Fraudulent Cloud Resource Consumption Attacks

Deepak Surana
12BCE018

NIRMA
UNIVERSITY

DEPARTMENT OF COMPUTER ENGINEERING
AHMEDABAD -382424

October 2015

Fraudulent Cloud Resource Consumption Attacks

Minor Project

Submitted in partial fulfillment of the requirements

For the degree of

Bachelor of Technology in Computer Science and Engineering

by

Deepak Surana
12BCE018

Under the Guidance of

Vivek Kumar Prasad

DEPARTMENT OF COMPUTER ENGINEERING
AHMEDABAD -382424

October 2015

CERTIFICATE

This is to certify that the Minor Project entitled *Fraudulent Resource Consumption attacks* submitted by Deepak Surana (12BCE018), towards the partial fulfillment of the requirements for the degree of Bachelor of Technology in B.Tech of Nirma University, Ahmedabad is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this Project, to the best of my knowledge, havent been submitted to any other university or institution for award of any degree or diploma.

Prof. Vivek Kumar Prasad
Assistant Professor
Computer Science and Engg.
Institute of Technology
Nirma University
Ahmedabad

Dr. Sanjay Garg
Professor and Head of Department
Computer Science and Engg.
Institute of Technology
Nirma University
Ahmedabad

Acknowledgements

I feel the need to express my sincere gratitude to each one of the people who helped us wholeheartedly to complete this report. A heartfelt vote of thanks for my guide **Prof. Vivek Kumar Prasad** whose constant and sincere support and guidance helped me to manage this undertaking efficiently. Wrapping things up, I also reserve my genuine gratefulness to my colleagues and peers for their help and support.

DEEPAK SURANA
12BCE018

ABSTRACT

A key feature that helps the adoption and expansion of public cloud computing services is its utility-pricing model, which calculates the cost based on resource utilisation. It inspires from the public utilities, such as electricity, gas and telephone, which bills the consumer based on the consumption. In the similar manner, cloud service users need to pay for the resources, such as, storage, CPU cycles, network bandwidth, computational hours, etc. based on the amount they utilize and for the period of time they use consume. As per the terms of agreement with the cloud service provider (CSP), every cloud consumer is accountable for all the resource consumption during their leased hours and need to pay for all the cost of leasing cloud services regardless of their intent behind leveraging resource.

As the downside of this utility-pricing model, web-based services hosted by cloud are open to Fraudulent Resource Consumption (FRC) attacks. It is intended to disrupt the financial expenses of operating in a cloud environment by extended use of its utility-pricing model. It makes this possible by fraudulently consuming the cloud services in considerable volume and in turn incurring significant fraudulent cost to the suffered consumer.

This report presents Anomaly Detection as a method to detect and encounter these FRC attacks. Anomaly Detection is the process of identifying any unexpected event that doesn't match with the normal profile of a user. These unusual behaviors can possibly a attack, thus demands a check. The anomaly detection mainly consists of two phases : training phase and detection phase. The labelled data is used to train the system about the user profile. Firstly, K-Means Clustering is implemented on the initial data in order to group them into multiple clusters. A probability based state-transition model, Hidden Markov Model (HMM), takes the next amount of resource consumption as input and return the probability with which it is possible. On comparing with the threshold probability, it is either categorized as a normal use or a anomaly, i.e. a possible attack.

Contents

Chapter 1

Introduction

Cloud Computing mutually benefits both providers and users by bringing economies of scale. It makes high and optimal utilization of resources possible from the perspective of provider whereas it means zero capital costs from the eye of consumer. The pay-as-you-go model requires user to pay only for the resources acquired by them and that too for the time they used them. This utility model seems attractive to a cloud user because of its low pricing and eliminating the need for any capital investment. However, this doesn't mean it is free of risks. This model too has its own security threats- financial accountability for all the resources consumed and it is unlimited. That is, a user has to pay the total cost of consumption regardless of the intent of use.

In case of a Fraudulent Resource Consumption attack, the cloud consumer is billed each time a cloud service is used by the attacker on behalf of the victim user. A extensive use of the resources can be too costly. A botnet consume cloud-based web resources by impersonating authorized user behavior. For a Web application, each incoming request is regarded the same way, the intent of the request is not a point of concern. Thus, each request is served with the respective reply, results in incurring the cost accordingly.

Fraudulent Resource Consumption's prevention, detection, auditing, and mitigation makes network traffic analysis a important task in cloud management. The anomalies are needed to be detected based on network analysis. The known network traffic occurred in a cloud computing system will be used to build up the normal behavior profile of a user. Then, deviations from these profiles will be considered as anomalies and will be considered as a possible threat.

Anomaly Detection is the process of identifying unusual behavior. Anomalies are patterns during network analysis that do not conform to a well-defined notion of normal behavior. There are situations where enormous impact can be made by identifying anomalies in a timely and right manner, e.g. early detection of fraudulent cloud resource consumption can prevent financial loss to the legitimate user. Its applications has generated the interest for designing of efficient methods of detections anomalies and in turn anomaly detection systems. Anomaly detection systems, a subset of intrusion detection systems, model the typical system/network behavior which make them extremely effective in finding and foiling both known as well as unknown or zero day attacks. Firstly, they are capable of detecting insider attacks by measuring the deviations from normal behavior. Secondly, the detection systems are custom made for each user profile thus it becomes very difficult for a fraud user to take out any movement without setting off an alarm. Finally, they are able to detect anomalies that are previously not known to the system.

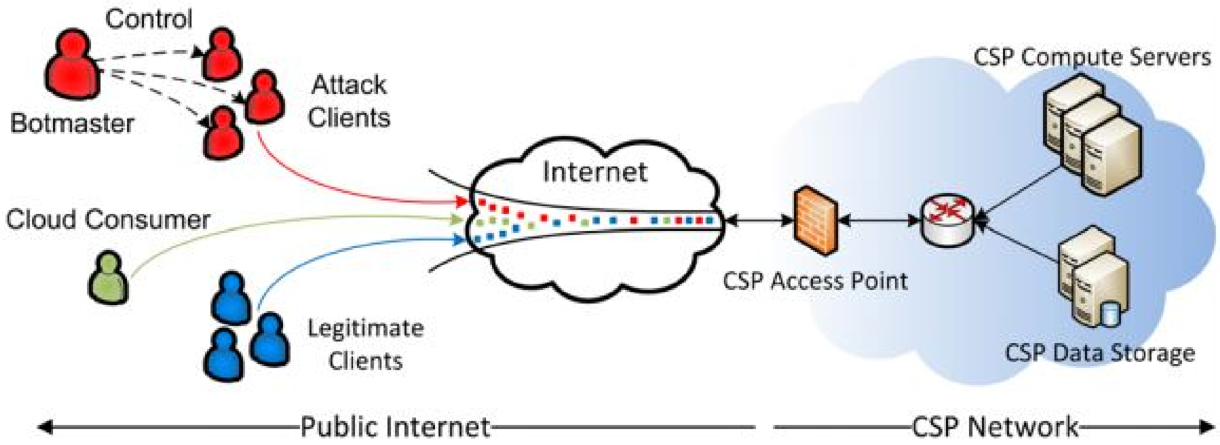


Figure 1: Cloud FRC Attack Illustration

Chapter 2

Security in Cloud

A cloud is liable to a few coincidental and deliberate security threats, including vulnerability to the confidentiality and availability, integrity, information and infrastructure. Additionally, when a cloud with huge computational ability and storage capacity is wrongly used by a poorly intentioned party for malicious purposes, the cloud possesses a danger against society. Insiders and external intruders creates intentional threats. Insiders are authenticated cloud consumers who misuse their permissions and privileges to use cloud services for the unintended purposes. An intrusion is a attack that exploits the possible security flaws and performs a subsequent breach that violates the security policy of the system. Although, an intrusion implies a successful attack, Intrusion Detection Systems do try to identify that don't lead to security compromise. The intrusion can affect the network infrastructure, operating system daemons, cloud applications, cloud middleware, etc.

