ADAPTIVE AUTHENTICATION: A CASE STUDY FOR UNIFIED AUTHENTICATION PLATFORM

Khairul Azmi Abu Bakar[1] , Nor Izyani Daud[2]
and Mohd Shafeq Md Hasan[3]

[1]Information Security Lab, MIMOS Berhad, Malaysia
khairul.azmi@gmail.com
[2]Information Security Lab, MIMOS Berhad, Malaysia
izyani.daud@mimos.my
[3]Information Security Lab, MIMOS Berhad, Malaysia
shafeq.hasan@mimos.my

ABSTRACT

Adaptive authentication is a risk-based authentication that identifies high-risk and suspicious illegitimate login attempts. User past login records which implicitly contains attribute factors context information are used to establish user behavior profile. Later if the user logins under different environmental context from that established profile, the identity of the user may be questioned. The system may challenge the user to present additional authentication method to get authenticated. We implemented such adaptive authentication system in our production server and collected user login records for more than six months. In this paper, we presents the analysis of the user login profile with regards to attribute factors such as geographical location and time of login. We also developed testbed system that uses the collected real data to evaluate the system for different ratio threshold values.

KEYWORDS

Adaptive Authentication, Web Application, Testbed Analysis

1. INTRODUCTION

Authentication is a process to confirm that someone or something is, in fact, who or what it is claimed to be. The process involves obtaining identification credentials or authentication method such as username/password from a user and validating the credential against some authority. Users who can present valid credential are considered authenticated identities. In general, there are three categories of credentials: something you know (password), something you have (ATM card) or what you are (fingerprint). To make it difficult for unauthorized person to gain access, the system may implement multi-factor authentication where the user needs to successfully present additional credentials from at least two of those three categories. If one factor is compromised or broken, the attacker still has at least one more barrier to breach to break into the system.

David C. Wyld et al. (Eds) : WiMo, ITCSE, ICAIT, NC - 2015
pp. 61–72, 2015. © CS & IT-CSCP 2015 DOI : 10.5121/csit.2015.51006

In traditional authentication system, the decision on the level of authentication credential required solely depends on the application that the user trying to access. High sensitive applications such as internet banking would demand the user to present stronger authentication credential than what insensitive applications would. The required authentication methods could also be a combination of two or more credentials, increasing the authentication security even more. However, in such system, the security requirement is static because it only depends on the application security requirement.

Adaptive authentication system uses login environmental characteristics and user behavioral profiling to identify high-risk login and dynamically customizes the authentication requirement accordingly. The system studies common behavior pattern of all users based on their past history login access. If a user follows the same patterns when logging into the system, the login experience may probably be a username/password indicating a low risk attempt. However, if a user tries to login under different behavior or environment, the identity of the user is questioned. The system may adaptively challenge the user to provide stronger or additional authentication credentials to get authenticated.

We built such system in our lab and put it on our production server for couple of months. The system had been collecting and storing users' login information into a database. We also developed a testbed system that uses those data as the input parameters. The testbed enables us to evaluate the behavior of our adaptive authentication system with different set of configurations are used. In this paper, we present the result.

The remainder of the paper is organized as follows. Section 2 presents our current Unified Authentication Platform (UAP) system that contains adaptive authentication component. Section 3 describes the formula to calculate User Attribute Score with regards to the relevant attribute factors. Section 4 explains the algorithm used for the testbed environment that have been developed to evaluate the system. Section 5 present the results from user login records analysis and different ratio threshold values. Finally, Section 6 draws the conclusions.

2. BACKGROUND

In MIMOS Berhad, we have developed an authentication system called Unified Authentication Platform (UAP). UAP is a centralized multi-factor authentication system with web-based single sign-on (SSO) capability to manage user authentication profiles. It is designed to manage front-end application authentication using an established protocol, Secure Assertion Markup Language (SAML), which provides a centralized authentication framework and aims to reduce significant application changes at the backend. The objectives of UAP are as the following:

1. provide an infrastructure that offers authentication service to applications
2. provide information technology that de-couples authentication function from application and authorization
3. grow indigenous authentication mechanism industry throughout the country
4. a unified authentication platform initiative for enabling government e-services application

UAP is derived from Shibboleth [7] which is a standard based, open source package for web single sign-on across or within organizational boundaries. In addition, UAP supports multiple authentication methods. Users can choose from a list of authentication methods to get

authenticated and be allowed to access various applications without having to go through the same authentication process again. The overall architecture of UAP is depicted in Figure 1[5].

