Adaptive Authentication based on Analysis of User Behavior

Khairul Azmi Abu Bakar
Information Security Lab
MIMOS Berhad
Kuala Lumpur, Malaysia
khairul.azmi@gmail.com

Galoh Rashidah Haron
Information Security Lab
MIMOS Berhad
Kuala Lumpur, Malaysia
rashidah@mimos.my

***Abstract*—Authentication is a mechanism to verify identity of users. Those who can present valid credential are considered as authenticated identities. In this paper, we introduce an adaptive authentication system called Unified Authentication Platform (UAP) which incorporates adaptive control to identify high-risk and suspicious illegitimate login attempts. The system evaluates comprehensive set of known information about the users from the past login history to define their normal behavior profile. The system leverages this information that has been previously stored to determine the security risk and level of assurance of current login attempt.**

Keywords—adaptive authentication; behavioral profile; multi-factor authentication; authentication strength

I. INTRODUCTION

Authentication is one of the fundamental issues of information system security. It is the first step in access control mechanism to protect resources from being accessed by unauthorized users [5]. A system verifies identity of users by requesting the users to present credentials which would then be validated against some authority. Users who can present valid credential are considered authenticated identities. Permissions, rights and privileges are then granted to the users based on their proven identity.

In general, credentials used in authentication process can be divided into three categories as the following:

- What you know - knowledge of some shared secret (password, PIN, etc.)
- What you have - possession of something given to the user (smartcard, token, etc.)
- What you are - either physical (fingerprint, retina, etc.) or behavioral characteristic of a user (typing patterns, habitual behavior, etc.)

In this paper, we focus on the last category which uses observation of users habitual behavior for authentication. The idea is to capitalize on the fact that humans are creatures of habit [6]. Users tend to carry the same behavior when performing their daily routines including when logging into a computer system.

By collecting login contexts for any attribute factor such as login time and geolocation, authentication system can perform analytics study to build rich and comprehensive information about the users which define their normal behavior. The system can leverage this information that had been previously stored to uncover expected behavior to determine the security risk and level of assurance of the current login attempt. If the users adhere to their normal behavior profile, the system will see it as a low risk attempt. Otherwise, the identity of the users will be questioned and the login attempt is considered as a high risk.

The remaining of the paper is organized as followed. Section II explains about Unified Authentication Platform (UAP) authentication system and introduces the Adaptive UAP. The two basic processes in Adaptive UAP are explained in Section III. In Section IV the key components of the system are discussed in details. The process in making the final authentication decision is described in Section V. To test our adaptive model, three type of scenarios are simulated which is explained in Section VI. Finally, Section VII draws the conclusions.

II. 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 free open-source project. 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 Fig. 1 [3].

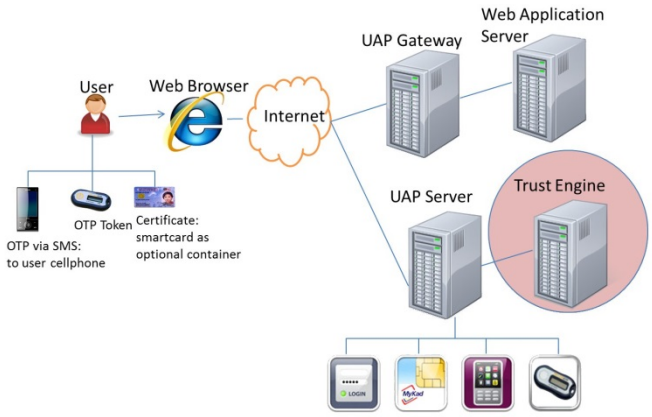

Fig. 1. Adaptive UAP Architecture

In the new version of UAP, called adaptive UAP, we introduce an additional component called Trust Engine which incorporate adaptive control based on security risk and level of assurance. When making authentication decision, Trust Engine takes into account attribute factors from the user normal profiles which had been previously analysed and stored.

III. Processes in Adaptive UAP

Adaptive UAP consists of two basic processes: Pattern Generating and Trust Evaluating [1].

