Intrusion Detection System Using Fog Computing

Aamenah Azher Bashir - 20801620
University of Waterloo
Waterloo, ON, Canada
aabashir@uwaterloo.ca

Megha Patel - 20754590
University of Waterloo
Waterloo, ON, Canada
mn7patel@uwaterloo.ca

Shafaq Iqbal - 20697578
University of Waterloo
Waterloo, ON, Canada
sz3iqbal@uwaterloo.ca

***Abstract*— Fog computing can be used as a deception-based active prevention technology, detecting and tracking effectively against external attack and internal threat. This technology proposes three security challenges: generation and deployment of decoy document, unidirectional transparent detection, and present corresponded improving measures. [1]**

Keywords*— *fog computing, decoy information, user behavior profiling, intrusion detection

I.Introduction

There are two focus points in this project that are Fog Computing and Intrusion Detection System. Fog computing is a recent computing infrastructure in the field of networking. It enables decentralization of the whole network providing distribution of data, computation, storage and applications at all efficient places between the data source and the cloud. Narrowing down the vast topic of fog computing to one of its application that is Intrusion Detection System gives an idea of how Fog computing can be used to achieve security of the systems [2].

Fundamentally, the development of fog computing frameworks gives organizations more choices for processing data wherever it is most appropriate to do so. For some applications, data may need to be processed as quickly as possible – for example, in a manufacturing use case where connected machines need to be able to respond to an incident as soon as possible.

Fog computing can create low-latency network connections between devices and analytics endpoints. This architecture in turn reduces the amount of bandwidth needed compared to if that data had to be sent all the way back to a data center or cloud for processing. It can also be used in scenarios where there is no bandwidth connection to send data, so it must be processed close to where it is created. As an added benefit, users can place security features in a fog network, from segmented network traffic to virtual firewalls to protect it [3].

II.Project Scope

Users should be able to store data on the fog without any concern for attacks and intrusion on their saved data. Two technologies can be used to provide additional security to the users while storing data on the cloud:

- User behavior profiling can be used to understand the user access patterns and therefore determine illegal access to the fog.
- Decoy technology can be used for the various types of files, such as images, data files, multimedia files, etc.

III.Fog Computing

Fog computing is an architecture that extends the scope and services of cloud computing. It was proposed by Cisco Systems Inc. in 2012 [5] to ease up the processing of the data at edge nodes i.e. by an intermediate architecture which is placed in between the cloud and the IoT devices but is also said to be present at the network edge. Fog nodes can be imagined as small clouds which are connected together to be a part of one big cloud. They can be any device from resource-constrained device to powerful servers or a part of end users' devices. The two major advantages of introducing fog computing over cloud computing are:

1. De-centralization of Cloud services

Fog computing is a distributed structure of fog nodes each of which provides services of cloud such as

- compute
- control
- storage
- networking
- communication

2. No real-time information exchange with the cloud:

With cloud computing the IoT devices are directly connected to the cloud and hence all the processing even the ones that are time critical or need a real time response had to be dealt by the cloud server thus increasing the load on the cloud and increasing the latency. Fog computing resolves both these issues of load balancing and reducing the latency significantly by placing the fog node device very near or with the end device. In this way there is no need to communicate with cloud every time which is a great achievement especially when real time exchange of information is involved.

Low Latency: In case of cloud computing the distance between the cloud and the end device is long and thus cause latency which is of great concern in time sensitive applications such as augmented reality. On the other hand, the fog layer is capable of providing such services of the cloud and is a lot nearer to the end device as compared to the cloud. Thus, better latency can be obtained using fog computing.

High Scalability: One of the major challenges when it comes to cloud computing is the available resources that has to be allocated to accommodate million if IoT devices that requires storage, computing capabilities and network bandwidth. Even though the computing resources can be increased by adding more hardware but to achieve enough network bandwidth for all the devices is not easy. Fog computing decentralize the cloud computing therefore dealing with only a specific number of devices which is not too large. Scalability and capacity can be achieved by adding as many fog nodes as are required.

