International Journal of Current Engineering and Technology
E-ISSN 2277 – 4106, P-ISSN 2347 - 5161

Available at http://inpressco.com/category/ijcet

Research Article

Fault-Tolerance and Transformation Analysis in Self-Tuning Servers

A. S. Alvi[Á] and Omprakash A. Jaisinghani[B*]

[Á]Computer Science, SGBAU Amravati, Amravati Maharashtra, India
[B]Information Technology, SGBAU Amravati, Rampuri camp Amravati, Maharashtra, India

Accepted 10 May 2014, Available online 01 June2014, **Vol.4, No.3 (June 2014)**

Abstract

We presenting a framework for intrusion tolerance in cloud computing environment which review how a number of suspicion and security techniques, and to improve it up to the framework serves as an excellent platform for making cloud services intrusion tolerant. tolerant server's to offer full functionality of self-healing which provides user self-governing and involuntary healing functionality, now here we are recommend new sketch for an intrusion-tolerant oriented self-healing server which move ahead performance quality of server. For secure data storage, 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. It provides hidden intrusion, and the vulnerable prerequisite of intrusion tolerance, and further enhances the consistency and survivability of the intrusion-tolerant server.

Keywords: *Malicious, Transformation, Intrusion, Self-tuning.*

1. Introduction

We present the intrusion tolerance with self transformation giving their definition, brief description, architectures etc. A responsible system is defined as one that is able to transport a service that can excusably be trusted. Attributes of reliability include availability (readiness for correct service), reliability (continuity of correct service), confidentiality (prevention of unauthorized disclosure of information), and integrity (the absence of improper system state alterations).

An intrusion is defined as malicious, superficially induced, operational fault. The word intrusion comes from the Latin and current usage covers both senses of illegitimate penetration. Self -Tuning is methodology to design an intrusion-tolerant oriented server in which initiate a self-healing functional components, including the discovery components and implementation components (XF. Zhang and F. Zheng eng, 2004).

However, if the intrusion is nonstop and unbeaten, it can wipe out the prerequisite for intrusion tolerance system. In addition, if the intrusion or attack does not be detected and tackle, then hidden intrusion occurred. The system turns into a state of impulsive dangers.

Moving serious applications and sensitive stuff to a public and shared cloud environment is a major task for corporations which are moving ahead of their data centre's network perimeter defense due to various security issues in cloud environment. Intrusion detection along with Tolerance in cloud is a fault tolerant design approach against malicious attacks. Under above assumptions, intrusion tolerance ensures that the overall system always remains functional and secure. Intrusion tolerance involves prevention, detection, removal, forecasting of attacks (Adelsbach et al. Conceptual model and architecture of Maftia 2003).

Intrusion Tolerance in cloud computing is very significant and vital tasking a fault tolerant design approach which protect cloud infrastructure is against malicious attacks. This paper aims at designing and realizing a framework for intrusion tolerance in cloud computing environment; testing the feasibility of the proposed framework against cloud environment and finally using the proposed framework for securing cloud applications and its services (D. Powell, R. Stroud 2003).

2. Aspects of System Architecture

In transformation analysis server design is projected in this paper. It combined self-healing software technology along with the 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 self tuning design approach to guard cloud infrastructure against malicious attack (Adelsbach et al. Conceptual model and architecture of MAFTIA 2003).

2.1 Intrusion analysis

Intrusion Tolerance muddle up of self-tuning with self healing which recovers the data which is affected by the intrusion. A justifiably be trusted system is defined as one

*Corresponding author: **Omprakash A. Jaisinghani**

that includes availability (readiness for accurate service), integrity (provide indistinguishable and approved data), confidentiality (anticipation of unauthorized leak of information) and integrity (the deficiency of improper system state variation).

Fault diagnosis is 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 review the degree of success of the intruder in terms of system corruption.

b) Vulnerability diagnosis, this will detect vulnerability in the channels through which the intrusion penetrated so that corrective maintenance can be carried out.

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

2.2 Intrusion segregation

In Cloud Computing situation intrusion segmentation is needed to make sure that the source of the sensing error is prevented from producing further error.

In terms of intrusions, this might involve:

a) Removing an infected file from the system or, with reference to the root vulnerability/attack causes.

b) Uninstalling software with newly-found vulnerabilities

We added self-tuning capabilities into the intrusion-tolerant server. It tackles present intrusion tolerant servers, such as the hidden intrusion and the vulnerable prerequisite of intrusion tolerance, and enhances the reliability of the intrusion-tolerant application server (Adelsbach et al. Conceptual model and architecture of Maftia 2003).

3. Secured Data Communication Channel

Based on IDC survey (International Data Corporation) the security and vulnerability market should exceed revenue of $4.4 billion by the end of 2013, with a climbing annual growth rate resulting in a compound annual growth rate (CAGR) of 10.8%. This study shows that products that reduce within the security and vulnerability management market will remain in high demand.

There are no flawlessly 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. An actually secure channel will not be essential 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 media. The most important advantage of public-key cryptography is improved security and convenience.

The 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 (Biswanath Mukherjee, L. Todd Heberlein, Karl N. Levitt 1994). Functionality provided by self motivating system architecture in favor of securities is 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

Sender can use public key encryption technique 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 sign a document. It is an acknowledgement of the message and thus, the sender cannot deny it (Meng Qiang, Zhou Rui-peng, Yang Xiao 2010).