A. Pattern Generating Process

The pattern generating process is responsible to analyze the users behavior from the past login records and produce the corresponding user attributes profile. As depicted in Fig. 2, there are three components in the pattern generator: events storage, patterns generator and patterns storage. The events storage contains past login records. Each record has contexts information about user login behavior. Examples of these records are shown in Table IV.

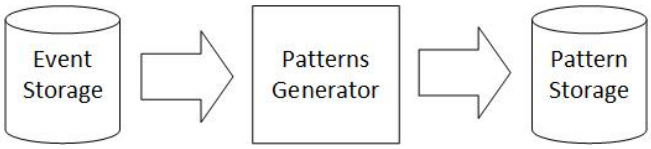

Fig. 2. Pattern Generating Process

The pattern generator is the main component in the pattern generating process. The pattern generator analyses context information inside the login records stored at the event storage. Not all login records are been used. Only login records that had occurred in the last predefined period of time will be taken into account. Login records beyond this period of time are considered obsolete and will be ignored. Contexts information for attribute factors such as login time and geolocation are converted into a meaningful data format before the output records along with their number of occurrence are stored at the pattern storage.

This information will be used in the next trust evaluating process. Examples of output records stored at pattern storage are shown in Table V. Each attribute factor is assigned an unique attribute ID (attrID) number and also a weightage value as depicted in Table III. User behavior profiles comprised of all of these attribute factors.

The pattern generating process analyses past records for all users in the system. This process consumes a lot of computing power since it needs to retrieve and analyze a great amount of data. As a result, in Adaptive UAP, the pattern generating process is scheduled to run only at midnight or when the system is not busy.

B. Trust Evaluating Process

The trust evaluating process is the second process in adaptive authentication. The trust evaluating process is responsible to analyze, decide and act upon every login request from users. As illustrated in Fig. 3, the trust evaluating process contains five components: contexts collector, patterns storage, trust calculator, challenger and events storage.

The context collector processes a collection of data reflecting the current user login attempt parameters. All of the data except time login are passed from the authentication server. The time login information context uses the time clock of the server. The geolocation of the user is identified based on the IP address sent by the authentication server. Patterns storage contains user attribute profiles generated from the first process mentioned before.

The trust calculator is the main component of the trust evaluating process. The trust calculator compares the current contexts processed by the context collector with the user attribute profiles retrieved from the pattern storage to decide the total trust score of the user. The challenger component determines the system response based on the final trust score of the user calculated by the trust calculator and the threshold level of the application that the user wants to access. The response could be either to grant access to the user or challenge the user to provide additional credential or in the extreme cases, block the user from logging into the system. The events storage stores the data from the context collector and the decision from the challenger. The events storage will be used by the pattern generating process which then completes the closed-loop mechanism for the two processes.

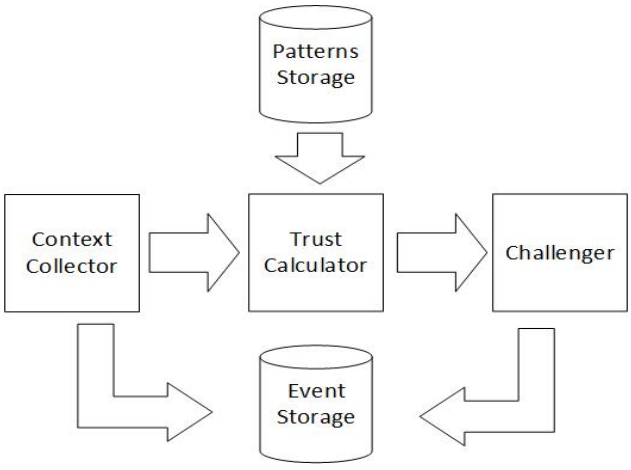

Fig. 3. Trust Evaluating Process

IV. System Configuration

This section discusses about key elements of adaptive UAP system. It also discusses about the configuration setting customized for the system.

A. User Factors

There could be many user attribute factors that can be analyzed from the past users login records but in this case study, only the following four factors are being considered:

a) Login Time

b) User Geolocation

c) Application been Accessed

d) Type of Browser and Operating System

1) Login Time

The time of the login is an important indicator to determine a high risk login attempt. Each individual user usually follows the same time pattern when to become active (working/socialize) or to become inactive (sleeping/leisure). For this system, the time period is divided into three blocks based on a standard working hours as shown in Table 1. The division of this time block should be seen as a starting point to design a working system and can be customized later to suit each system requirements.