Location Awareness: Cloud also takes care of providing location-based services to its end users. Location Based Services (LBS) tasks put more load on the cloud and also because the cloud is so far away from its users, users have to send their location information every time they need access to LBS services which increases the risk of leaks of location information. With fog computing the user only sends service requests without and location information because the fog nodes are very near to the end devices thus making it safe and secure [6].

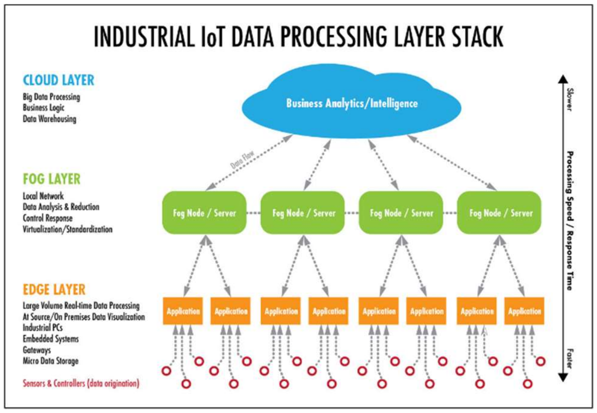

Figure 1 [15]

A. *Architecture of Fog Computing*

The fog network is divided into three major categories usually known as layers, the cloud layer, fog layer and edge layer. The cloud layer is the main cloud system which is the central point of the architecture where all the nodes join. The speed at the cloud layer is the slowest since it is farthest from the source node i.e., the IoT devices. The fog layer is the middle layer where several fog nodes exist. These fog nodes are the intermediate layer between the IoT devices and the cloud. Its presence reduces the latency in the transfer and processing of data. The last layer is the edge layer where all the IoT devices exist. The devices at the edge layer can also work as fog node if it has the capabilities of fog such as compute, control, communicate, storage and networking. This is why it is hard to explain the separation between the fog and edge layer. These layers are shown in the figure above [4].

This technique is having great importance for Internet of Things as the amount of data produced by a large number of sensors is huge. Also, the safety critical processes need high speed of data transfer which is achieved by the fog nodes places very near to the edge device to fulfil the requirement.

B. *Security issues with Fog Computing*

Security in communication systems have been a concern since decades. Security engineers are working constantly to come up with solutions to protect the system from the attackers but how much ever progress is achieved, it is always incomplete when it comes to the attackers who are smarter then what is thought of them. IoT devices that form the lowest layer of the fog network interact with the fog nodes mainly for processing and storage requests. On the other hand, two fog nodes will communicate when network resources are to be managed among the fog network. There are two possible communication links that needs to be secured in a fog network:

1. Communications between constrained-IoT devices and fog nodes
2. Communications between fog nodes.

Some very basic issues that are seen when it comes to security threats are:

1. Trust

All the devices connected in a fog network is expected to have a certain level of trust between each other. Authentication is necessary when important information is being transferred. It is the first step to build a secure connection between IoT devices and fog nodes. The authentication process is not considered to be enough to provide complete security as devices can malfunction as well as are susceptible to malicious attacks and threats. In such cases, a trust model is needed which is formed on the basis of previous interactions made between the device and node. Trust is needed both ways in a fog network i.e., from fog node to IoT and from IoT to fog node.

2. Authentication

Authentication plays an important role in building the secure connections within the fog network. Any new device that joins the fog network has to authenticate itself before entering the network to keep the unauthenticated devices away from the secure network. This is a challenge due to the constraints associated with the devices such as power, processing and storage all of which has to be authenticated.

3. End User Privacy

In any communication system the privacy of the end users is a major concern. The data being shared between two parties can be important information like identity, bank details, usage of utilities, location information or random talks but all of the information and the people having that communications needs to be secure. Since fog computing is not a centralized system architecture, controlling things from one central point is difficult. Low secured edge nodes can be an invitation for the attackers which is the threat for end user's privacy. [7]

4. Insider Attack

Insider attack also known as insider threat is a kind of attack which is done from inside of an organization. People who are working or have worked in a company knows the personal authentic information of that organization like passwords and other highly important details. If that person becomes the attacker and attacks the system this becomes an insider attack.

5. Masquerader attack

Masquerader attack is an attack done by people who disguise themselves under a mask to attack the system. These attackers use fake identities to hack a personal computer and steal important information. The have stolen passwords and other important details to breach data from inside the organization. The probability of masquerade attacks increases if the authorization process is not fully protected [8].

To secure the system from such attacks and threats, work is being done to develop an intrusion detection system that can detect from where and what information is leaking. The next part of the report will be focusing on how an intrusion detection system works.

IV.INTRUSION DETECTION SYSTEM

Intrusion Detection System is a system that is used to detect any intrusions from the attacker into the system. It is also known as anti-theft system. The aim of such a system is to prevent internal threats and data leakage before it occurs by generating documents (PDF file, MS Office file etc.) which are similar to the real files but having fake content to trick the attacker that he has recovered the information where in reality he has not got access to any real information. These documents are known as decoy. *Th*ese decoy documents are deployed in computers and file servers.

V.USER BEHAVIOR PROFILING

User behavior profiling is a technique whereby the volumetric information is analyzed to evaluate how often a user carries out certain events like accessing their files [9].

Blacklist Count Algorithm

Through the below algorithm, the system detects how many unauthorized people are attempting to access the account, increasing the blacklist count. Only authorized people have access to clear that blacklist count. When an unauthorized person attempts to clear the count, the system informs the authorized person [14].

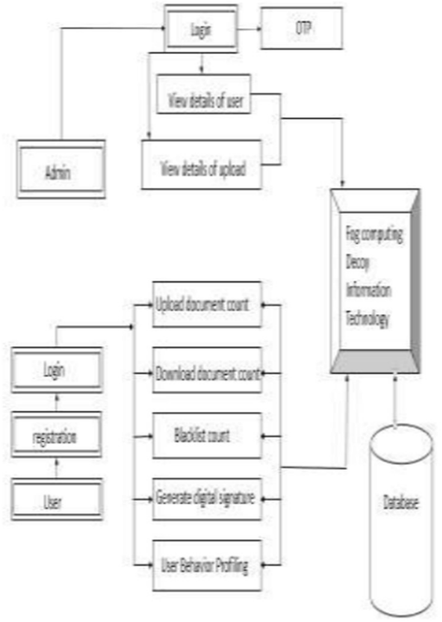

Figure 2 [14]

Mathematical Model

Let, G be the superset of all sets.

G ≡ {input, output, operations, success, failure}

Where, Input is set of parameters provided as input to system. Input ≡ {U, S, DS, F} U is set of users. It is infinite set of users.

U ≡ {U1, U2, U3……………Un} S is set of servers.

It is finite set of servers.

S ≡ {S1} DS is set of dataset parameters.

DS ≡ {P1, P2, P3, P4, P5}

P1 ≡ Session Time

P2 ≡ Duration

P3 ≡ File upload count

P4 ≡ File Download count

P5≡ Blacklist count F is set of files. It is Infinite set of files.

F ≡{F1, F2, F3…………………, Fn }

Output is set of results.

Output ≡ { Legal user/Unreal user, Decoy document, Alert user via mail, OTP via SMS } Operations is set of functions.

Operations ≡ { Op1, Op2, Op3, Op4, Op5, Op6, Op7, Op8, Op9}

Op1 ≡ Request received

Op2 ≡ Load user profile

Op3 ≡ Apply mining & calculate current request parameter

Op4 ≡ if invalid user then send the Decoy/Bogus data

Op5 ≡ Fetch file

Op6 ≡ Calculate digital signature

Op7 ≡ Compare with decoy file digitally

Op8 ≡ If similar, Alert admin

Op9 ≡ Update log, Blacklist

SUCCESS ≡ Desired input generated

FAILURE ≡ Desired output not generated [14]

VI.DECOY TECHNOLOGY

A decoy document consists of three major parts namely beacon, mark and content. Beacon is actually a code file that is executed to see if the person accessing the decoy file is attacker or the user. Different methods like hashing or HMAC is used to ensure the security in this part of decoy.

Mark is used to differentiate between the authentic documents from the decoy documents. A well-known algorithm to generate mark is the key-based HMAC, 128 bits; which can be embedded in the header section of any editable file, e.g. the OPC Properties Section of a PDF document.

The file content contains the document name of the decoy. These names are selected in such a way that it entices the attacker to attack those files as it contains information like transaction details, credit card information or private emails etc.

In the field of communication networking decoy documents are also known as honeypots. As described above, these honey pots act as baits for detecting masqueraders and malicious insider activity.

The decoy documents or honey pots that are created are expected to look as real as they can so that the attacker can be trapped. The researchers have proposed certain properties that defines a perfect decoy document which includes:

- Believability: The bait information should look authentic enough for the attacker to believe it.
- Enticingness: The document should be appealing enough to the attacker to access it because that is necessary for the detection.
- Conspicuousness: The location of the baits should be such that they are easily accessible to the attacker.

- Detectability: The document once accessed should be able to detect who and when accessed the document so that the attacker can be detected.
- Variability: They documents should be distinctive enough so that the attacker cannot predict which documents are authentic and which are not.
- Non. Interference: The fake documents should not be mistaken with the real ones by the authorized person.
- Differentiability: It is important that the authentic user is able to distinguish between the fake and original decoy documents to make their life easier.
- Shelf-life: Decoy documents should be kept active for a certain time period after which they should disappear from the system.

Also, honey tokens are files with inappropriate information that user should not access but if the attacker accesses them then this information can be used to direct the attacker towards more advanced baits called honeypot which helps in identifying the malicious intent of the attacker. The pattern that an attacker follows in accessing these honeypots helps in determining what an attacker is looking for and how vulnerable are his/her intentions.

There are different ways introduced by the researchers to introduce decoy files into the system. Yuill et al, a researcher proposed a way in which the system gave the privilege to the users to convert their files into decoy files on a network server with a record that has all the filenames with associated user ids for proper identification of the honey files. On the other hand, Bowen at el, who also worked on this notion developed a system which was capable of generating the decoy files automatically. This automated system kept in consideration all the properties that a decoy file should contain e.g. enticingness, variability etc. [10].

Deception-based Active Prevention (DAP) has gradually been a research emphasis in information security protection due to high pertinence and accuracy to anti-theft 1. Based on precedent experiments, existing data protections such as firewall, intrusion detection have been being poor at defending unknown intrusion pattern. DAP can make up for that, meantime it can be on the alert against internal threat or external theft action, saving precious response time for security administrator to cope with theft events. Fog computing 2 is a typical case of DAP, aiming at preventing internal threat and data leakage in advance by generating enticing decoy documents automatically (PDF file, MS Office file), embedding information collecting beacon, deploying decoy in computers and file servers even cloud services in targeted defense environment, monitoring decoys access records, receiving leaked beacon
alert, monitoring the misuse of decoy document. So, we can trace and locate the attacker precisely in time.

Fog computing consists of 3 key technologies:

(1) Believable decoy generation and distribution technology.
(2) Data leakage sensing, defending and tracing technology.
(3) Anomaly behavior modeling and mining technology.

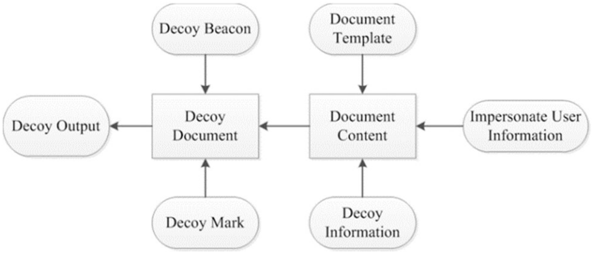

Figure 3: generation of decoy [1]

This decoy combined with user behavior profiling that is to track the behavior of the actual user about how they access their files can alert the system if any theft is happening. This method also helps in locating the attacker and secure the system from that attacker in the future.

C. *Decoy technology and HMAC*

The key modules of decoy document are beacon, mark and content. Beacon is executable script codes, being in charge of collecting attackers' information stealthily. Mark is used to distinguish decoy ones from authentic ones. File content contains decoy document name, aiming to entice the attacker to steal and check [11].

Beacon is actually a code file that is executed to see if the person accessing the decoy file is attacker or the user. Different methods like hashing or HMAC is used to ensure the security in this part of decoy. Beacon is executable script codes, being in charge of collecting attackers' information stealthily. Mark is used to distinguish decoy ones from authentic ones. File content contains decoy document name, aiming to entice the attacker to steal and check.

The decoy documents carry a keyed-Hash Message Authentication Code (HMAC), which is hidden in the header section of the document. HMAC uses two passes of hash computation. The secret key is first used to derive two keys – inner and outer. The first pass of the algorithm produces an internal hash derived from the message and the inner key. The second pass produces the final HMAC code derived from the inner hash result and the outer key. Thus, the algorithm provides better immunity against length extension attacks.

The HMAC is computed over the file's contents using a key unique to each user. When a decoy document is loaded into memory, we can verify whether the document is a decoy document by computing a HMAC based on all the contents of that document. Then, compare it with HMAC embedded within the document. If the two HMACs match, the document is deemed a decoy and an alert is issued. 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" behavior can be continuously checked to determine whether abnormal access to a user's information is occurring [11].

Mark is used to differentiate between the authentic documents from the decoy documents. A well-known algorithm to generate mark is the key-based HMAC, 128 bits; which can be embedded in the header section of any editable file, e.g. the OPC Properties Section of a PDF document [1].

The file content contains the document name of the decoy. These names are selected in such a way that it entices the attacker to attack those files as it contains information like transaction details, credit card information or private emails etc.

D. *Combining user behavior profiling and decoy technology*

User behavior profiling is best suitable technique to detect whether the access is illegitimate or not. Behaviors that shows deviations from the user baseline can be detected by monitoring to recognize legitimate users or masqueraders. Decoy technology comes into the picture when unauthorized access to information is detected. Decoy technology serves two purposes:

1. Validating the authorization of data accessed
2. Confusing the attacker

By combining these two techniques we can achieve a good security against masqueraders. Eighteen classifiers are trained with computer usage data using the search behavior anomaly detection over 4 days. Another 18 classifiers are trained using the combined detection approach to compare with. After analyzing the AUC scores achieved by both detection methods, it is shown that the combined detection approach achieve better results [12].

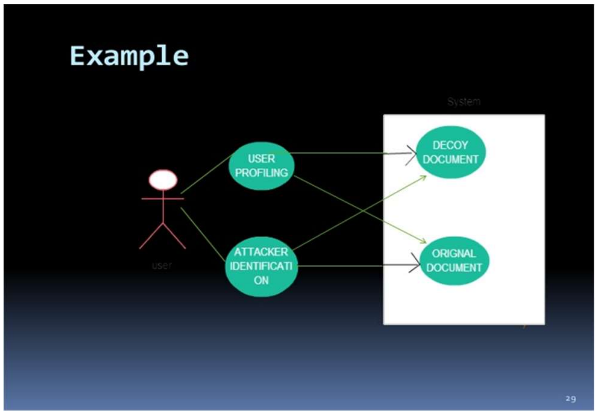

Figure 4 [12]

E. *Decoy Technology implemented SMS*

In this section, we elaborate enhancement in security using decoy proposed by Sonali Khairnar et al. with implementing SMS in decoy technology. In their suggestion, if the administrator or user wants to download any file from fog, he or she must answer the provided security question correctly (those are previously set by legitimate user at the time of very first login). If not, the content in the document would be replaced (with decoy document).

For instance, suppose the word “madam”, if the monitored access is identified as illegitimate access, the alphabet “m” would be replaced by “a”, and the word turns to “aadaa”. Though the attacker knows the principle “m->a” and applies reverse engineering, he will get the result as “Mmdmm” [13].

Sonali also suggested the best thing of Fog computing is after successful login the user can get alert by SMS on the mobile. If it is the access from masqueraders, the user can get SMS that contain attacker details like IP or server name. The approach proposed considered the threat from malicious insider, ensured that attacker can’t judge the correct content of replaced document. And the details disclosed of attackers make it easy to trace them. The approach makes Fog computing more secure than traditional Cloud computing.

VII.CHALLENGES

A. *Beacon Challenges*

The beacon needs to collect the attacker’s information stealthily without triggering any alerts. Current technology can collect the attacker’s IP address, OS version and the decoy document path, however the issue is that it triggers a document reader warning and an anomaly networking alert as well. It is also difficult to currently distinguish between a legitimate user and an attacker.

B. *Mark Challenges*

The mark needs to allow the decoy document to be distinguishable from the authentic document while still being able to trick the attacker. The mark is generated by the HMAC algorithm (128 bit) and is added to all decoy documents in a specific position of the header. This makes the mark easy to identify using decoy document scanning tools and therefore exposing the document anti-theft mechanism.

C. *File Content Challenges*

Currently the decoy documents are generated using a fixed file template that might result in a document that is very different from the authentic document therefore the document is unable to deceive the attacker.

D. *Decoy Document Deployment Challenges*

The assumption while deploying the decoy documents is that legitimate users will be familiar with the contents on their file system and therefore will not be attempting to access the decoy documents unlike an attacker. B. Bowen et al. researched on the usage of decoy documents and proposed methods for the generation and distribution of the decoy documents. The 8 properties of a perfect decoy document are:

1. Believability
2. Conspicuous
3. Enticing
4. Differentiability
5. Non-interference
6. Detectability
7. Variability
8. Shelf-life

Based on this research the ADAMS project developed the DDT (Decoy Distributor Tool) which can complete the deployment in 4 simple steps:

1. Selecting decoy document source
2. Selecting root distribution directory
3. Naming decoy documents as well as their directories
4. Modifying decoy document time

It has been found that DDT focuses on conspicuousness and non-interference, without paying enough attention to limited shelf life and naming variability. This may result in a reduction in the number of attackers getting deceived in a long distribution time.

E. Unidirectional Transparent Detection Challenges

Once the decoy documents are deployed, fog computing based anti-theft system installs the Decoy Document Access software to monitor any access to the decoys. However advanced attackers can locate and shutdown the DDA processes while at the same time legitimate processes may trigger false positive alerts (e.g. antivirus scanning processes) [1].

VIII.IMPROVEMENTS

The beacon needs to be variable and the hidden network processes need to be regularly updated to avoid detection by the attacker's defense software. Legitimate users can be distinguished from attackers through user behavior modelling to detect abnormal behavior. With regards to the mark, it should be positioned in a variable location to avoid easy detection by the attacker.

Instead of using a generic template for the decoy document creation, the system administrator can be involved to ensure that the decoy matches the authentic documents. Also, the documents should be deployed in directories which the attackers are most likely to try to access. Also, a valuable information keyword list provided by the administrator should be used and the files should be sorted according to inverted correlation to ensure that there is a higher probability that the attackers open the decoy document first.

A workaround to ensure that the DDA processes are not shut down is to have a guarding process in the background that restarts the DDA proves if it is shut down maliciously. False alarms can be reduced by regular scanning and whitelisting of the legitimate processes such as the anti-virus scanning software [1].

IX.CONCLUSION

In a nutshell, this project aims at discussing the security issues that occur in fog computing. There are several security issues that are common and are hard to solve in general as well as in an environment like fog computing. One of the most recent technologies that we have focused on in this paper is the decoy technology. As already described in detail, these are kind of fake documents with certain properties that entices the attacker to access them and with the help of this accessing of fake file the attacker and his intentions can be caught. A lot of researchers and engineers are still working on this area to discover more and to come up with an error free solution. Though the idea of decoy technology is very fascinating, there is still a huge amount of work that needs to be done to overcome the limitations of this technique.

