Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.217 – 222

RESEARCH ARTICLE

Alleviating Internal Data Theft Attacks by Decoy Technology in Cloud

I.Sudha[1], A.Kannaki[2], S.Jeevidha[3]

[1] Department of Computer Science and Engineering, Pondicherry University, India

[2]Department of Computer Science and Engineering, Pondicherry University, India

[3]Department of Computer Science and Engineering, Pondicherry University, India

[1] sudha123prem@gmail.com; [2] kannakianbu@ymail.com; [3] jeevidha21@gmail.com

Abstract— Cloud Computing enables several users to share common computing resources, and to access and store their personal and business information. The accessing includes so many things as well as can keep their private and industrial information. These new thoughts and innovations have pro's at the same time con's too. And there is new security challenges has been a raised. The increase in the number of cloud users are from the World Wide Web users means of internet. The users who have prospective valid credentials which contain username and password are treated as insiders. In security perspective, all the remote users are known as attackers. Some active security mechanisms fails to prevent data theft attacks and it should make sure that the remote user is not an attacker. We propose a new approach for securing data in the cloud by using user profiling and provoking decoy technology. When an unauthorized access is assumed and then confirmed using various challenge questions, we initiate a disinformation attack by returning huge amount of decoy information to the attacker. This approach protects against the misuse of the original user data. When a decoy document is loaded into memory, we authenticate whether the document is a decoy document by computing an HMAC based on all the contents of that document.

Keywords— Cloud Security; Fog Computing; Data Theft Attacks; Decoy Technology

I. INTRODUCTION

Cloud computing refers to a centralized site on the Internet that stores data, making it accessible anytime, anywhere, from any device. Small businesses have squeezed the cloud opting for outsourcing data and computation to the cloud because it has a number of benefits, including reduced cost, ease of use, elasticity and automation. Data theft attacks are augmented if the attacker is a malicious insider. Existing data protection mechanisms such as encryption have failed in preventing data theft attacks, especially those perpetrated by an insider to the cloud service provider. Many customers of cloud computing are aware of this kind of threat especially malicious insiders. However, the data stored must be protected in the cloud storage by the cloud service provider. There is a lack of transparency, problem in data dynamics and security related problem like authorization, authentication, and audit controls only exacerbates this risk. The Cloud Security Alliance has identified the top threats to cloud computing. The threats are discussed:-

- Nefarious and abuse use of cloud computing
- Insecure APIs

- Malicious insiders
- Shared technology vulnerabilities
- Data leakage or loss
- Hacking Account

The Twitter incident is one example of the data theft attack from the cloud service provider. This incident makes the customers to lose their sensitive data and documents. Several Twitter corporate and personal documents were expatriated to technological website Tech Crunch and customers' accounts, including the account of U.S. President Barack Obama, were illegally accessed. The attacker used a Twitter administrator's password to gain access to Twitter's corporate documents hosted on Google's infrastructure as Google Docs. The damage was significant both for Twitter and for its customers. While this particular attack was launched by an outsider, stealing a customer's admin password is much easier if perpetrated by a malicious Associate. In their work Rocha and Coria explored how to steal easy passwords through malicious insider of cloud service provider (CSP). They also demonstrated how to steal private keys and the confidential data which is saved on hard disk After stealing a customer's password and private key, the malicious Associate get access to all customer data, while the customer has no means of detecting this unauthorized access. Over a 160,000 Gmail users had their accounts deleted. The data was restored, but accounts remained unavailable for days. A hacker used Amazon's Elastic Compute Cloud, or EC2 service to attack Sony online entertainment systems.

We propose a completely different approach to secure the cloud with the decoy information technology and is called as "Fog Computing". We use this technology to instigate disinformation attacks against malicious insiders, which helps to prevent and distinguish the real sensitive customer data from fake worthless data. The Decoy Information Technology is used for validating whether data access is authorized when abnormal information access is detected. It helps in confusing the attacker with bogus information.

II. SECURING CLOUD WITH FOG

The basic idea is that we provide a security by following this preventive measure in the fig.1which ensures avoidance of insider data theft attacks. We conceive that cloud services can be implemented securely by two features, namely user behavior profiling and decoys.

