Fog Computing: Data Theft Attacks in the Cloud

Santosh Kushwaha[1]
M.Tech Scholor in Dept. of Computer Science and Engineering, S.R. Institute of Management and Technology, Lucknow, UP

Wasif Khan[3]
Assistant Professor in Dept. of Computer Science and Engineering, S.R. Institute of Management and Technology, Lucknow, UP

Nidhi Kushwaha[2]
Assistant Professor in Dept. of Computer Science and Engineering, S.R. Institute of Management and Technology, Lucknow, UP

Ram Krishna Param Hans Dubey[4]
Assistant Professor in Dept. of Computer Science and Engineering, S.R. Institute of Management and Technology, Lucknow, UP

***Abstract*: Cloud computing significantly alters the way we use computer with guaranteed access and storage of our overall information. These new computing and communication models face new data security challenges. Existing data conservation procedures such as encryption fail to prevent data from the attacks of theft especially in the cloud provider. So to overcome these problems we are proposing a new technology called Fog Computing: Data Theft Attacks in the Cloud. I have proposed a different approach in Fog computing to obtain data in the cloud using aggressive decoy technology and user behavior profiling. The users using the Cloud are trapped and their access patterns are recorded. Every User has a unique profile which is monitored and updated. We monitor data access in the cloud by the users and detect abnormal data entry patterns. When unauthorized access is suspected and challenged by challenge questions, we begin the wrong attack by returning the bulk of the information to the attacker. This protects user's real data from being misused. Experiments in a local file setting give evidence that this approach can provide an unprecedented level of user security in the cloud environment.**

INTRODUCTION

Fog computing, or "fogging", is a dispersed foundation in which certain application procedures or administrations are kept up at the edge of the system by keen gadgets, yet others are as yet kept up in the cloud. Presently a days in a business world cloud is utilized to store the more secret information. Security is an imperative issue in the cloud. Most as of late utilized innovations neglect to give security to cloud information. Insider Data Theft Attacks have frequently happened in the cloud, so most business zones know about security issues. So we are utilizing mist figuring for explaining the issues in the cloud by utilizing two advancements. We can utilize bait innovation to give security for cloud. In fake innovation we confound assailants by Sending counterfeit information's. As of late Tweeter account was hacked by the assailants. We propose an extraordinary technique to anchor cloud, called as Fog Computing. We utilize fake data and client conduct profiling to ensure information in the Cloud. We begin hostile assaults against vindictive Insiders utilizing these two advancements subsequently keeping the assailants from recognize the genuine delicate data from the phony information.

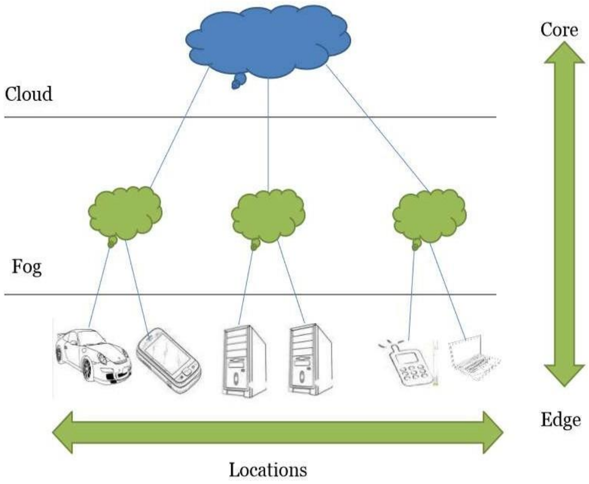

Fig. 1 Computing Process

We propose a totally unique way to deal with anchoring the cloud utilizing distraction data innovation that we have come to call Fog registering. We utilize this innovation to dispatch disinformation assaults against malevolent insiders keeping them from recognizing the genuine delicate client information from phony useless information. In this paper, we propose two different ways of utilizing Fog registering to forestall assaults, for example, the Twitter assault by conveying fake data inside the Cloud by the Cloud benefit client and inside close to home online long range informal communication profiles by individual clients.

Fog computing can be distinguished from cloud by its proximity to end-users.

Decentralised computing means dense geographical distribution and better support

Fog computing provides low latency, location awareness and improves quality of services and rea time applications.

Fig.2 : Working Process

EXISTING SYSTEM

The present framework gives just the single confirmation which isn't much anchor and can without much of a stretch be hacked by a programmer. The framework does not give any extra security like security inquiries for greater security. The programmer can without much of a stretch get into the cloud and scan for the records that are accessible. The present framework does not confirm whether the client is approved or not. The current framework gives security by encryption however it neglects to anchor the cloud.

