# A Review on Intrusion Prevention and Self-healing of Server side

**Priya N Jethani, Ankush S. Narkhede**

*Abstract* -**This paper describes a approach for the network security on server based on the combination of intrusion prevention, tolerance, detection and self-healing concepts. This Intrusion Detection System detects and analyzes the malicious activities in the network and then self-heals the system using mirroring. It means that damage that is detected by malicious activities is itself used to start the self - healing mechanism and helps in recovering the original data. It is an automated system that enhances the fault repair and system recovery. In this system we present a framework for intrusion tolerance, detection and self healing in server which is combination of integrity, authentication, availability, accountability and confidentiality.**

*Index Terms*--**Intrusion prevention, MD5, Intrusion detection, self-healing, network security.**

## I. INTRODUCTION

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. Like IDS, IPS can be divided into three types, i.e. Host-Based Intrusion Prevention Systems (HIPS); Network-Based Intrusion Prevention Systems (NIPS) and Distributed Intrusion Prevention Systems (DIPS)[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 along with other preventive security mechanisms, such as authentication and other controls. It is use as a second line of defense that protects information 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.INTRUSION TOLERANCE*

A dependable system is defined as one that is able to deliver a service that can justifiably be trusted. Attributes of dependability 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) [8]. We assume that,

1) The system remains to certain extent vulnerable.
2) Attacks on components and subsystems will happen and some of them will be successful.

Under above assumptions, intrusion tolerance ensures that the overall system always remains functional and secure. Intrusion tolerance involves prevention, detection, removal, forecasting of attacks.

Intrusion tolerance is not a new concept. The Delta-4 Project [10] was one of the earliest works on intrusion tolerance, in contrast with the traditional security approaches aiming at avoiding intrusions. It proposed an original approach for security in open distributed systems. One of the interesting features of the Delta-4 architecture is the fragmentation scattering technique that has been applied to file storage, security management, and data processing. For file storage, fragmentation is carried out using simple cryptographic techniques, and fragment naming employs a secret key one-way function. The fragments are sent over the network in a random order, which means that one of the hardest tasks for an intruder would be to sort all the fragments into the right order before being able to carry out cryptanalysis. For security management, the principle resides in the distribution of the authentication and authorization functions between a set of sites administered by different people so that failure of a few sites or abuse of privilege by a small number of administrators do not endanger the security functions. On these sites, non-sensitive data are replicated, whereas secret data are



ISSN: 2278 – 1323 

fragmented using threshold cryptographic functions. Finally, for data processing, two data kinds are considered:

1) Numerical and logical data, whose semantics are defined by the application, and
2) Contextual data (for example, character strings) that is subjected only to simple operations (input, display, concatenation, etc.).

In this scheme, contextual data is ciphered and deciphered only on a user site during input and display. In contrast, context data is subjected to successively finer fragmentation until the fragments do not contain any significant information. This is achieved using an object-oriented decomposition method.

Even if this intrusion tolerance notion is not a new concept, real research programs dedicated to this topic are relatively recent. Examples of such research programs are the *Defense Advanced Research Projects Agency* - Organically Assured and Survivable Information Systems (DARPA-OASIS) and OASIS Demonstration/Validation Programs (J. H. Lala et al. 2003). OASIS "seeks to provide defense capabilities against sophisticated adversaries to allow sustained operation of mission critical functions in the face of known and future cyber attack information systems". It focuses on different aspects of intrusion tolerance such as intrusion-tolerant storage, group communications, mobile code security, and intrusion-tolerant servers.

Partha Pal [12] presents analysis of potential impact of intrusion tolerance on SOA, cloud and semantic web technologies on intrusion tolerance. A number of defenses and security techniques, especially those providing availability, integrity and confidentiality, can possibly be encapsulated in the cloud or within the services, and offered as value-add; but new capabilities (e.g., the ability to dynamically manage the security aspects of Service Oriented Architecture (SOA) services and cloud resources or support for privacy preserving interaction) will also be needed.