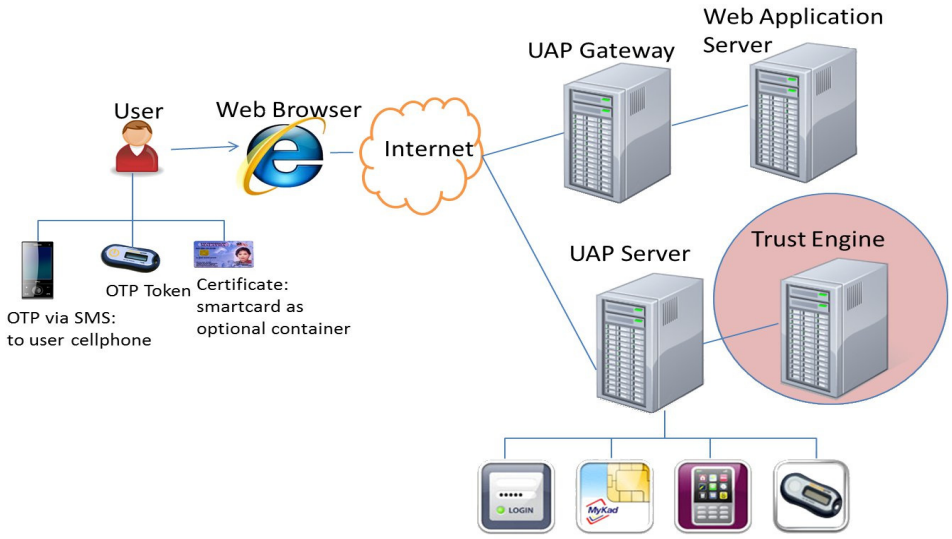

Figure 1. Adaptive UAP Architecture Diagram

For the new generation of UAP, called adaptive UAP, we introduce an additional component called Trust Engine that incorporates adaptive control based on security risk and level of assurance. To make informed authentication decision, Trust Engine takes into account attribute factors from the user behavior profiles which had been previously analyzed and stored.

Adaptive UAP consists of two processes: Pattern Generator and Trust Evaluator. Pattern Generator analyses context information from user past login records which had been stored at table data_log. Only login records from the last predefined period of time are processed. Context information from the records such as login time and IP Addresses are converted into a meaningful data format before the results along with the number of occurrence are stored at table common_attr as the user behavior profile. Pattern Generator is currently configured to get executed on every midnight.

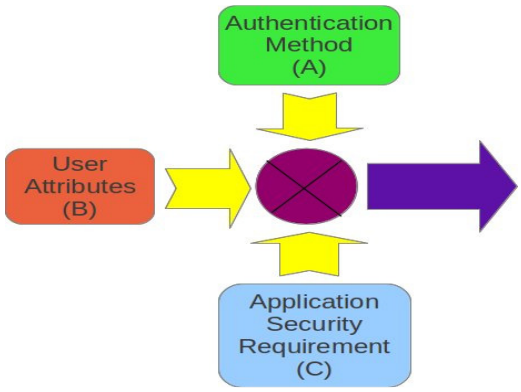

Figure 2. Decision Factors

Trust Evaluator is responsible to analyze, decide and act upon every login request from users. Trust Evaluator makes the final decision regarding on user authentication based on three factors as shown in Figure 2: Authentication Method Score, User Attribute Score and Application Required Trust Level. Authentication Method Score derives from the accumulated trust value of all user successfully presented credentials. User Attribute Score is the result value from the comparison process between the user current login contexts against the established user behavior profile. Finally, Application Required Trust Level is the minimum trust level set by the application that the user need to acquire to get authenticated. At the end of the process, Trust Evaluator stores the login records into table data_log as depicted in Table 1 next to be processed by component Pattern Generator.

Table 1. User Login Information.

No	Entry	Description
1.	uuid	User unique identity
2.	time_login	Data and time of the login
3.	browser_osname	User's browser and OS information
4.	ip_int	User's terminal IP Address
5.	sp_id	UAP Gateway Server ID string
6.	auth_method	Authentication method ID
7.	Tr	Application required trust level

Basically the requirement as in Formula 1 should be satisfied before the user is considered an authenticated entity by the system.

$$A - B \geq C \qquad (1)$$

Where
A: Authentication Method Strength
B: User Attribute Score
C: Application Security Requirement

If the established trust level does not meet the application required trust level, Trust Engine returns FALSE to Authentication Server. User needs to present additional authentication method to increase the established trust level. More explanation with some scenario examples can be read at [3] and [4].

