Security Time
50	4	1	2	5	3	3
100	4	1	2	10	8	3
150	4	1	2	15	13	3
200	4	1	2	20	18	3
250	4	1	2	25	23	3
300	4	1	2	30	28	3
350	4	1	2	35	33	3
400	4	1	2	40	38	3
450	4	1	2	45	43	3
500	4	1	2	50	48	3

TABLE II
PERFORMANCE COMPARISON BETWEEN EXISTING AND UDIS

Data Size	Total Time Exiting	Total Time UDIS	Performance Exiting	Performance UDIS
50	3	6	12	48
100	8	11	11	44
150	13	16	10.67	42.67
200	18	21	10.5	42
250	23	26	10.4	41.6
300	28	31	10.33	41.33
350	33	36	10.29	41.14
400	38	41	10.25	41
450	43	46	10.22	40.89
500	48	51	10.2	40.8

Sample Calculation

No of Connections (Existing)=1
No of Connections (UDIS)=4
Data Size=50kb
Request Time=2ms
Response Time=5ms
Security Time=User Validity time+Data validity time+Data avail time+Data log time=1+1+0.5+0.5=3ms
Mean Time=Response Time – Request Time=3ms
Total Time (Existed)=Mean Time=3ms
Total Time (UDIS)=Mean Time + Security Time=3+3 =6ms
Performance=(Total Time* No of Connections)/Data Size
Performance (Existing)=[(3*1)/50]*100=12%
Performance (Existing)=[(6*4)/50]*100=48%

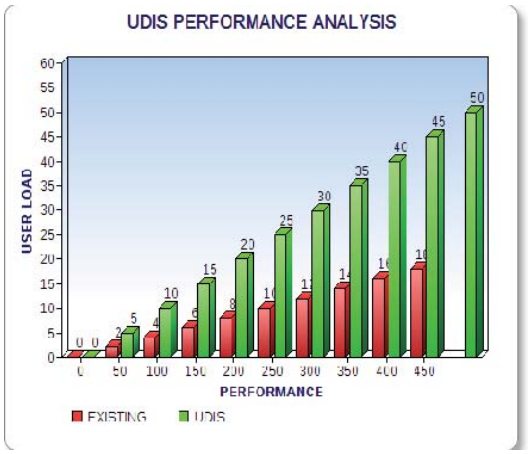

Fig. 8 Comparative Performance of UDIS

V. CONCLUSION

In this paper, the system UDIS improved the data availability and decreases the response time without violating any security mechanisms. Even though the security time increases the response time, the no of connections permitted increases the performance level compared to the previous system.

