iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 4 **Issue:** VI **Month of publication:** June 2016

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Authentication and behavior monitoring in an online vote Polling system using middleware Security

Rajasoundaran.S[1], Narayanasamy.P[2], Jeevakumar.R.[3]
Department of IT , Anna University

***Abstract:** An online vote polling system conduct by government of our country should not believe any third party software and utility program which are running in operating system of server. In server machine, a large scale software system is used to handle the large amount of overload conditions and a lot of HTTP requests and responses. As well as there is a chance of activation of residual faults which are escape from software testing phases and intentional faults can decrease the performance and increases the system downtime with huge costs. Most existing OS-Level anomaly detection techniques are based on static and worse thresholds or on lengthy profiling phases. So, these are not well-suited for systems subject to highly variable and non-stationary operating conditions. Hence, the core concept of proposed system is implementing both application security and system security for a specific web application can increase the reliability, security and robust. Application security encompasses with lot of vulnerable attacks on web application like remote code execution, SQL Injection and cross site scripting. System security includes various dynamic threshold values for non-stationary variables of all active processes and file system. Thus a strong reliable and robust vote polling system is proposed here.*

***Keywords :** Security, Behavior monitoring, Process, Mobile, Server.*

I. INTRODUCTION

A web application security is not enough for a highly sensitive data roaming application. Along with this the centralized server should also be secure. To increase the reliability of this proposed system, an OS-level security is implemented in server. hence, the performance issue in overload conditions due to changes in the workload can be rectified. The evolution of software system causes increase in their complexity, which forces the integrated subsystem in to complex. Most organizations totally rely on the internet and the partnering and data exchange opportunities in the network connectivity bridge. However, the current internet and networking environment is full of threats, both external and internal. This causes organizations and their published applications to be potential attack targets each and every second of a day. Thus the scope of this proposed system is to restrict the file system access and process memory access and changing their attributes from network attacks and system side attacks in a complex software system.

Applications are becoming more complex that tends to faults or bugs in every piece of software. Some of the bugs are security related, so their existence creates a security vulnerability that can be exploited to create a real security problem or incident. When these two facts are combined, it can be concluded with need of tools and mechanisms to detect the actions of potentially malicious code and well automated applications. Also need to identify the known and unknown software faults, system features to gain access, privileges against the published issues or implicit system security policy. Thus the anomaly detection is an important mean to design diagnoses procedures for timely identification and activation of proper countermeasures, since anomalies may be related to the activation of faults, for performance issues and to intentional faults. This system is an application specific anomaly detection, in which all variables of each and every process related to that particular application is being monitored and set a threshold value for each variable. It not only processes, but also monitors the file system which contains sensitive files. For this, kernel modules are developed which precisely get information of process and file system.

By applying system security in central server and application security in an application, robustness and reliability are improved. Here, online vote polling system for an election commission is implemented with the above solutions. E-voting system gains more and more public interest. Some countries offer their citizens to participate in elections using electronic channels. The term e-voting stands for the possibility of voting electronically in general. Thus, e-voting includes voting by the use of handheld devices and electronic voting machines in voting booths as well.

II. LITERATURE SURVEY

The feasibility study made to understand the issues in exiting system which adopts the OS-level monitoring techniques is to be discussed in this chapter.

Bovenzi, A. et. al (2015), have proposed OS level process monitoring to inspect the conditions and behaviors of air traffic systems . Here linux Redhat EL 5 and windows server 2008 platforms are taken for this system implementation.

Khanna, G. et. al (2006) have proposed the monitoring the dynamically changing parameters of process subsystem in OS including runtime behavior monitoring and anomaly detection etc.

Lim, G. et. al (2013) have discussed about the difficulties of application side development techniques for mobile devices. In addition, they have techniques related to use minimal resources of mobile devices for monitoring procedures.

Razzaq, A. et. al (2011) have concerned about intelligent intrusion detection systems and rapidly growing attacks and the cyber problems with the relevant solutions.