A. User behaviour profiling

User profiling is a familiar Technique has been applied to model how, when, and how often a user accesses their information in the cloud environment. This module is concerned about storing the user's request for files on the web application. The operations may include creating, read, write, delete. Normal user and his behaviour can be continuously monitored to check whether abnormal access has been occurred to the user data. This way of behavior-based security is commonly used in fraud detection applications. Such profiles naturally include volumetric information, how many documents are typically read and how often. We check for abnormal search behaviors that exhibit deviations from the user baseline the correlation of search behavior anomaly detection with trap-based decoy files should provide stronger evidence of malfeasance, and therefore improve detector accuracy.

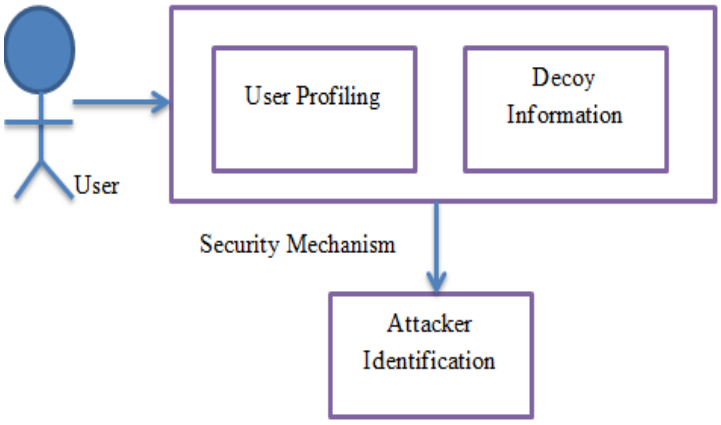

Fig 1. Securing Mechanism

B. Decoys

Any Decoy file in the fig.2 has certain properties as [BIAUDD] "Believable, Enticing Conspicuous, Non-interference, Detectable, Variability, Differentiability ". We use decoy files that contain "bad information" such as online banking logins, social security numbers, and web-based email account credentials. The Decoy File System has for each newly created folder or a file, the corresponding decoy file will be maintained. The directory and file structure are same for both the decoy file system and the original file system. The information contained in the decoy file is not original.

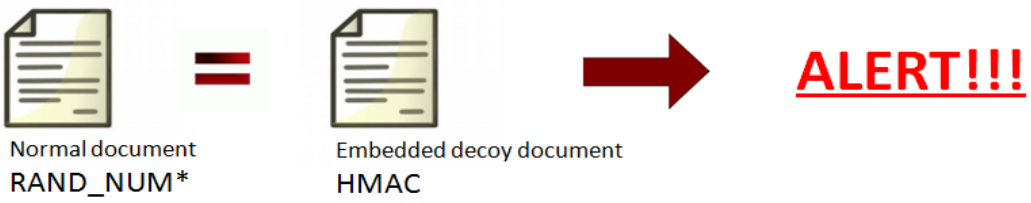

Fig 2. Decoy Document

Some researchers proposed the use of honeyfiles, a type of honeypot, to detect malicious insider activity [4]. They introduced the concept of perfectly believable decoys and proposed several properties to guide the design and deployment of decoys, namely:

1. Believability: The attacker would not use the bait information if it did not appear authentic to the attacker.
2. Enticingness: No attack detection is possible if the attacker does not access the bait information because it does not look attractive enough.
3. Conspicuousness: Decoys should be easily located or retrieved in order to maximize the likelihood that an attacker takes the bait.
4. Detectability: If the access to the bait asset is not detectable than the deployment of the decoys is useless.
5. Variability: Decoys should not be easily identifiable to an attacker due to some shared invariant.
6. Non-interference: Decoys should not interfere with the legitimate user's normal activity. Non-interference has been defined as the likelihood that legitimate users access the real documents after decoys are introduced [4].
7. Differentiability: Legitimate users should be able to easily distinguish decoy documents from authentic documents, which has a direct effect on non-interference.
8. Shelf-life: Decoys may have a limited time period during which they are effective.

