ISSN:0975-9646

Self Motivated Intrusion-Tolerant, Detecting and Healing Server

Omprakash A. Jaisinghani[#1], Dr. A. S. Alvi[#2]

[#1] *Dept.: Master in Engineering (I.T.)*
Prof. Ram Meghe Institute of Technology and Research
Badnera(India).
[#2] *Head of Dept. in Information of Technology*
Prof. Ram Meghe Institute of Technology and Research
Badnera(India).

***Abstract*— our main motive research paper is towards intrusion-tolerant server's to provide full functionality of self-healing which provides user independent and automatic healing functionality, now we propose new design for an intrusion-tolerant oriented self-healing server which improves performance quality of server. In this type of server self-healing functional components, including the detection components along with implementation components are added for better result and performance improvement, we add self-healing capabilities into the intrusion-tolerant server that will recovers original file from affected one. It solves lots of problems in the intrusion-tolerant servers, such as the hidden intrusion, and the vulnerable prerequisite of intrusion tolerance which is very common problem now a days, and further enhances the reliability and survivability of the intrusion-tolerant application server.**

***Key words* — Malicious, Intrusion, Self-healing, mirror & Checksum.**

I. INTRODUCTION

Intrusion tolerance is an integration of cryptography and fault-tolerant technology, which is a new network security technology, emphasizing certain parts of the system have been attacked even damaged or attacker who successfully control the system. How to continue to provide services overseas and to ensure that key system data confidentiality, integrity and availability, is considered the modern information security defense in depth in the last line of defense. However, if the intrusion is continuous and successful, it can destroy the prerequisite for intrusion tolerance mechanisms. In addition, if the intrusion or attack does not be detected and blocked, then hidden intrusion occurred. The system turns into a state of unpredictable dangers. In addition, with the long-running, there will be software aging in application server, the performance of the whole system will be declining, and the error will be increased. In all three cases, the application server will quickly lose the ability of normal services, and even crash [1, 2].

An intrusion is defined as malicious, externally induced, operational fault. The word "intrusion" comes from the Latin but current usage covers both senses of "illegal penetration" and "unwelcome act." Self -healing is method to design an intrusion-tolerant oriented server in which establishing a self-healing functional components, including the detection components and implementation components [1].

Thus Intrusion Tolerance combining the aspects of self-healing which recovers the data which is affected by the intrusion. A justifiably be trusted system is defined as one that includes availability (readiness for correct service), integrity (provide identical and correct data), confidentiality (prevention of unauthorized disclosure of information) and integrity (the absence of improper system state alterations).

Intrusion Tolerance in Cloud Computing is very important and critical task a fault tolerant design approach to protect cloud infrastructure is against malicious attacks. This project aims at designing and implementing a framework for Intrusion Tolerance in Cloud Computing; testing the feasibility of the proposed framework against cloud environment and finally using the proposed framework for securing cloud applications and its services [2].

II. FEATURE OF SYSTEM ARCHITECTURE

In response to this paper, an intrusion-tolerant self-healing application server design is proposed in this paper. It combined self-healing software technology and intrusion tolerance technology to make up for their deficiencies, to meet the user's high reliability and high survivability requirements.

Intrusion Tolerance in Cloud Computing is a fault tolerant design approach to defend cloud infrastructure against malicious attacks. This paper aims at designing and implementing a framework for Intrusion Tolerance in Cloud Computing; testing the feasibility of the proposed framework against cloud environment and finally using the proposed framework for securing cloud data and its services [2].

A. Fault Diagnosis:

Fault diagnosis is afraid with identifying that what type of fault is and locations of faults that needs to be isolated before carrying out system reconfiguration or initiating corrective maintenance. Fault detection is very crucial and important aspect of the self motivated server. If the case of intrusions fault diagnosis can be further decomposed into,

a) Intrusion diagnosis, i.e., trying to assess the degree of success of the intruder in terms of system corruption.
b) Vulnerability diagnosis, in this types of vulnerability the channels through which the intrusion took place so that corrective maintenance can be carried out.

c) Attack diagnosis, i.e., finding out who or what organization is responsible for the attack in order that appropriate litigation or retaliation may be initiated.

B. Fault Isolation:

In Cloud Computing environment fault isolation is needed to make sure that the source of the detected error(s) is prevented from producing further error(s).
In terms of intrusions, this might involve:

a) Blocking traffic from an intrusion containment region that is diagnosed as corrupt, by, for example, changing the settings of firewalls or routers.
b) Removing a corrupted file from the system or, with reference to the root vulnerability/attack causes.
c) Uninstalling software versions with newly-found vulnerabilities

We added self-healing capabilities into the intrusion-tolerant server. It resolves some problems present in the intrusion affected application servers, such as the hidden intrusion and the vulnerable prerequisite of intrusion tolerance, and enhances the reliability of the intrusion-tolerant application server [2].