3. USER ATTRIBUTE SCORE

In this section, the second component of the formula that contributes for the final authentication decision is discussed. User Attribute Score represents the uncertainty level of current login attempt with respect to the established user behavior profile. In other words, if the user logins under different behavior and environment from what the user normally had experienced, the score value would be high. In this case study, attribute factors used are user geographical location (city name), time login, type of browser and operating and targeted application.

There are two steps to calculate User Attribute Score. In the first step, the system needs to find out the user common context for each attribute factor. Common context should meet the following conditions:

1. the number of records for the last 14 days is more than 10
2. the frequency of occurrence of any particular context is more than a ratio threshold of the overall records

Trust Evaluator should have gathered reasonably enough number of user records to establish credible user attribute profile. In this case study, the minimum number of records required is set to 10 from which only login records for the past 14 days are been considered .

To get qualified as a common context, the occurrence of the context should also exceeds a ratio threshold. Presumably the threshold is set to 30%, if the user has 100 login records within the period of time, any context that has at least 30 records is labeled as the user common context. For example, if the user had been login from city of Kuala Lumpur for more than 30 times, Kuala Lumpur is considered as the user common context under geolocation factor. If none of the geolocation information meets both of the two conditions, the user is considered as not having common context for that attribute factor and geolocation element is omitted from the formula. Other attribute factors may still be applied subject to the same conditions.

If there exist common profile within the attribute factor, Trust Evaluator compares each of the common profiles with the user current login context. If there is a match, the value of that attribute factor is set to zero which gives no effect to the final score. Otherwise, the attribute factor is activated. As explained later in this paper, the number of event where the attribute factor is activated is recorded for analysis. The number reflects the rate of occurrence where the user logins under different environment from the user behavior profile for every attribute factor.
Each of the attribute factor is assigned with a weightage value. The weightage value is in fraction numbering format and represents the significance of the factor in user behavior profile. Higher significant factor is assigned with a higher value. User attribute factor is calculated by adding the weightage values for all activated factors as shown in (2) and multiply the result with a variable max_user_score to limit the maximum possible scoring number.

$$\begin{aligned} \text{attribute\_score} = (&(\text{time} * \text{weight}_{\text{time}}) \\ &+ (\text{geolocation} * \text{weight}_{\text{location}}) \\ &+ (\text{browserOS} * \text{weight}_{\text{browser}}) \\ &+ (\text{application} * \text{weight}_{\text{application}})) \\ * (\text{max\_user\_score}) & \end{aligned} \quad (2)$$

4. TESTBED SETUP

We have developed a testbed component for the Trust Engine that would able to take input parameter from the stored records in table data_log instead of from the UAP server. Both processes in Trust Engine take place as it is a real input data. The testbed allows us to evaluate the performance of the Trust Engine when different set of configuration is used. Algorithm 1 shows the pseudo code for the testbed. Trust Evaluator component processes each of the login record while Pattern Generator component is executed at the end of each day.

Algorithm 1. General Steps for TestBed

```
1: Generate list of days of table data_log
2: for Each day in the list do
3:     for Every login records on day day do
4:         process TrustEvaluator
5:     end for
6:     process PatternGenerator
7: end for
```

5. EXPERIMENT RESULT

We collected login information from our production server from 6 May 2014 until 15 January 2015. Total number of 171,045 login information from 1244 unique users have been recorded during those 254 days period.

5.1. Attribute Factor

In this section, we present the overall analysis of the collected user login information based on the attribute factor.

5.1.1. Geographical Location

The information about user geographical location is extracted from the IP Address of the user terminal. We use solution from a third party company ip2location [6] which provides database records that contain geographical information such as name of the city, region and country of origin for IP Addresses. Special IP Addresses (10.0.0.0 to 10.255.255.255, 172.16.0.0 to 172.31.255.255, or 192.168.0 0 to 192.168.255.255) are originated from internal network and are categorized as private IP Addresses.