2.1 Types of Intrusions

Different cloud intrusions can be classified as:

1). Unauthorized Access: A break-in conferred by an intruder that takes on the appearance of a genuine cloud user. This becomes possible by stealing the login credentials, brute-force all combinations, a conjecture, or a careless user exposes it.

2). Misuse: This may be an outcome of an unapproved access or the misuse of benefits by a legal client (insider) and this behavior is mostly anomalous to the normal profile.

3). Cloud Attack: These attacks are carried out using tools and by harming the script that correspond to vulnerabilities exist in cloud services, protocols and applications. They may show up in the form of Denial of Service (DOS) Attack, Malware Injection attacks, etc. and can effect the cloud infrastructure present at several locations.

4). Data Security: Users are relieved by outsourcing data on to the cloud, but Data on a Cloud is stored at different geographical locations and is liable for different intrusions, thus data integrity maintenance is challenging.

5). Flash Crowds: Sudden increment in the traffic generate by authenticated users. Cloud computing frameworks are utilized by numerous customers, along these lines, they produces enormous amount of logs. This huge auditing is hard to analyse and very much time consuming for a IDS. This eventually decreases framework efficiency. Rate of intrusions are mostly higher with noteworthy consequences and damages.

2.2 Security Methods

For maintaining security, Cloud framework basically implements the following:

1. Encryption: It encodes the data in such a way that only concerned persons can read this. It uses keys in order to encrypt and decrypt the information which is available with sender and receiver only. It must be based on some accepted norms for both structured as well as unstructured data.

2. Contextual Access Control: It allows access to the corporate data stored on a cloud based on user, device and geographical location. Thus, tries to limit the access through a random entity.

3. Application Auditing: It keeps track of cloud usage for individual users and detects them for any possible anomalous behavior. In case of any anomaly, it alerts the system about any possible threat.

4. Data loss prevention : It is a strategy that blocks the migration of sensitive and critical information. It keeps track of where your data is moving. Thus, prevents any exposure of the critical data to the public.

5. Extend Cloud Security: It is not only about mobile-to-cloud security and office-to-cloud security, the cloud-to -cloud matters too. It includes governance and security policies as data moves between different cloud services.

Chapter 3

Fraudulent Resource Consumption Attack

The Fraudulent Resource Consumption is the downside of the utility pricing cloud model. It exploits the pay-as-you-go pricing strategy by fraudulently consuming cloud services and in turn make the legitimate customer accountable for the uses and the billing amount. This attack is more subtle than DDoS and is carried out for longer period of time. It incurs a cost for the victim each time the intruder uses the application. With the increase in usage, it becomes too costly for the victim to sustain and that too with no associated business value.

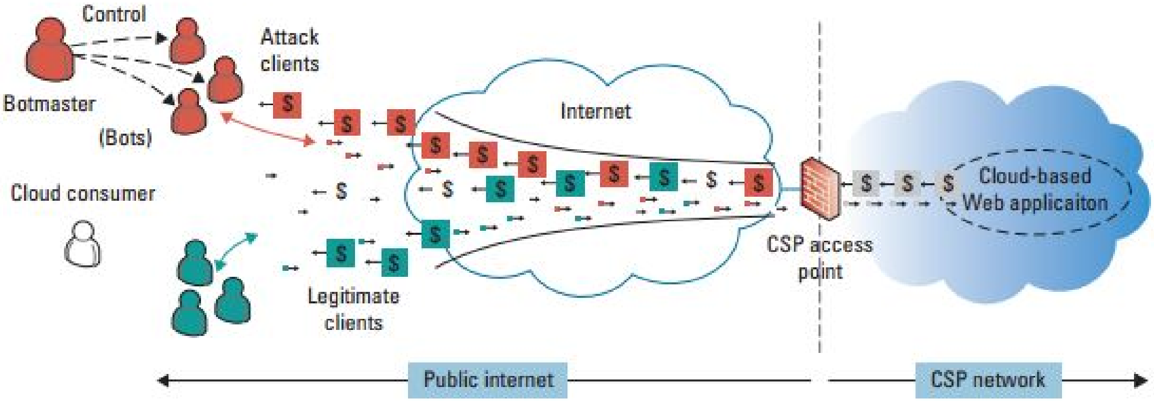


Figure 2: A cloud network-attack diagram. Botnets can exploit the cloud utility model to perform fraudulent resource consumption (FRC), making consumers incur unexpected costs from dishonest use

The attack formulates the proper mimicking of legitimate user's resource consumption profile. It then performs the similar behavior in considerable volume for a longer period of time that accounts for high costs unsustainable for the cloud user. One of the intention of the intruder in such a attack is to hide behind the shadow of legitimate activities in order to keep their malicious intents undetected. From the victim's point of view, these attacks become unsustainable when business objectives are failed regardless of the considerable cost paid.

The FRC is known as *free-rider problem* in the field of economics, the unlimited access to the cloud services to a attacker via the Internet in which not the requesting users, instead the legitimate cloud customer is responsible for the cost of this excessive resource consumption. In the context of economics, this unrestrained consumption leads to market failure for the victim and abandonment of the public cloud model.

3.1 Fraudulent Resource Consumption

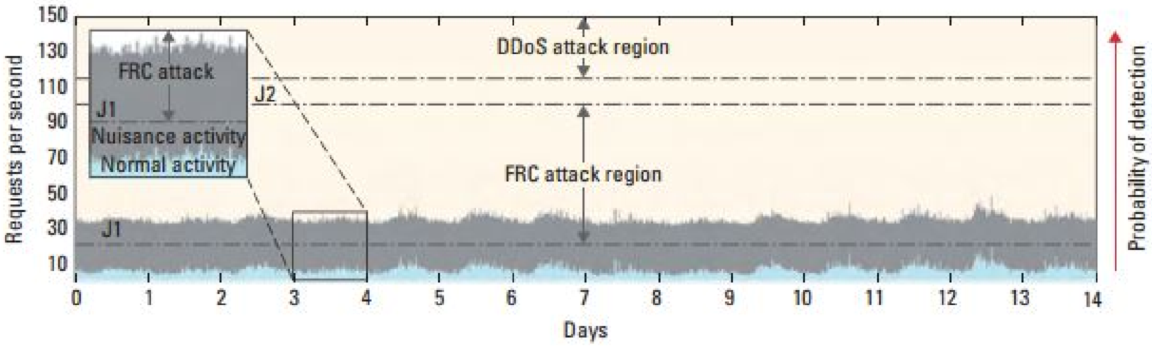


Figure 3: Division of Malicious Behavior. The initial attack intensity (J1) is insignificant in terms of cost for a cloud service user. However, as intrusion activity intensifies and enters FRC region beyond this region of nuisance activity, the consumption cost incurred to the consumer is a concern.