TABLE I. Time Block Duration

No	Block ID	Duration
1	A	12 am – 7 am
2	B	7 am – 6 pm
3	C	6 pm – 12 am

2) User Geolocation

Another user factor that the system is considering to determine user behavior is the geolocation (geographical location) of the user. If the user is logging at a different location from where the user normally did in the past, the system should be suspicious about the identity of the user. There are many methods to track the geolocation information. GPS (Global Positioning System) which may be integrated or attached to the machine the user is using can give very accurate location information. However this solution has significant implementation costs in term of hardware purchasing and distribution in the enrollment process. An economical ways is by using the IP address of the machine. There are numbers of free and paid subscription geolocation database records that can identify the real-world geographic location of an object based on its IP Address. Adaptive UAP uses solution from ip2location [4] which are available as database records to provide geographical information such as the name of the city, region and country of origin for the IP Address.

3) Application Attribute

Adaptive UAP consists of a centralized authentication server that allows Single Sign-On (SSO) capability to access multiple applications. Users do not need to have multiple login ID and this provides convenience and time saving for both organization and users. Once the users have been authenticated by the system, they may not need to go through the same authentication process to gain access to other applications. In Adaptive UAP, each of the application is assigned a unique application ID. The application ID is recorded every time user successfully login into the system to study the user's behavior. If the user tries to access an application that the user is not commonly used, the system will see this as a abnormal request and may raise the risk level.

4) Browser and Operating System

There are many types of browsers and operating systems available in the market. Some of the popular web browsers today are Microsoft's Internet Explorer (IE), Mozilla Firefox, Google Chrome and Apple's Safari. In term of operating system, popular examples include Microsoft Windows, OS X, BSD (Berkeley Software Distribution) and Linux. Users tend to have their own preference on browser and operating system. One reason for that is each type of browser and operating system has its own unique look and feel. Adaptive UAP is able to extract the information about the browser and operating system from the user agent string header send by the browser. Adaptive UAP studies that information to determine the user's preference browser and operating system.

B. Authentication Methods

Adaptive UAP can support various types of authentication methods but in this case study four authentication methods are being considered as the following:

1) Username/Password

Password is a secret word or string of characters most commonly used to authenticate a user. Authentication system simply checks if the person claiming to be the actual user knows the secret. The main problem with password is that it is generally static which makes it vulnerable when the password is copied, sniffed or even guessed by an attacker. Long and complex passwords are harder for the attacker to guess but also harder for the actual user to remember.

2) OTP Token

OTP token is a hardware device to display a secure One-time-Password that can be used only once and will be changed after every 60 seconds. Such short validity of the password can prevent attackers to use the password after valid intervals. The password is generated based on security cryptography that uses the current time and the initiated secret key. As a result, the real time clock of the user's token device and the authentication server need to be synchronized before the token is distributed to the actual user.

3) smsPIN

smsPIN is a key generated by the system that is only valid for one login session or transaction. In contrast to OTP Token, there is no time validity to use the password. Furthermore, there is no need to have a special hardware device at the user side to display the password. The key is sent by using SMS messaging system to the user's personal mobile phone that has been pre-registered at the system.

4) Digital Certificate

Digital certificate is an electronic document that uses a digital signature to bind a public key with an identity. The certificate normally contains the name of the certificate holder,

a serial number, certificate's expiration date, holder's public key and the digital signature of the certificate-issuing authority (CA). Digital certificate can be verified because it is issued by an official, trusted agency. The digital certificate can be stored inside a smart card or in the form of software file. Many digital certificates conform to the X.509 standard [2].

C. Sampling Duration

Each individual user could have different environment factor as time goes by. For example, a user who has just been transferred to a new place would have recorded new geolocation information. For this reason, adaptive UAP only consider sampling records from a predefined period of time. In our case studies, the time limit is set to 14 days. That mean, records older than 14 days are considered obsolete and not used to establish user behavior profile.