From the total number of records collected, 67.7% (115,816) have IP Address information. 97.8% (113,254) of those login records that contains IP Address information are from internal network. From the other 2,562 login records that comes from external network, 2,515 (98.1%) are from Malaysia where city of Kuala Lumpur is the most originating login access location with 2,116 records (84.1%). The remaining login records are from USA (37), Sweden (3), Thailand (3), Netherland (2) and Philippines (2).

5.1.2. Time Login

Time login entries stored in table data_log are the time of the server machine when the login requests were received from UAP server. In other words, if the users login at a location that has a different time zone, the server only stores the time of the server into the database. Time period is divided into three time blocks, each with a different ID based on a standard working hours as shown in Table 2.

The login records show that 95.4% of the login requests were received during the configured working hours which is the second time block in the table. 2.9% of the total records are from the first time block (7 pm- 12 am). Only 1.7% of the login access were recorded during the third time block (12 am - 8 am). Figure 3 shows the number of login events based on the time hours.

Table 2. Time Block Duration

No	Block ID	Duration
1	A	12am – 8am
2	B	8am - 7pm
3	C	7pm – 12am

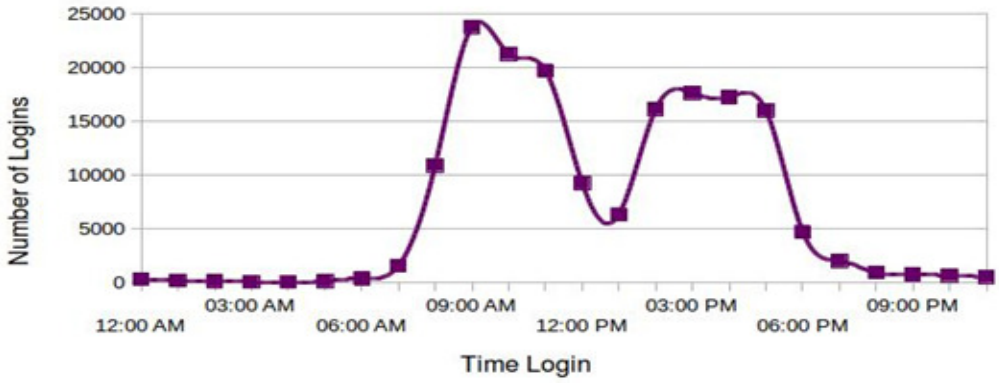

Figure 3. Number of Login with Respect to Time Hour.

The graph in Figure 3 also shows that the number of users login increases from 8 am onwards. After 6:00 pm, the number of login starts to decrease. This is normal since the standard working hours is between 8:30 am to 5:30 pm. In the afternoon, the number of login decreases for around 2 hours before increases back. It is a standard time for users to have their lunch break and stop operation.

5.1.3. Type of Browser and Operating System

Adaptive UAP is able to extract information about the user terminal by looking at the user agent string header send by the browser. There are ways to modify the string such as by using browser extensions (User Agent Switcher [1] and User Agent Selector [2]). However, in this case study, we assume that all user agent strings received by the system are original and unmodified.

Information such as type of browser and operating system running at the user terminal can be determined based on the user agent string. Figure 4 shows an example of a user agent string. In this example, the type of browser and operating system are Safari and iOS 8.1 respectively. It is also clear that the user was using an iPhone smartphone.

Mozilla/5.0 (iPhone; CPU iPhone OS 8_1 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) CriOS/39.0.2171.50 Mobile/12B411 Safari/600.1.4

Figure 4. Example of User Agent String Header.

Figure 5 and 6 show the distribution of type of browser and operating system the users used during the data collection period. In term of type of browser usage as shown in Figure 5, majority of login records are from browser Chrome (42.2%), followed by Firefox (33.7%) and Mozilla (16.7%). From the operating system stand point, as shown in Figure 6, 92.2% of the login records are from Windows operating system. The second most popular operating system is Mac OS (4.3%), followed by Linux OS (2.2%).

