ISSN 2395-1621

Fog Computing Mitigating Data Theft Attacks in Cloud

[#1]Ajinkya Dev, [#2]Tausif Firoj, [#3]Praveen Kumar, [#4]Shirshak Khandelwal,
[#5]Prof. Sunita Nandave

[1]ajinkya81194@gmail.com
[2]tausiffiroj@gmail.com
[3]pkumar45156@gmail.com
[4]shirshakkhandelwal@gmail.com

[#1234]Department of Computer Engineering
[#5]Prof. Department of Computer Engineering

G.H.Raisoni College of Engineering & Management,
Wagholi, Pune-412207

ABSTRACT

Cloud is basically a bunch of multiple appropriate servers attached within a network. It changes the approach to use computers and access and store our personal and enterprise information. It provides a manageable approach for accessing, managing and computation of end user information. Cloud computing gaining popularity in today worlds but cloud failed in preventing data theft attacks. To overwhelm this problem we familiarize a new technique which is called as Fog Computing. Fog Computing precept a single unit to concern with two different technologies i.e. User behavior Profiling and Decoy Information Technology. Fog computing improves the quality of services and reduces latency.

ARTICLE INFO

Article History

Received: 21st May 2016

Received in revised form :

21st May 2016

Accepted: 24th May 2016

Published online :

25th May 2016

Keywords: **Cloud Computing, Fog Computing, Decoy, Encryption.**

I. INTRODUCTION

We live in the age of big data, where the data volumes we need to work with on a day-by-day basis on outgrown the storage and processing capabilities of a single host. Flood of data is coming from many sources such as New York Stock Exchange, Facebook, Ancestry.com, Amazon, Flipkart etc. These sources generate two types of data public and private data. Cloud Computing assurance to significantly change the way to access and store these data. But in cloud the essential issues that occurs is security. And now a day's security and privacy both are main concern that needed to be considered. To overwhelm the issues of security we establish the new technique which is called as Fog Computing. It's not a substitute of cloud it is just add to the cloud computing by providing security in the cloud environment.

Fog computing, also known as fogging, is a distributed computing infrastructure in which some application services are handled at the network edge in a smart device and some application services are handled in a remote data center – in the cloud. The goal of fogging is to improve efficiency and reduce the amount of data that needs to be transported to the cloud for data processing, analysis and storage. This is often done for efficiency reason, but it may also be carried out for security and compliance reasons. [1]

In a fog computing environment, much of the processing take place in a data hub on a smart mobile device or on the edge of the network in a smart router or other gateway device.

The term fog computing is often associated with Cisco. "Cisco Fog Computing" is a registered name; "fog computing" is open to the community at large. The choice of the word "fog" is meant to convey the idea that the advantages of cloud computing can - and should - be brought closer to the data source. (In meteorology, fog is simply a cloud that is close to the ground.)[2]

II. EXISTING SYSTEM

Existing data insurance mechanisms such as encryption was failed in securing the data from the attackers. It does not verify whether the user was authorized or not. Cloud computing security does not focus on ways of secure the data form unauthorized access. Encryption does not provide much security to data. In 2009 we have our own confidential documents in the cloud. This file does not have much security. So, hacker gains access the documents. Twitter incident is one example of a data theft attack in the cloud. Difficult to find the attacker. In 2010 and 2011 Cloud computing security was developed against attackers. Finding

of hackers in the cloud. Additionally, it shows that recent research results that might be useful to protect data in the cloud. Nobody is identified when the attack happens. It is complex to detect which user is attacked. We cannot detect which files were affected during the attack.

III. PROPOSED SYSTEM

We proposed a completely new technique to secure user's data in cloud using user behavior and decoy information technology called as Fog computing. We use this technique to provide data security in the cloud. A different approach for securing data in the cloud using offensive decoy technology. We monitor data access in the cloud and detect abnormal data access patterns. In this technique when the unauthorized person try to access the data of the real user the system generates the fake documents in such a way that the unauthorized person was also not able to identify that the data is fake or real. It is identified thought a question which is entered by the real user at the time of filling the sign up form. If the answer of the question is wrong it means the user is not the real user and the system provide the fake document else original documents will be provided by the system to the real user.

IV. MATHEMATICAL MODEL

Let us consider that we have database 'D' and 'n' number of attribute such as user name, user id etc.