By analyzing these works the existing system has been designed for providing feasible solution in process monitoring system.

III. SYSTEM ARCHITECTURE AND MODULE DESIGN

A. System Architecture of Proposed System

The following system architecture of proposed system explains the overall functional view of this proposed system. Also the interaction between system security modules and web application modules are diagrammed.

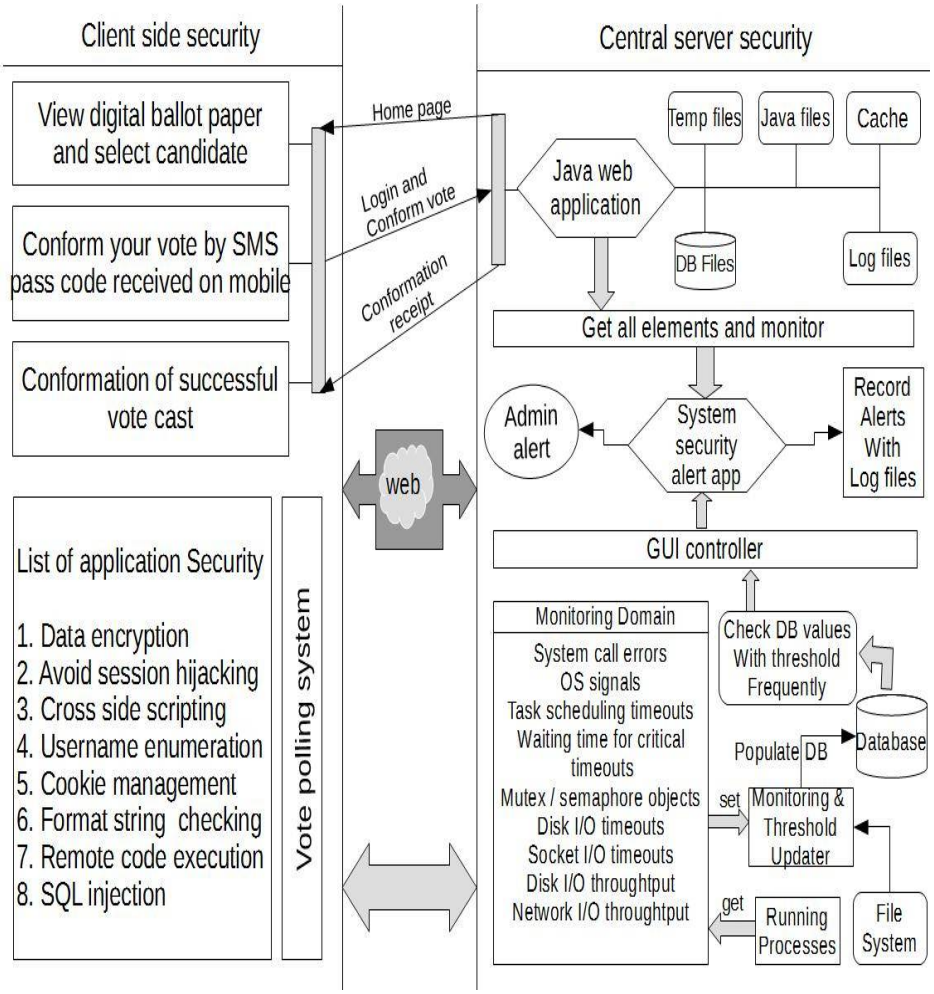


Figure 3.1: Secure Online Vote Polling System

Volume 4 Issue VI, June 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

B. Detail Design of System Security Module

Full internal architecture of System Security Module is designed as follows.

