IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 12, December 2015

Intrusion Detection, Prevention and Self-Healing using Mirroring on Server

Priya N Jethani[1], Prof. Ankush S Narkhede[2]

Student, Department of Computer Engineering, Pdm Dr.V.B.Kolte College of Engineering, Malkapur, India [1]

Professor, Department of Computer Engineering, Pdm Dr.V.B.Kolte College of Engineering, Malkapur, India [2]

Abstract: Current systems are prone to attacks caused by intruders due to increased connectivity (especially on the Internet). So there is need to handle these attacks and provide efficient solution to these attacks. Our project provides us the technique to tackle the problem caused by intruder and helps in achieving the few basic objectives of intrusion detection systems i.e Confidentiality, Integrity, Availability, and Accountability. It uses RSA algorithm of cryptography to prevent the data or we can also use the symmetric and asymmetric cryptography techniques to provide more security on data and MD5 to detect intrusion If we have detect any intrusion in our system, we can recover our data through mirroring technique with the help data present on our backup server.

Keywords: RSA algorithm, Mirroring, Intrusion detection system, cryptography,MD5.

I. INTRODUCTION

We present the intrusion prevention ,detection and self healing mechanism giving their definition, brief description, etc. A responsible system is defined as one that is able to execute a service properly and this will be trusted only. 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).

Intrusion Prevention System is an advance combination of IDS, personal firewalls and anti-viruses etc. The primary aim of an Intrusion Prevention System (IPS) is to detect an attack that is trying to interrupt and then to stop it by responding automatically such as logging off the user, shutting down the system or stopping the process and disabling the connection. The intrusion detection and self-recovery systems is a dynamic monitoring and protecting system which is used to identify and monitor unsafe activities, even new attacks. Like IDS, IPS can be divided into three types, i.e. Host-Based Intrusion Prevention Systems (HIDS), Network-Based Intrusion Prevention Systems (NIDS), Distributed Intrusion Prevention Systems (DIDS)[1].

Intrusion detection can be identify as individuals who use a computer system without authorization and those who have legitimate access to the system but misuse their privileges (i.e. insider threat)[2].Intrusion detection systems (IDSs) are usually combined with preventive security mechanisms, such as authentication and other controls. It is use as a second line of defense that protects data on systems. There are so many reasons that make intrusion detection an important part of the entire system. First, many systems and applications were developed without considering security techniques. In another cases, systems and applications were developed to work in an environment where they become vulnerable when deployed. Intrusion detection is different from these protective mechanisms that helps in improving the system security. However, even if the preventive security mechanisms can protect the information systems successfully, it is still require to know what intrusions have occured or are ocurring, so that we can understand the security threats with ease and be better prepared for future attacks. The intrusion detection and self-recovery systems in our implementation is a dynamic monitoring and protecting system which is used to identify and monitor unsafe activities, even new attacks.

II. LITERATURE REVIEW

A.RSA Algorithm

In this project, Before file is sent from client to datacenter(s), it is encrypted using RSA algorithm so that contents of file cannot be modified by unauthenticated person. Encrypted data is called as Ciphered data. This data is again decrypted at cloud co-ordinator using RSA algorithm. Working of RSA algorithm and example is given below.

KEY GENERATION

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key[10].

The keys for the RSA algorithm are generated the following way:

1) Choose two distinct prime numbers p and q.
For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.
2) Compute n = pq.
n is used as the modulus for both the public and private keys
3) Compute $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient function.

DOI 10.17148/IJARCCE.2015.41286

4) Choose an integer e such that $1 < e < \varphi(n)$ and greatest common divisor of (e, φ(n)) = 1; i.e., e and φ(n) are coprime.
e is released as the public key exponent.
e having a short bit-length and small Hamming weight results in more efficient encryption - most commonly 0x10001 = 65,537. However, small values of e (such as 3) have been shown to be less secure in some settings.[4]
5) Determine d as:

$$d \equiv e^{-1} \pmod{\varphi(n)}$$

i.e., d is the multiplicative inverse of e mod φ(n).
This is more clearly stated as solve for d given (de) mod φ(n) = 1 .This is often computed using the extended Euclidean algorithm.
d is kept as the private key exponent.
The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d which must be kept secret. (p, q, and φ(n) must also be kept secret because they can be used to calculate d.)