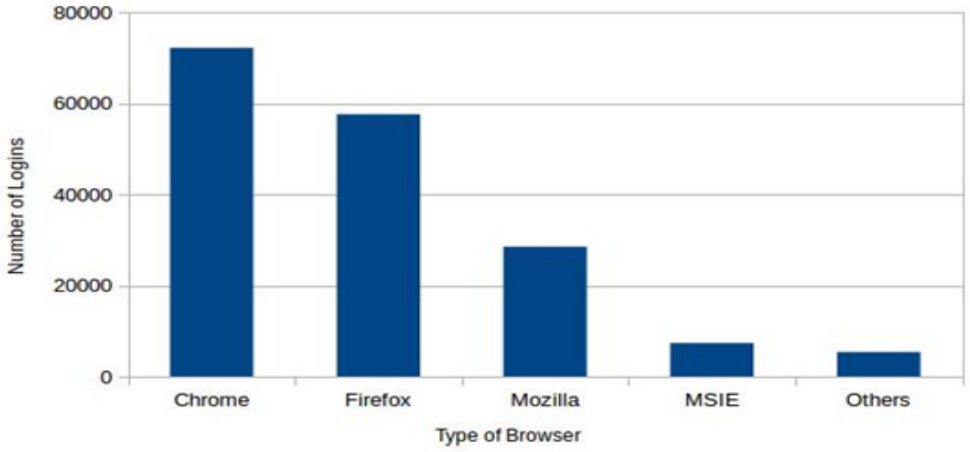

Figure 5. Number of Logins with Respect to Type of Browser

Trust Engine regards both entries of browser and operating system as one attribute factor which carry one weightage value. If one of them is different, Trust Engine consider it as another entry. Figure 7 shows the number of the paired value of browser and operating system derived from the collected login records. The top four most favorite operating system is all Windows 7. In those top list, the most popular browser is Chrome (36.4%), followed by Firefox (31.4%), Mozilla (16.5%) and Internet Explorer (3.9%). The fifth place is browser Chrome with Windows 8 operating system which is 1.5% from the total number of collected login records.

5.1.4. Application

Six UAP gateway servers have been used with only one of them was intended for high trust application. The threshold trust level for low trust and high trust application were set to 10 and 30 respectively. One UAP gateway server was purposely set to 0 so that the final authentication decision by the Trust Engine is not affected by the attribute score value. Table 3 shows the list of all UAP gateway servers used.