REFERENCES

[1] D.J. Day and Z. Zhao, "Protecting Against Address Space Layout Randomization (ASLR) Compromises and Return-to-Libc Attacks Using Network Intrusion Detection Systems.," International Journal of Automation and Computing, vol. 8, no. 4, pp. 472-483, Dec. 2011.
[2] W. R. Cheswick, S. M. Bellovin, and A.D. Rubin, "Intrusion Detection," in Firewalls and Internet Security: Repelling the Wily Hacker, 2nd ed. Boston: Addison-Wesley, 2003, pp. 279-283.
[3] Ryan Trost, "Intrusion Detection Systems," in Practical Intrusion Analysis: Prevention and Detection for the Twenty-First Century, Karen Gettman, Ed. Boston, USA: Addison-Wesley, 2010, ch. 3, pp. 53-85.
[4] P. M. Mafra, V.Moll, J. da Silva Fraga, and A.O.Santin, "Octopus-IIDS: An Anomaly Based Intelligent Intrusion Detection System," in IEEE Symposium on Computers and Communications, Riccione, Italy, 22-25 June 2010, pp. 405-410.
[5] S. Jajodia, Intrusion Detection Systems, R.Di Pietro and L.V. Mancini, Eds. New York, US: Springer, 2008.
[6] W.Li, Z.Li, H.Shi, and W.Li, "A Novel Intrusion Detection System for E-Commerce System," in International Conference on Management of e-Commerce and e-Government, Nanchang, China, 16-19 September 2009, p. 454.
[7] Z.Trabelsi and R.Mahdy, "An Anomaly Intrusion Detection System Employing Associative String Processor," in Ninth International Conference on Networks, Menuires, France, 11-16 April 2010, p. 220.
[8] C.C.Lo, C.C.Huang, and J.Ku, "A Cooperative Intrusion Detection System Framework for Cloud Computing Networks," in 39th International Conference on Parallel Processing Workshops, San Diego, USA, 13-16 September 2010, p. 281.
[9] J.Yang, X.Chen, X.Xiang, and J.Wan, "HIDS-DT: An Effective Hybrid Intrusion Detection System Based on Decision Tree," in International Conference on Communications and Mobile Computing, Shenzhen, China, 12-14 April 2010, p. 70.
[10] J.Mallery et al., "Intrusion Detection and Response," in Hardening Network Security, Jane K. Brownlow, Ed. Emeriville: McGraw-HIll, 2005, pp. 365-386.
[11] E.Flior et al., "A Knowledge-Based System Implementation of Intrusion Detection Rules," in IEEE Seventh International Conference on Information Technology, Las Vegas, USA, 12-14 April 2010, pp. 738-739.
[12] S.Ohtahara, T.Kamiyama, and Y.Oyama, "Anomaly-based Intrusion Detection System Sharing Normal Behavior Databases among Different Machines," in Ninth IEEE International Conference on Computer and Information Technology, Xiamen, China, 11-14 October 2009, pp. 217-219.
[13] D.L. Prowse, "Computer Security," in CompTIA Security+ SY0-201 Cert Guide. Indianapolis, USA: Pearson Certification, 2011, ch. 2, p. 35.
[14] L.Gui-Xiang and G.Wei-Min, "Research on Network Security System Based on intrusion Detection," in International Conference on E-Business and E-Government, Guangzhou, China, 7-9 May 2010, p. 2096.
[15] F.Haddadi, S.Khanchi, M.Shetabi, and V.Derhami, "Intrusion Detection and Attack Classification Using Feed-Forward Neural Network," in Second International Conference on Computer and Network Technology, Bangkok, Thailand, 23-25 April 2010, p. 262.

S.Balaji, M.E,MISTE doing his Ph.D in Anna University, Chennai and currently working as a HOD in dept of CSE at SCAD College of engineering and Technology, Tirunelveli. His research interests are Wireless networks Mobile Computing, Network Security with apps, Wireless Sensor Networks, Cloud Computing. He has presented many papers in National and International conferences in network security, Mobile Computing, network security, and Cloud Computing. He has organized and conducted various national and international conferences, International Seminars and National Workshops. And also his methodology of teaching about TCP & UDP is hosted on Wipro Mission 10x portal. He is a life time member of ISTE.

Prof. Dr.M.Rajaram M.E.,Ph.D is the Vice-Chancellor of the Anna University, Chennai. He obtained his B.E. in Electrical and Electronics Engineering from Alagappa Chettiyar College of Engineering and Technology, Karaikudi, in 1981, and started his career from 1982 as a Lecturer in Government College of Engineering, Tirunelveli, to satisfy his passion for teaching and research. Subsequently, he obtained his M.E. in

Power Systems from Government College of Technology, Coimbatore, in 1988. Besides having a strong technical expertise and analytical skills, he acquired his Ph.D degree from PSG College of Technology, Coimbatore, in 1994. He has contributed to the areas of Computer Networks, High Voltage Engineering, Measurement and Instrumentation, Adaptive Controller, Electro-Magnetic Theory, and Intelligent Computing with his 157 publications in renowned research journals, 111 research publications in International Conferences, 73 research publications in National Conferences, more than 100 technical reports and six technical books some of which he has co-authored. As a research guide, Dr.M.Rajaram produced 30 Ph.D's (besides six candidates who are awaiting evaluation reports) and four M.S. scholars in various fields. At present, 10 research scholars are pursuing their Ph.D. under his direct supervision.