ENCRYPTION
Alice transmits her public key (n,e) to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.
He first turns M into an integer m, such that 0<m<n by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext c corresponding to

$$c=m^e \bmod (n)$$

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

DECRYPTION
Alice can recover m from c by using her private key exponent d via computing

$$m=cd \bmod (n).$$

Given m, she can recover the original message M by reversing the padding scheme.
Example of RSA algorithm
1. Choose two distinct prime numbers, such as
p=61 and q=53.
2. Compute n=pq giving
n = 61 × 53 = 3,233.
3. Compute the totient of the product as φ (n)=(p-1)(q-1) giving
Φ (3233)=(61-1)(53-1)=3,120
4. Choose any number 1<e<3,120 that is coprime to 3,120. Choosing a prime number for e leaves us only to check that e is not a divisor of 3120.
Let e=17.
5. Compute d, the modular multiplicative inverse of e(mod φ(n)) yielding
D=2753.

The public key is (n=3222, e=17). For a padded plaintext message m, the encryption function is m^{17}(mod 3,233) .

The private key is (n=3,233, d=2,753). For an encrypted ciphertext $\mathbb{C}$, the decryption function is $c^{2,753}$(mod 3,233) .

For instance, in order to encrypt m=65, we calculate

$$c=65^{17}(\text{mod } 3{,}233)=2{,}790$$

To decrypt c=2,790, we calculate,

$$m=2{,}790^{2{,}753} (\text{mod } 3{,}233)=65$$

B .Checksumming
Checksumming is a well known method for performing integrity checks. Checksums can be computed for disk data and can be stored persistently. Data integrity can be verified by comparing the stored and the newly computed values on every data read. Checksums are generated using a hash function. The use of cryptographic hash functions has become a standard in Internet applications and protocols. Cryptographic hash functions map strings of different lengths to short fixed size results. These functions are generally designed to be collision resistant, which means that finding two strings that have the same hash result should be infeasible. In addition to basic collision resistance, functions like MD5 and SHA1 also have some properties like randomness. In this project, MD5 hashing algorithm is used to generate Checksum. Working of MD5 is shown below.

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit little endian integers); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with a 64-bit big endian integer representing the length of the original message, in bits, modulo 2^{64}. The bytes in each 32-bit block are big endian, but the 32-bit blocks are arranged in little endian format.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted *A*, *B*, *C* and *D*. These are initialized to certain fixed constants. The main algorithm then operates on each 512-bit message block in turn, each block modifying the state. The processing of a message block consists of four similar stages, termed *rounds*; each round is composed of 16 similar operations based on a non-linear function *F*, modular addition, and left rotation[11].

C.MIRRORING
It is recognized that disks are an inherently unreliable component of computer systems. Mirroring is a technique to allow a system to automatically maintain multiple copies of data so that in the event of a disk hardware failure a system can continue to process or quickly recover data. Mirroring may be done locally where it is specifically to cater for disk unreliability, or it may be done remotely where it forms part of a more sophisticated disaster 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.

DOI 10.17148/IJARCCE.2015.41286

III. PROJECT METHOLOGY

In the working of our project , we present the framework for intrusion prevention, intrusion detection and self healing using mirroring on server. We have added features like uploading the files, sending the files to others, preventing the files from intrusion ,intrusion detection if it has occurred in our files, self heals the files if infected by intrusion. The primary concentration is on the data security from intrusion which alters our files we have uploaded on the server.

To prevent the data from intrusion attacks, there will be provision for foe encryption of data using cryptographic techniques. We have used the two different tools for encryption of data-symmetric and asymmetric . Using these tools data can be encrypted and then stored to prevent from outsiders attack. We can also apply the combination of both tools that is symmetric and asymmetric for proving more security to our data .security level can be improved through applying both tools to our data.

For detecting intrusion in our files that is for integrity checks we apply the checksum MD5 algorithm while storing the file or while accessing the file. As there is the possibility of the attack on our data that can be detected by using stored MD5. When we store the data at that time ,we apply the MD5 on our data and stored it on the server . for detecting the intrusion we again apply the MD5 on the data and then compare it with the stored MD5 if both MD5 match then we can say no intrusion is present in our file .If in case,both MD5 doesn't match ,we say that file is infected and intrusion is present in our file.

After detecting the intrusion the present in our files we apply the self-healing mechanism.self healing mechanism consist of mirroring of the data when the file is uploaded on the server initially.we store the duplicate copy from the server to the backup server through mirroring .If we find our file is infected on primary server then backup server will replace the file and we get the original file as it is.

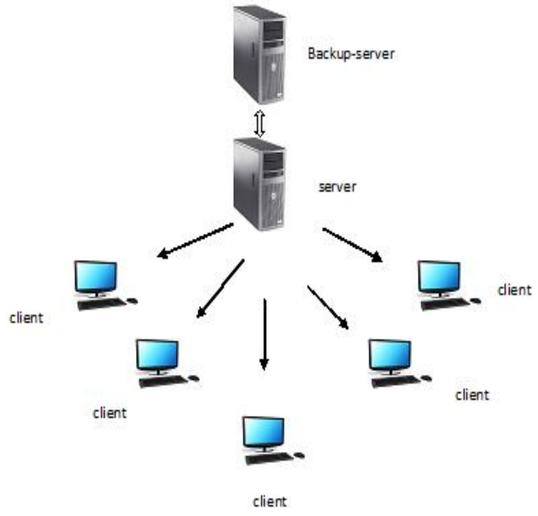

Fig.1 Structural design

A. Implementation Facts

We present here a framework for intrusion prevention in data server which helps in achieving basic principles of intrusion detection systems that are availability, integrity, authentication and confidentiality .

To check the integrity of data, we have used Checksum MD-5 code while storing and retriving the file. Checksum MD-5 code is used to provide intrusion prevention for data servers . Performance shows that the proposed scheme is efficient at good level against malicious data modification hit on data.

For security of data , we have used here Checksum MD-5 code while storing and accessing the file. Checksum MD-5 code is used to provide intrusion prevention for data servers. Performance of the proposed intrusion prevention data storage is measured in terms of evaluating checksum value for every data storage on datacenter and will display these value for datacenter individually[8].

Authentication is operation to verify whether the user is genuine or not. This involve confirming the identity of a software program or user. Here, for verification we use id and password given to legitimate user. Confidentiality is a set of rules that limits access or places limitations on certain types of information. To ensure the confidentiality of data there will be encryption of data using cryptography tool .

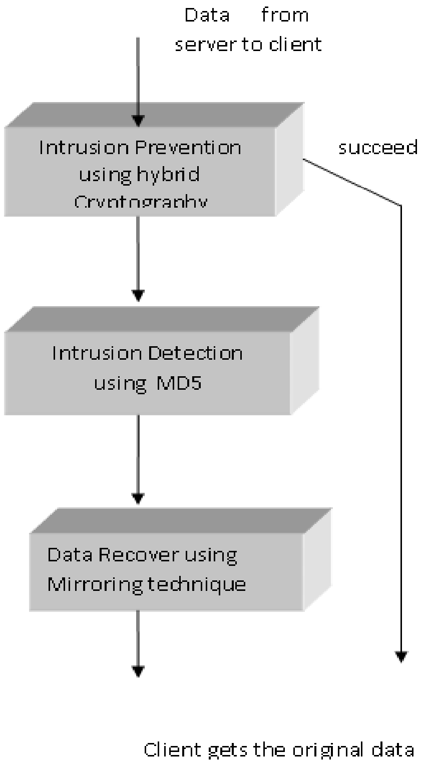
Data from server to client
Intrusion Prevention using hybrid Cryptography
succeed
Intrusion Detection using MD5
Data Recover using Mirroring technique
Client gets the original data

Fig.2 Workflow of the actual project

IV. CONCLUSION

Here, we have designed a framework for intrusion prevention based on the layered design of computing architecture. For the validation of framework, we will simulate Intrusion Prevention environment with security controls and techniques required for intrusion prevention. We will use Intrusion Prevention via threshold cryptography mechanism for validation. Different level of cryptography can be achieved using different type of

DOI 10.17148/IJARCCE.2015.41286

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 prevention data storage will measured by evaluating checksum value for every data storage on other datacenter of server and will display these value for each datacenter individually. Performance of this workflow shows that the proposed scheme is highly efficient against malicious data modification attack.

REFERENCES

[1] Ting SUN, XingchuanLIU(2013).” Agent–based Intrusion Detection and self–recovery system for wireless sensor networks” Proceeding of IC BNMT2013,IEEE,2013

[2] Prachi Jain, Pramod Kumar Singh, Prachi Jain, Pramod Kumar Singh”Intrusion Detection and Self Healing Model for Network Security” *2011 7th International Conference on Next Generation Web Services Practices,IEEE2011*

[3] Saidane, A., Nicomette, V., Deswarte, Y.: The Design of a Generic Intrusion Tolerant Architecture for Web Servers. IEEE Trans. 6, 45–58 (2009)

[4] Cuppen, F. &Miege, A. (2002). Alert Correlation in a Cooperative Intrusion Detection Framewok. In Proceeding of the 2002 IEEE Symposium on Security and Privacy. IEEE, 2002]

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

[6] 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.

[7] “Handbook of mathematics”,Thierry Vialarr

[8] “Ensuring Data Integrity in Storage: Techniques and Applications.”Gopalan Sivathanu,Charles P. Wright,and Erez Zado k Stony Brook University

Copyright to IJARCCE

DOI 10.17148/IJARCCE.2015.41286