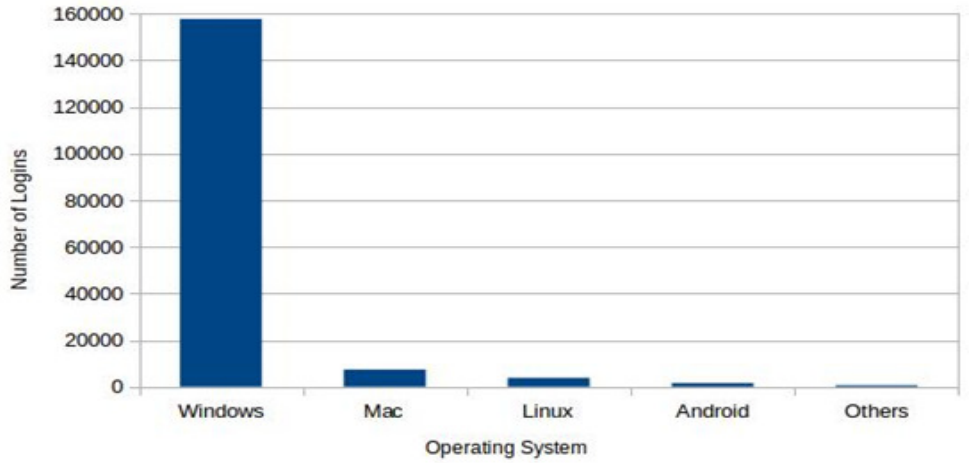

Figure 6. Number of Login with respect to Operating System

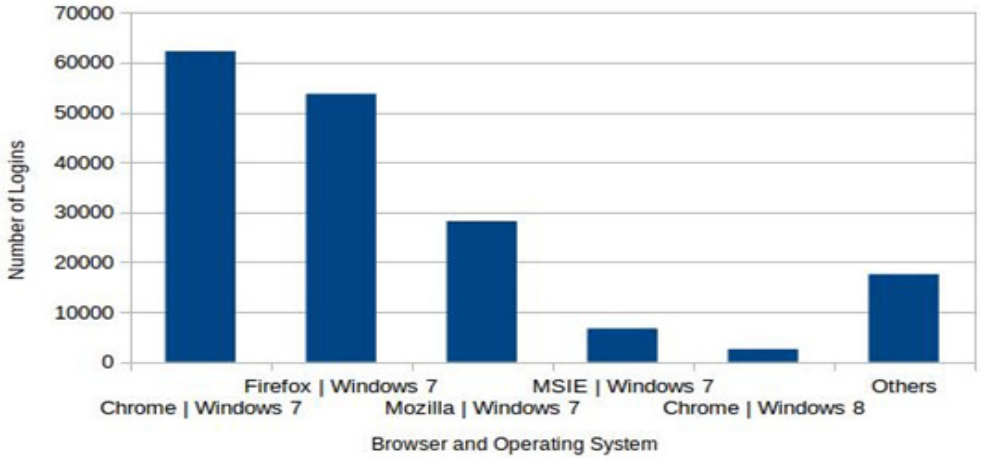

Figure 7. Number of Logins with Respect to paired Browser and OS

Table 3. List of all UAP Gateway Servers.

No	UAP Gateway Server ID	Required Trust
1.	https://uap.mimos.my/shibboleth	0
2.	https://sp.essweb.mimos.my/shibboleth	10
3.	https://bmyhdw.mimos.com/8647196938	10
4.	https://ipms-uap-gw-prod/vfdst35hj3120	10
5.	https://misso.mimos.my/shibboleth	10
6.	https://HT-MIESS1.mimos.my/shibboleth	30

There could be more than one application associated with one UAP gateway. If a user accesses two different applications which are served by the same UAP gateway, Trust Engine gets the same UAP gateway information for both login attempts.

Based on the collected login records, we found that majority (87%) of the logins were for accessing UAP gateway labelled as https://sp.essweb.mimos.my/shibboleth. The second most popular UAP gateway is https://HT-MIESS1.mimos.my/shibboleth which is assigned for high trust application. High trust applications are used to view high confidential documents. Examples of high trust applications are e-payslip to view the salary details online and e-PCB that shows employer's monthly income tax payment to the government tax return agency.

5.2. Ratio Threshold

In this section, we present the effect of User Attribute Score when different configuration of ratio thresholds are used. As explained in Section 3, the value assigned to the ratio threshold determines the qualification of the common context for every attribute factor. The higher the assigned value is, the more strict for any entry to be qualified as a common context. Each individual user may have different pattern of behavioral profile.

The same collected data that had been stored in table data_log was used as the input parameters to our testbed. The experiment setup allows us to see the effect of ratio threshold value to the number of activated attribute factor for every login access. The configuration of the testbed is shown in Table 4.

Table 4. Testbed Configuration

Entry		**Value**
Attribute		**Weightage**
1.	Geographical location	8
2.	Time	6
3.	browserOS	4
4.	Application	2
Credential		**Weightage**
1.	password	13
2.	smsPin	20
3.	otp	20
4.	certificate	40
5.	tck	20
6.	tckbar	20
Time Interval		14 days
Threshold Ratio Percentage		10,30 and 50

The number of events when the attribute factors are activated is recorded and analyzed. In each of those events, the respected attribute factors affect the outcome of the Attribute Score which then reduces the final result of the user established trust. The summary of the results from the experiment is depicted as in Figure 8. The last column labelled 'none' represents the login events where no single attribute factor was activated. In this experiment we assume that there is no adversary that wish to attack our system. In additional, all of the participating users were not aware of the profile tracking and login under their own normal behavior.

We used different values (10, 30 and 50) for ratio threshold percentage to see their affect to each Attribute Factor. From the Figure 8, we can see that in general for every attribute factor, the higher the ratio threshold is, the higher number of events where the attribute factors are activated. The number of events where no attribute factor is activated decreases when higher threshold ratio percentage is used.

Type of browser contributes the highest number of events followed by application, operating system, time login and location. We can conclude that users as individual have more tendency to use different type of browser compared to other factors. Location is the least factor. One main reason is because most of the users are the employers of the company and most of the applications can only be accessed from the company private network.