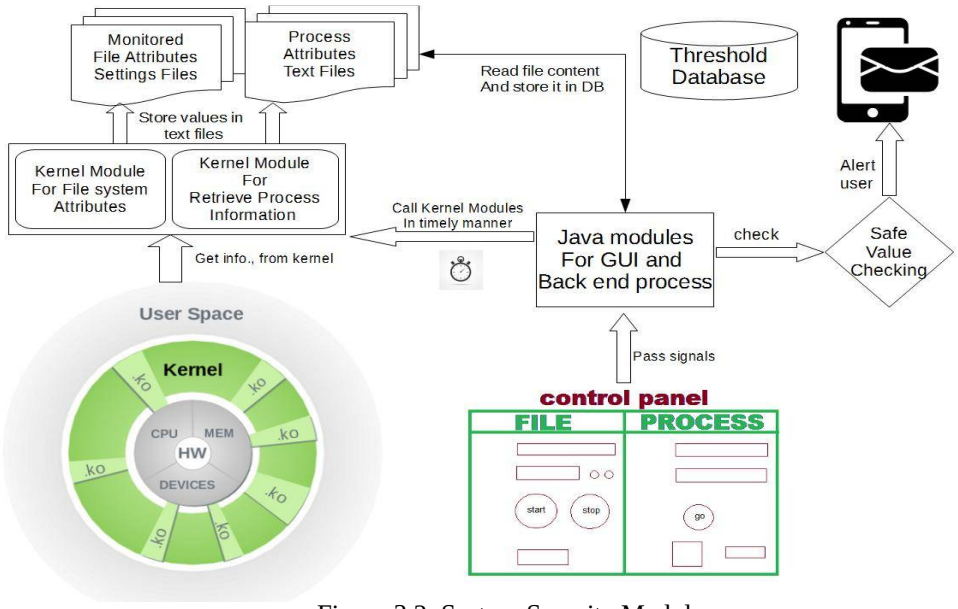


Figure 3.2: System Security Module

For retrieving each and every information of currently running process or out own interested process, some kernel modules are developed which able to directly access the kernel of linux. When coming to file system attribute retrieving, it's also need of kernel module create. So only here, two separate modules are created. Additionally, on these modules IO manipulation code are written to store the retrieved information into file system. After this java module take responsibility for storing those information into a database and fix threshold values for each attributes. Also for Administrator a GUI is develop which having options of selecting process to be monitored, files to be secured. In case of huge variation threshold value is found this module automatically sent mail and SMS about this. In future enhancement, this module will able to resolve the problem.

C. Detail Design of Web Application Module

Full internal architecture of Web Application module is designed as follows.

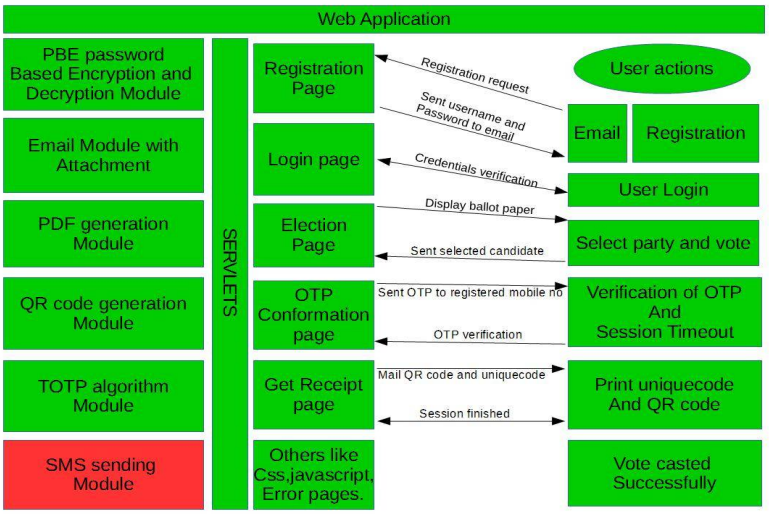


Figure 3.3: Online Vote Polling System

Election Vote polling system is taken as web application because the data roaming on this system should have security because of its sensitivity. This web application is developed with hibernate framework, backend as oracle. also application security like form validation, user authentication, vote conformation, TOTP algorithm, PBE algorithm, QR code generation and other encryptions decryptions are implemented successfully. In future enhancement network security modules will be implemented with robust.