D. User Behavior Profile

In order for any entry in the attribute factor to be labeled as a user common behavior, it should satisfy the following requirements.

1) the number of user records for the last 14 days is more than 10

2) the frequency of occurrence for any particular context is more than 30% of the overall records

Adaptive UAP should have enough number of user records before it generates user behavior profile. In this case study, we choose the minimum number of records required to 10. To qualify as a common behavior, the occurrence of the context should exceed a minimum ratio which is set to 30%. For instance, if the user has login 100 times for the last 14 days and from those 100 records, the user has used browser Chrome and Windows Operating System more than 30 times, that browserOS entry is considered as the user common behavior for the attribute factor. Since threshold of 30% is used, at one time, there could be a maximum number of three different entries for any user attribute factor. In the example above, if each browser Firefox, Chrome and Safari exists more than 30 times in the 100 records, all of them are labeled as normal browsers.

V. Decision Making Process

The decision whether the user who is gaining access is authenticated or not depends on three factors as shown in Fig. 4.

A. Authentication Methods

Adaptive UAP authentication system supports multiple authentication methods including username password combination and digital certificate. Users have the option to select any of these authentication methods to get authenticated. Each of the method is assigned authentication strength. The initial authentication strength for a user is zero. Each time the user presents a valid authentication method to the system, the user would acquire additional authentication strength. The final authentication strength is the accumulated strength of all different valid authentication methods that have been presented. The values of the authentication strength for the authentication methods are depicted in Table II. For example, if the user presents valid username/password and smsPin, the authentication strength is 31.

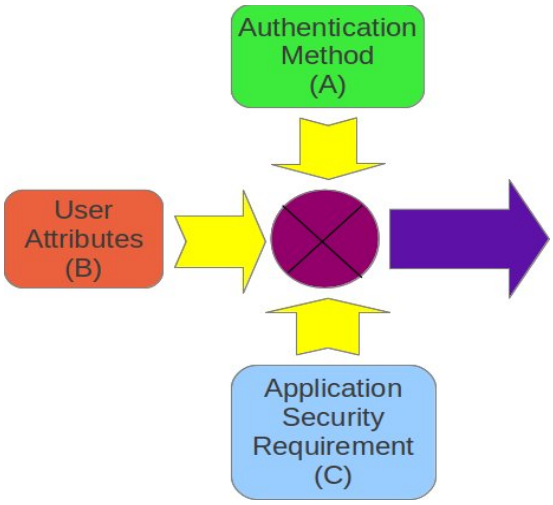

Fig. 4. Decision Factors

TABLE II. Authentication Strength for Authentication Methods

No	Authentication Method	Strength
1	Username/Password	13
2	smsPin	18
3	OTP Token	20
4	Digital Certificate	40

B. Application Security Requirement

The second factor the system takes into consideration when making the authentication decision is the application security requirement. Each application has a threshold level which denotes the minimum required trust score the user need to acquire to be authenticated.

The value of the required trust level is assigned by system administrator and depends on the sensitivity level of the application. Sensitive applications such as online banking would require a high trust score while insensitive applications as leave application would require a low trust score. As described later in this paper, we use both high and low trust applications with required trust level of 30 and 10 respectively to test our system model.

C. User Attribute

The final factor used in the authentication decision is the user attribute. The acquired trust score gain from the authentication method can be reduced based on the security risk level derived from the user attribute factor. If a user logins under different environment from the user's formed attribute profile, the system will consider the login attempt as a risk and reduce the user's trust score. The user may have to present additional authentication method to raise the trust score to reach the minimum trust level required by the application. The penalty or the reduction weightage assigned to the user attributes are shown in Table III.