REFERENCES

[1] Fei Huayu, He Jun, Wang Menglin, Research on fog computing based active anti-theft technology, Procedia Computer Science, Volume 111, 2017, Pages 209-213, ISSN 1877-0509, (http://www.sciencedirect.com/science/article/pii/S1877050917312280).

[2] IoT Agenda. (2018). What is fog computing (fog networking, fogging)? - Definition from WhatIs.com. [online] Available at: https://internetofthingsagenda.techtarget.com/definition/fog-computing-fogging [Accessed 11 Nov. 2018].

[3] Brandon Butler on what is fog computing? Connecting clouds to the things in network world, IDG press. (https://www.networkworld.com/article/3243111/internet-of-things/what-is-fog-computing-connecting-the-cloud-to-things.html).

[4] Win system article on fog computing, knowledge hub, tech library on March 28, 2018. (https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/).

[5] Bonomi, Flavio; Milito, Rodolfo; Zhu, Jiang; Addepalli, Sateesh (2012-08-17). "Fog computing and its role in the internet of things". ACM: 13–16. doi:10.1145/2342509.2342513. ISBN 9781450315197. (https://dl.acm.org/citation.cfm?id=2342509.2342513).

[6] Yunguo guan, Jun Shao, Guiyi Wei, Mande Xie on data security and privacy on fog computing in IEEE Network (Volume: 32, Issue: 5, September/October 2018) (https://ieeexplore.ieee.org/abstract/document/8315211).

[7] M. Mukherjee et al., "Security and Privacy in Fog Computing: Challenges," in IEEE Access, vol. 5, pp. 19293-19304, 2017. doi: 10.1109/ACCESS.2017.2749422 (https://ieeexplore.ieee.org/abstract/document/8026115).

[8] Technopedia article on Masquerade attack (https://www.techopedia.com/definition/4020/masquerade-attack).

[9] S. Virushabadoss, Dr.C. Bhuvaneswari on Analysis of Behavior Profiling Algorithm to Detect Usage Anomalies in Fog Computing in International Journal of Engineering Science Invention (IJESI) (http://www.ijesi.org/papers/NCIOT-2018/Volume-4/3.%2014-19.pdf).

[10] Malek Ben Salem, Salvatore J. Stolfo on Decoy Document Deployment for Effective Masquerade Attack Detection in International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment 2011. (https://link.springer.com/chapter/10.1007/978-3-642-22424-9_3).

[11] Mahajan, H. (2014). 'Threats to Cloud Computing Security' International Journal of Application or Innovation in Engineering & Management, Special Issue for International Technological Conference-2014 (IJAIEM), (http://www.ijaiem.org/ITechCON-2014/CMPN-01.pdf).

[12] Hosseinpour, Farhoud & Vahdani Amoli, Payam & Plosila, Juha & Hämäläinen, Timo & Tenhunen, Hannu. (2016). An Intrusion Detection System for Fog Computing and IoT based Logistic Systems using a Smart Data Approach. International Journal of Digital Content Technology and its Applications. 10. (https://www.researchgate.net/publication/312868380_An_Intrusion_Detection_System_for_Fog_Computing_and_IoT_based_Logistic_Systems_using_a_Smart_Data_Approach).

[13] Y. Wang, T. Uehara and R. Sasaki, "Fog Computing: Issues and Challenges in Security and Forensics," 2015 IEEE 39th Annual Computer Software and Applications Conference, Taichung, 2015, pp. 53-59. doi:10.1109/COMPSAC.2015.173 (https://ieeexplore.ieee.org/abstract/document/7273323).

[14] Gayatri Kalaskar, Purva Ratkanthwar, Prachi Jagadale, Bhagyashri Jagadale, "FOG Computing: Preventing Insider Data Theft Attacks in Cloud Using User Behavior Profiling and Decoy Information Technology", International Journal of Engineering Trends and Technology (IJETT), Volume 32, Number 7, February 2016.

[15] Cloud, Fog And Edge Computing – What's The Difference? (https://www.winsystems.com/cloud-fog-and-edge-computing-whats-the-difference/)