III. SECURED DATA COMMUNICATION CHANNEL

Network security is protecting networks and their services from unauthorized alteration, destruction, or denial, and provision of assurance that the network will performs its critical functions correctly and there will not even a harmful side-effects.
There are no perfectly secure channels in the real world. Channel is a media through which data is travels. There are only ways are present to convert insecure channels into less insecure. It is important to note that most cryptographic techniques are trivially breakable if keys are not exchanged securely. An actually secure channel will not be required if an insecure channel can be used to securely exchange keys to overcome this problem here we will going to use asymmetric cryptography in which there is not a need to transfer key between two communicating medias. The primary advantage of public-key cryptography is increased security and convenience.
Main advantages of asymmetric encryption are that the two users don't need to have secretly stored their keys in order to communicate using encryption and that both authentication of message or file and non-repudiation are achieved using this type of encryption [3].
The public key is made publicly available and is used to encrypt messages by anyone who wishes to send a message to that public key belongs to. The private key is kept secret and is used to decrypt received messages. An example of asymmetric key encryption is RSA (Rivest Shamir Adleman) algorithm which is very popular and powerful tool and provides highly secured encryption and it is an algorithmic tool which provides data transferring without sharing secret key with another person.

Features provided by self motivating system architecture in favor of security are as below,

a) Convenience:

Major problem of distributing the key for encryption is solved here. Everyone provide their public keys in open domain and private keys are kept secret.

b) Provides for message authentication:

User can use Public key encryption for digital signatures which enables the recipient of a message to verify that the message is received from a particular sender.

c) Detection of tampering:

Digital signatures in public key encryption allow the receiver to detect if the message/ file were tempered or not. A digitally signed message cannot be custom-made without invalidating the signature.

d. Provide for non-repudiation:

Digitally signing a message or file is akin to physically signing a document. It is an acknowledgement of the message and thus, the sender cannot deny it [4].

To secure data from unauthorized access file sent from client to server should be encrypted form and security level must be high. It should be encrypted using RSA algorithm so that contents of file cannot be decipherable by unauthenticated user. The best way of RSA algorithm is Asymmetric encryption, which allows Alice to send Bob an encrypted message without a shared secret key; there are two keys are present a secret key (private key), but only Bob knows what it is, and he does not share it with anybody, including Alice. And second key is public key, public key is shared with each and every user that is sharing data or will be share. Figure1 provides an overview of this asymmetric encryption, working of an asymmetric encryption as follows which is consist of single sender and single receiver end:

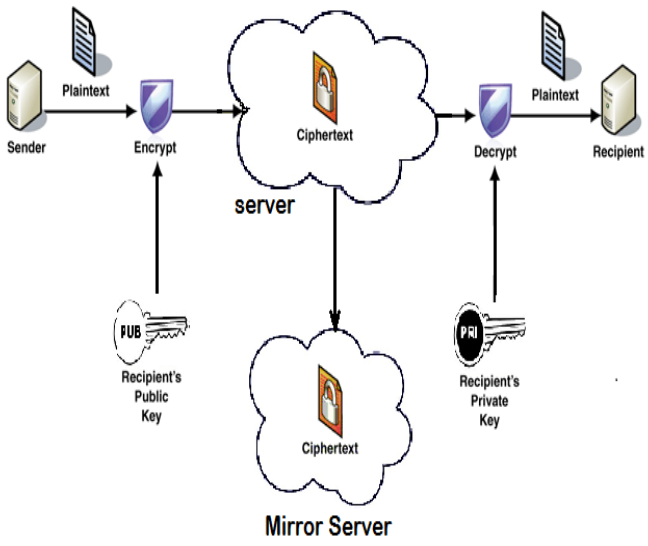

Fig. 1 Data encryption and server storage

IV. SYSTEM DESIGN

Most self-healing system is composed of detect components, implement components and manage components. They made a self-healing ring with the application server, Detect components are used to detect intrusion and fault, and the results on to the manage components; and manage components are used to analyze the results, and made the healing programs, and sent to the implement components; implement components are used to heal the application server based healing programs[3].