D = {A|A ε Information of user}

Here D is the set of all A such that A is information of user which is to be store on server.
Consider following function STORE (D, SERVER): Here admin enter the user information into database at server.

Let us consider that the receiver provide us with value "X" for every input it obtain from the every time login account of the particular user .so we can further assume to have a set 's to have value 'n' number of detect value at particular instance.
Let us denote the current situation in the following manner.

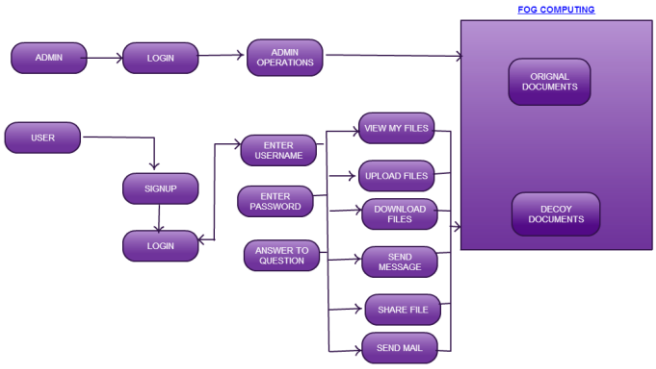
S = {X| X ε D ∃ ID for attacker}

Here S is the set all X such that for all X there exits Id for user. Now, for some X value that match with some value inside the database when admin check user account update.

1. GET (D, X, SERVER): Admin get all information about the user account from server.
2. PUT(X, ATK, SERVER): Here admin will upload attacker's information on server.
3. PUTP(X, REPORT, SERVER): Here admin upload daily report on server.

V. SYSTEM ARCHITECTURE

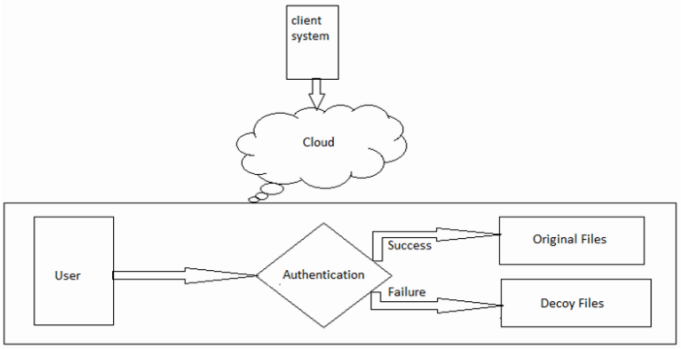
Fog Computing system is trying to work against the attacker especially malicious insider. Here malicious insider means Insider attacks can be performed by malicious employees at the providers or users site. Malicious insider can access the confidential data of cloud users. A malicious insider can easily obtain passwords, cryptographic keys and files. The threat of malicious attacks has increased due to lack of transparency in cloud providers processes and procedures .It means that a provider may not know how employees are granted access and how this access is monitored or how reports as well as policy compliances are analyzed.



VI. TECHNOLOGIES

1. User Behaviour Profiling

We use this technology to detect the behaviour of the user and compare it with the normal user behaviour. User profiling is a well-known technique that can be applied here to model how, when, and how much a user accesses their information in the cloud. Such 'normal user' behaviour can be continuously checked to determine whether abnormal access to a user's information is occurring. This method of behaviour-based security is commonly used in fraud detection applications. [3] Online Behaviour profiling is based purely on a limited set of user actions collected by detection systems. That is why current detection systems have opted to analyse normal user behaviour, define a normal user profile and then raise a red flag if an action outside of that "normal" profile occurs. It's called white-listing.



2. Decoy Technology

Decoy documents or fake date are generated on demand when server encounters unauthorized access to information. Serving decoys will confound and confuse an adversary into believing they have ex-filtrated useful information, when

they have not. Decoy documents are automatically generated. Hashed Message Authentication code (HMAC) algorithm used by decoy information technology to generate bogus information. If the hacker gets the success to hack the username and password he tries to access the files but before that he has been randomly set by the user. Even if the hacker tries and enters anything he gets the access to the account but the data displayed he gets the access to the account but the data displayed will be in the encrypted format. Here the terminology is that a key will be generated every time the key generated during previous login. If the security question entered is correct then same key will be generated and will have access to the data but if the security question falls to be wrong then the key will not be same and thus will have data displayed in encrypted format and the original data will be kept safe on cloud. This will prevent the unauthorized user to hack the data [4].