THREATS IN CLOUD:

1 **Data breaches** – This led to the loss of personal data and credit card information of about 110 million people, it was one of the theft during processing and storage of data.
2 **Data loss** – Data loss occurs when the disk drive dies without any backup created by the cloud owner. It occurs when the encrypted key is unavailable with the owner.
3 **Account or service traffic hijacking** – Account can be hacked if the login credentials are lost.
4 **Insecure API's** – Application Programming Interface controls the third party and verifies the user.
5 **Denial of service** – This occurs when millions of user request of same service and the hackers take this advantage for hacking.
6 **Malicious insiders** – This occurs when a person close to us knows our login credentials.
7 **Abuse of cloud services** – By using many cloud servers hacker can crack the encryption in very less time.
8 **Shared technology** – This occurs when the information is shared by the many sites.

PROPOSED WORK

Proposed structure is to anchor data using antagonistic diversion advancement. We screen data access in the cloud and distinguish odd data get to plans design. Right when unapproved get to is suspected and after that affirmed using challenge questions, we dispatch a disinformation strike by restoring a great deal of lure information to the aggressor. This anchors against the maltreatment of the customers real data. This will be repeated even at the period of downloading from the cloud. Diversion information for instance bait reports, nectar records, nectar pots and distinctive phony information can be created on intrigue and fill in as a technique for distinguishing unapproved access to information and to harm the cheat's ex-filtrated information. Serving impersonations will clutter and puzzle an aggressor into confiding in they have ex-filtrated accommodating information when they have not. This development may be composed with customer lead profiling advancement to anchor a customer's information in the Cloud. At whatever indicate unordinary get to a cloud organization is seen, diversion information may be returned by the Cloud and passed on in order to show up absolutely genuine and conventional.

Key Hashed Message Authentication Code Algorithm (HMAC):

Decoy Information innovation deals with the calculation Key Hashed Message Authentication Code (HMAC). In the event that the programmer gets the accomplishment to hack the username and secret phrase he endeavors to get to the records however before that he needs to cross one more obstruction of security question which has been haphazardly set by the client. Regardless of whether the programmer attempts and enters anything he gets the entrance to the record however the information showed will be in the scrambled organization. Here the wording is that a key will be created each time amid entering the security question. This key will be coordinated each time the key created amid past login will be coordinated with the key produced amid next login. In the event that the security question entered is right then same key will be produced and will approach the information however on the off chance that the security question tumbles to not be right then the key won't be same and along these lines will have information shown in encoded design and the first information will be remained careful on cloud. This will keep the unapproved client to hack the information.

MATHEMATICAL MODEL

Give us a chance to consider that we have database 'D' and 'n' number of trait, for example, client name, client id and so on.

D = {A|A ε Information of user}

Here D is the arrangement of every one of the A with the end goal that An is data of client which is to be store on server Consider following capacity

STORE (D, SERVER): Here administrator enter the client data into database at server.

Give us a chance to consider that the collector furnish us with esteem "X" for each info it get from the each time login record of the specific client .so we can additionally expect to have a set 's to have esteem 'n' number of identify an incentive at specific occurrence.

Give us a chance to indicate the present circumstance in the accompanying way

S = {X| A X ε D Ǝ ID for attacker}

Here S is the set all X with the end goal that for all X there ways out Id for client.

Presently, for some X esteem that match with some an incentive inside the database when administrator check client account refresh.

1. GET(D,X,SERVER): Admin get all data about the client account from server.

2. PUT(X,ATK,SERVER): Here administrator will transfer aggressor's data on server.

3. PUTP(X,REPORT,SERVER) : Here administrator transfer every day investigate server.

The methodology on the most proficient method to finish this theory is partitioned into three stages. In the limits of the individual advances, poise circles will evaluate how the function fits in with the defined prerequisites

Investigation of Existing methodologies - The first step is gathering data about momentum explore materials in the fields of distributed computing, mist registering and the IoT in blend with asset provisioning, benefit arrangement and assignment of loading. After the methodical social occasion of the materials is done, the important data must be separated and dissected to construct the hypothetical cutting edge foundation for the outline of the structure. The extraction of the data will be finished by searching for predefined themes specified. Facilitate an investigation of practical and non-utilitarian necessities must be performed.

Engineering Design- As a subsequent stage, the plan of the mist figuring system design is finished. The plan is a vital point in the advancement of a system and along these lines must act naturally checked and looked into all through the entire improvement process. This will guarantee the accuracy of the plan choices.