Talking about an FRC attack,let us consider the visualisation of a Web server log as shown in Figure 3. The y-axis shows the number of requests/second and the x-axis depicts a two-week period. The server capacity is sufficiently over-provisionedthis represents an orthodox estimate, given the capacity of these servers. Placed on top of normal Web activity are requests from an FRC attack. As shown in the figure, the starting attack intensity beyond the usual activity is in the region known Nuisance activity, because the costs have no relevance to the cloud consumer. As harmful activities increase, the cost to consumer also increases. The quality of service will degrade heavily if attack increases over J2. But one thing that is positive from an increasing attack intensity is that detection becomes easier. From the perspective of an attacker a modest attack spread out over a large period would guarantee better results.This is a dangerous situation because the current technology is not apt for these type of attacks. Hence with the arrival of this utility pricing model, and in turn introduction of FRC attacks changes the fraud detection mechanism for cloud.

Chapter 4

Anomaly Detection

Anomaly Detection is the process of identifying unusual behavior. Anomalies are patterns or events that do not matches with a well-defined notion of normal behavior.These non-conforming patterns are often termed as anomalies, outliers, exceptions, aberrations, discordant observations, surprises, peculiarities or contaminants in numerous application domains.

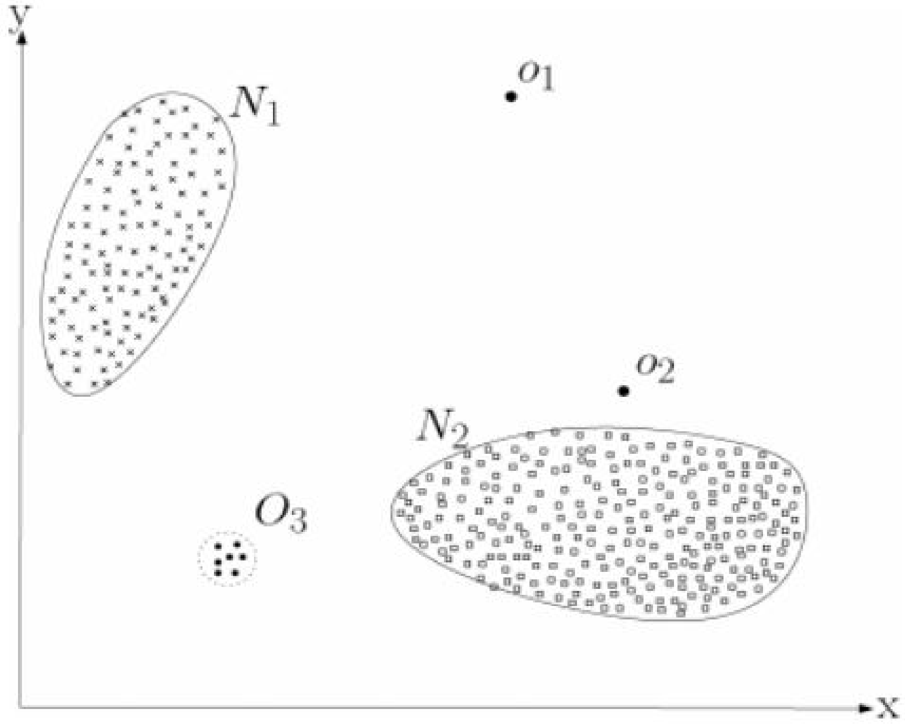


Figure 4: Anomalies are shown in a two-dimensional dataset.

Figure 4 depicts anomalies in a 2D-dataset. It illustrates two regions N1 and N2 that counts for most of the points, and represents the normal regions, whereas points o1, o2 and points in regions 03, that lies far from these concentrated normal regions are regarded as anomalies. These unexpected behaviors cannot always be categorized as attacks but they can just be the events previously not known. Thus, may or may not be possible threat.

Anomaly detection system is modelled on the normal data and may or may not contain the previously known anomalies dataset, it then looks for any aberrations in the observed data with respect to the model designed. With the input of available dataset as a training set, and given a new test entry, the objective of the system is to determine whether that new entry imitates the ”normal” pattern or corresponds to some ”anomalous” event. These systems do have an advantage of detecting any new type of anomaly by considering deviation from normal profile. However, they do have high false negatives.

4.1 Problem Statement

To terminology of normality is required to represent the issue of anomaly detection in any framework. A formal model inculcates the ”normality” by defining relationship between the key parameters included in the system. Therefore, an activity is classified as an anomaly in light of the fact that its level of deviation with respect to the profile of characteristic behavior, as depicted by the model, is sufficiently high. Formally, an anomaly detection system A is defined as A = (M,D), where M defines the normal profile of resource consumption and D is a similarity index which outputs the degree of deviation with regard to the trained model M when a input activity record is given. In this way, the elementary perspective of the system comprises of two fundamental modules: the modelling module and the detection module. The modelling subsystem comes to the picture during the training phase and scans through the available resource consumption activities in order to train and finally result a model M which imitates the normal user resource consumption profile. This resulted model is presented to the detection subsystem to check the upcoming new amount of consumption to get the deviation related to the trained model. These two operations are generally carried out independently. Moreover, user behavior do evolves and, along these lines, the model must be updated periodically so that the model can adapt to the new environment.

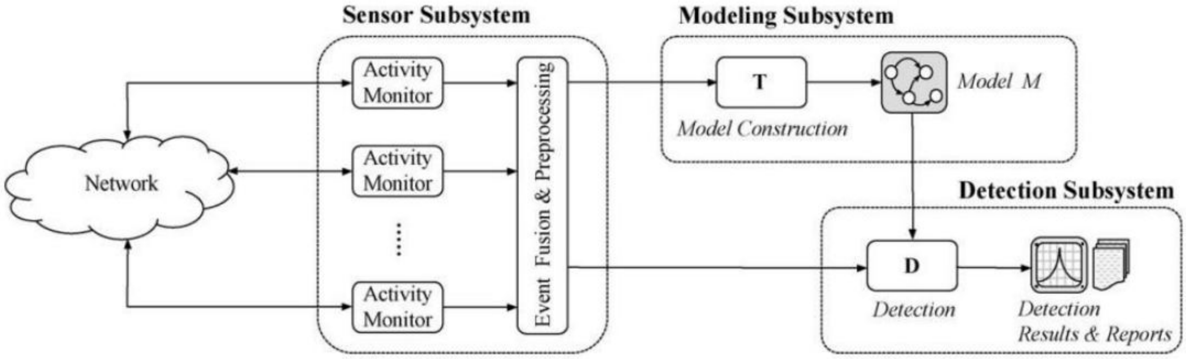


Figure 4: Architecture of a typical anomaly detection system.