In this paper, we expanded the server for intrusion-tolerant self-healing function. This design concept is as follows: Firstly, we combined the component technology and interceptor technology which is expanded a single intrusion-tolerant application server by self healing; then the expanded intrusion tolerance application servers are made of for the collaboration cluster to achieve self-healing target in the system level.

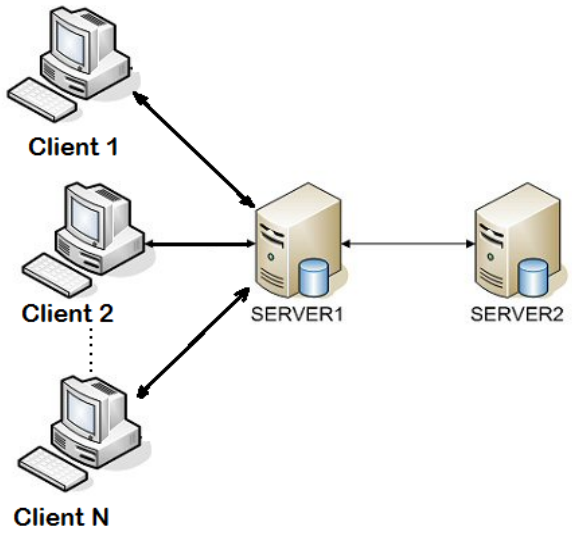

Fig. 2 Structural design

As depicted in Figure 2 the intrusion-tolerant self-healing application server platform is run by the expanded N-mode redundancy servers from the main server, which uses semi-active service model of external services. The application server is a cluster of group members, all group members via a secure group communication system to communicate. This is perception of N client with single server and single backup server. Here all users make request to server1 and server1 communicate with server2 for its mirroring and for file backup.

A. Details of Implementation:

In this paper we present a framework for intrusion tolerance in data server which summarizes availability of integrity, authentication and confidentiality can possibly be integrated in the cloud or within its services.

For integrity of data, we have used Checksum MD-5 code while storing and accessing the file. Checksum MD-5 code is used to provide intrusion tolerance for data servers in the cloud. Performance analysis shows that the proposed scheme is highly efficient and resilient against malicious data modification attack.

Authentication is the act of confirming the truth of an entity or legitimate user. This might involve confirming the identity of a person or software program. Here we will provide user id and password for validation of legitimate user. Confidentiality is a set of rules or a promise that limits access or places restrictions on certain types of information. To maintain confidentiality of data there will be provision for encryption of data using cryptography tool [4, 8].

Implement components include:

(1) Checkpoint actuator, if the application server process by setting checks points, will have a greater overhead, Therefore, this thesis used in your application server component level on checkpoint set, preservation and restoration application server running the correct state of the business component for the transfer and restart the implementation of the actuator uses. The checkpoint setter set the rules through the checkpoint and checkpoint consistency rules on the application server is running the business components of the checkpoint set, and saved to the checkpoint after the notification from the storage medium in the next self-healing Manager step operation; the check point reduction, receivers to self-healing management restore the signals emitted from the storage medium to extract checkpoint restore operation. [8]

B. Design of detect components:

Failure detector, for different fault levels, divided into a server-level failure detection and component-level failure detection. The former use of Secure Group Communication System, through the heartbeat cluster system technology application server processes failure detection. The server-level failure detectors include:

Failure state table, failure trigger and failure monitor Fault status table according to the received "heartbeat" define the real-time updates on each application server cluster fault condition; fault monitor the rules under the heart, is responsible for receiving and sending "heartbeat"; Fault trigger fault status table according to the fault condition information, send and receive results of application server failures, and trigger the application server's self-healing manager.

Check-summing is a well known method for performing check integrity of document or text. Checksums can be computed for disk data which stored persistently and useful to check data existence. Data integrity can be confirmed by comparing the stored and the newly computed values which generated at every data read time. Checksums are generated using a hash function. Every time data is accessed, checksum is again created and matched with saved checksum in this file. If it is matched then it shows that data integrity is persisting. If it is not matched then it shows that data is corrupted or any alteration is done in that file. The use of hash functions has become a standard in Internet applications and protocols [5].

Hash functions map strings of different lengths to short fixed size results. It processes a variable-length message into a fixed-length output which is 128 bits. These functions are usually intended to be collision resistant,

which means that finding two strings that have the same hash result should be infeasible. Besides to basic collision resistance, functions like MD5 (Message digest) have some properties one of them is randomness. Here MD5 hashing algorithm is used to generate Checksum MD5 hashing algorithm is one way encryption in which we able to do only encryption decryption is not possible for MD5 output. Once the data is encrypted into message digest then it is not feasible to retrieve same input from encrypted message digest MD5 output is unique for each unique input given to Message digest algorithm.