Execution and Evaluation- As the usage will be done inside a one individual group and no specific predefined forms or workflows exist, a few parts of surely understood programming improvement forms are separated and customized to deal with the remaining task at hand.

CHALLENGES AHEAD

There are many open issues that should be routed to make the mist a reality. It is important to plainly distinguish them so future research works have these issues into record. The arrangement of open difficulties for the mist to end up a the truth is:

1. Discovery/Sync: Applications running on gadgets may require either some concurred unified point (e.g. build up an upstream reinforcement if there are excessively few companions in our capacity application).
2. Compute/Storage confinement: Current patterns are enhancing this reality with littler, more vitality effective and all the more great gadgets (e.g. one of the present telephones is more great than numerous top of the line work areas from 15 years prior). Still new upgrades are allowed for non-purchaser gadgets.
3. Management: Notwithstanding setting up the correspondence courses crosswise over end hubs, IoT/universal processing hubs and applications running on top should be legitimately setup and designed to work as wanted. Having possibly billions of little gadgets to be arranged, the haze will vigorously depend on decentralized (versatile) administration components that are yet to be tried at this remarkable scale. One thing that can be anticipated with certain level of certainty is that there will be no full control of the entire mist and asymptotic definitive arrangement methods will turn out to be more typical.
4. Security: A similar security worries that apply to current virtualized situations can be predicted to influence mist gadgets facilitating applications. The nearness of secure sandboxes for the execution of bead applications presents new intriguing difficulties: Trust and Privacy. Prior to utilizing different gadgets or smaller than expected mists in the system to run some product, separation and sandboxing components must be set up to guarantee bidirectional trust among coordinating gatherings. The mist will enable applications to process clients information in outsider's equipment/programming. This obviously presents solid worries about information security and its perceivability for those outsiders.
5. Standardization: Today no institutionalized systems are accessible so every individual from the system (terminal, edge point...) can declare its accessibility to have others programming parts, and for others to send it their product to be run.
6. Accountability/Monetization: Having clients ready to share they save assets to have applications is vital to empower new plans of action around the idea of the mist. A legitimate arrangement of motivating forces should be made. The motivating forces can budgetary or something else (e.g. boundless free information rates). Then again the absence of focal controlling element in the haze makes it hard to affirm if a given gadget is without a doubt facilitating a part (bead) or not.

7. Programmability: Controlling application lifecycle is as of now a test in cloud conditions. The nearness of little useful units (beads) in more areas (gadgets) requires the correct reflections to be set up, so developers don't have to manage these troublesome issues. Simple to utilize APIs for software engineers will vigorously depend on basic Management components that give them the correct deliberations to conceal the gigantic complexity of the fog. Some vendors like Microsoft have already taken some steps in positioning themselves in this space.

CONCLUSION

Consequently in this paper we propose a particular innovation to make the cloud more secure by anchoring the individual and the essential information of the business firms. We give observing of the entrance to the record by checking the conduct of the client. We give get to by login certifications as well as by test addresses which would be just known to the client. In the event that the entrance observed to be unapproved in this way giving the phony information with the goal that the genuine information of the client can be spared. This innovation would include a level in anchoring the information on the cloud. In this methodology client information is anchoring in the cloud we propose information get to designs by profiling client conduct to distinguish when insider assailant get to other client information in cloud benefit distraction data put away in cloud alongside the client unique information and furthermore utilized as sensor to recognize unapproved get to once unapproved information get to identify and confirmed by test question for example we give sham information to malignant insider to ensure clients unique information.

REFERENCES

[1] Salvatore J. Stolfo, Malek Ben Salem, Angelos D. Keromytis, "Haze Computing Mitigating Inside Data Theft Attacks In The cloud",IEEE Base Paper, 2013

[2] F. Rocha and M. Correia, "Lucy in the sky without jewels: Stealing private information in the cloud," in Proceedings of the First International Workshop on Dependability of Clouds, Data Centers and Virtual Computing Environments, Hong Kong, ser. DCDV '11, June 2011.

[3] M. Van Dijk and A. Juels, "On the difficulty of cryptography alone for protection saving distributed computing," in Proceedings of the fifth USENIX meeting on Hot subjects in security, ser. HotSec'10. Berkeley, CA, USA: USENIX Association,pp. 1– 8, 2010.

[4] M. Ben-Salem and S. J. Stolfo, "Displaying client searchbehavior for disguise identification," in Proceedings of the fourteenth International Symposiumon Recent Advances in Intrusion Detection. Heidelberg: Springer,pp. 1– 20,September 2011.

[5] D. Takahashi, "French programmer who spilled Twitter archives to TechCrunch is busted," March 2010.