4.2 Evaluation of Anomaly Detection

The fundamental thought behind anomaly detection is that intrusive activities are the activities that are anomalous. In case of intruder who is unknown to the users profile trends, probability of detecting this intruders activity as anomalous is very high. Although in ideal circumstances, all anomalous activities corresponds to intrusive events. This means that they will not be any case of false positives and false negatives.But in real-life it is not always the case that intrusive activities coincide with anomalous activities. As a result there can be four possibilities as follows, all with some probability:

1). Intrusive but not anomalous: This case is referred as false negatives as the detection system falsely denied the presence of intrusion. The intrusion detection system fails to detect this intrusion because it was not found anomalous.

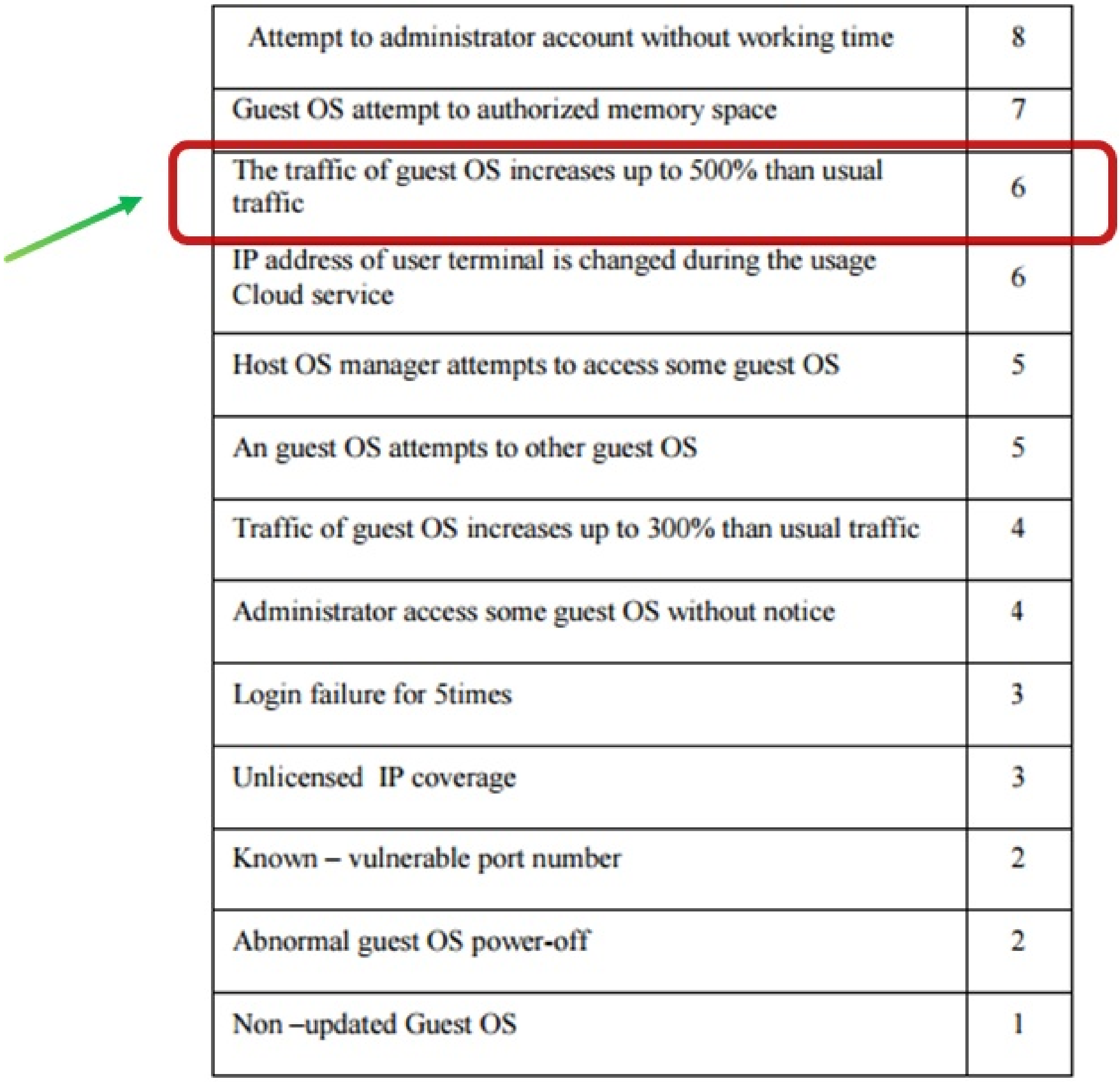
2). Not intrusive but anomalous: This case is referred as false positives as the detection system falsely reports the presence of intrusion. In true, it is not an intrusive activity, but it is reported as anomalous so in turn the IDS detected this as intrusion.

3). Not intrusive and not anomalous: This case is referred as true negatives as neither the activity is intrusive nor it is reported as intrusive as it non-anomalous.

4). Intrusive and anomalous: This case is referred as true positives. It is the most desired case as the activity is actually intrusive and since it is anomalous too, it is also reported as intrusive.

To minimize false negatives, lower threshold values are set to define anomalies so that more anomalies can be captures. But, this results into larger number of false positives and reduces the efficacy and increases the overhead of investigating each such incident and discard false positive cases. Thats why it leads to a trade-off between detection rate (minimize false negative, thus increase the chance of detective anomalies) and false alarm rate (higher false positives, overhead of checking).

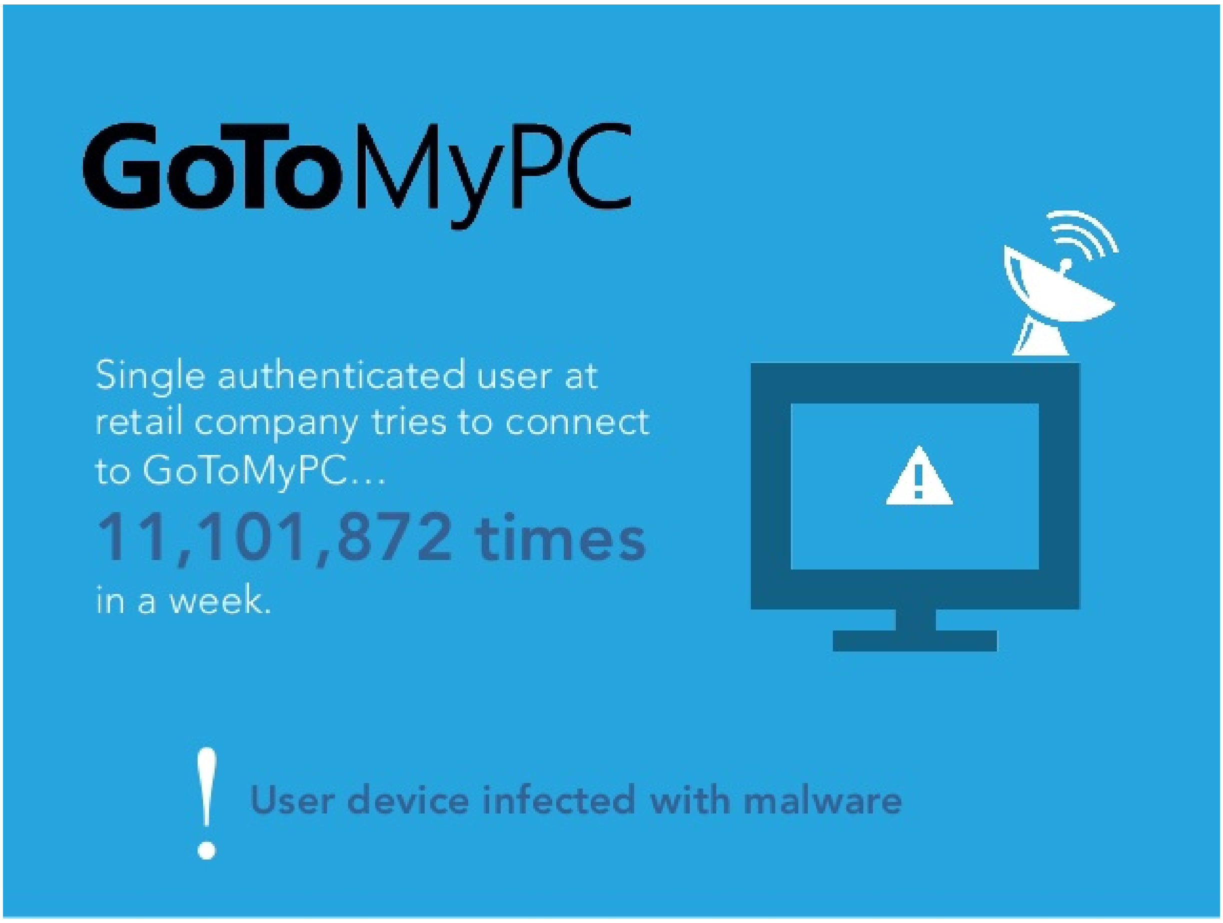
4.3 Evaluation of User Anomaly Level



Attempt to administrator account without working time	8
Guest OS attempt to authorized memory space	7
The traffic of guest OS increases up to 500% than usual traffic	6
IP address of user terminal is changed during the usage Cloud service	6
Host OS manager attempts to access some guest OS	5
An guest OS attempts to other guest OS	5
Traffic of guest OS increases up to 300% than usual traffic	4
Administrator access some guest OS without notice	4
Login failure for 5times	3
Unlicensed IP coverage	3
Known – vulnerable port number	2
Abnormal guest OS power-off	2
Non –updated Guest OS	1

Figure 5: Different Levels of Anomalies based on threat they possess.

4.4 Real-Life Examples of Anomalies



GoToMyPC

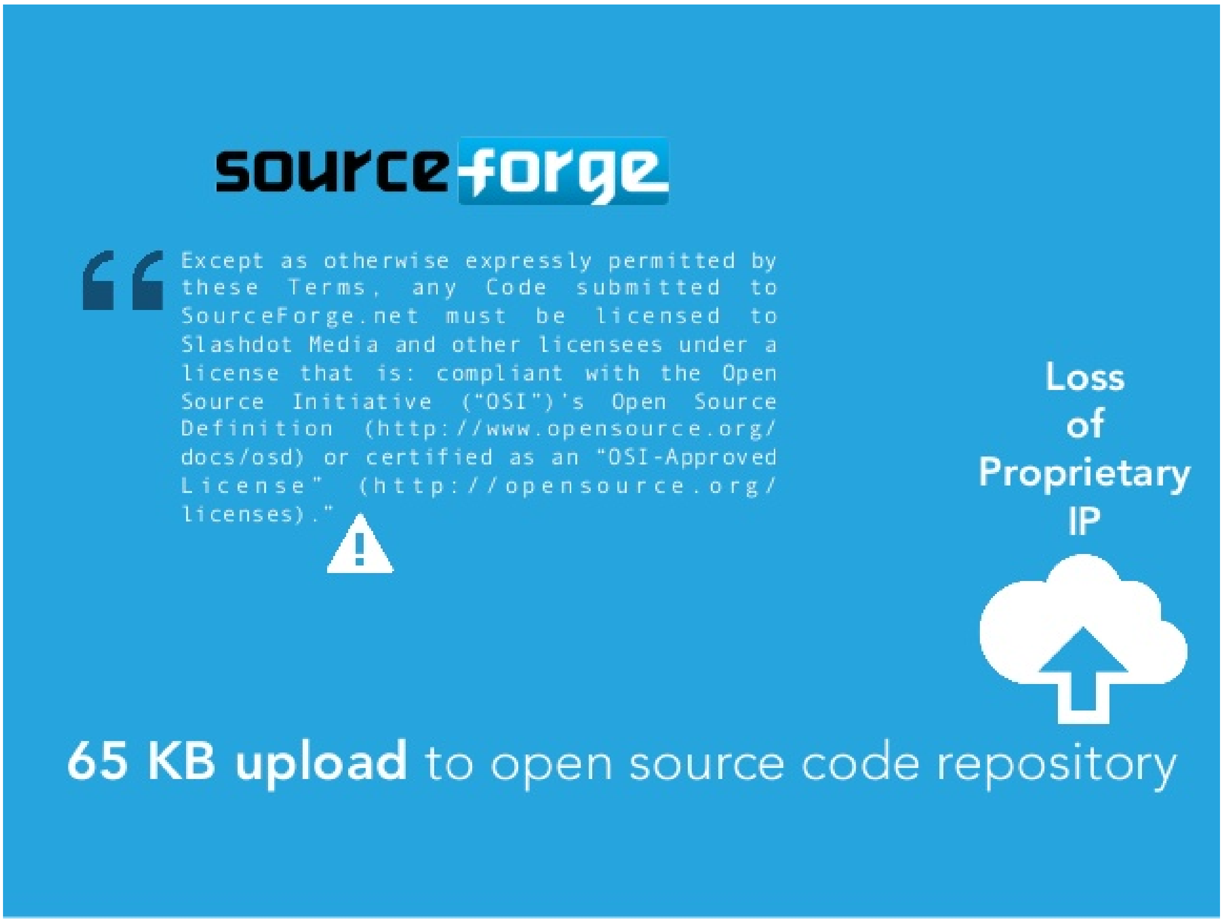
Single authenticated user at retail company tries to connect to GoToMyPC...

11,101,872 times

in a week.

! User device infected with malware

1. Uses increases as compare to normal.



sourceforge

"Except as otherwise expressly permitted by these Terms, any Code submitted to SourceForge.net must be licensed to Slashdot Media and other licensees under a license that is: compliant with the Open Source Initiative ("OSI")'s Open Source Definition (http://www.opensource.org/docs/osd) or certified as an "OSI-Approved License" (http://opensource.org/licenses)."

Loss of Proprietary IP

65 KB upload to open source code repository

2. Uses decreases as compare to normal.

Chapter 5

Fraud Detection Model

5.1 Traditional Systems

1). The traditional fraud detection systems includes a training phase that inputs the available labelled data of cloud resource consumption. They do contain consumptions with fraud cases may be because of stolen authentication, lost credentials, application fraud, mail-order fraud or counterfeit fraud.

2). A metaclassifier is then trained by using the correlations highlighted by predictions of base classifiers.

3). The real-life data is used as a representation of fraud density and confidence value to generate the relative fraud score for new data to reduce number of False Positives and True Negatives.

But the problem lies in these are:

a). Classifier's training requires the labeled data for both genuine and fraudulent consumptions.Availability of real-world labelled fraud data is a worry in these type of detection systems.

b). Since, they detect based on the trained data, they cannot detect any new type of fraud consumption for which the labelled data is not introduced.

In contrast, a Hidden Markov Model (HMM) based resource consumption Fraud Detection System, doesn't want fraud signatures and still these model can look for frauds by considering a users consumption behavior.

HMM based systems also improves on reducing the number of False Positives consumption that are identified as intrusion but they are genuine infact.

5.2 Model for Fraudulent Resource Consumption Detection

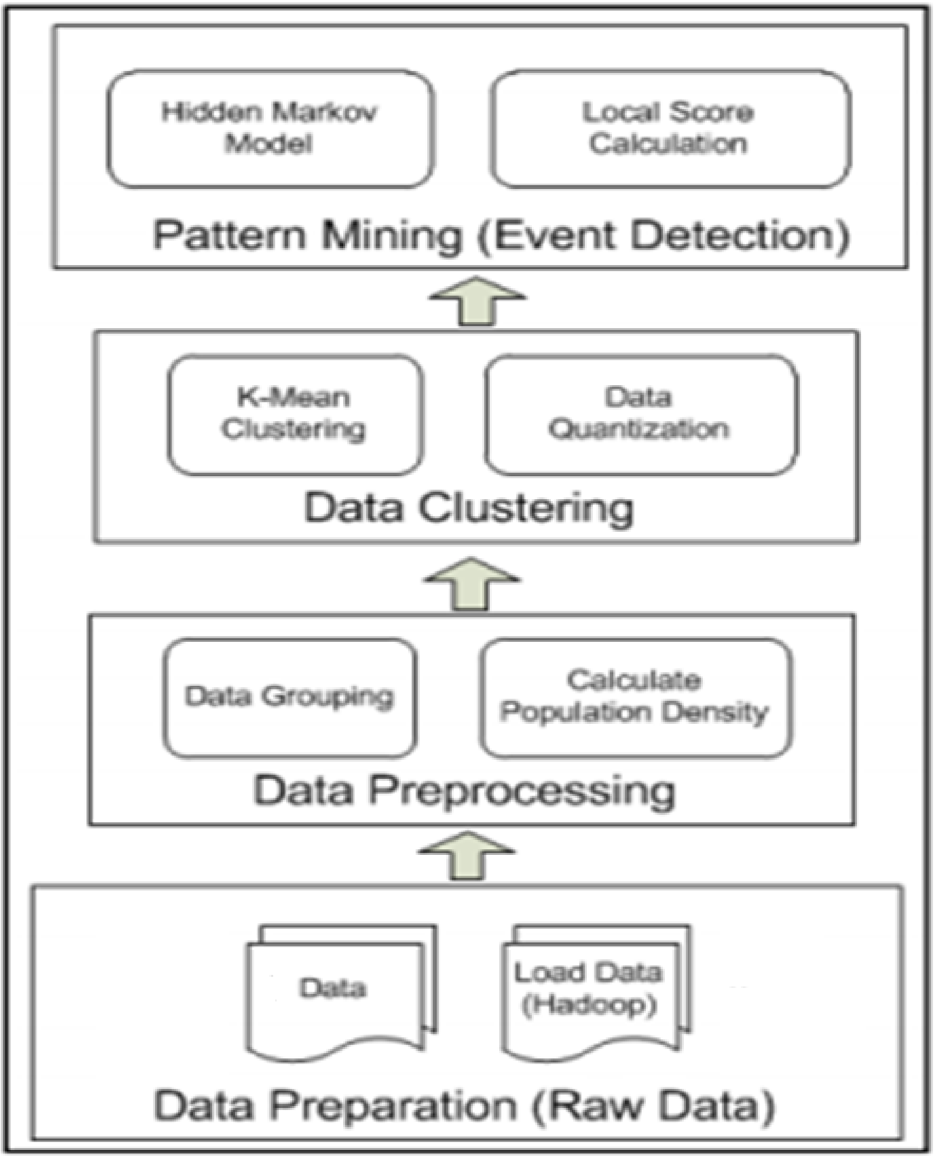


Figure 6: Fraud Detection Model

The model is based on feeding the system with the available resource consumption data in order to build the user's normal behavior profile. This data will be the real-world consumption data. We are using Hadoop as a platform to implement this model. Firstly, this training data set is loaded onto Hadoop. Grouping is done on the whole data and the respective population density is measured.

We are using K-Means Clustering in order to form different clusters from the data available. Clustering is a process where we group data objects into unconnected clusters so that the data in one cluster are similar, but data of any other two clusters will differ. A cluster is a collection of similar data objects belonging to the same cluster and non-similar to the other clusters. Data Quantization is the performed to classify them into different categories. The resource utilization will be categorized under three labels or observatiion symbols : high consumption, medium consumption and low consumption.

Hidden Markov Model (HMM) which is a probability based state-transitional model, is used to determine the transactional probability between different states. Inspired by cloud service model, the states are categorized as : SaaS, PaaS and Iaas. The initial probability of state transition and observation symbol can be detemined by running Baum-Welch algorithm. This algorithm uses forward-backward reasoning to detemine the unknown parameters of Hidden Markov Model.

A threshold probability is already set according to the need. The output probability of HMM is compared with this threshold probability. In case of it exceeding the threshold, it represents a anomalous behavior deviating from the normal user profile. It may be a case of probable fraud resource consumption. So, a alarm is raised to alert the system for the respective resource consumption and a suitable action can be taken.

Chapter 6

Hidden Markov Model and K-Means Clustering

The word ”hidden” in the Hidden Markov Model is meant for the sequence of states through which the model goes, and not to the state transition probabilities of the model.

The Model imitates the cloud service user behavior in terms of states and different transition probabilities, then anomalies are detected which tells about the possible attack. In these cases, a alarm is set up to alert about the deviation.

Hidden Markov Model characterizes:

(a) There are N states in the model denoted as S = S1, S2, , SN. The state at any time ’t’ is denoted by qt.

(b) There are M distinct observation symbols per state. The set of symbols are represented as V = V1,V2,..,VM).

(c) Probability matrix for different State transitions, A=[aij], where

$$a_{ij} = \mathrm{P}(q_{t+1} = S_j | q_t = S_i), 1 \leq i \leq N, 1 \leq j \leq N; t = 1, 2, \ldots.$$

(d) Probability matrix of different observation symbols, B=[bj(k)], where

$$b_j(k) = \mathrm{P}(V_k | S_j), 1 \leq j \leq N, 1 \leq k \leq M \text{ and}$$

$$\sum_{k=1}^{M} b_j(k) = 1, 1 \leq j \leq N.$$

(e) The initial probability vector = [i], where

$$\pi_i = \mathrm{P}(q_1 = S_i), 1 \leq i \leq N, \text{ such that } \sum_{i=1}^{N} \pi_i = 1.$$

(f) O = O1, O2, O3,.., OR,is the observation sequence in which Ot corresponds to symbols in V, and R is the total count of observations making up the sequence.

Now, Consider a example sequence

$$\mathrm{Q} = \mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_{\mathrm{R}},$$

Now the probability that this state sequence generates O is,

$$\mathrm{P}(\mathrm{O}|\mathrm{Q}, \lambda) = \prod_{t=1}^{R} \mathrm{P}(O_t|q_t, \lambda),$$

It can be expanded as,

$$\mathrm{P}(\mathrm{O}|\mathrm{Q}, \lambda) = \mathrm{b}_{q_1}(\mathrm{O}_1).\mathrm{b}_{q_2}(\mathrm{O}_2)\ldots\mathrm{b}_{q_R}(\mathrm{O}_{\mathrm{R}}).$$

Probability of having a state sequence Q is given by,

$$\mathrm{P}(\mathrm{Q}|\lambda) = \pi_{q_1}\mathrm{a}_{q_1q_2}\mathrm{a}_{q_2q_3}\ldots\mathrm{a}_{q_{R-1}q_R}.$$

Thus, Probability of generation of the observation sequence O by the HMM = all Q (Probability of generating O from a state sequence Q) (Probability to have a state sequence Q). i.e

$$\mathrm{P}(\mathrm{O}|\lambda) = \sum_{all\ Q} \mathrm{P}(\mathrm{O}|\mathrm{Q}, \lambda)\mathrm{P}(\mathrm{Q}|\lambda).$$

HMM Model for Users Cloud Resource Consumption:

The cost incurred is quantized into M cost labels V1, V2, ..., VM, which becomes the observation symbols for each state. The actual labels for a particular expense range is derived based on the consumption profile of each cloud service user.

In our case, the cost incurred is quantized as low cost(l), medium cost(m) and high cost(h). The observation symbol set V is ”l,m,h”. In a real world scenario, the user profile is more consistent in the sequence of types of service utilization as compared to the sequence of consumption.

An unsupervised learning algorithm, K-Means is used to form groups out of a given input data by considering their similarity. This is achieved by associating individual data points to cluster with which its distance to centroid is minimum. These centroid values are the used to determine the cluster label for a next resource consumption input.

The resource utilization profile of a cloud service user corresponds to his normal behavior. This is categorized into high-consumption(hc), medium-consumption(mc) and low consumption(lc).

Consumption profile reflects the cluster to which most of the amount of consumption of the client belongs.

Initialy, each state is equally likely, so with N states in our model, the initial probability of each state will be 1/N. Three sets of initial probability for observation symbol generation for three consumption groups: lc, mc, and hc. Based on the users consumption profile, initial observation probabilities at each state is decided.

Let,

$$O_{1,}\ O_{2,}\ O_{3,}\ .\ .\ .\ O_{R}$$

be a sequence of observation symbols of length R formed from the resource utilization activity up to time t whose probability of acceptance by the Hidden Markov Model will be:

$$\alpha_1 = P\ (O_{1,}\ O_{2,}\ O_{3,}\ .\ .\ .\ O_R \,|\, \lambda\ )$$

Now, the next input amount of consumption is denoted by OR+1 at time t+1. In order to keep the length of sequence as R, we removes O1 from the previous sequence and append OR+1. This makes the sequence as,

$$O_{2,}\ O_{3,}\ .\ .\ .\ O_{R,}\ O_{R+1}$$

Now, this new sequence becomes teh input for HMM and acceptance probability is calculated. Let this new acceptance probability be:

$$\alpha_2 = P\ (O_{2,}\ O_{3,}\ .\ .\ .\ O_{R,}\ O_{R+1}\ |\lambda\ )$$

$$\text{Let } \boldsymbol{\Delta\alpha = \alpha_1\text{-}\alpha_2}$$

If,

$$\Delta\alpha > 0,$$

, it is concluded that this new sequence has low acceptance probability and it could be a possible instance of attack.

For declaring a resource consumption as a fraudulent case, we determine percentage change in the acceptance probability, that is,

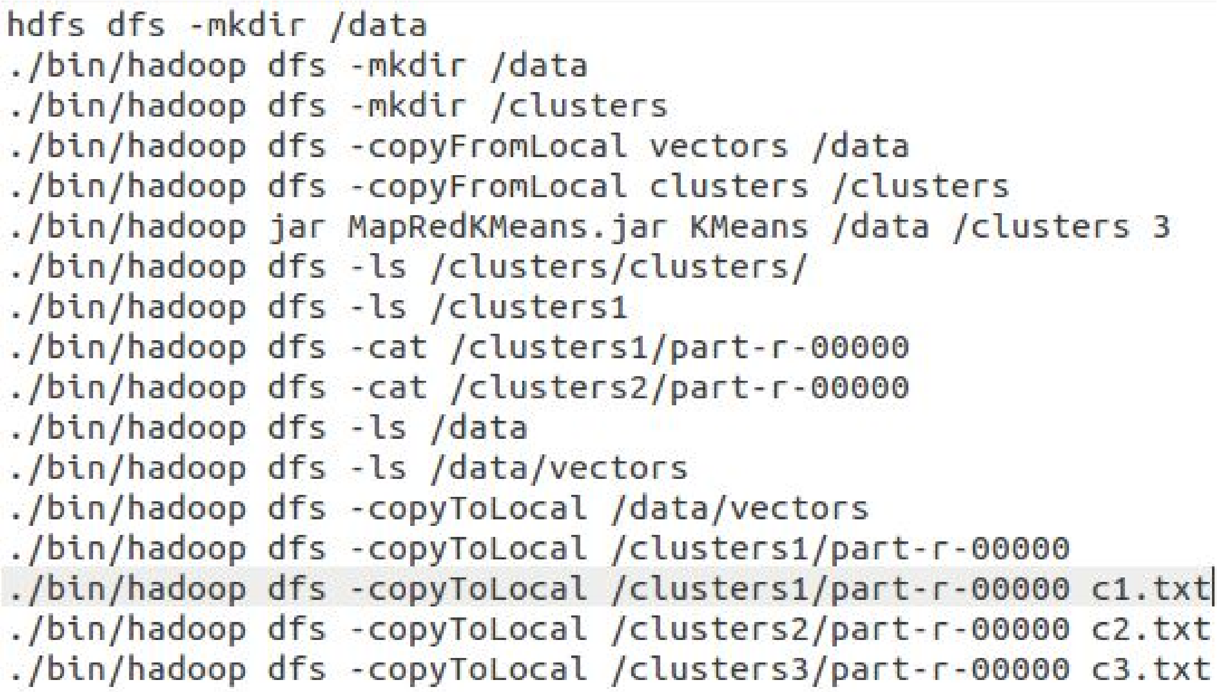
$$\Delta\alpha\ /\ \alpha_1 \geq \text{Threshold.}$$

Each accepted new non-malicious symbol is added in the sequence so as to update the user profile with any change in consumption behavior. In real-world, there exists a trade-off between True Positives and False-Positives, so the difference between these is considered as a performance metric.

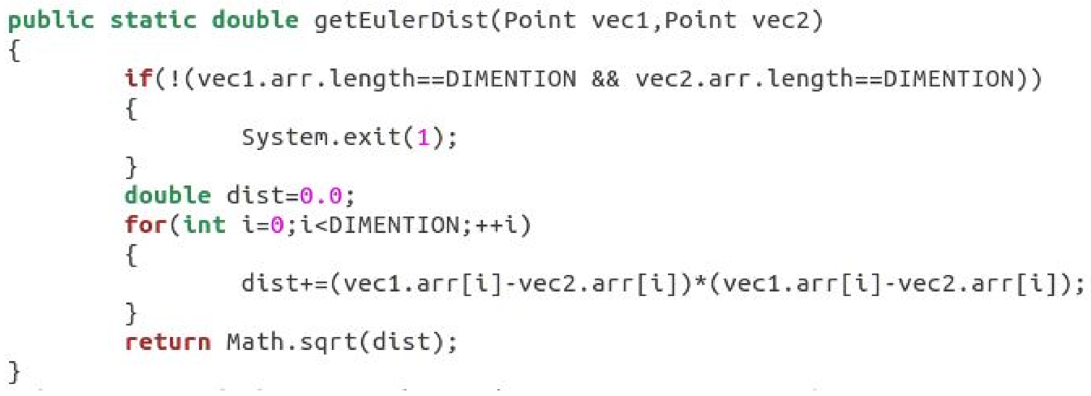
Chapter 7

Implementation

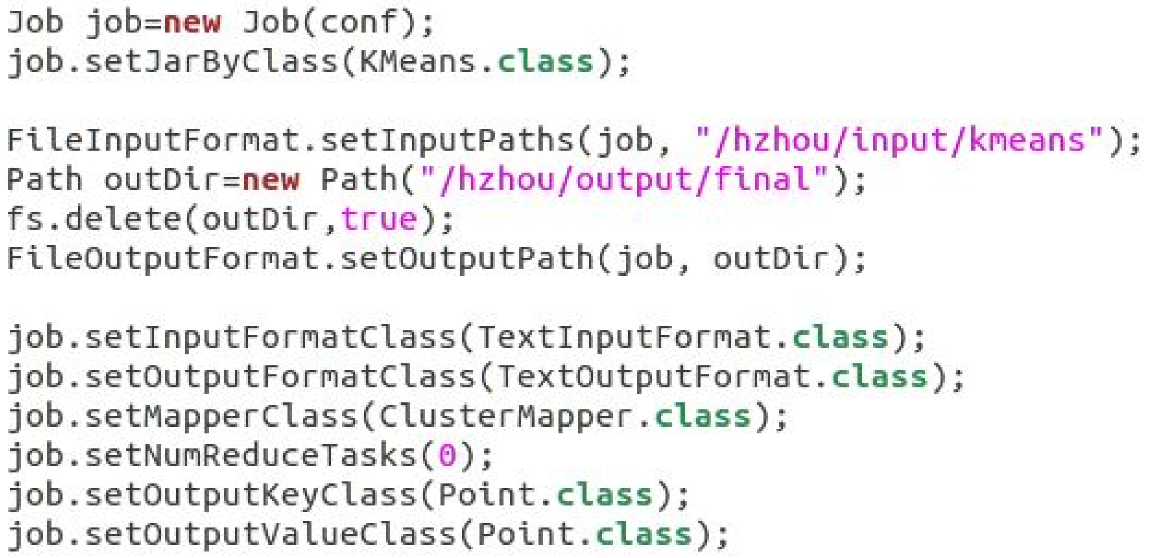
7.1 Screenshots for K-means



```
hdfs dfs -mkdir /data
./bin/hadoop dfs -mkdir /data
./bin/hadoop dfs -mkdir /clusters
./bin/hadoop dfs -copyFromLocal vectors /data
./bin/hadoop dfs -copyFromLocal clusters /clusters
./bin/hadoop jar MapRedKMeans.jar KMeans /data /clusters 3
./bin/hadoop dfs -ls /clusters/clusters/
./bin/hadoop dfs -ls /clusters1
./bin/hadoop dfs -cat /clusters1/part-r-00000
./bin/hadoop dfs -cat /clusters2/part-r-00000
./bin/hadoop dfs -ls /data
./bin/hadoop dfs -ls /data/vectors
./bin/hadoop dfs -copyToLocal /data/vectors
./bin/hadoop dfs -copyToLocal /clusters1/part-r-00000
./bin/hadoop dfs -copyToLocal /clusters1/part-r-00000 c1.txt
./bin/hadoop dfs -copyToLocal /clusters2/part-r-00000 c2.txt
./bin/hadoop dfs -copyToLocal /clusters3/part-r-00000 c3.txt
```



```
public static double getEulerDist(Point vec1,Point vec2)
{
        if(!(vec1.arr.length==DIMENTION && vec2.arr.length==DIMENTION))
        {
                System.exit(1);
        }
        double dist=0.0;
        for(int i=0;i<DIMENTION;++i)
        {
                dist+=(vec1.arr[i]-vec2.arr[i])*(vec1.arr[i]-vec2.arr[i]);
        }
        return Math.sqrt(dist);
}
```



```
Job job=new Job(conf);
job.setJarByClass(KMeans.class);

FileInputFormat.setInputPaths(job, "/hzhou/input/kmeans");
Path outDir=new Path("/hzhou/output/final");
fs.delete(outDir,true);
FileOutputFormat.setOutputPath(job, outDir);

job.setInputFormatClass(TextInputFormat.class);
job.setOutputFormatClass(TextOutputFormat.class);
job.setMapperClass(ClusterMapper.class);
job.setNumReduceTasks(0);
job.setOutputKeyClass(Point.class);
job.setOutputValueClass(Point.class);
```

Chapter 8

Conclusion

Utility models on the downside creates an issue of Fraudulent Resource Consumption attacks where intruder mimics the profile of a legitimate user in order to hide his own malicious intents behind it. Normally, these produces anomalies as it is hard to mimic a user's profile completely.

A anomaly based detection system is used to encounter these attacks. A Hidden Markov Model is used to determine the probability of any consumption event. This is compared with threshold to detect whether it is a normal event or anomalous.

References

[1] *Credit Card Fraud Detection Using Hidden Markov Model*; Abhinav Srivastava, Amlan Kundu, Shamik Sura

[2] A SURVEY ON DATA SECURITY IN CLOUD COMPUTING: ISSUES AND MITIGATION TECHNIQUES; Satarupa Biswas, Abhishek Majumder

[3] Securing Cloud from Attacks based on Intrusion Detection System; Soumya Mathew1, Ann Preetha Jose2

[4] Attribution of Fraudulent Resource Consumption in the Cloud; Joseph Idziorek, Mark Tannian, Doug Jacobson

[5] Anomaly Detection in the Cloud: Detecting Security Incidents via Machine Learning; Matthias Gander, Basel Katt, Michael Felderer, Adrian Tolbaru, Ruth Breu , and Alessandro Moschitti

[6] Anomalous Event Detection on LargeScale GPS Data from Mobile Phones Using Hidden Markov Model and Cloud Platform

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

[8] *Exploiting Cloud Utility Models for Profit and Ruin*Joseph Idziorek

[9] *Improving Network Traffic Anomaly Detection for Cloud Computing Services*

[10] *IEEE Cloud Computing*

[11] *Detecting Fraudulent Use of Cloud Resources*; Joseph Idziorek, Mark Tannian, Doug Jacobson