Design of a Risk Based Authentication System using Machine Learning Techniques

Mohammed Misbahuddin[1], B S Bindhumadhava[2], B. Dheeptha[3]
[1,2]Computer Networks and Internet Engineering (CNIE) Division,
Centre for Development of Advanced Computing, Electronics City, Bangalore, India – 560100
[3]Dept. of Computer Science & Engineering, Sastra University, Thanjavur, Tamil Nadu – India - 613402
misbah@cdac.in, bindhu@cdac.in, dheepthab1210@gmail.com

Corresponding Author: Mohammed Misbahuddin (mdmisbahuddin@gmail.com)

***Abstract*—Authentication provides a means to verify the legitimacy of a user trying to access any confidential or sensitive information. The need for protecting secure data hosted on the web has been rising exponentially as organizations are moving their applications online. Static methods of authentication cannot completely guarantee the genuineness of a user. This has led to the development of multi-factor authentication systems. Risk-based authentication, a form of multi factor authentication adapts itself according to the risk profile of the users. This paper puts forth the design of risk engine integrated with the system to examine the user's past login records and generate a suitable pattern using machine learning algorithms to calculate the risk level of the user. The risk level further decides the authentication method that the user will be challenged with. Thus the adaptive authentication model helps in providing a higher level of security to its users.**

Keywords— Multi-factor Authentication; Risk Based Authentication; User behavior; Risk engine; Machine Learning Algorithms

I. INTRODUCTION

Online systems and applications are increasingly encountering a plethora of cyber threats. This is majorly due to the conventional methods of authentication, like username/password prevalent in these systems. With access to several hacking tools, these systems are highly prone to attacks [8] including masquerade attacks, DNS attack and phishing attacks. Moreover, passwords are simply knowledge-based information that can be shared amongst users. Hence, single factor authentication does not completely ensure the authenticity of users. Although single factor authentication seems to be an effortless process for the user, two factor authentication or multi factor authentication is preferable due to its improved security levels.

User's credentials used for authentication are categorized according to what the user knows, what the user has, and what the user is. Passwords fall under the category of what the user knows, while tokens like dongle/phone are examples of what the user has. Biometrics is an example of what the user is. Passwords can easily be hacked and tokens can be stolen or reused. Biometric applications are expensive to implement. However, a combination of these factors can promote higher security to online systems. Such a form of authentication is called multi-factor authentication.

Risk based authentication is one type of multi factor authentication that adapts according to the user's risk profile. Risk profile is obtained by comparing the user's profile retrieved at the time of login and past login records of the user. A model, characterizing the user's behavior, is built by a risk engine that is integrated with the risk based authentication system. The proposed work utilizes machine learning algorithms to build such a model. Machine learning algorithms are trained to learn patterns from available data and predict the unknown value when provided with new set of data.

Depending on the risk profile generated, user is challenged with different authentication methods [9]. Thus a genuine user is not required to pass multiple factors of authentications to prove his genuineness, while a suspicious user needs to pass all the authentication methods he is challenged with. This ensures that the system is usable and security as opposed to the majority of the existing systems that strike a trade-off between usability and security.

The remaining paper is structured in the following manner- Section 2 briefs about the existing research works, analyzing their strengths and drawbacks. The detailed elaborations of the proposed method are elicited in section 3. Section 4 presents the experimental results and section 5 presents the conclusion.

II. RELATED WORKS

This section explains the various methods that have been used to analyze user behavior in a risk based authentication mechanism. These systems adjust themselves depending on the current context as opposed to the static authentication systems. Diep et al. [1] proposed a mathematical risk based technique called Multifactor Evaluation Process that assigned numerical weights to risk elements based on confidentiality, integrity and availability of the outcomes. The risk is measured based on contextual information, requested services and the type of transaction. A similar approach based on risk adaptive control was presented by Cheng et al. [2]. The model (Quantified Risk Adaptive Access Control) involved risk calculation based on fuzzy logic. The evaluated risk was divided into several ranges, each associated with a particular action.