To secure data from unofficial access file sent from client to server should be encrypted form and security will be high. It should be encrypted using the RSA algorithm so that contents of files cannot be decipherable by authenticated user. The best way of RSA algorithm is Asymmetric encryption, which permit Alice to send Bob an encrypted text message without a shared secret key; there are two keys are present a secret key (private key), but only Bob knows that key, and he does not share it with any person or to 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 gives a general idea of asymmetric encryption, working of an asymmetric encryption as follows, which is consisting of single sender and single receiver end:

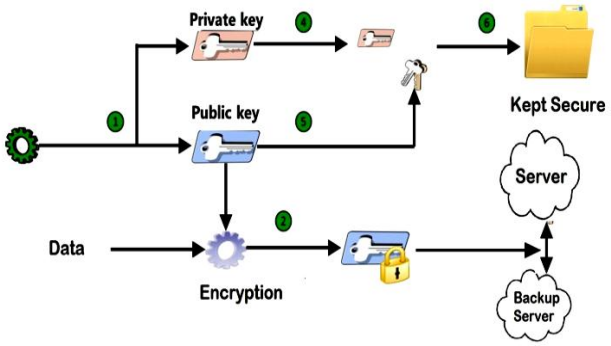

Fig. 1 Data encryption and server storage

4. Proposed System Design

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

To maintain confidentiality of data there will be provision for encryption of data using cryptography tool. There are two different encryption tools are available

symmetric and asymmetric. Using that tools data can be encrypted and using the combination of both tool security levels can be improved which will tolerate intrusion to get an original text from encrypted text (Adelsbach et al. Conceptual model and architecture of MAFTIA 2003).

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 integrity for data which is in servers. Performance analysis shows that the proposed scheme is highly efficient and resilient against malicious data modification attack (Meng Qiang, Zhou Rui-peng, Yang Xiao 2010).

As there is no perfectly secure system is available in practical so there is a possibility for the attack on the data that can be detected by using stored MD-5 which is generated at the initial time while data is stored on the server and current file's MD-5 if both MD-5 are match with each other then file is detected as intact file otherwise it is treated as intrusion attack is detected. Then self healing mechanism is applied on this file. Self healing mechanism is consisting of the mirroring of the data at an initial time while file is uploaded on the backup server. Now at this time when file is infected on primary server then back – up server will replace that file with integral initial file using this file is healed.

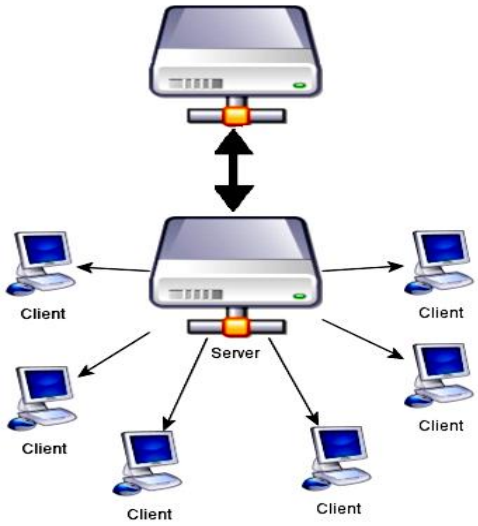

Fig. 2 Structural design

4.1 Facts about Implementation

Here 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 hit on data.

For secure data storage, we have also 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 of the proposed intrusion tolerant data storage is measured in terms of evaluating checksum value for every data storage on each datacenter of the cloud and will display these value for each datacenter individually. Performance analysis shows that the proposed scheme is highly efficient and resilient against malicious data modification attack (Popovic Kresimir, Hocenski, Zeljko 2010).

Authentication is operation of verify the truth of an entity or genuine user. This might involve confirming the identity of a software program or person. Here we will provide user id and password for validation of legitimate user. Confidentiality is a set of rules that limits access or places limitations on certain types of information. To maintain confidentiality of data there will be provision for encryption of data using cryptography tool (F. Cristian and C. Fetzer1999).

Implement components include

Checkpoint actuator, if the application server process by setting checks points, will have a greater overhead, Therefore, In this application server component level on checkpoint set, preservation and restoration application server running the correct state of the business component for the transport and restart the execution 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 (Popovic Kresimir, Hocenski, Zeljko (2010)).

4.2 Design of Self Tuning:

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 (Meng Qiang, Zhou Rui-peng, Yang Xiao 2010).

Hash functions map strings of different lengths to short fixed size results. 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.

Conclusions

Here, we designed 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. We will use Intrusion Tolerance via threshold cryptography mechanism for validation. Different level of cryptography can be achieved using different type of encryption. Encryption using symmetric along with asymmetric key encryption will improve tolerance of data confidentiality issues. Our framework had given successful implementation of the different securities for protecting digital stuff which is present in the server. Securities are like integrity, confidentiality, authenticity, availability and self healing.

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.

Also, Performance of the proposed intrusion tolerant data storage will measured in terms of evaluating checksum value for every data storage on other datacenter of the cloud and will display these value for each datacenter individually. Performance analysis shows that the proposed scheme is highly efficient and resilient against malicious data modification attack.

References

XF. Zhang and F. Zheng eng (2004) Intrusion Tolerance Technology-Survey and Direction, Information Security, (31):19-22.

Adelsbach et al. Conceptual model and architecture of Maftia (2003) Tech- nical Report DI/FCUL TR-03-01, Dept.Comp. Sci. Univ. Lisbon, February.

Biswanath Mukherjee, L. Todd Heberlein, Karl N. Levitt (1994) Network Intrusion Detection, IEEE Network, pp.26-41, May/June.

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

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.

Popovic Kresimir, Hocenski, Zeljko (2010) Cloud computing security issues and challenges. In proceedings of the 33rd International Convention, IEEE Transactions.

F. Cristian and C. Fetzer.(1999) the timed asynchronous distributed system model. IEEE Transactions on Parallel and Distributed Systems, 10(6):642–657, June.