D. Password Based Encryption Algorithm (PBES)

A PBE algorithm compute a secret key from the password selected by user. Two standards are used to create a secret key from given password. First one mix a random number called salt with password to create the key. Second one creates the key without salt which may be vulnerable against brute force attack

E. Time-Based One-Time Password Algorithm (TOTP)

TOTP creates one time password from generated timestamp value . The time stamp value is modified for every request transmitted to server. The timestamp value based secret one time password is created which is used to authenticate the user at different time intervals.

F. Rule Based Algorithm (RBA)

Gather information of all files related to web application and each and every details of all processes that are currently running. After, build database by this information and update it dynamically. From about database create several threshold values and encrypt it and store it in secure manner. Whenever the behavior of our monitored elements is exceeds this threshold values alert the admin.

IV. IMPLEMENTATION AND RESULTS

A. Registration

All the form validation of registration are performed in this module . All required values from voters are collected, database operations with hibernate framework and the form validations are done in this page. This registration form module validate the name and surname fields, the e-mail, username, age, zip code and everything else, which restricts unnecessary calls to the server.

B. Election Attributes

After successful login, a session is started with username and password. This module adds the user selected party name with that existing session. Party name and emblem are fetched from oracle database. And then CRUD operations for party registration added with admin rights. Behind this operations, session is maintained, once a session expires, the voter has to re-cast their vote. After a successful casting of vote a receipt is generate. By this the vote is validation is conformed and database is updated with true flag. All dataroaming from client to server are encrypted with AES algorithm and transferred in secure manner.

C. Email And Registration Conformation

Once registration completed successfully then this page is appeared. If there is anything fault with central server DB operations or user credentials then the error page is displayed. In future, the problem will be identified precisely and details of that problem will be displayed.

D. One Time Password Confirmation

For checking the valid user OTP is generated to the registered mobile number. Here the TOTP algorithm is used to generate OTP. And Way2sms and 160by2 API for java is used to sent SMS. This module is yet to complete, because of error code 500 for server in 160by2 and also cannot find the action hashvalue from way2sms page. Action string for their URLs are taken and integrated with web application, by this action string a particular end-user login is detected. If needed a GSM module can be integrated along with this application.

E. QR and Unique Code Generation

Successful casting of a end-users vote is conformed with this receipt generation module which contain QR and hash value. Conformation of casted vote is ended with this module and all details are put in to session. From current session username, party

name, and OTP are gathered and built with a string and it is converted into a hash code and a QR code. After this, these are mailed to voter registered mail. For generating QR code One-dimensional matrix is used and barcodes are designed. After the QR code is scanned successfully, it creates one two dimensional image which is analyzed by processor using suitable image processing technique. Once the noises are eliminated from the image, the QR code image is converted in to binary format for verification process.

F. Result Publication

Live result is developed in this module. To do that, auto refreshing of page is used. Also in this page, top leading parties are displayed dynamically. A vote may be casted from various electronic devices are collected by the central server and calculate the first three position political parties in servlet. This values are updated frequently while that particular page was refreshed.

G. Statistical Reports

Last five years statistical report aretabulated in this module with an interactive that details of constituencies in Tamilnadu. Also it contain winner margin of each parties and election history from 1977 to 2011 is detailed. And electionparticipants parties, ranking, vote count of all parties, etc are integrated in this module. Tamil Nadu has 234 assembly constituencies. All these constituencies statistical information are populated.

H. Process Information Gathering and File System Securing

Here, linux kernel 4.0 is taken to retrieving process information using proc file system. This provides a linkage between kernel side and the user side spaces. By accessing procfilesystem all information about linux is gathered. Also kernel module are designed which is store information of file system in correct manner. With above information GUI is created and from java modules all the information about file system and processes are processed successfully. As explained above, the overall process information are dynamically inserted in to the database. Then, all these data are processed and a defined threshold value is fixed for each processes. With this value the system monitors those processes and when threshold is overflows the specific peoples are alerted.