TABLE III. USER ATTRIBUTE PENALTY WEIGHTAGE

No	User Attribute	Ratio	Weightage
1	Application	0.1	4
2	browserOS	0.2	8
3	Time login	0.3	12
4	Geolocation	0.4	16
	TOTAL	1.0	40

The formula to calculate user attribute score is shown in (1).

$$\text{ATTRIBUTE\_SCORE} = ((\text{TIME} * \text{WEIGHT}_{\text{TIME}}) + (\text{GEOLOCATION} * \text{WEIGHT}_{\text{GEOLOCATION}}) + (\text{BROWSEROS} * \text{WEIGHT}_{\text{BROWSEROS}}) + (\text{APPLICATION} * \text{WEIGHT}_{\text{APPLICATION}})) * (\text{MAX\_USER\_SCORE}) \quad (1)$$

The formulating of the attribute score depends on the applicable user attributes during user login attempt. For instance, browserOS attribute factor would be applicable in the formula only when it meets the following two conditions:

1) user has common behavior with respect to browserOS attribute as explained in Section IV.D.

2) user logins using browserOS different from that common behavior

If all those four attribute factors do not meet the conditions, the final attribute score will be zero. There will be no reduction to the trust score. The maximum possible user attribute score can be set at max_user_score parameter which is currently configured to 40 in this case study.

D. Final Decision

The final decision depends on the values of all three above factors (Authentication Methods, Application Security Requirement and User Attributes). Basically the requirement as in Formula (2) below should be satisfied before the user is considered an authenticated entity by the system.

$$A - B >= C \quad (2)$$

WHERE

A: AUTHENTICATION METHOD STRENGTH

B: USER ATTRIBUTE PENALTY

C: APPLICATION SECURITY REQUIREMENT

VI. EXPERIMENTAL RESULTS

To test our system model, we consider three consecutive scenarios. For simplicity of this case study, we assume that the user always login under the following attributes:

1) User ID = '04ce397'

2) Application ID= 'spid5' (low trust)

3) Location = MIMOS Berhad, Kuala Lumpur

4) Time login = 7am - 6pm

In the first scenario, a user logins with no assigned attributes profile. The user presents a valid username/password credential to access a low trust application. In this case, only factors from authentication method and application security requirement contribute to the authentication decision. User attribute factor is not applicable since the user does not have attributes profile formed yet. Based on Table II, the acquired trust score value for username/password credential is 13 which is sufficient for the user to be authenticated to access the low trust application which has a security requirement value of 10. The system then stores the login records into event storage which resides in table data_log as shown in Fig. 3.

During pattern generating process which is scheduled to run every midnight, the system retrieves the user's login records for the last 14 days. For simplicity, let assume in the last 10 records, the user had always login under the same attribute factors. The contents of events storage in table data_log are as depicted in Table IV.

Adaptive UAP system analyzes the attribute factors mainly the geolocation, time login, application accessed and type of browser/OS used. Note that data fields time_login and ip_int contain raw data of the user records. The data are converted into meaningful entries during pattern generating process as described in Section III.A. Table common_attr now contains the user's login contexts along with the number of their occurrences as shown in Table V.

TABLE IV. DATA_LOG AFTER 10 USER INTERACTION

Uuid	Time_login	browserOS	Ip_int	appID
04ce397	09:24:53	Chrome|Windows	1023804686	Spid5
04ce397	11:02:23	Chrome|Windows	1023804686	Spid5
. . . . .				
04ce397	16:10:01	Chrome|Windows	1023804686	Spid5

TABLE V. USER'S ENVIRONMENT FACTORS IN TABLE COMMON_ATTR

uuid	attrID	entry	count
04ce397	1	2	10
04ce397	2	Spid5	10
04ce397	3	Kuala Lumpur	10
04ce397	4	Chrome|Windows	10

In the second scenario, the same user logins into the system when the user has established behavior profile. The user chooses the same username/password credential to access the same low trust application. The user also login under the same attribute factors as before but this time the user uses browser Firefox instead of Chrome. From Table V, the user has browser Chrome which is qualified as a common browser in the attribute profile. Browser Firefox has never been used by the user before and thus is not the user's common browser. In this case, user attribute factor of browser/OS is applicable in the

authentication decision. The security requirement value for a low trust application is 10. Username/password has weightage value of 13. From Table III, browser/OS factor has a penalty value of 8. After deduction of 8 from browser/OS factor, the user final acquired trust score is reduced to 5 which is not sufficient for the user to be authenticated. The system will show the user again login display where the user can choose another authentication method. The username/password option is disabled to the user as it had been chosen before in the login session.

