International Journal of Advanced Technology & Engineering Research (IJATER)
www.ijater.com

FILE MONITORING ON CLOUD USING SECURE TECHNIQUE

Kalyan Bamane and Ashish Saxena
Asst Prof. Dept of IT, DYPCOE, Pune

Abstract

Now a day's providing security to the files stored on cloud and to cloud resources is a recent area of research in computer science. Cloud resources include mainly two types of files, system specific files and user specific files. Tampering with the content of these files can lead to compromise in the security of Cloud. The designed tool "Secure file monitoring on cloud" helps in maintaining the integrity of the system specific files by recalculating the hash and stores the hash value within the file hence, reducing the cost of the tool. This tool is platform independent as it does not require external database to store the hash values. The tool has a file integrity monitoring module which is periodic in nature and an alert generation module which alerts the admin whenever a file is modified by unauthorized user. Alerts can be in the form of email or SMS to the admin. The main focus of the tool is to store the hash values in the file itself to avoid the need of external database for storing hash.

Keywords— Cloud, File integrity, Hash value, Integrity Establishment, Integrity Monitoring

I. INTRODUCTION

Cloud computing is mostly used in large as well as in small organizations for the purpose of data storage and it provides its resources on a pay per usage basis. Since, organizations are most likely accepting Cloud computing for fulfilling their purpose; it is very important to maintain the security of cloud and provide means to maintain the integrity of files. On Cloud, security needs to be provided to the individual hosts, networking infrastructure and to files and important data.

As Cloud services are remotely used facilities and are used by various users from different domains, it becomes a requirement to look at its security related aspects. Important files/resources stored on cloud require security from unauthorized access. If these file content get changed, they will affect the operation of cloud services and can result in breaches in cloud security.

After analyzing the above security aspects we understood that providing security to the files and important data is more of an importance as tampering with the file contents may cause the whole system to fail. Keeping in mind the security aspect of Cloud files we are proposing the tool named "Secure file monitoring on Cloud". The paper explains the process of developing the tool and making it a low cost, platform independent tool.

II. LITERATURE SURVEY

Important files/resources stored on cloud require security from unauthorized access. Now, it is very important to look after security related aspects of cloud, because cloud services are used by various users from various domains. These aspects generally divided into three categories as follows:

1. Security of individual hosts
2. Security of networking infrastructure
3. Security of important files and data.

For securing important files and data on cloud environment, integrity checking becomes imperative. If these file contents get changed on cloud, it will affect the operations of cloud environment. So, to prevent from the risks arising in cloud environment, researchers have proposed many solutions including:

Table I: Existing systems

Technique	Description
1. XenFTT	Records the system call log, and sends it to the privileged VM**.**
2. Flogger	A tool for the end user to check if their files have been tampered.
3. Tripwire	A Host based IDS that alerts on macro changes to the files and folders.
4. Storage based IDS	Allows the storage systems to watch for data modification.

III. PROPOSED WORK

Our proposed tool for monitoring files on cloud has the following characteristics:

1. The integrity establishment is one time (until the file has no authorized modifications or unwanted subversions) while monitoring is periodic in nature. In the Integrity Establishment module the encrypted hash value of the contents of the file is calculated which adds integrity to the files. The encrypted hash value is stored in the file itself well protected by pre-defined tags. This operation is performed over all the files that are configured for integrity establishment.
2. File is retrieved from a remote location to generate the hash value (checksum) of the contents of the file. The encrypted hash value is stored in the file itself well protected by pre-defined tags.
3. Monitoring the file integrity includes again calculating the checksum of the original contents of the file and comparing it with the checksum extracted from the file between the predefined tags. If they are unequal an alert is generated by the module to replace the file with its latest verified replica and new integrity established for the file. This operation can also be done automatically. This Integrity Monitoring process is periodic and runs over files in a sequential manner. The integrity monitoring can be set as one time per day or one time per week based on the needs of a specific configuration or execution environment and is controlled by admin.
4. There is response and alert generation module which informs the user of the file whenever the content of the files is modified by unauthorized user. In such cases it replaces the file with its latest replica.

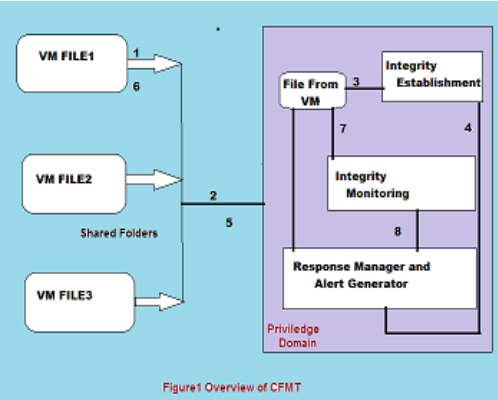

Figure I: Block Diagram of SFMT

The working of the tool can be explained in steps which are as follows:

1. The first step is to locate the files on the Cloud.
2. The files are then accessed by the CFMT (Cloud File Monitoring Tool) Domain.
3. Integrity is established over all the accessed files i.e. the encrypted hash codes are generated for the files and appended into file itself in between well-defined tags.
4. Integrity monitoring module checks for the integrity of the files on the cloud. For each file, the hash is recalculated and checked for equality with the hash value stored inside the predefined tags. If the values are found to be unequal, it implies that the contents of the file are modified and hence information is given to Response and Alert Generator Module.
5. After getting result of Integrity Monitoring, Response Manager and Alert Generator module reports alerts pertaining to manipulation in files by sending emails and SMS to respective user.