I. Performance Analysis

Approximate time taken to complete the several tasks are tabulated in able 5.1.

TABLE I: EXECUTION TIME FOR VARIOUS TASKS

SN.No	Tasks	Time to complete
1	Casting a vote	15 seconds
2	TOTP Algorithm execution	3 seconds for generating a OTP code
3	AES Algorithm execution	4 seconds for 20,527 bytes of input
4	Sending E-Mail phase	8 seconds
5	Generating hash and QR code	4.21 mille-seconds with 20 and blurriness ratio
6	OTP Validation	5 to 8 seconds

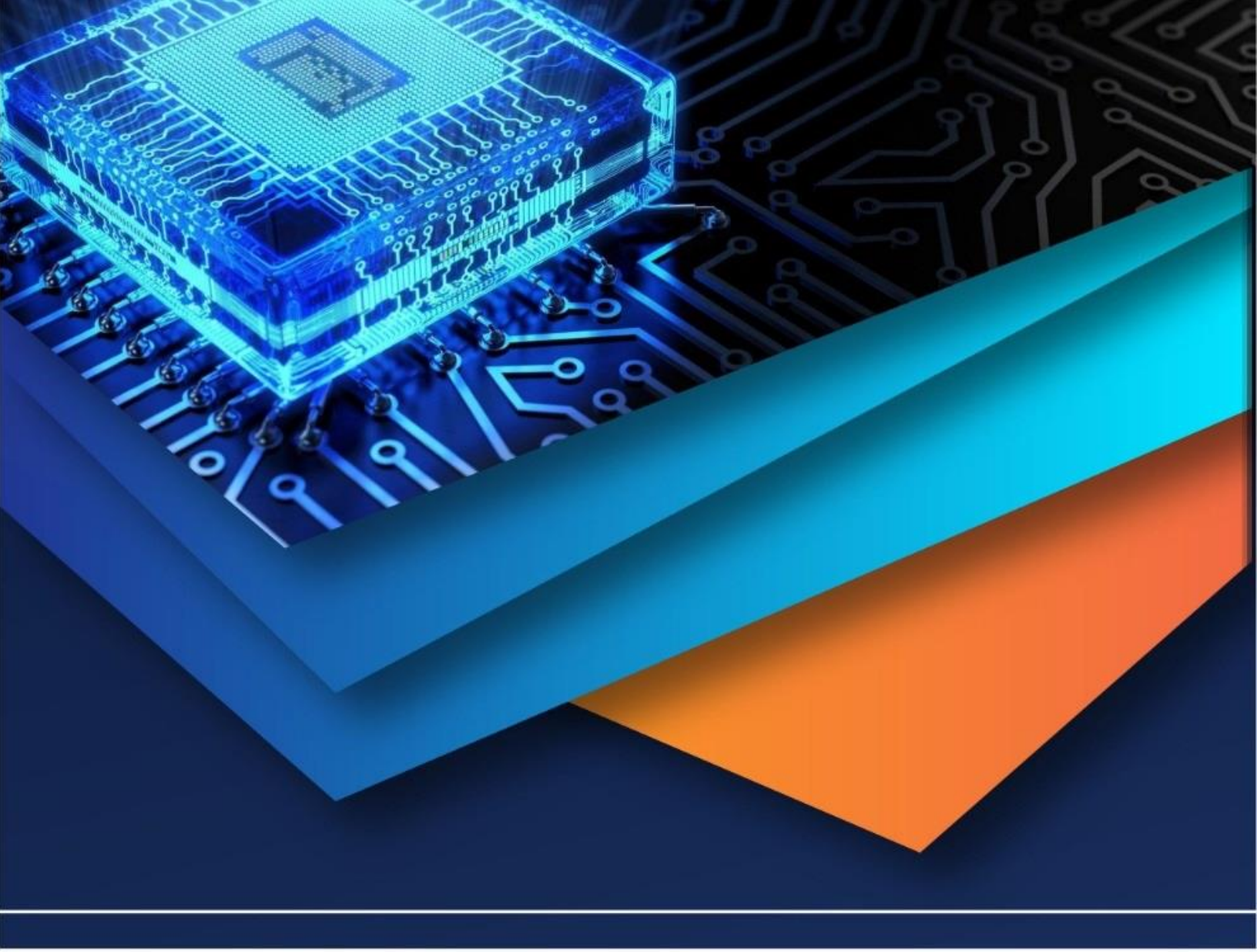
V. CONCLUSION

In this work , a method for identifying anomalies in complex software system is developed with a specific application of online vote polling system which contain sensitive data and have to securely transfer from one node to another node. For achieving this several web application vulnerabilities are prevented to increase client side security as well as in server side file system and processes are

monitored with kernel modules. All the detected threats are transferred to particular administrators and authority persons. So far, only the various application level and system level threats are identified but not fixed. In future, this system will be updated with huge number of vulnerable fixing capabilities and able to work in different kind of platforms. To identify network related vulnerabilities, the proposed system will be updated with various network security modules and to increase ease of access, mobility control features will be added.

REFERENCES

[1] Yau, S.S. and Chen, F.C. 1980, An approach to concurrent control flow checking, Proceedings of 1980 IEEE Transactions on Software Engineering, pp.126-137.
[2] Huang, K.H. and Abraham, J.A. 1984, Algorithm-based fault tolerance for matrix operations, Proceedings of 1984 IEEE Transactions on Computers, Vol. 100, No 6, pp. 518-528.
[3] Board, I. S. 1993, IEEE Standard 1044. 1993, Proceedings of 1993 IEEE Standard Classification for Software Anomalies, 1993,