III. DETECTION APPROACH THROUGH SENSORS

All Host-based sensors are used to detect user activity profiling and embedded markers. Embedded markers are tagged documents with keyed-hash message authentication codes (HMACs) which makes them matching with the definition of "perfect secrecy". Detectable by the host-level or network decoy sensor. A host based sensor monitors the accesses performed by a process or an application at HMAC-embedded decoy documents. It has two features, namely, behavior modeler, which tracks the legitimate user activity in order to get a baseline of what is considered a normal behaviour and document access sensor to detect when documents containing embedded markers are read, copied or transmitted.

In this approach, we define a modified sensor to detect data theft attempts and "need-to-know" policy violations perpetrated by traitors on multi-user systems like file systems. We classify this sensor as the RUU (Are You You?) Sensor. The sensor is composed of two sub-sensors namely User Search Behavior (USB) sensor for user search behavior, and Decoy Documents Access (DDA) sensor for which it has been used to monitor any access to the decoy documents embedded in the file system. It also serves as an oracle for the USB sensor. We express how each component of the RUU sensor works, and how the USB and DDA sensors are integrated in order to detect masquerade attacks shown in the fig. 3. RUU sensor endows with three mitigation strategies when it suspects malevolent masquerade activity. These strategies can be selectively implemented depending on the confidence level of the sensor that malicious activity takes place:

- Sending an alert message to a remote server.
- Displaying a set of challenge-response questions that the user must correctly respond to.
- Stealthily recording audio and taking a picture if a webcam is available.

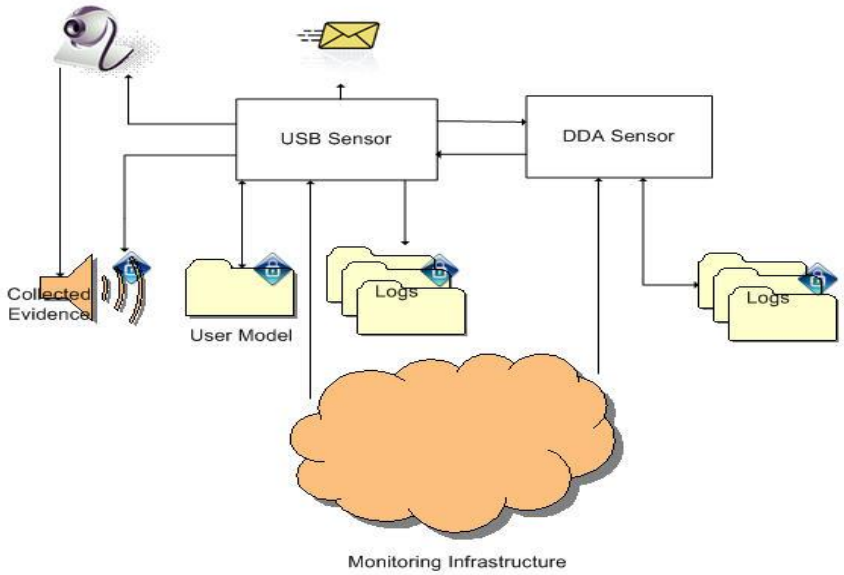

Fig 3. Architecture of the RUU Masquerade Attack Sensor

A. Profiling Search Behaviour

A Masquerader is an individual who has not authorized to use the computer and who penetrates a system's access control to exploit a legitimate user's account. However, the masquerader is likely to be an outsider who gets access to the victim's system illegally and unlikely to be familiar with the structure and the contents of the file system. Thus, their search seems to be pervasive and integrated. Based on this key assumption we define USB sensor to detect for abnormal search behaviors that exhibit large deviations from the baseline. Such deviations signal a potential masquerade attack. This sensor builds a one-Class Support Vector Machine (ocSVM) model that models the user's search behavior. Vectors with three search-related features are extracted for each two-minute period of user activity. The three search behavior-related features are:

- Number of automated search-related actions.
- Number of file touches, including fetch-, read-, write-, or copy- operation.
- Percentage of file system navigation user action.

We categorize two thresholds per user model to organize the observed user search activity as normal, abnormal, or non-identifiable. The first threshold t_{hr1} is determined empirically, so that the miss rate or false negative rate is minimized and the second threshold t_{hr2} is also set to minimize the FP rate. We evaluate the deviation d between actual user behavior and the historical user behavior as defined by the user model u. The distance d is compared to t_{hr1} and to t_{hr2} in order to determine whether there is enough proof for masquerade activity.

B. Monitoring Access to Decoy Documents

The Decoy files are well suited to the challenge of detecting insider threats because they can be used to issue alerts when attackers start accessing files even after all other defenses are circumvented. Decoy files can be downloaded from a fog computing site that serves as an automated service that offers several types of decoy documents such as tax return forms, credit card statements, medical records,e-bay r eceipts, etc.These files can be downloaded from the Decoy Document Distributor (D^3).A masquerade access these decoys files without having an idea about the file system contents if they search for sensitive information, such as the bait information embedded in these decoy files. While monitoring access to the decoy files usually it leads to indicate masquerade activity on the system. We use decoy documents that carry a keyed-Hash Message Authentication Code (HMAC) embedded in the header section of the document. It is visible only if the document is opened using a hex editor The HMAC is computed over the file contents using a unique key to each user. The DDA sensor detects when decoy documents are being read, copied, or zipped. Whenever the decoy document is loaded into the memory, it checks whether the file is a normal or decoy file by computing an HMAC based on all the contents of that document. We compare it with HMAC embedded within the document. If the two HMACs match, the document is deemed a decoy and an alert is issued. The advantages behind the

insertion of decoys in a file system are three- fold: (1) the detection of masquerade activity (2) the perplexity of the attacker and the additional costs incurred to differentiate real from trick information, and (3) the deterrence effect which, although hard to measure, plays a vital role in preventing masquerade activity by risk-averse attackers.

C. Integrated Masquerade Detection Approach

In this approach, we detect anomalous search and decoy traps together for an effective masquerade detection system. Combining the two techniques improves detection accuracy and provides unique levels of security to detect insider theft attacks. We apply these concepts to detect illegitimate data access to data stored on a local file system by masqueraders, *i.e.* Attackers who pose as legitimate users after stealing their credentials. When a rogue insider tries to use the Cloud for data dynamics, he gets attracted to bogus information which appears sensitive and useful to hackers. This way the proposed application deceives malicious users to behave that way and avoid insider theft attack. Our experimental results in a local file system setting show that combining both techniques can yield better detection results, and our results suggest that this approach may work in a Cloud environment, as the Cloud is intended to be as transparent to the user as a local file system. In this approach, we detect anomalous search and decoy traps together for an effective masquerade detection system. Combining the two techniques improves detection accuracy.

We define two detection thresholds for each user search model thr_1 and thr_2. If the user behavior captured in feature vector v is similar enough to the user model u, which captures the user's historical behavior, then the user behavior should be deemed normal. In other words, if the distance d between the v and user model u is smaller than thr_1, then no masquerade activity is suspected, and no alert gets generated. If, on the other hand, feature vector u exhibits a highly abnormal search, i.e. If $d > thr_2$, then an alert is generated. However, if $thr_1 < d \leq thr_2$, then the USB sensor checks whether any excessive access to decoy documents has been recorded by the DDA sensor. If so, then an alert is generated and the right mitigation strategy is initiated. Otherwise, the user search activity is not deemed suspicious enough. The Fig. 4 describes the overall decision process related to masquerade alert generation using the two sensors.

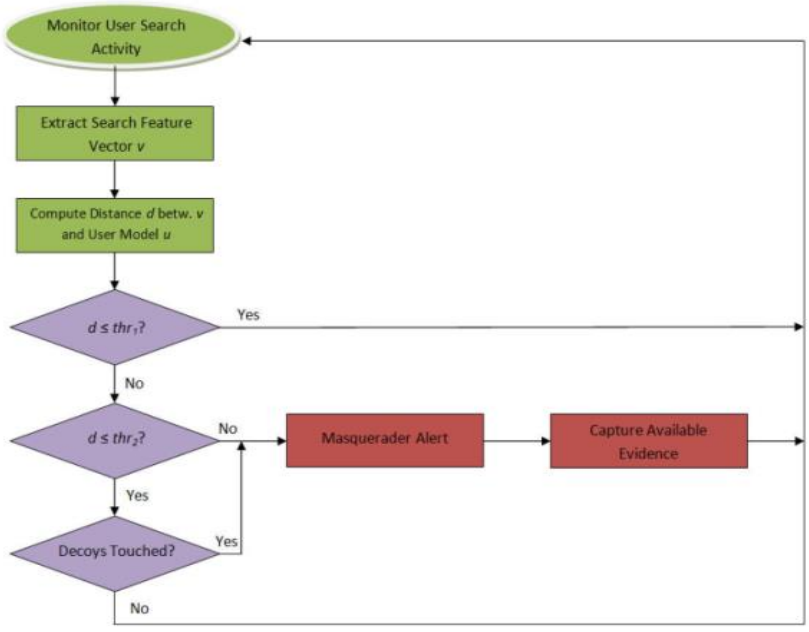

Fig 4. Alert Generation Decision Process

We qualified 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. The figure 5 displays the AUC scores achieved by both.

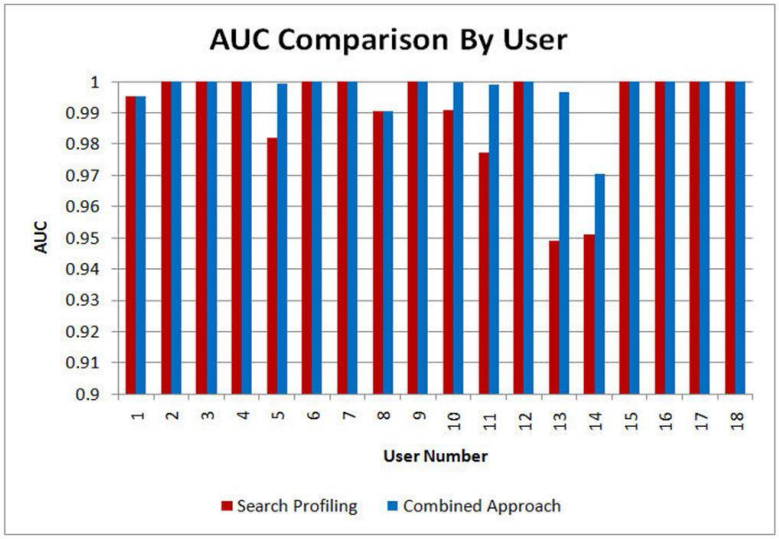

Figure 5. AUC score by both approach

IV. CONCLUSIONS

The Masquerade attacks cause a computer security problem. In this paper, we present an integrated approach to prevent insider data theft attacks in the Cloud. We propose the user profile management ensures that the legitimate users' behavior and navigational patterns are recorded. The decoy technology allows the application to keep decoy information or bait information in the file system to deceive insider data theft attackers. Once unauthorized data access or exposure is suspected, and later verified, with challenge questions, for instance, we overwhelm the malicious insider with bogus information in order to dilute the user's real data.

REFERENCES

[1] X Bowen, B. M., Hershkop, S., Keromytis, A. D., And Stolfo, S. J. Baiting inside attackers using decoy documents. In SecureComm'09: Proceedings of the 5th International ICST Conference on Security and Privacy in Communication Networks (2009).

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

[3] F. Rocha and M. Correia, "Lucy in the sky without diamonds: Stealing confidential data 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.

[4] New IDC IT Cloud Services Survey: Top Benefits and Challenges. Retrieved April 8, 2011 from http://blogs.idc.com/ie/?p=730

[5] D. Takahashi, "French hacker who leaked Twitter documents to TechCrunch is busted," March 2010. [Online].Available:http://venturebeat.com/2010/03/24/french-hacker-wholeaked-twitter-documents-to-techcrunch-is-busted/