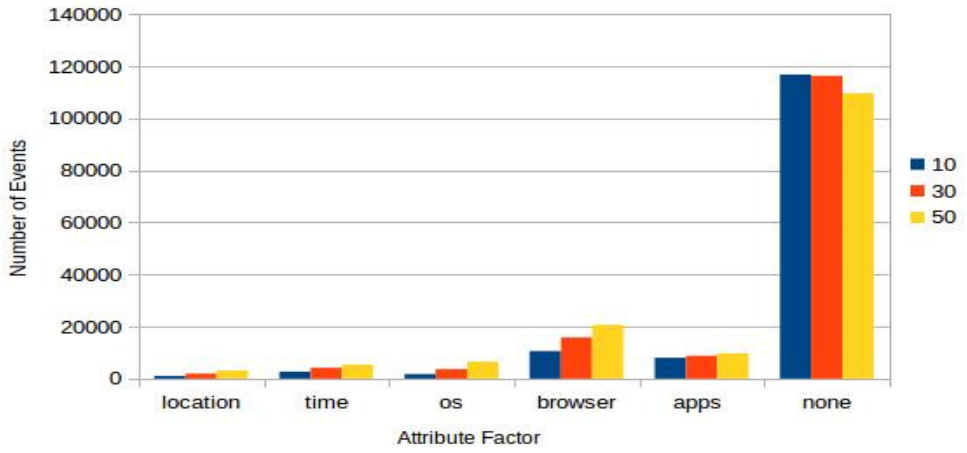

Figure 8. Total Number of Events for every Attribute Factor.

6. CONCLUSION

Adaptive authentication is an additional security layer that uses risk factor analysis to make authentication decision. Users' past login records are used to form user behavior profile. Context information in login records such as user geographical location and time login are important parameters to reflect user common behavior profile. If the user logins from a different environment from the established behavior profile, adaptive authentication calculates the risk associated with the deviation and may request the user to present additional authentication method to be authenticated. We collects actual login records from the production environment for more than 6 months. The login records profiling with regards to relevant attribute factors is analyzed. We also develop a testbed environment that uses those records to evaluate our system when different percentage of threshold ratios are used. In this paper, we present the results of the experiment. For the future works, we plan to evaluate our testbed system for other parameters setup.

ACKNOWLEDGEMENTS

We acknowledge the support provided by Ministry of Science, Technology and Innovation (MOSTI) in funding the MIMOS Unified Authentication Platform (UAP) project through the Tenth Malaysia Plan (10MP). The completion of the project allows the delivery of a centralized authentication infrastructural platform for web applications.

REFERENCES

[1] http://chrispederick.com/work/user-agent-switcher/. [Retrieved 30 January 2015].
[2] https://chromeuseragentselector.wordpress.com/. [Retrieved 30 January 2015].
[3] Khairul Azmi Abu Bakar and Galoh Rashidah Haron. Adaptive authentication: Issues and challenges. In World Congress on Computer and Information Technology (WCCIT), pages 1–6, June 2013.
[4] Khairul Azmi Abu Bakar and Galoh Rashidah Haron. Adaptive authentication based on analysis of user behavior. In Science and Information Conference (SAI), pages 601–606, August 2014.
[5] Galoh Rashidah Haron, Dharmadharshni Maniam, Vijayakumari Sadavisam, and Wong Hon Loon. Re-engineering of web reverse proxy with shibboleth authentication. In The 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012), pages 325–330, 2012.

[6] ip2location. ip2location home page. http://www.ip2location.com. [Retrieved April 2012].
[7] Shibboleth. Shibboleth documentation. https://wiki.shibboleth.net. [Retrieved January 2014].

AUTHORS

Khairul Azmi Abu Bakar received the degree in Computer Engineering from Iowa State University, USA in 1995 and master degree in Communication and Computer from National University of Malaysia in 2002. He was awarded Ph.D. degree in Electrical Engineering from University of Strathclyde, United Kingdom in 2012 for the study on free-riding nodes in an open MANET. He is currently a staff researcher at MIMOS Berhad where he has been since 1996. He has been involved in several R&D projects in the field of micro-controller, smartcard, security systems under open source platform. His primary research interests include wireless ad hoc security, authentication system and computer network.

Nor Izyani Daud was born in Kuala Lumpur, Malaysia. She received the B.A Hons Degree in Information Technology, Artificial Intelligence majoring from the Universiti Utara Malaysia in 2000; and Master Degree in Real Time Software Engineering from Universiti Teknologi Malaysia in 2006. In 2006, she joined MIMOS Berhad as a Senior Engineer. She has been working in Information Security areas; for example smart card programming, security scanning and analyzing. She also involved with CMMI-Capability Maturity Model Integration implementation in the organization.

Mohd Shafeq Hasan was born in Kuala Lumpur, Malaysia in 1987. He received the B.A Hons Degree in Information Technology, Java majoring from the Kuala Lumpur Metropolitan University College in 2011, He joined MIMOS Berhad as a Junior Engineer. He has been working in Information Security areas; for example adaptive authentication, J2EE and Spring Security