978-1-5386-0435-9/17/$31.00 ©2017 IEEE

Few researches considered location and time for contextual modeling. Cristiano et al. [3] used device and spatial-temporal context to model user behavior. User behavior was analyzed using explicit profile, implicit profile, session profile and a Vector Space Model (VSM) filter. The users are classified as normal, suspicious and abnormal. Issa et al. [4] built Bayesian models for user's keystroke dynamics and mouse dynamics. The model built during enrolment stage was compared with the one built during verification stage to arrive at a risk score for the user.

An adaptive authentication mechanism was proposed by Abu Bakar and Haron [5], which used a Unified Authentication Platform that integrated multifactor authentication and Single sign-on. The model consisted of a trust engine to generate patterns and evaluate trust score. Trust score evaluation was based on the authentication method strength, user login parameters and application security requirement. Shi et al. [6] applied a learning algorithm to let the machine learn the past behavior of the user and create a user model. The probability of the user being genuine is calculated based on the user model and the recently observed behavior.

In a recent study, Dasgupta et al. [7] developed a design for selecting factors dynamically for multifactor authentication. Trustworthy values were calculated for each factor using mathematical objective functions. The selection of factors was affected by the device, media and external conditions like noise and light. This technique reduced the repetitive selection of the same set of authentication factors.

The existing methods, however, do not employ machine learning techniques to implement the adaptive authentication system. This paper proposes a technique to model user behavior based on contextual information using three different machine learning algorithms. Supervised and unsupervised learning algorithms have been used to analyze different types of input.

III. PROPOSED METHOD

The Risk based authentication system consists of three blocks: Profile retrieval block, Risk engine and an adaptive authentication block as depicted in figure 1.

A. Profile Analysis Block

When the system receives a request for authentication, the user enters his password and the authentication server retrieves the user parameters required to model user behavior. The user factors include the following. These parameters form the inputs to the risk engine as shown in figure 2.

- IP address
- Geo location (retrieved using DB-IP database [11])
- Time zone
- Login time
- OS version
- Browser version
- Device type
- Number of failed attempts

Every user's past login records that serve as the user's baseline profile according to [12] are stored at the authentication server. Once the user enters the correct password, risk engine is activated. The recently retrieved contextual information of the user and user's past login records are made available to the risk engine.

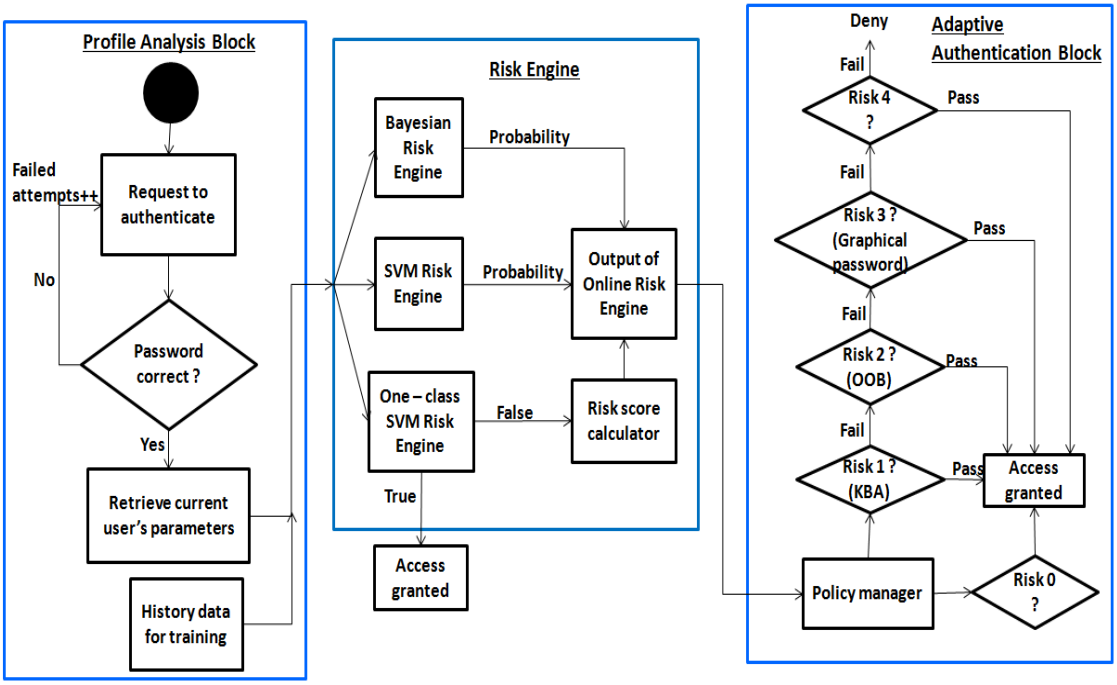

Fig. 1 Architecture of Proposed Risk Based Authentication System

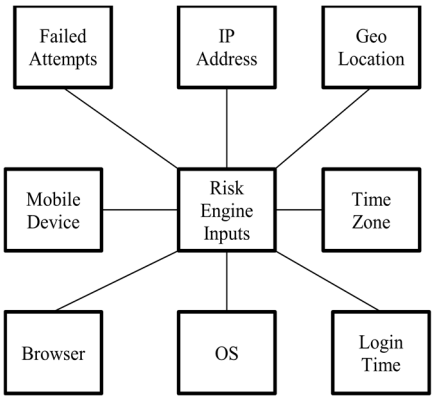

Fig. 2. Risk engine inputs

B. Risk Engine

The risk engine is the core component of risk based authentication scheme which can predict the risk level of a user. In this paper, the risk engine is designed using three different machine learning algorithms (SVM, one-class SVM, and Naïve-Bayesian).

The main aim of any machine learning algorithm is to learn the structure of data and predict the outcome when fed with new data. **SVM** is among the best supervised learning algorithm that is used for classification and regression. In the given scenario, we wish to classify a user as genuine or suspicious. SVM can be used if we have a dataset containing login patterns of a user (user parameters) with genuine and fraudulent accesses. User access is considered to be fraudulent when he is denied access to the resource requested. Now, if the SVM risk engine is fed with a new set of user parameters, it should be able to classify the user as genuine/fraudulent. It does so by drawing a hyperplane that separates genuine patterns from fraudulent patterns. A hyperplane is used by the SVM algorithm in cases where linear separation of data points is plausible, while a kernel function is used for high dimensional data. It is also possible for SVMs to predict a probability value. Hence, the output of the SVM risk engine is programmed to indicate the probability of the user being fraudulent.

The main objective of the risk engine is to detect anomalies in the data. For such purposes, labeled training set is not essential. For unlabelled training dataset, supervised learning algorithm cannot be applied. This calls for the use of unsupervised learning algorithm. **One-class SVM** is one such unsupervised learning algorithm. Therefore, one-class SVM risk engine is trained using data with only genuine user patterns. It can be used when there is a dearth of anomaly data. The output of one-class SVM is a true-false value. Hence, 'true' output indicates that the user is genuine, while 'false' output indicates that the user may be fraudulent. If the output results in a false value, table1 should be used to estimate the risk score of the user. Risk score is calculated using equation 1. A higher risk score corresponds to a higher risk level.

$$Risk\ score = \sum user\_parameter\_value\ X\ user\_parameter\_weig\square t \quad (1)$$

,

where User_parameter_value = 0 if behavior exists in the past login records

= 1 Otherwise.

User_parameter_weight is assigned according to table 1. These weights have been assigned after considering the impact each parameter would have in determining potential risk.

Naïve Bayesian classifier is based on Baye's theorem, which is stated as

P(h|d) = (P(d|h) * P(h)) / P(d), where

- P(h|d) is the probability of hypothesis h given the data d, called the posterior probability.
- P(h) is the probability of hypothesis h being true (regardless of the data), called the prior probability of h.
- P(d|h) is the probability of data d given that the hypothesis h was true.
- P(d) is the probability of the data

In our project, hypothesis 'h' refers to the user being genuine or fraudulent and data'd' refers to each user parameter stored in the server database. The calculation of the posterior probability relies on an important assumption that the parameters are conditionally independent. Naïve Bayesian model also uses a supervised learning algorithm. Training the model takes very less time compared to SVM, as it involves simple probability calculation. However, it is only suitable for small amount of data. The output of the naïve bayes classifier indicates class probabilities. Class probability refers to the probability of the user belonging to genuine class and fraudulent class.

TABLE I. USER PARAMETER WEIGHTS

User parameters	**Weight**
Browser	1
OS	2
Login time	3
IP address	4
Mobile Device	5
No. of failed attempts	6
Location	7
Time zone	8

C. Adaptive Authentication Block

The risk score/probability calculated by risk engine is fed to the policy manager, which classifies risk level based on the risk score/probability. The user is challenged with an authentication method like security questions, or OTP, or graphical password

[13, 14] associated with each risk level as mentioned in table 2. Lower risk levels have been associated with shorter probability and risk score ranges as against the larger threshold ranges for higher risk levels. A successful login is stored as a genuine record at the server, while an unsuccessful login attempt is recorded as a fraudulent pattern.

TABLE II Authentication methods for each risk level

SVM/Bayesian Probability (P)	One-class SVM Risk Score (S)	Risk Level	Authentication Method
50≤P≤60	1≤S≤6	1	Security Questions
60<P≤75	7≤S≤18	2	OTP token
75<P≤90	19≤S≤29	3	Graphical Password
90<P≤100	30≤S≤36	4	Digital Signature

IV. Experimental Results and Discussions

The proposed technique has been implemented using Android, java and R programming [10]. For login, the user provides username and password for his first login attempt after registration. The machine then learns the user behavior by analyzing the user parameters retrieved at every login attempt, and raises the risk level whenever it encounters new user behavior.

The paper assumes the following constraints:

a) A change in login time is considered as a risk only if it exceeds a margin of two hours from the usual login time of the user.

b) Number of unsuccessful attempts affects the risk level if the user exceeds 3 login attempts.

Suppose that the user's recent 10 transactions are as specified in table 3. The table indicates that the user has logged in using the same parameters at different time intervals and hence has been classified as a genuine user. During initial stages of user login, the risk engine remains inactive. Once the historical dataset is populated with a minimum of 10 records per user, the risk engine is activated to predict the risk level of the user. Table 3 shows that the user has mostly logged in from Bangalore using a windows10.0 laptop and chrome browser at different times of the day.

One-class SVM is an unsupervised learning algorithm. Hence, genuine patterns are sufficient to train the model. However, SVM and Bayesian models are required to be trained using genuine and fraudulent patterns. Examples of fraudulent patterns in the user's historical dataset are shown in table 4. The table shows that the user has requested access with parameters different from that of table 3. These patterns have been classified as fraudulent by one-class SVM risk engine since the user could not pass the additional methods of authentication that were requested of the user due to the elevated risk level.

Suppose that the user tries to login with the user parameters as shown in table 5 after having established a behavior profile. Table 5 considers four different scenarios for all the risk engine types. The first scenario explains a situation where the user accesses the service from a different location. The second scenario corresponds to a state wherein the user logs in using a different OS and a browser. The third set of user parameters indicates that the user has accessed the service from a different location using different OS and browser after 3 failed login attempts. In the last situation, the user logs in from a different time zone with different OS, browser after three unsuccessful login attempts. The corresponding SVM and Bayesian risk engine outputs indicate the probability of the user being fraudulent whilst one-class SVM risk engine output gives the risk score of the user. The risk score is directly proportional to the risk level of the user. Depending on the obtained values, the user is requested for further credentials as cited in table 2.

It is evident from table 5 that the risk score calculated by one-class SVM is more relevant to the given scenarios than Bayesian probability values, which appear extreme for all the example cases i.e. a change in OS alone has resulted in the least risk level, while a change in time zone has resulted in the highest risk level. Bayesian probability values for the first two scenarios indicate that the user is genuine despite the change in location or change in OS/browser. The values generated by SVM model are considerable although, a higher risk level would have been preferred for change in location. Hence, one-class SVM risk engine is more preferred for our test data. However, it cannot be generalized that one-class SVM algorithm would always perform better in detecting anomalies as the performance of any machine learning based engine depends on the training data.

To summarize the working of the engine, when the user's past login records, as shown in table 3, are fed to the risk engine, it learns the usual behavior of the user and builds a model to reflect the user profile. Thus, when the user requests access with different parameters, the engine predicts a higher risk level to avoid masquerading attacks. If the user is able to login after providing the extra credentials, the machine records the new pattern as safe. Else, the user's behavior is marked as fraudulent.

V. Features and Advantages of the Proposed Method

Despite the existence of several approaches to multi factor authentication, not many studies have been effectuated to determine selection strategy for multiple authentication factors using machine learning algorithms. Machine learning algorithms offer the best means to model user behavior and finding the risk level associated with each user. This paper employs three different algorithms (SVM, Naïve-Bayesian and one-class SVM) to analyze user behavior. Depending on the accuracy of results and history records available, a suitable model can be chosen.

Although Bayesian algorithm has been used in the past to model user behavior [7], the model uses supervised learning algorithm and therefore cannot provide efficient results unless it is trained using genuine and fraudulent patterns. This paper hence proposes the use of one-class SVM algorithm in situations where both types of records may not be available. The risk score determined using equation 1 depends on the

number of non-matching parameters and their corresponding weights. This improves the effectiveness of the model in predicting he risk level.

Furthermore, this study considers nine user login parameters before deciding the genuineness of the user. The combination of all parameters helps in determining risk score. The determined risk level of the user facilitates dynamic selection of authentication factors as opposed to the static methods of authentication prevalent in most multi factor authentication systems.

The proposed procedure also promotes usability in addition to security. A genuine user is not required to pass multiple factors of authentications to prove his authenticity, while only a suspicious user needs to pass all the authentication methods he is challenged with. Consequently, users are less vexed with the authentication process.

TABLE III User Login Records

UID	IP address	Location	Time Zone	Login Time	OS	Browser	Mobile	Failed Attempts	Class
DDAF35A1	103.5.19.128	Bangalore	IST	9:11:44	Windows10.0	Chrome	Motorola	0	Genuine
DDAF35A1	103.5.19.128	Bangalore	IST	11:24:31	Windows10.0	Chrome	Motorola	0	Genuine
…..									
DDAF35A1	103.5.19.128	Bangalore	IST	22:53:13	Windows10.0	Chrome	Motorola	0	Genuine

TABLE IIV Example for fraudulent patterns

UID	IP address	Location	Time Zone	Login Time	OS	Browser	Mobile	Failed Attempts	Class
DDAF35A1	103.5.19.11	Bangalore	IST	10:17:46	Windows8.1	Chrome	Motorola	4	Fraudulent
DDAF35A1	103.5.19.11	Bangalore	IST	10:43:23	Windows8.1	Firefox	Motorola	0	Fraudulent
DDAF35A1	103.5.19.2	Bangalore	IST	13:5:5	Linux	Firefox	Motorola	0	Fraudulent
…..									