The advantages of placing decoys in a file system are threefold:

(1) The detection of masquerade activity
(2) The confusion of the attacker and the additional costs incurred to distinguish real from bogus information.
(3) The deterrence effect which, although hard to measure, plays a significant role in preventing masquerade activity by risk-averse attackers.

Combining the Two Techniques

The correlation of search behavior anomaly detection with trap-based decoy files should provide stronger evidence of malfeasance, and therefore improve a detector's accuracy. We hypothesize that detecting abnormal search operations performed prior to an unsuspecting user opening a decoy file will corroborate the suspicion that the user is indeed impersonating another victim user. This scenario covers the threat model of illegitimate access to Cloud data. Furthermore, an accidental opening of a decoy file by a legitimate user might be recognized as an accident if the search behavior is not deemed abnormal. In other words, detecting abnormal search and decoy traps together may make a very effective masquerade detection system. Combining the two techniques improves detection accuracy. We use decoys as an oracle for validating the alerts issued by the sensor monitoring the user's file search and access behavior. In our experiments, we did not generate the decoys on demand at the time of detection when the alert was issued. Instead, we made sure that the decoys were conspicuous enough for the attacker to access them if they were indeed trying to steal information by placing them in highly conspicuous directories and by giving them enticing names. With this approach, we were able to improve the accuracy of our detector. Crafting the decoys on demand improves the accuracy of the detector even further. Combining the two techniques, and having the decoy documents act as an oracle for our detector when abnormal user behavior is detected may lower the overall false positive rate of detector.

We use decoys as an oracle for validating the alerts issued by the sensor monitoring the user's file search and access behavior. In our experiments, we did not generate the decoys on demand at the time of detection when the alert was issued. Instead, we made sure that the decoys were conspicuous enough for the attacker to access them if they were indeed trying to steal information by placing them in highly conspicuous directories and by giving them enticing names. With this approach, we were able to improve the accuracy of our detector. Crafting the decoys on demand improves the accuracy of the detector even further. Combining the two techniques, and having the decoy documents act as an oracle for our detector when abnormal user behavior is detected may lower the overall false positive rate of detector. We trained eighteen classifiers with computer usage data from 18 computer science students collected over a period of 4 days on average. The classifiers were trained using the search behavior anomaly detection described in a prior paper. We also trained another 18 classifiers using a detection approach that combines user behavior profiling with monitoring access to decoy files placed in the local file system, as described above. We tested these classifiers using simulated masquerader data. Figure 1 displays the AUC scores achieved by both detection approaches by user model1. The results show that the models using the combined detection approach achieve equal or better results than the search profiling approach alone.

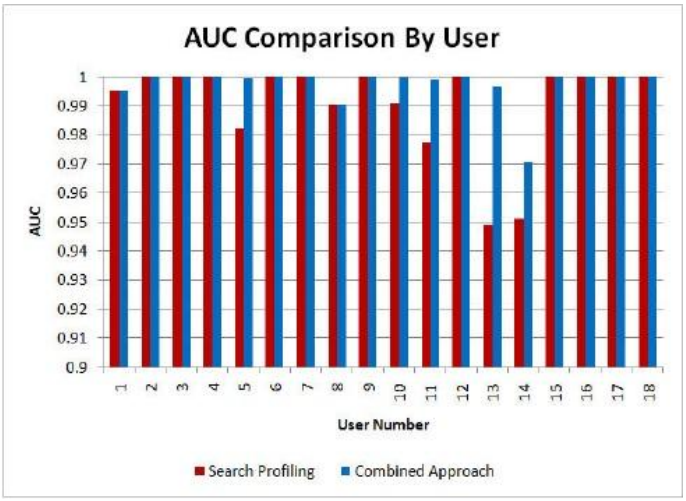


Fig. 1. AUC Comparison By User Model for the Search Profiling and Integrated Approaches

3. AES (Advanced Encryption Standards)

The Advanced Encryption Standard (AES) is a symmetric-key encryption standard approved by NSA for top secret information and is adopted by the U.S. government. AES is based on a design principle known as a substitution permutation network. The standard comprises three block ciphers: AES-128, AES-192 and AES-256. Each of these ciphers has a 128-bit block size, with key sizes of 128, 192 and 256 bits, respectively. The AES ciphers have been analysed extensively and are now used worldwide; AES was selected due to the level of security it offers and its well documented implementation and optimization techniques. Furthermore, AES is very efficient in terms of both time and memory requirements. The block ciphers have high computation intensity and independent workloads (apply the same steps to different blocks of plain text).