[4] Madeira, H. and Silva, J.G. 1994, Experimental evaluation of the fail-silent behavior in computers without error masking, Proceedings of 1994 Twenty-Fourth International Symposium on Fault-Tolerant Computing, pp. 350-359.
[5] Chen, W., Toueg, S. and Aguilera, M.K. 2002, On the quality of service of failure detectors, Proceedings of 2002 IEEE Transactions on Computers, Vol. 51, No. 1, pp. 561580.
[6] Woo, S.W., Alhazmi, O.H. and Malaiya, Y.K. 2006, Assessing Vulnerabilities in Apache and IIS HTTP Servers, Proceedings of 2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing, pp. 103-110.
[7] Khanna, G., Varadharajan, P. and Bagchi, S. 2006, Automated online monitoring of distributed applications through external monitors, Proceedings of 2006 IEEE Transactions on Dependable and Secure Computing, Vol. 3, No. 2, pp. 115-129.
[8] Carrozza, G., Cinque, M., Cotroneo, D. and Natella, R. 2008, Operating System Support to Detect Application Hangs, Proceedings of 2008 Second international conference on Verification and Evaluation of Computer and Communication Systems, pp. 117 - 127.
[9] Razzaq, A., Hur, A., Masood, M., Latif, K., Ahmad, H.F. and Takahashi, H. 2011, Foundation of Semantic Rule Engine to Protect Web Application Attacks, Proceeding of 2011 10th International Symposium on Autonomous Decentralized Systems, pp. 95-102.
[10] Kook, J., Hong, S., Lee, W., Jae, E. and Kim, J. 2011, Optimization of out of memory killer for embedded Linux environments, Proceedings of 2011 ACM Symposium on Applied Computing, pp. 633-634.
[11] Lim, G., Min, C. and Eom, Y. 2013, Virtual Memory Partitioning for Enhancing Application Performance in Mobile Platforms, Proceedings of 2013 IEEE Transactions on Consumer Electronics, Vol. 59, No. 4, pp. 786 - 794.
[12] Bovenzi, A., Brancati, F., Russo, S. and Bondavalli, A. 2015, An OS-level Framework for Anomaly Detection in Complex Software Systems, Proceedings of 2015 IEEE Transactions on Dependable and Secure Computing, Vol. 12, No. 3, pp. 366 - 372.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY