5-2018

Implementing Resiliency of Adaptive Multi-Factor Authentication Systems

Kim Phan
St. Cloud State University, kimgwen@gmail.com

Implementing Resiliency of Adaptive Multi-Factor Authentication Systems

by

Kim Gwen Phan

A Starred Paper

Submitted to the Graduate Faculty of

St. Cloud State University

in Partial Fulfillment of the Requirements

for the Degree of

Master of Science in

Information Assurance

May, 2018

Starred Paper Committee:
Abdullah Abu Hussein, Chairperson
Susantha Herath
Jie Hu Meichsner

Abstract

Multifactor authentication (MFA) is getting increasingly more popular to safeguard systems from unauthorized users access. Adaptive Multi-Factor Authentication (A-MFA) is an enhanced version of MFA that provides a method to allow legitimate users to access a system using different factors that are changing based on different considerations. In other words, authentication factors include passwords, biometrics among others are adaptively selected by the authentication system based on criteria (e.g., whether the user is trying to log in from within system boundary, or whether the user is trying to access during organization operating hours). The criteria (i.e. triggering events) that A-MFA uses to select authentication factors adaptively are usually pre-defined and hard-coded in the authentication system itself. In this paper, the graphical user interface application is designed to add more resiliency to the existing Adaptive Multi-Factor Authentication (A-MFA) method by enabling system administrators to rank the triggering criteria based on the users' roles, system assets, tolerance to risks, etc. The proposed tool allows system administrators to determine when to tighten and soften user access to the system. The tool uses multiple criteria decision making (MCDM) method to allow system admins to access the trustworthiness of user. Based on the trustworthiness of the user, the tool selects the number and complexity of the authentication methods. This tool will help to utilize the systems administrator situational awareness to improve security. This work aims to preserve the AMFA strengths and at the same time give system administrators more flexibility and authority in controlling access to systems.

Keywords: Adaptive Multi-Factor Authentication (AMFA), One-Time Password (OTP), Biometrics, Security, Authentication, Integrity, Threat, Situational Awareness, Multiple Criteria Decision Making (MCDM), Access Control, Role-Based Security.

Acknowledgements

I would like to express my sincere gratitude to my starred paper advisor Professor Abdullah Abu Hussein for his continuous support, guidance, and immense knowledge, which led to the successful completion of my starred paper and master program. I also extend my thanks to the rest of my starred paper committee: Professor Susantha Herath and Professor Jie Hu Meichsner for their advice, encouragement and support throughout my starred paper and education at St. Cloud State University.

I would like to thank the Information System, Computer Science, Computer Networking and Application Faculty at St. Cloud State University for providing me high education, technical skills and necessary resources required to write this research paper and throughout my master program.

I also express my deeply thanks to my family and friends for supporting me to do the best to complete my master program.

Table of Contents

Page

List of Tables ... 6

List of Figures ... 7

Chapter

I. Introduction ... 9

Introduction ... 9

Problem Statement ... 11

Nature and Significance of the Problem ... 11

Objective of the Research ... 13

Research Questions/Hypotheses ... 13

Definition of Terms ... 14

Summary ... 16

II. Background and Review of Literature ... 17

Introduction ... 17

Background Related to the Problem ... 17

Literature Related to the Problem ... 18

Literature Related to the Methodology ... 24

Summary ... 35

III. Methodology ... 36

Introduction ... 36

Design of the Study ... 36

Data Collection ... 43

Chapter Page

Proposed System 43

Tools and Technology 44

Hardware and Software Environment... 45

Summary 45

IV. Data Presentation and Analysis.. 46

Introduction... 46

Data Presentation.. 46

Data Analysis... 59

Results 59

Summary 69

V. Results, Conclusion, and Recommendations 70

Introduction... 70

Discussion 71

Conclusion... 72

Future Work... 73

Recommendation 75

References.. 76

Appendix: User Interface Programming Codes 81

List of Tables

Table Page

1. Trustworthiness Scores 42

List of Figures

Figure Page

1. Sample Scenario of Authentication Process 19

2. Ransomware Spreading 22

3. Data Breach Continue to Rise 23

4. Consumers Learning 2FA 24

5. Smart Card Credential Insurance 33

6. MFA Everywhere 34

7. Flow Chart of Implementing Resiliency of Adaptive Multi-Factor Authentication Systems 38

8. Login Page 39

9. Access Data of Operation Criteria 39

10. Implementing Resiliency of Adaptive Multi Factor Authentication Systems Flow Chart 49

11. Adaptive MFA Administration Controller Flowchart 52

12. Adaptive Multi Factor Authentication Administration Controller Interface 53

13. User Sequence Diagram 58

14. Login Page Implementation 62

15. Registration Page 63

16. SQL Server Database Entity 63

17. Security Questions 64

18. Random Number Generation 65

19. Random Number Output 65

Figure Page

20. Confirmation Number 66

21. Entity Relationship Diagram 68

Chapter I: Introduction

Introduction

In the computing environment, business applications had been used by many users worldwide. With the advancements of information technology, most user online access to the online accounts had counted on various online services, which needed to be secured and trusted in a way to prevent the thorny issues of illegal access, identify theft and data breaches. According to O'Leary (2017), the authentication problems were still increasing dramatically due to dynamic threats, the application security statistics reported that 81% hacking breaches of stolen passwords, and 93% financially compromised by criminals. These incidents affected user's tremendous burdens and insecure accesses the online system. Authentication method was the mandatory factor to address the trustworthiness, to identify user credentials and to restrict illegal and unauthorized access to the system. For instance, authentications through a single factor with user ID and password. If a single factor authentication mechanism failed, the users could not get access to the online systems until a system administrator checked and recovered the actual system. Thus, the single factor authentication was suffering from some significant pitfalls.

To improve the single factor authentication issue, authentication through additional factors was needed for system administrator to enforce data security policies and procedures on all database levels. So that only legitimate users could have right permissions to get access to computing systems. The use of multiple authentication factors with various weights associated with pre-defined criteria made it harder for intruders or malicious attackers to gain unauthorized access to the systems. Most of authentication systems in use nowadays verify a user's credentials during the login time to the systems. For example, two-factor authentication systems used in different email servers that had been checked for two separate factors at the time of accessing the online services for the first time but did not validate the second time throughout the ongoing session; thus, this scenario could increase the chance of compromising user credentials and the authentication was not verified throughout the ongoing session of any user who opened a back door for hackers to imitate the actual user to login to the systems. In addition, mobile technology continued to increase user's access to online systems. Thus, checking the authenticity of the registered users daily was very important for system administrator to protect sensitive data from tampering or unauthorized attempts. Therefore, the trustworthiness algorithms enhanced the need for system administrator to increase or decrease the resiliency of adaptive multi-factor authentication system.

Multifactor authentication was a secure authentication that was required one more methods of authenticate technique, which was selected from further criteria selections. This method was used to double check the users' identity prior to accessing the sensitive and confidential data (Centrify). MFA added a layer of security that allowed system administrator to link two or more types of authentication to provide better way of authenticating users. By doing this technique, it protected against the compromised data. The most common four types of authentication factors were: the first one was "something the user knows", for example: username, password, PIN or security questions. The second one was "something the user have" that was the device of user possesses like the smartphone device or smart card. The third one was "something the user are" that was a user's physiological traits, for instance, biometrics, fingerprint, retina scans or voice recognition. The last one was "where the user is" that was a user's location, for example IP address to identify the geographic location of the users (Bolle et al., 2004).

Problem Statement

The criteria that A-MFA was used to select authentication factors adaptively were usually pre-defined and hardcoded in the authentication system. The goal of this program approach was to give system administrators more authority to make decision and to control over tightening and loosening triggering events by enabling the system to change the importance and assessment of triggering events. These events were based on the organization requirements, user access roles, system assets, and factors of authentication.

Nature and Significance of the Problem

The challenges that organizations and/or individuals faced many events were to safeguard systems against the unauthorized access and/or malicious attacks. Adaptive Multi-Factor Authentication (A-MFA) provided a method to allow legitimate users to access a system using different authentication factors. These factors were changing triggering events based on different considerations. The criteria (i.e., triggering events) that A-MFA used to select authentication factors adaptively were usually pre-defined and hard-coded in the authentication system itself. In this paper, the user interface program was analyzed and designed as a prototype to add more resiliency to the existing Adaptive Multi-Factor Authentication (A-MFA) method by enabling system administrators to rank the triggering criteria based on users' roles, system assets, tolerance to risks, security network, etc.

The number of incidents continued to rise significantly, and data breaches were making alerts to the media and online users weekly. Adaptive MFA had become the norm to prevent unauthorized users from accessing corporate data and/or individual accounts. Based on the above incidents, the system administrator would take actions to play a vital role to decide the importance of triggering events based on the MAC address or IP location, time frame, and IP address. This approach was important for information security purposes. because it helped the system administrators to decide important scenarios, triggering events via the user interface to figure out which events were important for increasing the trustworthiness scores or decreasing the trustworthiness scores based on the factors of authentication. Thus, it helped organizations to increase the complexity, flexibility and the number of authentication factors. Managing information security was a major challenge in business organization, thus system administrators should protect information and network security from unauthorized user who attempted to capture legitimate user’s credentials stored in the system.

In this paper, the trustworthiness of the user was measured and calculated based on the criteria or triggering events in which the A-MFA uses to select authentication factors adaptively to the computer systems. System administrators had authorities to change the existing pre-defined coded to assign criteria via user IP address, time login, MAC address, and to grant access control privileges associated with the authentication rules to user. The access control privileges were based on the user’s roles, system assets, time login and device location.

Objective of the Research

The objective of this study was to analyze the trustworthiness of the user roles, system assets to increase resiliency of A-MFA systems, which were highly important for system administrators to define proper access control levels of adaptive authentication for user privileges. The tools allowed the system administrators to determine when to increase or decrease appropriately the resiliency of A-MFA method to grant user access to the systems.

The logical algorithms were analyzed and designed as a prototype with sequence diagram, entity relationship diagram designed associated with attributes, entity relationships of relations resided in database. The user interface was based on the scales of triggering events to increase resiliency of A-MFA method for system administrators to evaluate the effectiveness of authentication factors. This study was very critical for the system administrator to improve the accuracy and complexity of adaptive MFA systems.

Research Questions/Hypotheses

The proposed study’s research questions could be answered upon completion of the research study to impellent the interface programs, the questions illustrated below:

1. Any threats or risks when the organization or individuals used online applications to access to the computer systems?
2. Legitimate user could efficiently login to the system on a regular basis?
3. Where was the location of user’s device to login?

4. Device belongs to the organization or not belong to the organization?

5. Does authentication system recognize the location of the device in which the user attempted to login?

6. How was the system detected unknown user? What would had happened and how it was occurred?

Definition of Terms

Adaptive Multifactor Authentication (A-MFA): A-MFA is to adapt dynamically security and authentication policies to leverage insight from user credentials, network devices and to integrate with applications and network infrastructure.

One-Time Password (OTP): OTP is a valid code to be used for only one login session on a computer system or any digital device for securely accessing into systems (One-Time Passwords," n.d).

Computer Security: The process of preventing and detecting unauthorized users to safeguarding against intruders from using computer resources for malicious intents.

Authentication: Authentication is the fundamental defense against any illegitimate access to a computing devise or any sensitive online applications. In other words, authentication is a process of giving individuals access to the system based on user's identity via a username and password.

Integrity: Integrity is a method to ensure the accurate data from users and to safeguard from unauthorized user modification.

Threat: The potential to cause serious harm and to attack to a computer system and networks.

Situational Awareness: The ability to identify a process, to comprehend information, and to be awareness of what happening in the information technology services.

Multiple Criteria Decision Making (MCDM): MCDM is a sub-discipline of operations research to evaluate multiple conflicting criteria in decision making Multi-Factor Authentication.

Access Control: A control type of selective restriction of access to the computer resources to control access by users.

Role-Based Security: The approach to restricting system access to authorized users.

RSA Security Tokens: A type of device for displaying One-Time Passwords with a six-digit number shown on the device’s LCD screen. One-Time Passwords are only effective for a fixed period, (e.g. 60 seconds) and become invalid once the user logs in. By using a One-Time Password in combination with user name and password, the user would be able to further secure login account.

Soft Tokens: The software security token applications that generate one-time password, which is any random numbers launched on a smartphone or text, land phone code ("One-Time Passwords," n.d.).

Biometrics: The unique physical authentication methods such as retina scans, iris scans, fingerprint scans, facial recognition, and voice recognition can be used for automated recognition (“Biometric Authentication,” 2017).

Keystroke Recognition: The keystroke recognition is also the biometric authentication modality. It is used to identify the typing pattern, the rhythm of an individual.

Summary

In this chapter, the objectives of the proposed system were discussed as well as the nature of the problem and how it was overcoming the drawbacks existing in current MFA systems. The coming chapter would have described in detail about the background and literature of the paper.

Chapter II: Background and Review of Literature

Introduction

Computer system required successful user authentication before providing user access. For example, a user was requested to provide a combination of a username, a password and a geographic location to obtain access to the system. During authentication method, authentication circuitry retrieved the user profile from a database based on username, password, geographic location provided by user input. If the user was either on campus or outside of campus, system administrator would provide authentication rules to the computer system to allow that user continued to login in. If the authentication circuitry found the credentials that were not match the credentials in the system or any biometric reading do not match, then adaptive authentication rules were unsuccessful, and the user could not provide access to the computer system. Furthermore, the system administrator would provide more adaptive to the authentication system until the user was successful to get access to the online system.

Background Related to the Problem

Multiple adaptive methods of multi factor authentication improved authenticate techniques, which involved the use of network forensics of user login. The system administrators would monitor and detect any user log in, traced the log files to find if any unauthorized users and any attacks would have occurred simultaneously. The paper proposed the application for system administrators to define the weights and scores of multiple criteria, such as user IP address, login time, and MAC address. Then system admins would calculate the trustworthiness scores based on the scores of selected criteria. By calculating the trustworthiness scores, system admins would define the authentication rules and grant secure access permission to user’s roles. In controlling the authentication rules permissions assigned to the user, it would improve the network security, secure data, and to reduce the major concern of data breaches by unauthorized attempts, such as hackers, malicious attacks, insider threats, internet vulnerabilities.

Literature Related to the Problem

According to the article, (“143 Million Equifax customers affected by data breach. Here’s what you should know.,” n.d.), threats cyber security for Equifax web application compromised via customer names, SSN, birthday, address and driver’s license numbers. Hijacking attackers gained unauthorized access to the Equifax data files where 143 million of US customers hacked, 209,000 customers' credit card numbers, 182,000 customers were exposed. Cyber criminals used stolen data to access online banking accounts, insurance accounts and emails.

Additional cyber security threats occurred in Bell Canada organizations, there were over 560 million login credentials leaked online via database breaches at Yahoo, LinkedIn, MySpace, Tumblr, and Dropbox. Also, there were 17 million Zomato customer accounts compromised, encrypted passwords, email addresses. Thousands of health records compromised in the breach at a Coney Island hospital, 1.9 million customers hacked, 3,500 patient accounts were compromised, and 120,000 hashed passwords decrypted, and United Airline confidential codes leaked (Nicholas, 2017).

According to the recode reports, which revealed Yahoo’s 2013 security Breach affected three billion users hacked to steal the sensitive data of more than 145 Americans. The Senate committee requested Yahoo and Verizon to testify on Capitol

Hill with official executives (“Recode Daily: Hackers got into three billion Yahoo accounts - Recode,” n.d.).

In addition, a recent survey conducted by AICPA, information security breaches targeted to victims’ financial accounts. Cybersecurity attacks were a fraud alert to consumers. About 25% of respondents said they had been victims of cyberattacks. 82% of respondents said cybersecurity was a big concern, they were also afraid of changing their on-premise shopping to internet shopping. For small business, security threats were even more critical for online consumers (Vien, 2015).

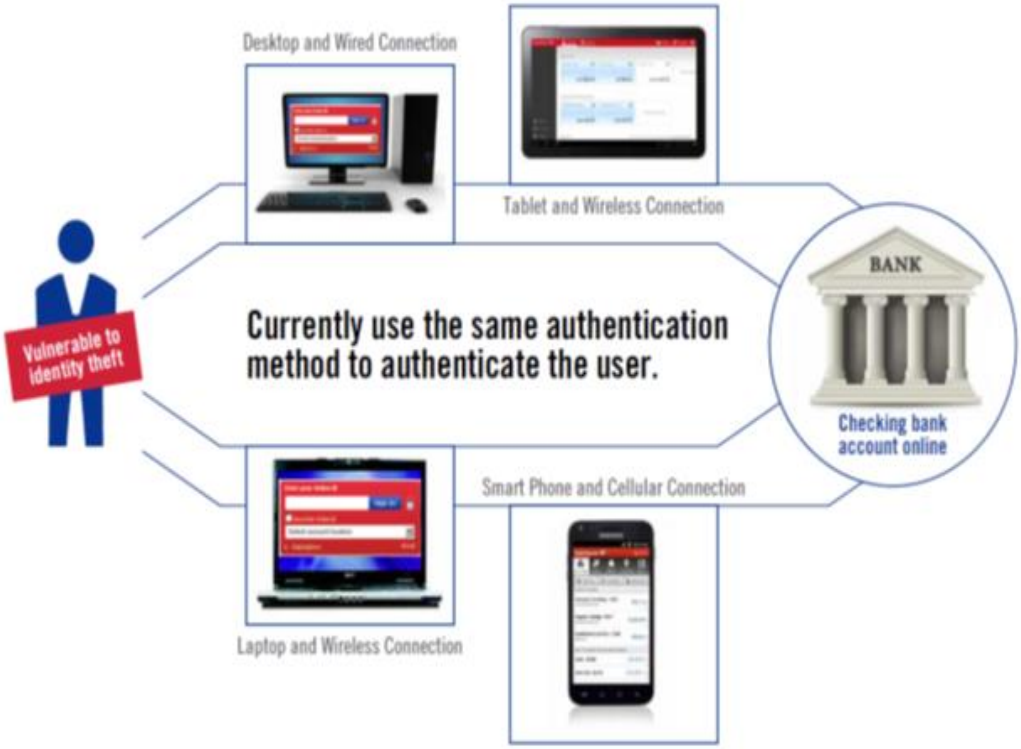

Desktop and Wired Connection
Tablet and Wireless Connection
Vulnerable to identity theft
Currently use the same authentication method to authenticate the user.
BANK
Checking bank account online
Smart Phone and Cellular Connection
Laptop and Wireless Connection

Figure 1: Sample Scenario of Authentication Process (Dasgupta, 2017).

Figure 1 shows the authentication system had been used the same authentication method to authenticate the same legitimate users in different conditions, which could lead to user’s credentials compromised. The issues were malicious thefts who could attempt to predict the possibly predictable situations from the previous history of the users' login to mimic the password in the same operating conditions, the same device at the same time. And malicious theft could get access to corporate network remotely easily to steal individual’s user ID and password. Furthermore, the same factors of generating a random number stored in the authentication system itself, the intruders could obtain the guessing number from the random selection of authentication factors stored in computer cookies. Thus, less trustworthiness were the big concerns to the end users. This would lead to the managers, system admins’ concerns regarding the security breaches; thus, system admins would involve thinking the best strategies to improve the security concern. This was an ongoing need to design a promising prototype for increasing the resiliency of A-MFA to validate users’ credentials at any given time with at different locations.

One of the issues for MFA was how to select the better way of authentication factors out of all possible choices in pre-defined events in the online system. The selection of any better set of trustworthiness scales determined the better performance of adaptive MFA solutions to provide significant benefits to the end users. This strategy enhanced system admins to develop and to implement the application scenarios to increase or to decrease the resiliency of adaptive MFA system. This would make more adaptively to mobile devices, and the diversity of authentication systems when verifying the legitimate users' credentials.

According to the latest news, the article indicated that Ransomware attacks was on the rise to encrypt the hard drive, then required the victim to provide a password to access their encrypted data. The application could be loaded through different methods in the cybersecurity space of the attack vectors to fraudulently gaining information or access to a device (Cullen, 2016). In this case, the system administrator duties were highly important to use the proper A-MFA methods to ensure that sufficient security technologies were in place to protect computer network system from the compromised data and to block the intruder attacks.

Security professionals addressed cyber security incident issues occurred recently. Due to business technology trends, the quantity of cyber security incident was increased over 80% vulnerabilities, data breaches, which led to highest risks to the corporation. During 2016, 62 families were compromised by ransomware attacks. The number of ransomware attacks increased by a factor of 11, from 2,900 to 32,091 in 2016. The duration of time that ransomware attacks tightened faster for every 10 seconds compared with 20 seconds previously. As business users increased their uses of mobile devices and data centers, which expanded the use of cloud services and it made Ransomware attacks to launch to the computer systems easier. To threaten to the end users, the attackers either searched for ransoms or persuaded the end users to provide users' name and password credentials (Cullen, 2016). These incidents illustrated the higher risks in computer security areas. Also, some businesses computer network was attacked by ransomware and data recovery was never retrieved data.

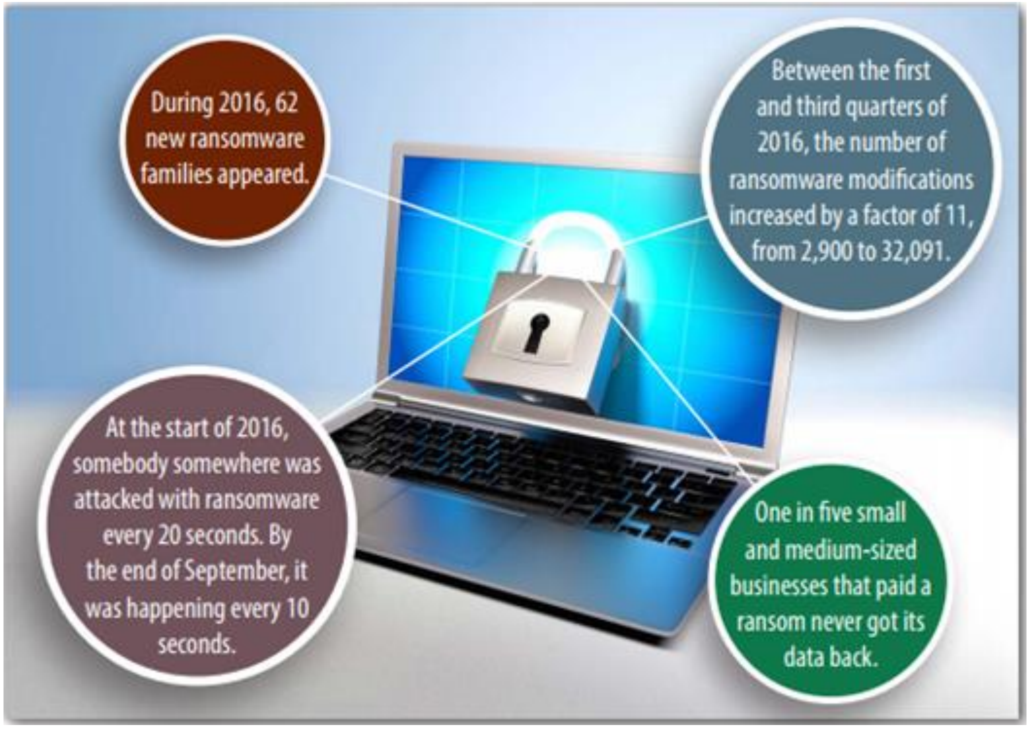

During 2016, 62 new ransomware families appeared.
Between the first and third quarters of 2016, the number of ransomware modifications increased by a factor of 11, from 2,900 to 32,091.
At the start of 2016, somebody somewhere was attacked with ransomware every 20 seconds. By the end of September, it was happening every 10 seconds.
One in five small and medium-sized businesses that paid a ransom never got its data back.

Figure 2: Ransomware Spreading (Pescatore, 2017).

Figure 2 shows the number of incidents occurred by ransomware attack. The ransomware is a form of DoS to use malware to encrypt critical information from consumers like personal account, social security number, user ID and password. Based on the above incidents, this proposal would be designed to improve the authentication system. The system administrator and/or any information technology professionals would monitor the systems continuously to increase the resiliency of authentication methods, to use flexible choices for additional authentication factors to safeguard the entire system.

Furthermore, in recent reports showed data breaches occurred in the US double compare to 2015 to 2017.

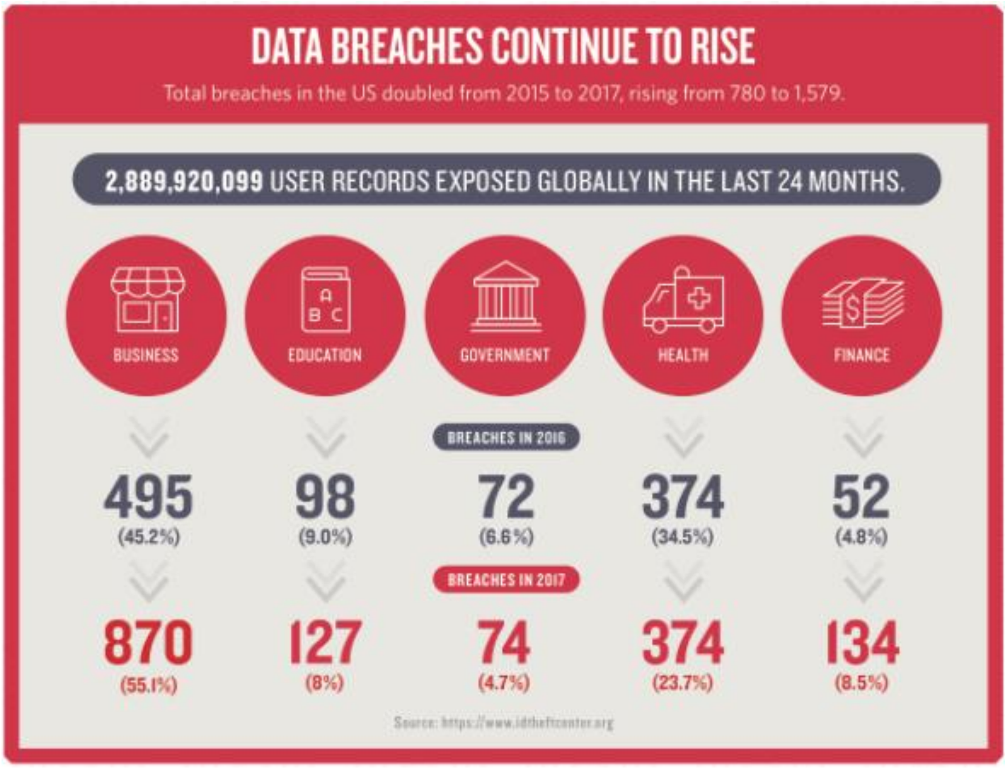
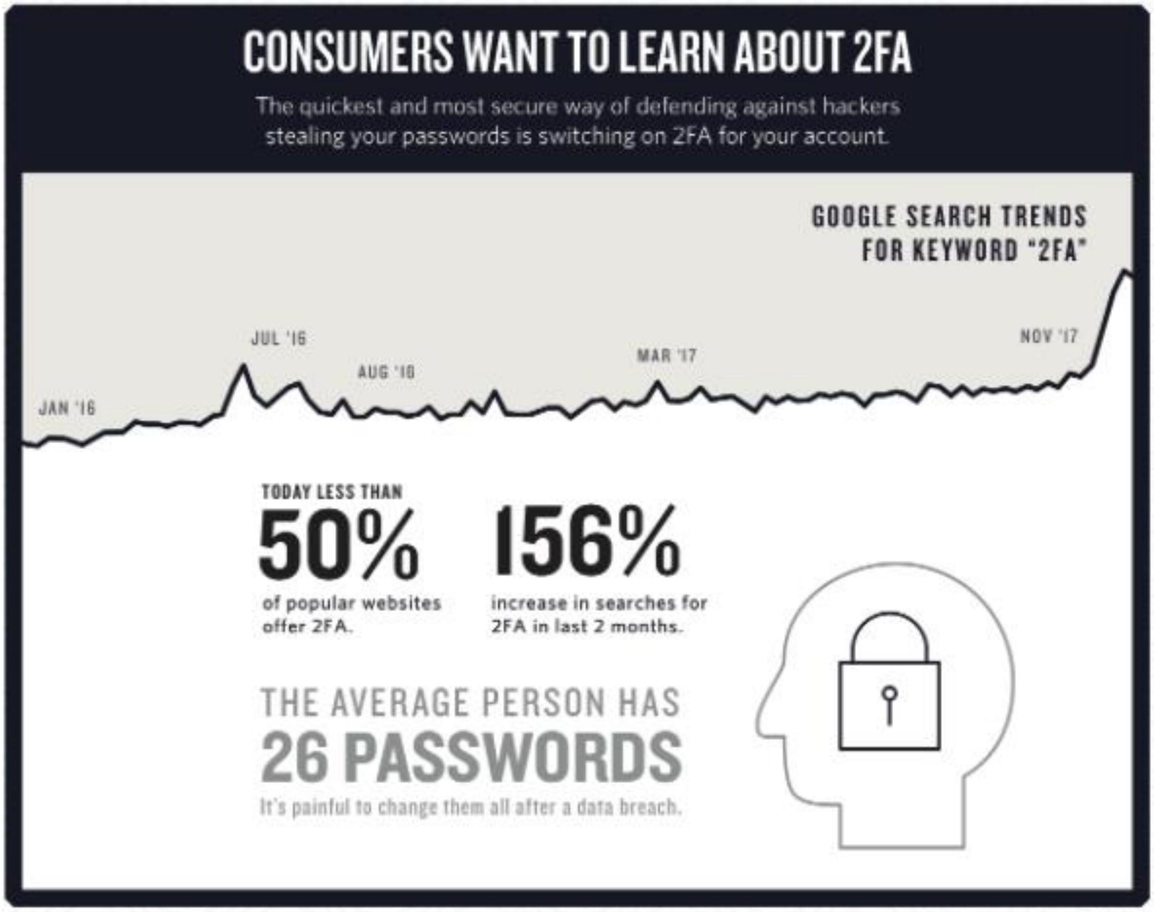


Figure 3: Data Breach Continue to Rise ("[Infographic] Is the internet getting safer?," n.d.)

Figure 3 shows that 2,889,920,099 user records exposed globally, data breaches increased rapidly recently from 2015 to 2017. Therefore, data breach was an alarm to alert consumers to enable two factor authentication methods on consumers' account or on consumers' device to activate on his/her device to avoid personal data was lost.



CONSUMERS WANT TO LEARN ABOUT 2FA

The quickest and most secure way of defending against hackers stealing your passwords is switching on 2FA for your account.

THE AVERAGE PERSON HAS 26 PASSWORDS

It's painful to change them all after a data breach.

Figure 4: Consumers Learning 2FA (InWebo, 2018).

Figure 4 shows recent reports found that consumers were encouraged to learn how to use the two-factor authentication method that was the most secure way to implement the network security in customers' device. It showed 156% consumers to increase in searching for two factor authentication methods and to learn how to use the two-factor authentication via their own devices.

Literature Related to the Methodology

This authentication algorithms were used to grant access to the user online access with policy-based access control for sign-in and password protection. This password protection was based on the triggering events that was identified by IP address of user’s login device to recognize the user location, the time the user login whether it was during daytime or evening time and MAC address. The MAC address was a unique identifier of the hardware address assigned to network interfaces at the data link layer of a network communications (Beal, 2004). The MAC layer was connected directly with the network medium, so each different type of network medium required a different MAC layer. By observing the MAC address of the network device where the user tried to login, the system administrators monitor the authentication system and defined the weights and scores of these triggering events.

Matyas Jr. et al. in “Toward Reliable User Authentication through Biometrics” proposed a new layer model for user authentication through biometrics to verify the accuracy rates for user authentication and discussed advantages and disadvantages of using biometric features. Two basic types of biometric systems were used in the model. The first model called “Automated Identification Systems”, which was used by police departments to identify the thefts found at the crime scenes. The second model called “Biometric Access Control Systems” that was used by any users to obtain permission to get into the system. The drawbacks of these models were the inaccurate performance of biometric techniques, and false rejects for an identical twin case to prevent biometric accuracy system when users attempted to authenticate themselves.

Also, the adaptive Multi-Factor Authentication helped mitigate potential threats, real-time alert to notify the system administrator of suspicious account credentials and provided multiple authentication options to secure access to the online applications.

Strategies for Adaptive Multi-Factor Authentication selection mechanism described in the article, the authors designed an approach to calculate the trustworthiness based on the type of devices. Each factor carried different trustworthiness for each device, e.g. fixed device, portable device and hand-held device. The drawbacks of the strategies were that it did not measure the burden on user while using this approach for adaptive MFA (Dasgupta et al., 2017).

According to Nag et al. (2015) illustrated that authentication was the mechanism to defense against illegitimate access to get access sensitive data in the cloud. Many recent security threats occurred, authentication using only a single factor was not reliable to protect the device of organization or individuals. Thus, to facilitate continuous protection of computing devices and other online devices from malicious attacks or unknown users. There were many authentication mechanisms with variety of authentication accuracy were available to be used. These mechanisms could get connected with various communicating devices. There were several factor authentication strategies had been used actively to enhance the security of applications for organization and individuals.

In addition, the authors also indicated that the design of a robust and scalable framework for authenticating legitimate users. This framework had many stages to proceed the authentication modalities associated with many features in time operating situations on a regular basis. The article focused on the creative framework of trustworthiness to quantify different authentication factors in terms of different types of devices. Furthermore, the trustworthy values were retrieved from previous history data in which user logged in. The history data was also based on the surrounding events or multiple conditions. These conditions were selected via the adaptive strategy to make sure the incorporation of the existing conditions within the adaptive authentication process. By doing the proposed solutions, the authentication strategy provided more flexible, better diversity in the selection of authentication factors. This would improve security, authentication, availability in terms of confidentiality of users (Dasgupta et al., 2017).

In this paper, the prototype of this program was designed and analyzed a mechanism to add resiliency to the A-MFA method. The mechanism included steps described as follows: first, the application was designed to help system administrators identify user's credentials to login the system based on geographic locations whether the user credentials was in the organization profile or outside of the organization. Secondly, the application enabled the system administrator to assign access roles for that user to login. Third, the authentication application helped system administrator identify the situations where trustworthiness of a user increased. Finally, the application helped system administrators identify the situations where trustworthiness of a user decreased. Based on four scenarios above, the application eventually enabled system administrator to define and to compute the trustworthiness scores of users who was trying to login. System admins would use multiple selection criteria to computer trustworthiness scores based on weights and scores chosen via user IP address, time login in, MAC address. By implementing this new approach of resiliency of authentication, it would be very important for managers, system admins, and executive staff in the organization to enforce security policies, security standards, security compliance. Also, system admins would proceed to increase the numbers of authentication modalities and the complexity of making decision of which triggering events was important, such as MAC address, IP location or time login.

In addition, the login page interface allowed user to login, which would be showed below. Whenever user entered his/her credentials to login, the computer system then stored user's information in the authentication system itself. The access database would execute the criteria based on the pre-defined by system administrators. System admins would select authentication rules and saved in the authentication system, then application would grant permissions to users the authentication rules thereafter.

The authentication methods would allow users to get access to the system based on the trustworthiness scores, which were measured carefully by system admins. The trustworthiness scores would be stored in the system. Then the authentication system executed further operations to grant users' rights to execute in the 1st authentication method, or the 2nd authentication method, or the 3rd authentication method, or the 4th ones, or the 5th ones based on the pre-dined trustworthiness scores designed by system administrators. If the trustworthiness score was less than 5, the user could not access to the system.

Authentication systems. Authentication system was the process of verifying user’s identity to verify who the users were. It involved validating the proof of identity of a person by their valid documents, genuine physical objects. In computer system, it was supposed to assign only authorized users to get access to the computer systems. To get access to the computer, the system was controlled by authentication procedures to establish with some degree of confidentiality of the users’ identity, to grant privileges for that users’ identities. The access control was in the 8th layer - the user layer on top of the OSI model architecture of data communication of networked computer. The 8th layer was referenced to physical controllers and external hardware device which interacted with an Open Systems Interconnection (OSI) model network. Thus, authentication system was very important in the computer security (Rouse, 2015).

Single factor authentication. Single Factor Authentication System performed one action for user identity (Feltner, 2016). This also meant that this method was easy, did not require too much user cooperation and it was executed fast. A single factor was always easier for a malicious to receive other users' profile than multiple factors, and the possibility of passing a security measure with an obtained factor was inversely proportional to the number of factors required. Using single factor authentication could be suggested to use at any places, where high security levels were less important to use it in their organizational performance.

Two factor authentication (Two FA). The two-factor authentication was adopted by software companies such as Amazon, Google, yahoo, Dropbox, Facebook, LinkedIn, Twitter, Microsoft, and others. Two FA was a method of confirming some users claimed identify by utilizing a combination of two different components, which were the password/username combination. In addition, the user would be asked to verify who a person was by using something only he or she owns, such as a computer device, mobile device, etc. The two FA is used two factors to confirm an identity. Also, two FA was a type of multi-factor authentication (Dyer et al., 1992).

Multi Factor Authentication (MFA). Multi-factor Authentication (MFA) was the process of authenticating a user after successfully presenting several evidences to an authentication system. That was the MFA was a method to identify the legitimate users in multiple ways through an active authentication process, which consisted of user credentials, passwords, security token, biometrics, cognitive behavior metrics, software and hardware devices, etc.

A user was granted access through authentication mechanism, these categories must be verified include:

1) *Knowledge* (something the users know) like a user ID, PIN numbers.

When presenting a knowledge factor to authenticate, user must prove that he or she knew a secret, like a password or four-digit pin number.

2) *Possession* (something the users have) like a hardware device, RSA token, a one-time passcode.

Possession factor was another way of authenticating users where a user must prove the possession of something like smart card, Short Message Service code, or a key to verify himself or herself.

3) *Inherence* (something the users are) such as a finger-print or some other physical bio-metric (Feltner, 2016).

In addition, user provided proof of who he/she was like biometrics, unique physical or behavioral characteristics. Then, the identity was verified using technology of fingerprint, iris, voice and other unique features.

MFA was used to add an extra layer on top of the user layer - user name and password (the first factor – what they know) as well as for an authentication code from MFA (the second factor – what they have). The combined factors provided safeguard access and important for the user authentication process (Nag, 2014).

Adaptive Multi-Factor Authentication (A-MFA). Professors Abhijit Kumar Nag and Dipankar Dasgupta invented the Adaptive Multi-Factor Authentication (A-MFA) that used a combination of user credentials, passwords, biometrics, and human factors to build a trustworthy authentication system to validate the proper authentication factors when users log in the systems (Dasgupta et al., 2017).

A-MFA was the method to authenticate legitimate users in a system, which was recognized as a new way to prevent the weakness of password and traditional multifactor authentication. The A-MFA was used in online access and identify management systems where authentication modalities were selected adaptive through sensing many characteristics of the user's behaviors while the users attempted to log in the systems. For example, smartphone-based on multi-factor authentication, the authentication method was important to verity the legitimate users' identity, finger print, a smartphone' unique identity. Thus, A-MFA was critical for security concerns underlying the authentication methods (Nag, 2014).

According to Bolle et al. (2004) explained that a user could use a portable device to transmit wirelessly the stored biometric for authentication purposes or a user could locally measure a biometric by using the portable device and matched it against a biometric which was already stored locally in the computer systems like portable device. Various methods were also proposed in the article to build a biometric authentication system and to implement the authentication methods.

In the experiments for Multi-Factor Authentication. He proposed a new Adaptive MFA mechanism by mathematically calculating the trustworthiness of each authenticating modality. They proposed adaptive selection strategies based on what they tested trustworthy algorithms. The shortcoming of this article was that it didn't mentioned about industrial mechanism and lack of business sense (Dasgupta et al., 2017).

In addition, Nag et al. (2014) proposed an approach for A-MFA selection mechanism. Trustworthiness of devices based on various type of devices. Each factor carries different trustworthiness for each device like fixed devices, portable device and handheld device, and the media like wired, wireless and cellular. Based on the approach of these authors' experiments, the drawbacks of this approach were that it did not measure the burden on users while using this approach for adaptive MFA.

According to Saha (2015), the article illustrated that CAPTCHAs had a significant role in recognizing humans and machines via online authentication mechanism. With the technology advancement, the computer recognized human traits, images to extract the characters shown in CAPTCHAs. The CAPTCHAs provided many mathematical, logical, and inference problems that only humans could understand and answered accurately. The framework provided questions to ask human beings many kinds of questions. The more complexity questions being asked, the more accuracy of the authentication could be used. The study showed the implementation of the computer system to illustrate the adaptive MFA based on biometrics and human factors.

A-MFA via smart cards or workstation to authenticate user credentials for access to workstations, mobile devices, cloud and on-premises apps needed to be complied with security regulations of the organization, to enforce strong password was mandatory, to request users to enroll credentials to the authentication system ("Multiple Criteria Decision Analysis," n.d.).

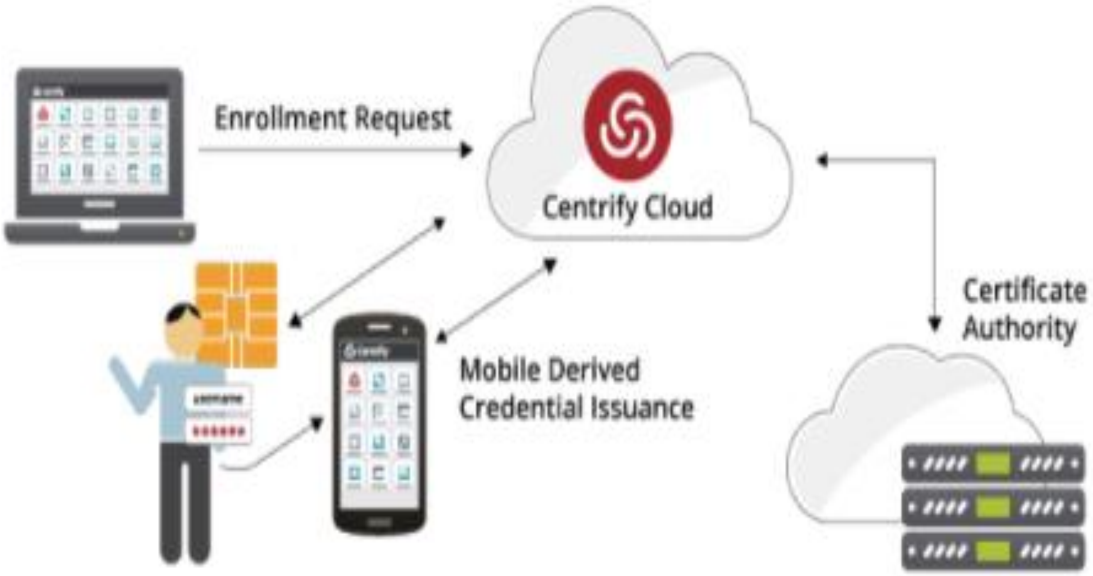


Figure 5: Smart Card Credential Insurance ("Adaptive MFA and Strong Authentication," n.d.).

Figure 5 shows the smart card system that was a highly secure alternative to passwords and comply with security regulation. The system enforced to use strong authentication techniques via smart cards for access to Mac and Linux workstations, mobile device in the cloud and on-premise locations.

To enforce A-MFA more efficiently, Ping ID approach described that Ping ID could match the security risks included policies for applications, session and devices based on geographic location and trusted networks. Security policies could be followed by any scenarios to get access to the system (Khandelwal, 2018).

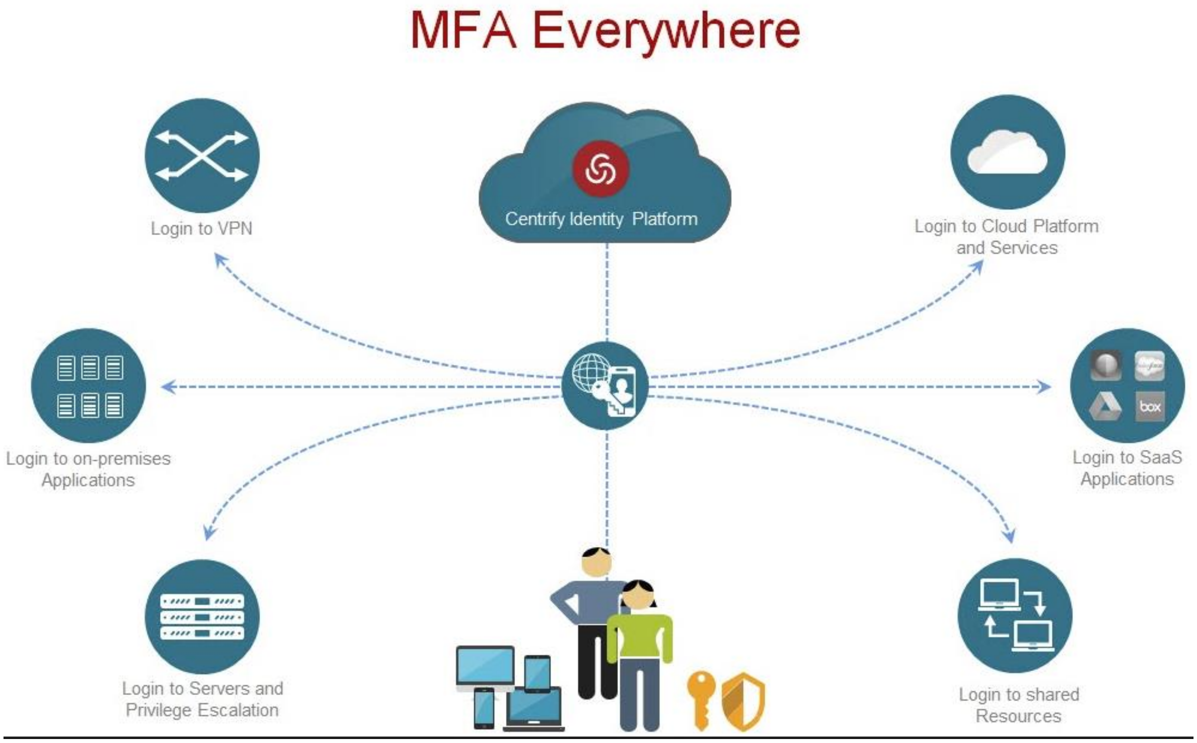

MFA Everywhere
Centrify Identity Platform
Login to VPN
Login to Cloud Platform and Services
Login to on-premises Applications
Login to SaaS Applications
box
Login to Servers and Privilege Escalation
Login to shared Resources

Figure 6: MFA Everywhere (Zindel, 2017)

Figure 6 shows the MFA could be useful for consumers everywhere. The MFA could be configured and deployed via Identity Service Provider (ISP) system. Users could use the correct multiple authentication factors to login to the system depending on a user's profile and biometrics. ISP could set static policies for different factors, such as user roles, resources, locations, time of day or day of week. Thus, A-MFA could provide the use of OTP tokens like RSA, secure ID to user.

Summary

In this chapter, the Background and Literature Review of the proposal paper was illustrated in this chapter. The methodology of the paper would be covered in the next chapter.

Chapter III: Methodology

Introduction

This chapter would briefly cover how the proposed application to be analyzed and implemented. It would also cover various subsections, tasks and functions in the proposed resiliency of AMA system.

The adaptive MFA was considered as a best practice to protect users' sensitive data from fraudulent access. Users used smartphones to access emails, financial transactions, etc. at different location and different time. Cyber criminals exposed most of computer system, they did not only steal sensitive information but also modified the programs, and they injected the malicious code into the system and made the system compromised. Based on the recent incident issues described above, dynamic authentication techniques provided a continuous method of protecting user's identity and avoided major security breaches. The prototype of this program was designed on demand, and system admins could enforce application security to define weights and scores of multiple selections, such as user IP address, time login, MAC address. Trustworthiness model was designed for system admins to calculate the trustworthy values associated with weights, scores, probabilities of three criteria of user IP address, time to login, MAC address, so that system administrator could decide the trustworthiness scores and apply one of three authentication rules to grant access to user login to the system properly.

Design of the Study

This study applied the qualitative approach to analyze the various authentication methods. The criteria like triggering events that A-MFA used to select authentication factors adaptively were usually pre-defined and hardcoded in the authentication system. Also, this study focused on designing the application to implement the adaptive MFA applications to evaluate the best results achieved though this study and system administrators would proceed to assign the weights and scores to increase the resilience of A-MFA systems to rank the triggering events based on the user's roles, the weight of IA address, the time user's login and the MAC address. All these criteria should be authenticated through the authenticate system. This research study had been worked better for the study of qualitative approach because it would have illustrated how system admins made the authentication process harden or soften based on different important security objectives.

According to the article, which indicated cybersecurity awareness solution was a module and powerful platform so that the system professionals in the organization could effectively learn and manage the human cybersecurity risk at the right time, right place. To harden the infrastructure was the best solution to improve resilience to cyber incidents and reduce the threat ("Countering Advanced Persistent Threats with Cyber Forensics," n.d.).

To improve the secure access to the computing network, system administrators aimed particularly at the interface programs designed by organizations to allow authorized users to get access into the systems. This approach improved and brought various benefits to business performance and productivity (Khalig, 2013).

The program would be designed and implemented corresponding with the flow chart to describe the resiliency of A-MFA program as follows:

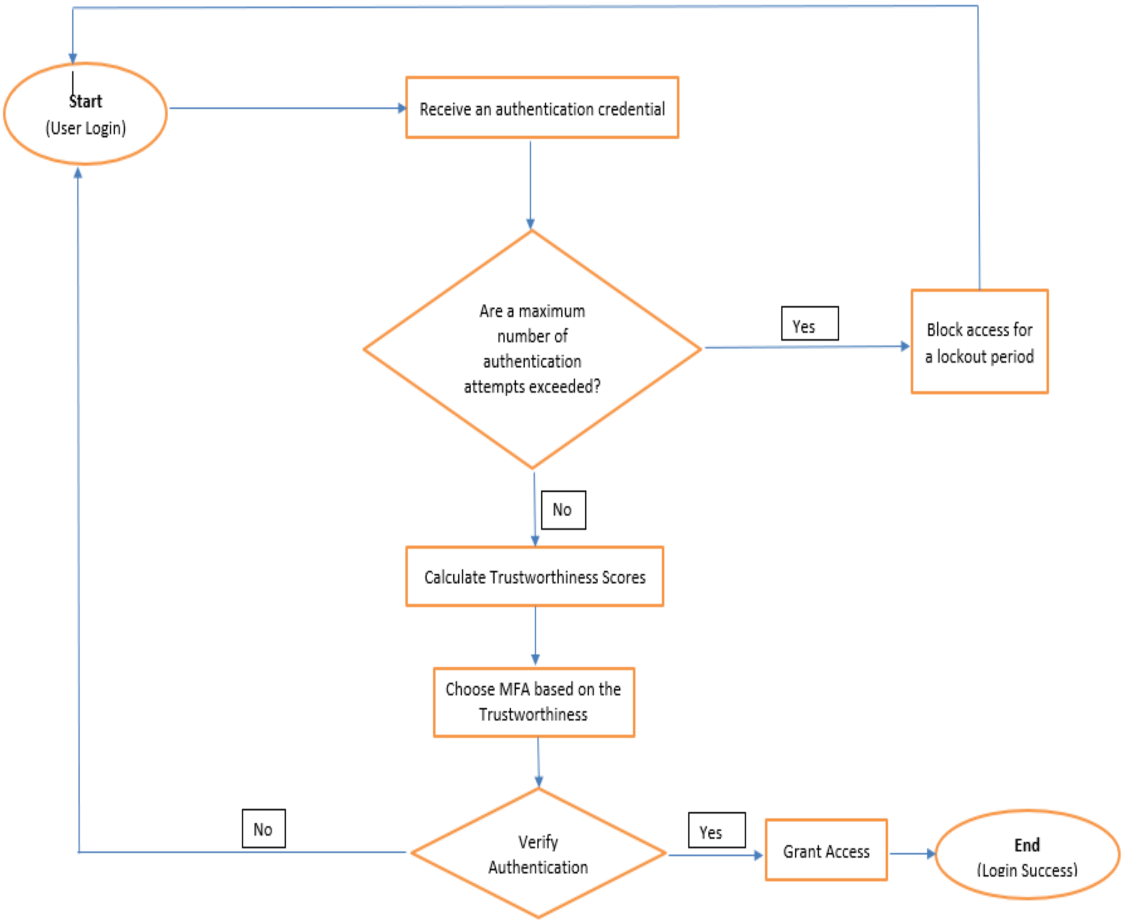

Start
(User Login)
Receive an authentication credential
Are a maximum number of authentication attempts exceeded?
Yes
Block access for a lockout period
No
Calculate Trustworthiness Scores
Choose MFA based on the Trustworthiness
Verify Authentication
No
Yes
Grant Access
End
(Login Success)

Figure 7: Flow Chart of Implementing Resiliency of Adaptive Multi-Factor Authentication Systems.

Figure 7 shows the flow chart of program to illustrate the proposal prototype of this research. The flow chart designed for system administrators, application programmers, Information Technology manager, and internal employees to visualize the whole program and to comprehend how the program executed to authenticate authorized users to login the systems. This flow chart demonstrated the data flow from start to finish, so that reader would follow the sequential steps easily.

Below is the GUI login page designed for users would like to login.

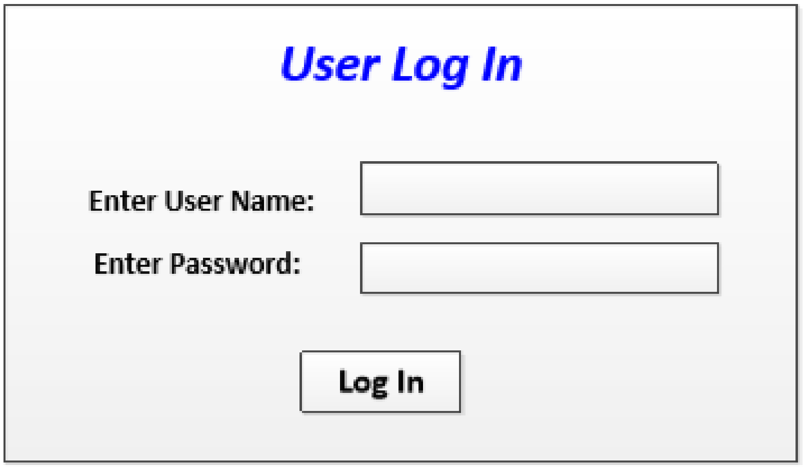


Figure 8: Login Page

Figure 8 shows the GUI login page for user to login. In the process of authentication techniques, the trustworthiness scales would be defined by system admin to decide the authentication methods granted to the user thereafter.

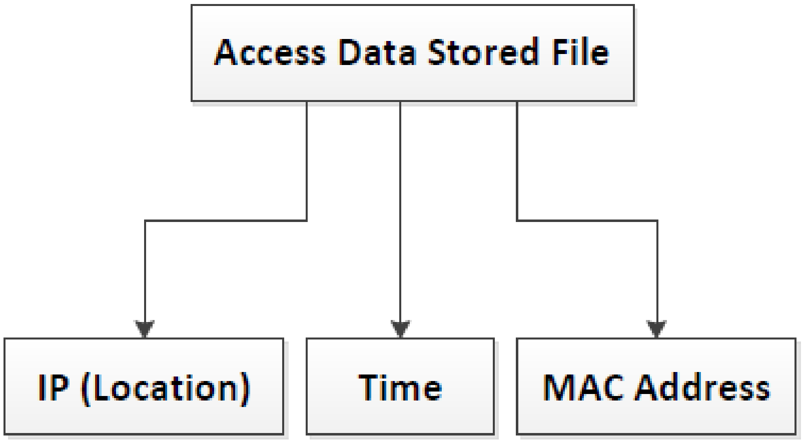


Figure 9: Access Data of Operation Criteria

Figure 9 shows the back end of the system how to store user’s credentials in database. After user login to the system, the user credentials stored in the data file in the database server based on the triggering events via IP address (location), time frame and MAC Address. These events designed to calculate the trustworthiness scales to help system admins to assign authentication methods associated with the scales.

A table illustrated the trustworthiness modalities, which would be described below to show how to compute the trustworthiness values for each individual factor based on IP address, time frame and MAC address when the user's attempted to login to the system. The trustworthiness scores were then calculated by the computational complexity of the selected modalities via the multiplication of the scores and weights.

These numbers 10, 5, 0 were examples designed in this program for system administrator to enter into the authentication system. The scores from 0-10 corresponding with user IP address, time login, and MAC address were also examples to demonstrate the score values.

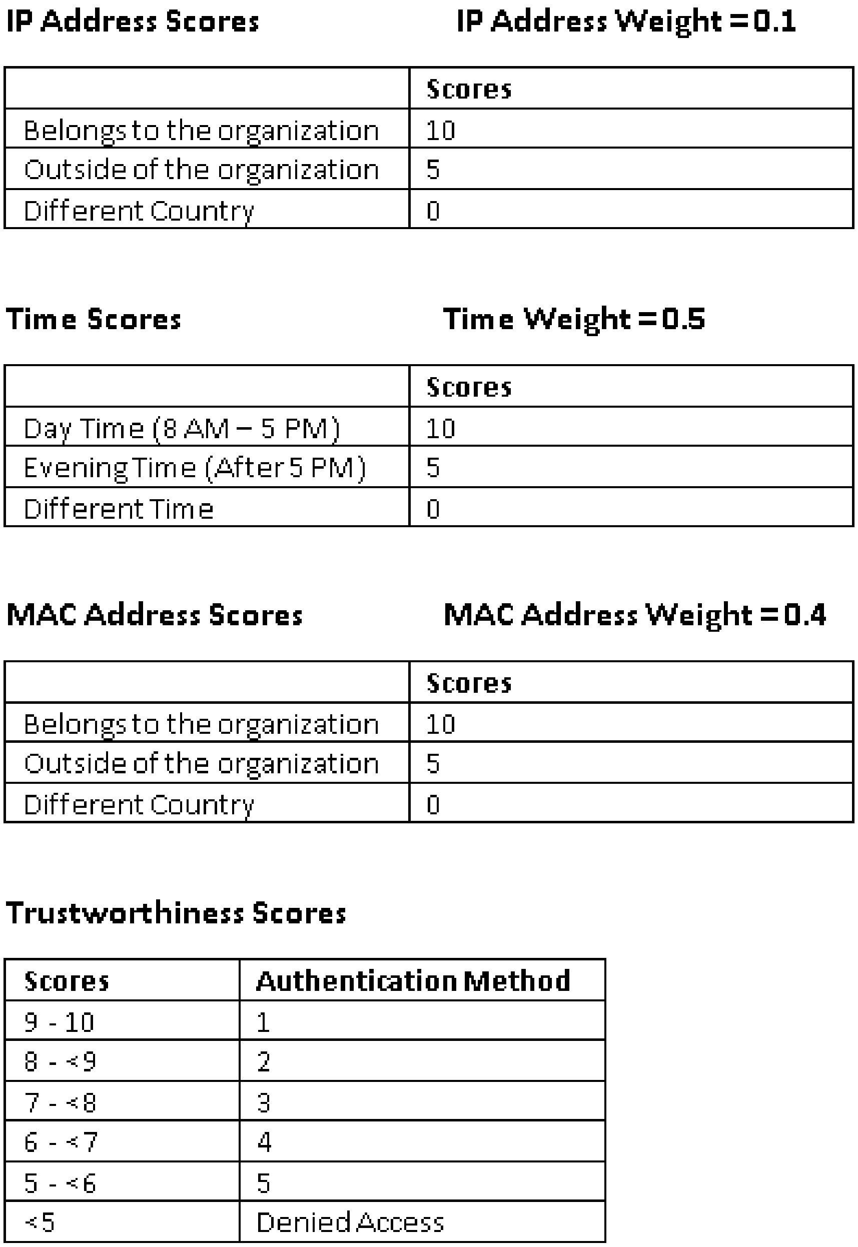
Trustworthiness scores were calculated via the probability of three criteria values, from these values system admins could define authentication methods. There were six authentication methods would be used based on the following trustworthiness scores:

- If trustworthiness scores were greater than 9, the system admin grand access roles to the user.
- If trustworthiness scores were from 8 to less than 9, one authentication method would be granted to user.
- If trustworthiness scores were from 7 to less than 8, two authentication methods would be used.
- If trustworthiness scores were from 6 to less than 7, three authentication methods would be granted to user.

- If trustworthiness scores were from 5 to less than 6, four authentication methods would be granted to user.
- If trustworthiness scores were less than 5, user could not log in to the system because of the denial access defined by system administrator.

Therefore, the higher numbers of authentication methods would be executed, the harder authentication access would be used, so that this scenario would be limited unauthorized access to the system. To protect information from possible threats, it was very important for system admin and organizations to identify all possible vulnerabilities and manage risks. By designing the flexibility of calculating the scores and weights, it would minimize possible risks.

Table 1: Trustworthiness Scores



IP Address Scores **IP Address Weight = 0.1**

	Scores
Belongs to the organization	10
Outside of the organization	5
Different Country	0

Time Scores **Time Weight = 0.5**

	Scores
Day Time (8 AM – 5 PM)	10
Evening Time (After 5 PM)	5
Different Time	0

MAC Address Scores **MAC Address Weight = 0.4**

	Scores
Belongs to the organization	10
Outside of the organization	5
Different Country	0

Trustworthiness Scores

Scores	Authentication Method
9 - 10	1
8 - <9	2
7 - <8	3
6 - <7	4
5 - <6	5
<5	Denied Access

Table 1 shows all criteria of triggering events and show how to calculate the trustworthiness scores. The trustworthiness values for authentication modalities with various features in different devices and time are tabulated and are shown in table 1 above. To calculate the trustworthiness values for combination of different criteria, the formula is illustrated below:

The trustworthiness = (IP Scores * Weight $_{\text{of IP address}}$) + (Time Scores * Weight $_{\text{of login time}}$) + (MAC address * Weight $_{\text{of MAC address}}$)

Data Collection

The data for the proposed methodology was created to demonstrate the purpose of the proposed prototypes of AMFA for system administrator to define the trustworthiness scores and authentication rules. Resources had been collected from articles, journals. The secondary resources would be collected and analyzed from internet source and books.

Proposed System

- *User Login GUI*: The proposed system for user login access was written and designed to capture user's credentials and retrieved user ID and password, security questions via one-time passcode. The program was designed to get a generated random number. When user retrieved generated random number from the interface, user could enter the generated random number into the program, it then allowed user to get into to the system. This program was a simulation to implement a standard login process for user to get access to the system.
- *AMFA Administration Controller GUI*: This was a prototype designed for system administrator to enter the weight and score values of user ID address, time login, and MAC Address. Based on the trustworthiness scores, system admins would decide the authentication rules and grant access to the user.

This prototype had not been coded yet, it was designed in a diagram to show how the process working in the prototype.

Tools and Technology

The following tools were used in the process of implementing the proposed system of user login interface:

- *User Login GUI*: a proposed system for user login access was written in ASP.NET, HTML, XML, and SQL Server and C #.
- *AMFA Administration Controller GUI*: the prototype would be coded to execute the interface in ASP.NET, HTML, XML, C# and SQL server in the future work. This prototype aimed to design authentication algorithms for end users to understand how the A-MFA to have more flexibility and resiliency to execute the program.

Potential applications of the project included the simulation of the logical algorithms of Adaptive MFA to increase the resiliency of user authentication related to online banking, financial transactions, access to critical and sensitive electronic database, access to cloud services, etc. This project used Webpages for user and system admin to login to the system, system admins also controlled the authentication system to validate user’s credentials, to calculate trustworthiness scores, to send access token to user and in return system admins would receive acknowledgements access token from user to verify if the passcode was valid or invalid. If the passcode was valid, user could login to the system, otherwise, access denied.

Hardware and Software Environment

This study involved the use of hardware and software installed on the researcher’s workstation included Microsoft Visual Studio 2015, Microsoft Business Management, and Microsoft Visio.

- Programs written in HTML, XML, ASP. Net, C#, and cascade style sheets (CSS). CSS is a stylesheet language used to describe the presentation of a document written in HTML or XML. CSS was designed in Web applications to make the GUI have the same functions across all screens.
- Databases used MS SQL Server.

Summary

This study was designed to collect information resources, recent incidents, literature review, methodology related to the adaptive multifactor authentication. The hardware and software equipment requirements and specifications were mentioned above. Project schedule tasks had been prepared for analyzing the methodology and the logical algorithms weekly and/or biweekly. All authentication algorithms mentioned in the paper would be studied in the due course and implementation had been implemented as a basis program written and executed on the webpage.

The prototype of system administration GUI and the system administration sequence diagram were prototypes. The prototypes had been designed to illustrate the process of resiliency of A-MFA that system administrator had defined the trustworthiness scores and authentication method rules based on the user ID address, time login and MAC address of the user’s credentials.

Chapter IV: Data Presentation and Analysis

Introduction

Most webpages rely on user IDs and passwords for access to the system. In case of billions of stolen credentials had been used, it was clear that a user ID and passwords alone were not secure in the system. Thus, increasing the resiliency of AMFA would be very beneficial in place to provide more security for user’s rights and authentication methods. This research included the implementing resiliency of AMFA systems that gathered many factors entered by users’ login interface like user ID address, time login and MAC address. To increase security network, the system administrators took actions to decide more additional factors whether tightened or softened user access to the system. The system administrator’s accountabilities also computed trustworthiness scores and defined authentication method rules to provide authentication access to the user. This secure scenario would provide the authentication process to prevent unauthorized users with stolen credentials from accessing applications. This study included the flow chart of implementing resiliency of adaptive multi factor authentication systems, the user login application, adaptive MFA controller Graphical User Interface, and Adaptive MFA Administration Controller flow chart.

Data Presentation

Resiliency was an increasingly adaptive process in academic research and in all companies and is closely connected to the complexity of AMFA systems. Resiliency of AMFA should take multiple processes to execute in the user’s login application to grant authentication access rules to the user’s credentials to get access to the system more securely.

Applications that processes sensitive information of user credentials should have created the need of secure software development to maintain high level C.I.A. (confidentiality, integrity, and availability) to the computer system. Especially for this proposed prototype, it should be more secure to implement the resiliency of A-MFA administration approach, system admin had better to comply with the organization’s rules and objectives to increase or to decrease more resiliency of user access. By analyzing and executing this approach, it would minimize the change for malicious hackers to intrude the systems.

According to Grembi (2008), creating a software design was the most important design for quality projects to uncover issues with security, requirements, and functionality (Grembi, 2008, p. 134). This concept was relevant to this research project to increase the resiliency of A-MFA Administration application. The criteria that A-MFA had been used to select authentication factors would help system analyst, software developer, and system administrator to understand the overall application of the project. This research was designed as a prototype to enhance the authentication methods, to make the A-MFA functionality more resiliently, to utilize the application execution more effectively, to make systems administrator situational awareness to improve application security.

A prototype is a type of proposed and small programs with little to no business logic or supporting databases. The prototype would provide end user with the general concepts and understanding the final output (Grembi, 2008, p. 136). This research paper included various static activities, multi factor authentication security concepts, input fields, output fields, and navigation features to connect to other related entities. For example, the first user interface was executed via the event actions as a standard interface to authenticate user username and password, then it was connected or related to the next interface, which was a user validation interface to verify user credentials to make sure user credentials were valid or invalid. Thereafter, the user validation interface was connecting to the A-MFA Administration Controller System interface, so that system admins would define the weights and scores of three criteria, such as user IP address, time login, and MAC address to calculate the trustworthiness scores.

The data for this experiment was categorized into the following categories:

- *User login*: all user login credentials were entered to the login system.
- *User Validation*: to validate correct user credentials in the system.
- *A-MFA Administration Control System*: all the admins were logged into the system to enter the values of weights and scores based on user IP address, time login and MAC Address. Then system admin would decide the rules of authentication methods and saved it to the system administration controller system.

Below was a flow chart of implement resiliency of adaptive MFA systems that was designed to illustrate the data flow from start to finish.

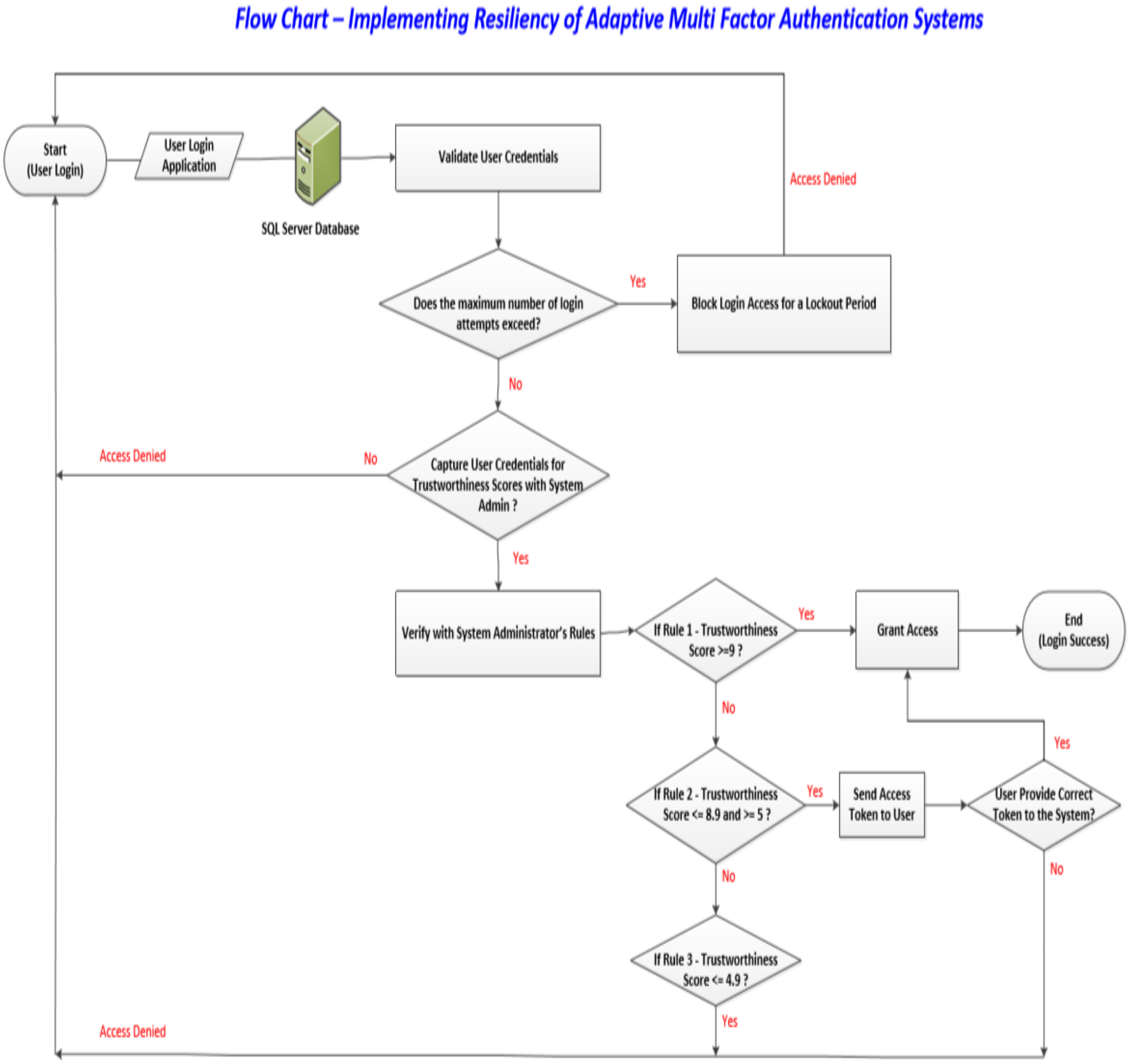

Flow Chart – Implementing Resiliency of Adaptive Multi Factor Authentication Systems
Start (User Login)
User Login Application
SQL Server Database
Validate User Credentials
Does the maximum number of login attempts exceed?
Yes
Block Login Access for a Lockout Period
Access Denied
No
Capture User Credentials for Trustworthiness Scores with System Admin ?
No
Access Denied
Yes
Verify with System Administrator's Rules
If Rule 1 - Trustworthiness Score >=9 ?
Yes
Grant Access
End (Login Success)
No
If Rule 2 - Trustworthiness Score <= 8.9 and >= 5 ?
Yes
Send Access Token to User
User Provide Correct Token to the System?
Yes
No
No
If Rule 3 - Trustworthiness Score <= 4.9 ?
Yes
Access Denied

Figure 10: Implementing Resiliency of Adaptive Multi Factor Authentication Systems Flow Chart

Figure 10 explains the data flow of the whole program which involves the authentication system had been developed. It gives the high picture/model of the authentication application.

The flow chart demonstrated when a user tried to login into the authentication system until the users accomplished to get access to the systems. When user would like to login to the systems, he/she entered user ID and passwords via the user login GUI, data then stored in SQL database. Validation process was implemented to verify if the user ID and password are valid or invalid. The application applied the maximum number of three times for the user to login, if users' credential failed or invalid, the system blocks login access for a block out period. By restricting the login time constraints, it will minimize intruders' attempts to use other users' accounts. For instance, there was a lawsuit case of David Kernell went on trial for hacking into Alaskan Governor Sarah Plain's personal account, David found Alaskan's emails on a website and posted the password in the media so others could access the account. Thus, malicious attacker got other credentials through the media to get access into personal individual's account. This was a serious crime, a computer fraud that impacted financial institutions, like banks or the U.S government, etc. ("Is Email Hacking Is a Serious Crime – Lawyers.com," n.d). Therefore, this proposal applied the maximum login of three times for any users attempt to login to the system. If any hackers or malicious thefts would try to log in, they would fail in attempt to get access to corporate network system.

System administrator then captured user credentials to calculate the trustworthiness scores via user IP address, time login, and MAC address. system administrator then defined the authentication rules based on the trustworthiness values, which was a strategy for calculating the trustworthy values of different factors of triggering events in three different setting of criteria in the following ways:

(1) *IP address*: to specify if the computer belongs to the organization or not, or from different county.

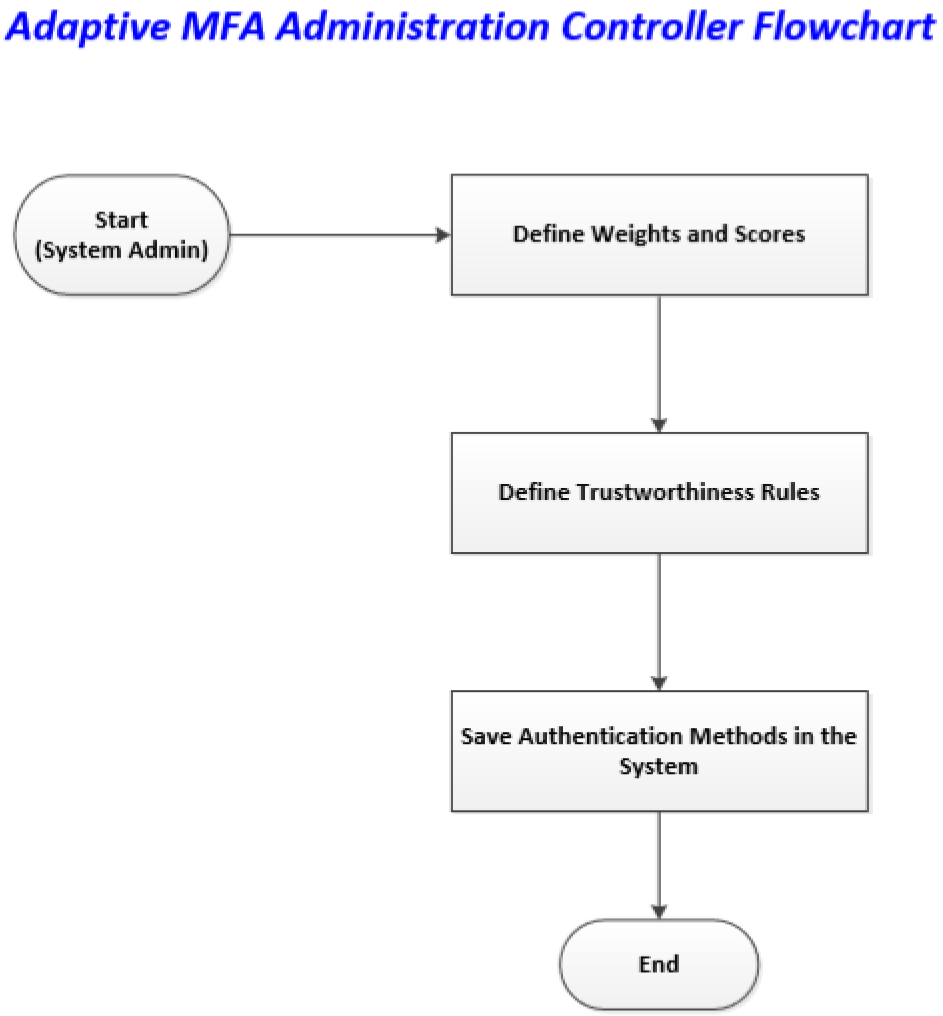
(2) *Time login*: to specify the time during day time from 8 AM to 5 PM, or evening time after 5 PM, or any different geographical time zone like Central Time (CT) or Easter Time (ET) zones.

(3) *MAC address*: to define the device like fixed device, handheld device, and portable device provided by organization or not.

When A-MFA system had been defined the weights and scores of criteria described above. System administrator would focus on deterministic approaches to calculate the trustworthiness value of the authentication modalities. There were three authentication method rules defined by System Administrators, which included:

- *Rule 1*: If Trustworthiness score>=9, grant access to the system.
- *Rule 2*: If Trustworthiness score <=8.9 and >=5, send access code to the user for verification.
- *Rule 3*: If Trustworthiness score <=4.9, deny login access.

Below was the Adaptive MFA Administration Controller flowchart for system admins to login to the authentication system to define weights and scores, trustworthiness scores, then the scores would be saved in the system administration controller system.