Key Length	Number of Rounds
128	10
192	12
256	14

AES is based on a design principle known as a Substitution permutation network. It is fast in both software and hardware. Unlike its predecessor, DES, AES does not use a Feistel network. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits, whereas Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits. The block size has a maximum of 256 bits, but the key size has no theoretical maximum. AES operates on a 4×4 column-major order matrix of bytes, termed the *state* (versions of Rijndael with a larger block size have additional columns in the state). Most AES calculations are done in a special field. The AES cipher is specified as a number of repetitions of transformation rounds that convert the input plaintext into the final output of cipher text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher text back into the original plaintext using the same encryption key.

High-level description of the algorithm

1. Key Expansion: Round keys are derived from the cipher key using Rijndael's key schedule.

2. Initial Round: AddRoundKey: Each byte of the state is combined with the round key using bitwise xor.

3. Rounds:

i. SubBytes—a non-linear substitution step were each byte is replaced with another according to alookup table.
ii. ShiftRows—a transposition step where each row of the state is shifted cyclically a certain number of steps.
iii. MixColumns—a mixing operation which operates on the columns of the state, combining the four bytes in each column.
iv. AddRoundKey Final Round (no MixColumns)
v. SubBytes
vi. ShiftRows
vii. AddRoundKey

VII. CONLUSION

With the increase of data theft attacks the security of user data security is becoming a serious issue for cloud service providers for which Fog Computing is a paradigm which helps in monitoring the behavior of the user and providing security to the user's data. The system was developed only with email provision but we have also implemented the SMS technique. In Fog Computing we presenting a new approach for solving the problem of insider data theft attacks in a cloud using dynamically generated decoy files and also saving storage required for maintaining decoy files in the cloud. So by using decoy technique in Fog can minimize insider attacks in cloud. Could provide unprecedented levels of security in the Cloud and in social networks.

REFRENCES

[1]. Thogaricheti Ashwini, Mrs. Anuradha.S.G, "Fog Computing to protect real and sensitivity information in Cloud", IJECSE,2011.

[2]. Salvatore J. Stolfo, Malek Ben Salem, Angelos D.Keromytis, " Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud", IEEE CS Security and Privacy Workshops, 2012.

[3]. Viraj.G.Mandlekar, VireshKumar, Sanket, Maaz s.Rais, Survey on Fog Computing Mitigating Data Theft Attacks in Cloud, IJIRCST ,Nov 2014.

[4]. Cloud Security Alliance, "Top Threat to Cloud Computing V1.0," March 2010. [Online].Available: https://cloudsecurityalliance.org/topthreats/csathreats.v1.0.pdf.

[5]. Prevention Of Malicious Insider In The Cloud Using Decoy Documents by S. Muqtyar Ahmed, P. Namratha, C. Nagesh.

[6]. Cloud Security: Attacks and Current Defenses Gehana Booth, Andrew Soknacki, and Anil Somayaji.

[7]. Overview of Attacks on Cloud Computing by Ajey Singh, Dr. Maneesh Shrivastava.

[8]. D.Jamil and H. Zaki, "Security Issues in Cloud Computing and Countermeasures," International Journal of Engineering Science and Technology, Vol. 3 No. 4, pp. 2672-2676, April 2011.

[9]. K. Zunnurhain and S. Vrbsky, "Security Attacks and Solutions in Clouds," 2nd IEEE InternationalConference on Cloud Computing Technology and Science, Indianapolis, December 2010.

[10].W. A. Jansen, "Cloud Hooks: Security and Privacy Issues in Cloud Computing," 44th Hawaii International Conference on System Sciences, pp. 1–10, Koloa, Hawaii, January 2011.

[11].F. Bonomi, "Connected vehicles, the internet of things, and fog computing,"in The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET), Las Vegas, USA, 2011.