C. Design of manage components:

The self-healing algorithm and parameter selection will provide the infected component present in server side. These components are executed on regular interval at the time when server traffic is low or when server is still mode or it is in halt condition. Halt condition is that when server has no pending request, the self-healing configuration table is marinated on regular interval; Component manager is used for all self-healing components registration and cancellation; Reactive recovery device, received the output signal of fault detection and voting, and configured according to self-healing algorithm. It controls checkpoints, migration and restart actuator to reactive recovery; proactive recovery device, through the security group system, linked to all the proactive recovery devices, and triggered clock synchronization for each application server proactive recovery; synchronous clock, send a synchronization time.[6,7]

It is recognized that disks are an inherently unreliable component of computer systems. Mirroring technique allow a system to automatically sustain multiple copies of data so that in the event of a disk hardware failure a system can continue to process or rapidly recover data. Mirroring may be done in the neighborhood where it is specifically to cater for disk unreliability, or it may be done remotely where it forms part of a more sophisticated failure recovery scheme, or it may be done both locally and remotely, especially for high availability systems. Normally data is mirrored onto physically identical drives, though the process can be applied to logical drives where the underlying physical format is hidden from the mirroring process [7].

V. CONCLUSION

With the extensive use of self healing servers, it also services the survivability and reliability requirements are high, the use of a security technology alone is difficult to prevent all attacks, this paper oriented intrusion-tolerant self-healing application server design and implementation of effective state for intrusion tolerance unknown state, for periodic recovery, so that each application server to prevent the occurrence of a hidden intrusion, but also to avoid the phenomenon of software aging; also has a certain continuity and success for the intrusion of the introduction of a reactive recovery, to ensure that the application server cluster can tolerate the intrusion of the voting mechanism to meet the prerequisite for the design and realization of the user on the application server to meet the high reliability and high survivability requirements.

In this paper we had propose a framework for intrusion tolerance based on the layered design of computing architecture. For the validation of framework, we will simulate Intrusion Tolerant environment with security controls and techniques required for intrusion tolerance and along with recovery module which recovers any infected data. We will use Intrusion Tolerance via Threshold Cryptography mechanism for validation which increases functionality of server.

This framework will capable of detecting and recovering data which is infected by intrusions in the server environment. This detection and recovery process is held on regular interval when server is in still mode or in redundant condition which make server busy for all time mainly this is done when server has no request to resolve or to respond. Performance analysis of framework shows that the overhead of integrating intrusion detection and recovery mechanism in Cloud Computing environment.

REFERENCES

[1] XF. ZHANG and F. ZHENG eng, “Intrusion Tolerance Technology-Survey and Direction,” Information Security, 2004 (31):19-22.

[2] HariGovind V. Ramasamy, Adnan Agbaria, William H. Sanders (2004). “CoBFIT: A Component-Based Framework for Intrusion Tolerance”. In proceedings of the 30th EUROMICRO Conference. (pp. 591-600).

[3] Partha Pal, Rick Schantz, Michael Atighetchi, Joseph Loyall (2009). “What Next in Intrusion Tolerance”. In proceedings 3rd Recent Advances in Intrusion Tolerance Workshop at the IEEE/IFIP Distributed Systems and Networks Conference (DSN 2009), 29th June - 2nd July, 2009, Estoril, Portugal

[4] Y. Deswarte, L. Blain, and J.-C. Fabre (1991). “Intrusion Tolerance in Distributed Computing Systems”. In proceedings of the Int’l Symp. Security and Privacy (S&P ’91). (pp. 110-121).

[5] Meng Qiang, Zhou Rui-peng, Yang Xiao” Design and Implementation of an Intrusion-Tolerant Self-healing Application Server” 2010 International Conference on Communications and Intelligence Information Security.

[6] D. Powell, R. Stroud (2003). “Malicious-and Accidental-Fault Tolerance for Internet Applications: Conceptual Model and Architecture”. Technical Report 03011, Project IST-1999-11583 MAFTIA, Deliverable D21, LAAS-CNRS.

[7] Popovic Kresimir, Hocenski, Zeljko (2010). “Cloud computing security issues and challenges”. In proceedings of the 33rd International Convention, IEEE Transactions. (pp. 344 – 349).

[8] James C. Reynolds, James Just, Larry Clough, Ryan Maglich (2003). “On-Line Intrusion Detection and Attack Prevention Using Diversity, Generate-and-Test, and Generalization”. In Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS’03), Track -9, Volume 9.