IV. IMPLEMENTATION

The implementation details of the proposed tool are as follows:

1. The proposed tool is developed in Java on windows 64 bit platform.
2. The cryptographic checksum or the hash value is calculated using the MD5 algorithm.
3. For integrity establishment the files are first accessed from the remote location then fed as input to the MD5 algorithm where hash value is calculated and stored in the pre-defined tags. For example, <securevalue>hashvalue <securevalue> where <securevalue> is the start and end tag.
4. To provide more security, we have used two step verification process to authenticate the Cloud user. After login the user/admin has to enter the randomly generated password which is automatically generated and is send to the person's registered email-id.
5. After integrity establishment, integrity monitoring can be periodically carried out. In monitoring, the files are once again fed as input to MD5 for calculating hash. The newly calculated hash and the hash value already stored in the file are then checked for equality. If the values do not match, then it means that file content is modified by

unauthorized user and authorized user should be informed.

6. The alert and response generation module will then alert the user about the tampering of the file content. The alert will be sent in the form of email and SMS to the file owner.

A. Message Digest Algorithm (MD5)

Suppose a b-bit message as input, and that we need to find its message digest.

Step1: append padded bits .The message is padded so that its length is congruent to 448, modulo 512. Means extended to just 64 bits shy of being of 512 bits long. [A single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits equals 448 modulo 512.]

Step2: append length. A 64 bit representation of b is appended to the result of the previous step.[The resulting message has a length that is an exact multiple of 512 bits]

Step3: Initialize MD Buffer. A four-word buffer (A, B, C, D) is used to compute the message digest. Here each of A, B, C, D, is a 32 bit register. These registers are initialized to the following values in hexadecimal:

word A: 01 23 45 67
word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10

Step 4: Process message in 16-word blocks. Four auxiliary functions that take as input three 32-bit words and produce as output one 32-bit word.

F(X,Y,Z) = XY v not(X) Z
G(X,Y,Z) = XZ v Y not(Z)
H(X,Y,Z) = X xor Y xor Z
I(X,Y,Z) = Y xor (X v not(Z))

[If the bits of X, Y, and Z are independent and unbiased, the each bit of F(X,Y,Z), G(X,Y,Z),
H(X,Y,Z), and I(X,Y,Z) will be independent and unbiased]

Step 5: output

The message digest produced as output is A, B, C, D. That is, output begins with the low-order byte of A, and end with the high-order byte of D

V. CONCLUSION

The application is a file monitoring tool for system specific, configuration and user files stored on cloud. Application accesses files from remote location i.e. cloud and it establishes integrity in the form of hash value. Integrity monitoring is periodic, so if the file gets tampered, an alert is sent to the admin. Tampered files are replaced with a replica of the original file and integrity is re-established. Hash values are appended into files so that external database is not required, and this is main advantage of the application.

REFERENCES

[1] Sanchika Gupta, Anjali Sardana, and Padam Kumar. A light weight centralized file monitoring approach for securing files in cloud environment. In ICITST, pages 382 to 387, 2012.

[2] Ashutosh Kumar Dubey, Animesh Kumar Dubey, Mayank Namdev, and Shiv Shakti Shrivastava. "Cloud-user security based on RSA and MD5 algorithm for resource attestation and sharing in java environment". In IEEE Paper, 2012.

[3] Mr. Pritesh Jain, Prof. Vaishali Chourey, and Prof. Dheeraj Rane3. "An analysis of cloud model-based security for computing secure cloud bursting and aggregation in real environment ". In IEEE Paper, 2012.

[4] P.Varalakshmi and Hamsavardhini Deventhiran. "Integrity checking for cloud environment using encryption algorithm". In IEEE Paper, 2012.

[5] Zhidong Shen and Qiang Tong. The security of cloud computing system enabled by trusted computing technology. In Signal Processing Systems (ICSPS), 2010 2nd International Conference on, volume 2, 2010.

[6] W. Stallings and L.V. Brown. Computer Security: Principles and Practices. Always Learning. Pearson Education, Limited, 2012.

[7] Debdeep Mukhopadhyay. "*Cryptography and network security*".Tata McGraw Hill, New Delhi, 2nd edition, 2010.