TABLE V Four User login scenarios with Probability/Risk score

Scenario	I	II	III	IV
UID	DDAF35A1	DDAF35A1	DDAF35A1	DDAF35A1
IP address	1.22.247.55	103.5.19.128	1.22.247.55	192.154.1.11
Location	New Delhi	Bangalore	New Delhi	California
Time Zone	IST	IST	IST	PST
Login Time	16:09:57	16:41:33	16:55:03	03:15:19
OS	Windows10.0	MAC	MAC	MAC
Browser	Chrome	Safari	Safari	Safari
Mobile	Motorola	Motorola	Motorola	Motorola
Failed Attempts	0	0	3	3
SVM	59.918	48.08	65.144	74.406
Bayesian	0	20.328	99.99	99.99
One-class SVM	11	3	20	31

VI. CONCLUSION

Risk Based Authentication offers further security to an online application. It considers various parameters before deciding whether or not to grant access to the user. Besides, the user is challenged with extra factors of authentication if his behavior appears suspicious to the risk engine. Any variation in the login parameters is viewed as a potential risk by the risk engine and the system may demand additional verification from the user. Past login records of the user comprise a major implication in the design of the risk engine. Besides, the proposed technique offers three choices for risk engine to allow operation during different situations. For instance, lack of training records for fraudulent patterns requires the selection of one-class SVM risk engine. Moreover, as an additional measure of security, the authentication procedure is designed to be carried out on the user's mobile device. Hence, a user's account cannot be compromised unless the imposter uses the legitimate user's smart phone.

Four scenarios have been discussed in the paper to bring out the four probable risk levels and the outcome of each risk engine has been compared when a change in the user behavior is encountered. The paper thus proposes a highly secure method to protect online accounts from cyber threats.

REFERENCES

[1] Diep N. N., S. Lee, Y.-K. Lee, H.J. Lee, "Contextual Risk-based Access Control", Security and Management, pp. 406-412, 2007.

[2] Cheng P.-C., P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, "Fuzzy Multi–Level Security: An Experiment on Quantified Risk–Adaptive Access Control", IBM Research Report RC24190, 2007

[3] Cristiano C. Rocha, Joao Carlos D. Lima, M. A. R. Dantas, and Iara Augustin. "A2best: An adaptive authentication service based on mobile user's behavior and spatio-temporal context". In IEEE Symposium on Computer and Communication (ISCC), pages 771–774, 2011.

[4] Issa Traore, Isaac Woungang and Mohammad S. Obaidat. " Combining Mouse and Keystroke Dynamics Biometrics for RiskBased Authentication in Web Environments". Fourth International Conference on Digital Home, 2012.

[5] Khairul Azmi Abu Bakar and Galoh Rashidah Haron. "Adaptive Authentication based on Analysis of User Behavior". Science and Information Conference, 2014.

[6] Elaine Shi, Yan Niu, Markus Jakobsson, and Richard Chow. Implicit authentication through learning user behavior. Information Security, 6531, pp. 99–113, 2011.

[7] Dipankar Dasgupta, Arunava Roy and Abhijit Nag. "Toward the design of adaptive selection strategies for multi-factor authentication". computers & security, pp. 85–116, 2016.

[8] Mohan V. Pawar, Anuradha J. " Network Security and Types of Attacks in Network". Procedia Computer Science 48, pp. 503 – 506, 2015.

[9] Kumar Abhishek, Sahana Roshan, Prabhat Kumar and Rajeev Ranjan. "A comprehensive study on Multifacto Authentication Schemes". Advances in Computing and Information Technology, 177, pp. 561-568, 2013.

[10] Rob Kabacoff. "R in Action". Manning publications, 2010.

[11] DB-IP - IP geolocation and Network Intelligence homepage. http://www.db-ip.com

[12] Oded Peer, Yedidya Dotan, Yael Villa and Marcelo Blatt. "USING BASELINE PROFILES IN ADAPTIVE AUTHENTICATION". US Patent 8,621,586 B1, December 31, 2013.

[13] Mohammed Misbahuddin, P. Premchand, A. Govardhan,"A User Friendly Password Authenticated Key Agreement for Multi Server Environment", in International Conference on Advances in Computing, Communication and Control (ICAC3'09)

[14] Mohammed Misbahuddin, Dr. P. Premchand, Dr. A. Govardhan "A User friendly Password Authenticated Key Agreement for Web based Services", in the proceedings of International Conference on Innovations in IT (IIT '08) Al-Ain, UAE, Dec '08, Pg. 633 - 637.