In the last scenario, the system adapts the required trust score based on the new change of user behavior. In this scenario, the user continues login into the system using browser Firefox. Each time, the user has to present two authentication methods since browser Firefox is not considered as normal browser under the user established attribute profile.

When the system executes pattern generating process, the system perform the analyses again on the user past records. Let say the user has login another 5 times by using browser Firefox and those login records are still within the 14 days' time frame. Pattern generating process will construct a new user behavior profile manifested by the contents of common_attr table as shown in Table VI.

TABLE VI. COMMON_ATTR AFTER THIRD SCENARIO

uuid	attrID	entry	count
04ce397	1	2	15
04ce397	2	Spid5	15
04ce397	3	Kuala Lumpur	15
04ce397	4	Chrome|Windows	10
04ce397	4	Firefox|Windows	5

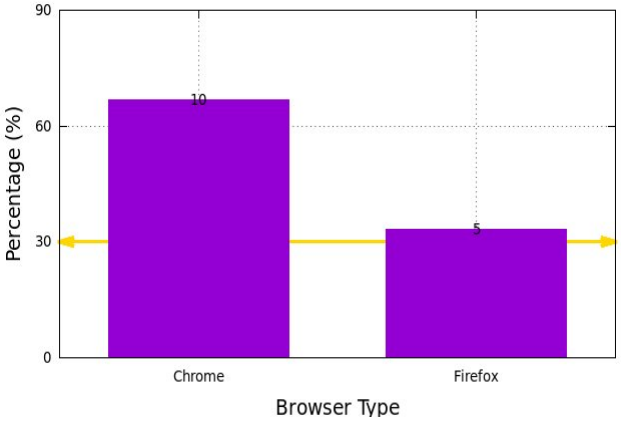

Fig. 5. Analysis on User's Registered Browser

As shown in Fig. 5, at this point, browser Firefox has been used 5 times in 15 login records which is more than the minimum threshold (30%) set for a normal attribute factor. The next time the user login again by using either browser Firefox or Chrome, browser/OS factor is not applied in the trust calculation since both browsers are now assigned as user's normal browsers.

Assuming all other attribute factors remain the same, the user can now login into the low trust application just by presenting a valid username/password credential.

VII. CONCLUSION

Adaptive authentication is a means of providing additional security layer to an authentication system. Adaptive authentication system looks at various parameters such as the time the logon request is taking place and the location of the users to form their behavior profile. It will then takes into consideration that behavior profile to confirm the identity of the users. Any deviation from the profiles is considered as a potential risk and the system may request the users to perform additional steps to verify their identity. In this paper, we presented our Unified Authentication Platform (UAP) which incorporates adaptive element. We introduce two general processes that are used in the adaptive UAP. The system defines behavioral profile based on the contexts from the past history records. The profile is then compared against the current context to come out with the final trust score which characterizes the login attempt. All of these processes are transparent to the end users. The key components in the new system involved in the authentication decision are explained in details.

To explain how the trust the system works, three different scenarios are used in our case study. The three scenarios simulate how the system generates behavior profile to measure trust value of the user login and adaptively adjusts the value when change of behavior is detected. For future work, we plan to use actual users' login data to evaluate the effectiveness and usability of adaptive UAP.

ACKNOWLEDGMENT

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] 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.

[2] Network Working Group. Internet x.509 public key infrastructure certificate and crl profile rfc:2459, January 1999.

[3] 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 Transanctions (ICITST-2012), pages 325–330, 2012.

[4] ip2location. ip2location home page. http://www.ip2location.com. [Retrieved April 2012].

[5] Pierangela Samarati and Sabrina De Capitani di Vimercati. Access control: Policies, models and mechanisms. In Foundations of Security Analysis and Design. Springer-Verlag, 2001.

[6] Elaine Shi, Yan Niu, Markus Jakobsson, and Richard Chow. Implicit authentication through learning user behavior. Information Security, 6531:99–113, 2011. Lecture Notes in Computer Science.

[7] Shibboleth. Shibboleth documentation. https://wiki.shibboleth.net. [Retrieved January 2014].