### *B.Intrusion Tolerance Techniques*

A system can be made intrusion tolerant using following techniques [13],

1) Diversity
   - Space Redundancy
     - Several copies of the same component
     - Same information stored in several disks
     - Different nodes compute same result in parallel
     - Messages disseminated along different paths
   - Time Redundancy
     - Several copies of the same component
     - Same information stored in several disks
     - Different nodes compute same result in parallel
     - Messages disseminated along different paths
2) Generate and Test
   - Value redundancy
     - Adding extra information about the data being stored or sent
     - Codes that allow the detection or correction of integrity errors
     - Parity bit or ECC added to memory chips or disk structures
     - Cryptographic message signatures

### *C.FRAMEWORKS FOR INTRUSION TOLERANCE*

The Malicious and Accidental Fault Tolerance for Internet Applications (MAFTIA) Project, funded by the European Union, systematically investigated the "tolerance paradigm" for security in order to propose an integrated architecture built on this paradigm and to realize a concrete design that can be used to support the dependability of many applications [9]. MAFTIA was the first project that uniformly applied the tolerance paradigm to the dependability of complete large-scale applications in a hostile environment and not just for single components of such systems. Its major innovation was a comprehensive approach for tolerating both accidental faults and malicious attacks in large-scale distributed systems, including attacks by external hackers and by corrupt insiders. The framework proposed is strongly inspired by the MAFTIA framework, but we applied it to an emerging cloud computing environment.

A Component Based Framework for Intrusion Tolerance (CoBFIT) provides a platform for building and testing a variety of Intrusion tolerant distributed systems [11].The CoBFIT framework, by virtue of its design and implementation principles, can serve as a convenient base for building components that implement intrusion-tolerant protocols and for combining these components in an efficient manner to provide a number of services for dependability. The design and implementation principles of the CoBFIT framework stress characteristics that are essential for dependability in the face of attacks. These characteristics include portability, reconfigurability, flexibility, and adaptability. The framework components were designed using well-scrutinized software patterns with the goal of providing a framework for building intrusion-tolerant services that is itself robust. Our project involves refining and extending the prototype implementation of the CoBFIT framework components, and exploring additional supporting software mechanisms for intrusion tolerance that can be added to the proposed framework.

### *D. Intrusion Detection Techniques*

An intrusion detection system can be divided into two approaches which are behavior based (anomaly) and knowledge based (misuse) . The behavior based approach is also known as anomaly based system while knowledge based approach is known as misuse based system .The misuse or signature based IDS is a system which contains a number of attack description or signature that are matched against a stream of audit data looking for evidence of modeled attack. The audit data can be gathered from network traffic or an application log. This method can be used to detect previous known attack and the profile of the attacker has to be manually revised when new attack types are discovered. Hence, unknown attacks in network intrusion pattern and characteristic might not be capture using this technique .Meanwhile, the anomaly based system identifies the intrusion by identifying traffic or application which is presumed to be normal activity on the network or host . The anomaly based system builds a model of the normal behavior of the system and then looks for anomalous


ISSN: 2278 – 1323 

activity such as activities that do not confirm to the established model. Anything that does not correspond to the system profile is flagged as intrusive. False alarms generated by both systems are major concern and it is identified as a key issues and the cause of delay to further implementation of reactive intrusion detection system .

Therefore, it is important to reduce the false alarm generated by both of the system. Although false alarm is a major concern in developing the intrusion detection system especially the anomaly based intrusion detection system, yet the system has fully met the organizations' objective compared to the signature based system . The false positive generated by the anomaly based system is still tolerable even though expected behaviour is identified as anomalous while false negative is intolerable because they allow attack to go undetected An attack that uses a large amount of packet or connection within a few second scanning attack, DOS attack, DOS attack worm attack are some of fast attack Code Red Worm and NIMDA worm are another breed of DoS attacks on Internet infrastructure after the Morris Worm. Code Red Worm has a fast rate of propagation and infection via network scanning to detect and automatically exploit.

## III. FUTURE WORK

In this working scenario primary concentration is on the data security against the intrusion which performs alteration or deletion of particular data and also prevents data file deletion from server. The detection and prevention of changes can be done on regular basis by our server. If the server detect any intrusion in the file which cases change in the file or deletes whole file then server will uses the self healing concept which recover original file.

To detect the intrusion in data we check its integrity through check summing and MD5 .To perform integrity check we use Check summing method. Checksums can be computed for data accessed by user and can be stored persistently. Data integrity can be checked by comparing the stored and the newly computed values on every data read by user. Checksums are generated using a hash function. The cryptographic hash functions that are used nowadays has become a standard in Internet applications. Cryptographic hash functions map the different lengths strings to short fixed size results. These hash functions are basically designed to be collision resistant that means that finding two strings that have the same hash result should be infeasible. Along with the basic collision resistance, functions like MD5 and SHA1 also have some properties like randomness. In this system, MD5 hashing algorithm is used to generate Checksum. Working of MD5 is given below.

MD5 process on a variable-length message into a fixed-length output of 128 bits. The input message is first 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 values. 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.

Now there is need to compute checksum of hashed value obtained from MD5 algorithm. This is created by following code.

```
for (int i = 0; i < b.length; i++)
{
        result += Integer.toString((b[i] &
0xff) + 0x100, 16).substring(1);
}
return result;
```

Checksum obtained from above code is then saved in file. Every time data is accessed, checksum is again created and matched with saved checksum in this file. If it is matched then it indicates that data is not corrupted and can be used. If it is not matched then it shows that data is corrupted and the data can be obtained from other datacenters. For convenience, datacenters are numbered form 1 to n and first data integrity of datacenter1 is verified. If data is found not corrupted then this data is returned to user. If data is found corrupted then data from datacenter2 is accessed.

Intrusion-Tolerant & Self-healing Server is mechanism very crucial and very important part of the server security to prevent from any undesirable events which can affect the user data which is stored on the server. If it occurs then self healing mechanism can be recover the complete original data.

## IV. CONCLUSION

In this paper we had proposed model for intrusion prevention. For the validation , we will simulate Intrusion prevention environment with security controls and techniques required for intrusion tolerance and along with recovery module which recovers any infected data.

We will use Intrusion Prevention via Threshold Cryptography mechanism for validation which increases functionality of server. This model 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.

## REFERENCES

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



ISSN: 2278 – 1323 

[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] Cabrera, J.B.D., Ravichandran, B &Mehra R.K. (2000). Statistical Traffic Modelling for Network Intrusion Detection. In Proceeding of the IEEE Conference.
[6] Yeophantong, T, Pakdeepinit, P., Moemeng, P &Daengdej, J. (2005).Network Traffic Classification Using Dynamic State Classifier. In Proceeding of IEEE Conference
[7] Farah J., Mantaceur Z. & Mohamed BA. (2007). A Framework for an Adaptive Intrusion Detection System using Bayesion Network. Proceeding of the Intelligence and Security Informatics, IEEE, 2007.
[8] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr (2004). "Basic Concepts and Taxonomy of Dependable and Secure Computing" *In proceedings IEEE Transactions on Dependable and Secure Computing. Volume 1 Issue 1.* (pp. 11-33)
[9] 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.
[10] D. Powell, G. Bonn, D. Seaton, P. Verssimo, and F. Waeselynck (1988). "The Delta-4 Approach to Dependability in Open Distributed Computing Systems". *In proceedings of the 18th IEEE Int'l Symp. Fault-Tolerant Computing Systems*. (pp. 46-51)
[11] 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)
[12] 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.
[13] 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)

**Priya N Jethani**
She is pursuing M.E. in computer engineering from Padm. V.B.Kolte college of engineering,Malkapur.

**Ankush S.Narkhede**
He has done M.Tech in computer network. Also he had research experience of 3 years. He have published 7 international papers in various recognized journals.


ISSN: 2278 – 1323 