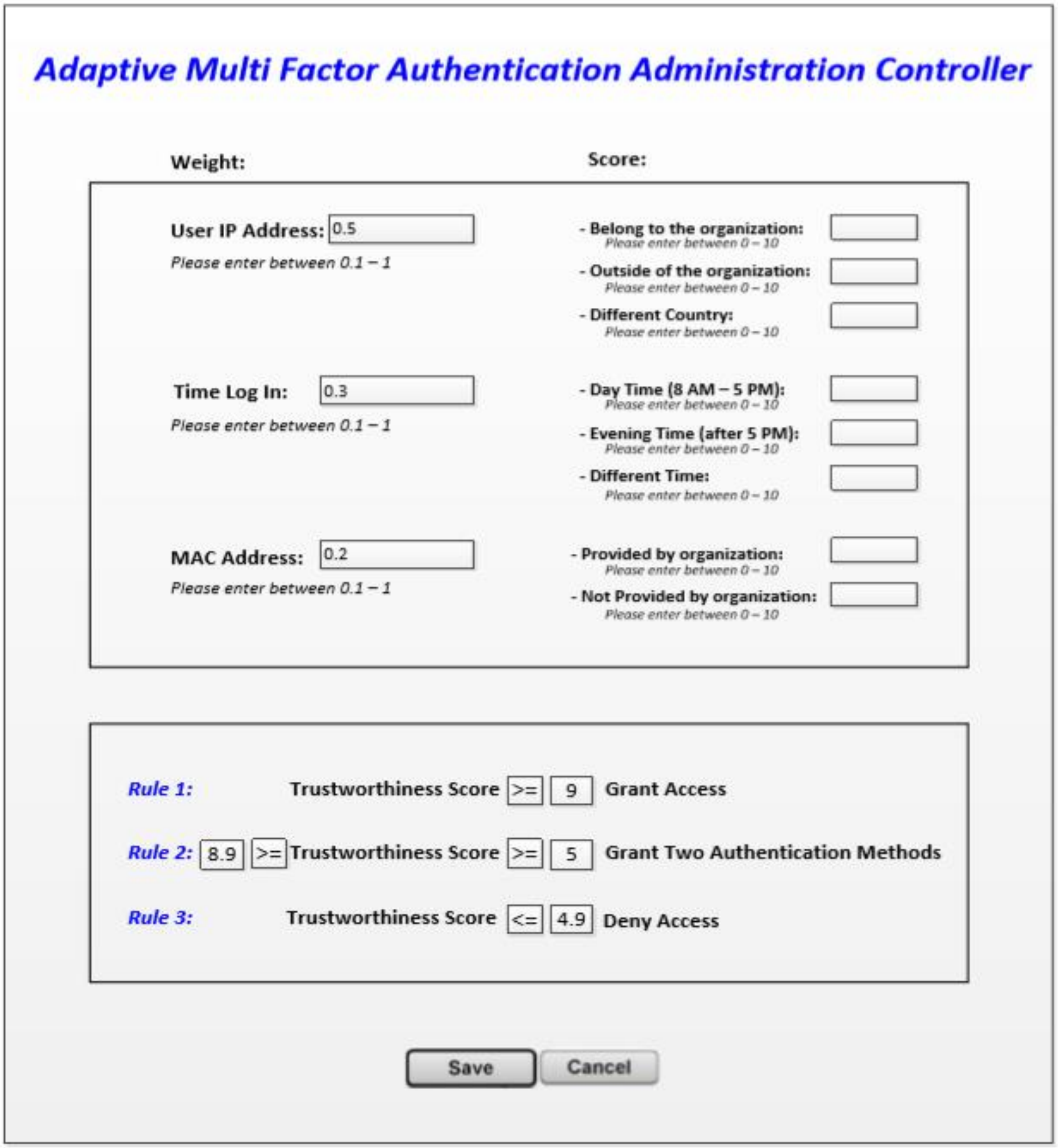


Adaptive MFA Administration Controller Flowchart

Start
(System Admin)
Define Weights and Scores
Define Trustworthiness Rules
Save Authentication Methods in the System
End

Figure 11: Adaptive MFA Administration Controller Flowchart

Figure 11 shows the data flow of A-MFA Controller for system admin to define weights and scores, trustworthiness rules, and defined the authentication methods based on the probability of trustworthy scores.



Adaptive Multi Factor Authentication Administration Controller

Weight:

User IP Address: 0.5
Please enter between 0.1 – 1

Time Log In: 0.3
Please enter between 0.1 – 1

MAC Address: 0.2
Please enter between 0.1 – 1

Score:

- Belong to the organization: *Please enter between 0 – 10*
- Outside of the organization: *Please enter between 0 – 10*
- Different Country: *Please enter between 0 – 10*
- Day Time (8 AM – 5 PM): *Please enter between 0 – 10*
- Evening Time (after 5 PM): *Please enter between 0 – 10*
- Different Time: *Please enter between 0 – 10*
- Provided by organization: *Please enter between 0 – 10*
- Not Provided by organization: *Please enter between 0 – 10*

Rule 1: Trustworthiness Score >= 9 Grant Access

Rule 2: 8.9 >= Trustworthiness Score >= 5 Grant Two Authentication Methods

Rule 3: Trustworthiness Score <= 4.9 Deny Access

Figure 12: Adaptive Multi Factor Authentication Administration Controller Interface

Figure 12 shows the design of Adaptive MFA Administration Controller Interface. The GUI was used for system admin to define the probability of weights and scores of user’s IP address, time login, and MAC login, which were relate to the authentication algorithms. The calculated sum of three criteria to measure the probability must equal to 1.

(1) *IP address:* scores associated with User IP address should be defined between 0-10.

(2) *Time Login*: scores associated with time login should be defined between 0-10.

(3) *MAC address*: scores associated with time login should be specified between 0-10.

After system administrators defined the weights and scores, system administrator would specify the rules of authentication methods. The purpose of the authentication methods was to increase or decrease the resiliency of adaptive MFA system to provide secure authentication for legitimate users considering various triggering events. System administrator had authority to define the three authentication rule scenarios as follows:

- *Rule 1*: If trustworthiness >=9, grant access to the system.
  - The 1st scenario would be evaluated like an example below:

Weight	**Score**	**Probability**
IP address 0.8	10	8
Time login 0.25	0	0
MAC Address 0.25	5	1.25
		Total = 9.25

Therefore, this user earned a probability total of 9.25 points that were assigned the optimal authentication method as the first rule to get access to the system based on the selected constraints of this authentication values. The first rule of trustworthy values showed high performance of trustworthiness scores.

- *Rule 2*: If trustworthiness <=8.9 and >=5, send access code to the user for verification and email verification.
    - The 2nd scenario would be used if user provided correct these features (access code and email verification) to the authentication system, the user then retrieved two authentication methods to login. The effects of selecting a set of authentication factors which would satisfy different optimal criteria to do authentication. An example score illustrated below:

Weight	**Score**	**Probability**
IP address 0.2	5	1
Time login 0.6	10	6
MAC Address 0.3	5	1.5
		Total = 8.5

This user had a probability total of 8.5 points, which were granted two authentication methods: (1) access code. (2) email verification.

- *Rule 3*: If trustworthiness <=4.9, deny login access.
    - The 3rd scenario applied to illegitimate user or malicious attackers. The objectives of this authentication rule would make it harder for

the user to login to prevent any chance of compromising authentication selection patterns of the attackers.

- An example of weights and scores shown below:

Weight	**Score**	**Probability**
IP address 0.5	0	0
Time login 0.1	1	0.1
MAC Address 0.7	5	3.5
		Total = 3.6

In this case, this user had a probability total of 3.6 points, which showed that the trustworthiness values were so slow, and the system denied user access.

In three scenarios described above, a strategy for calculating the trustworthy values of different authentication factors quantified the effects of different criteria. The criteria provided system admin's authority to select decisions of different authentication rules in different operating conditions. The highest trustworthy values for any authentication triggering events, the better chance for user to get access to the system quickly.

To proceed the operating procedures of the application, system administrators should be aware of how to provide authentication methods to the end user. A sequence diagram was then designed to illustrate the sequential events in which system administrator managed user credentials to login to the system with correct access token.

System administrator would validate the user credentials to verify if the user ID and passwords were valid or invalid. If user credentials were valid, system administrator would send the access token to user. Users would then receive access code from the system, and then entered to the login page. Then, system administrators would validate the access token to verify the access token was valid or not. After that, system admins would send authentication rules to user for them to get access to the system.

A sequence diagram was designed for this research paper, this sequence diagram was an interaction diagram in Unified Modeling Language (UML) that showed the objects, communication outline and events to illustrate how processes operated with one another and followed sequential order. It was a construct of a Smartdraw to show the relationships and connections between entities arranged in a time sequence. Sequence diagrams were also called event diagrams, event scenarios, and timing diagrams ("Data Modeling and Entity Relationship Diagram (ERD)," n.d.).

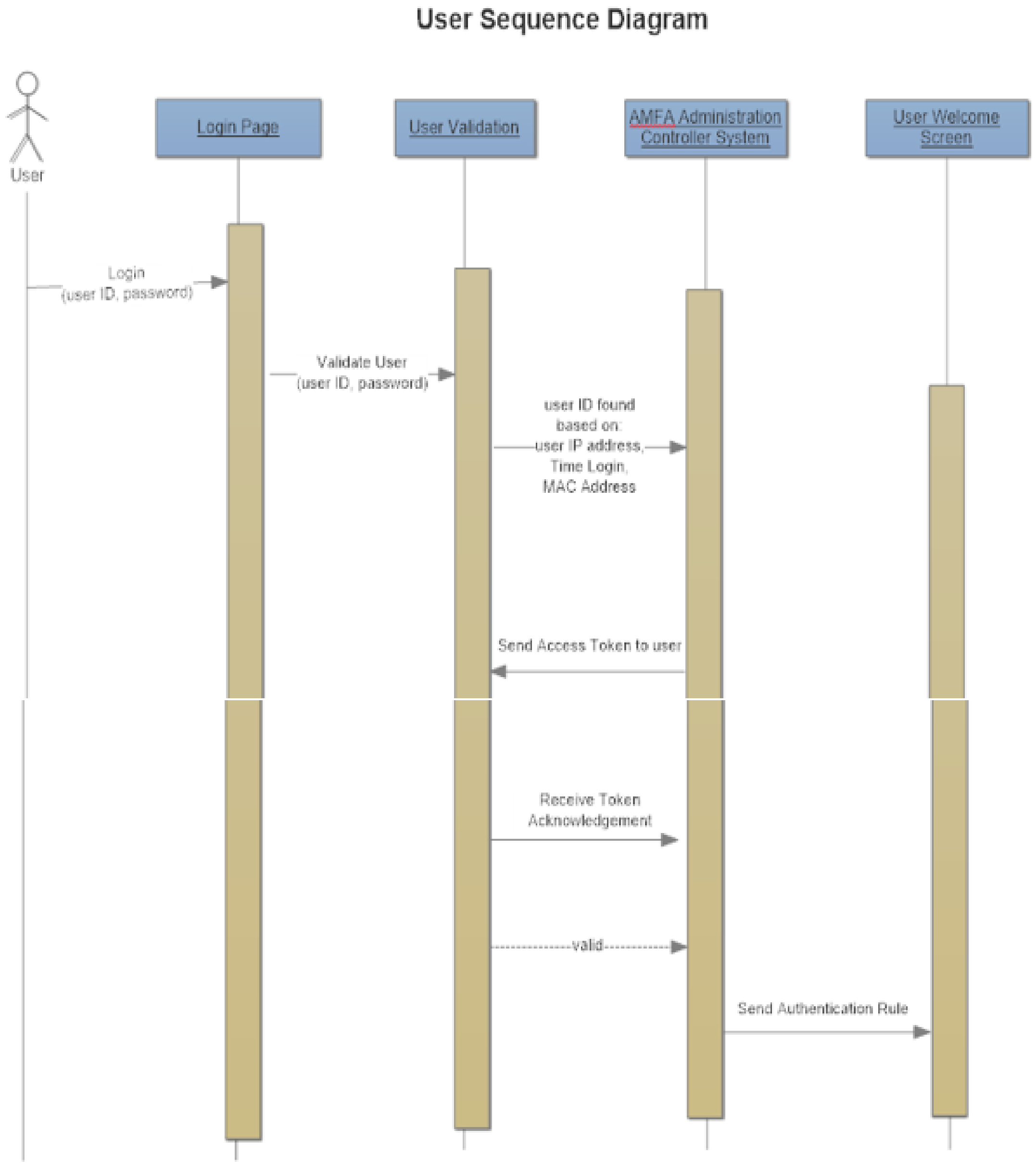

User Sequence Diagram
User
Login Page
User Validation
AMFA Administration Controller System
User Welcome Screen
Login (user ID, password)
Validate User (user ID, password)
user ID found based on: user IP address, Time Login, MAC Address
Send Access Token to user
Receive Token Acknowledgement
valid
Send Authentication Rule

Figure 13: User Sequence Diagram

Figure 13 shows the user sequence diagram to illustrate the sequential process from start to finish when user attempted to login to the system. The first interface was a "login page" for user to login, then the next interface was called "user validation" to validate username and password to see if data was valid or not. The next GUI was called A-MFA Administration Controller System, which was an interface controlled by system admin to define weights and scores of triggering events, such as user IP address, time login, MAC address. After the system admins validated user criteria, system admin then sent access token to user via one-time-passcode to user and requested user to acknowledge the access token via his or her device to login. By approaching this process, system admin would have ability to double check the validity of access token to identify that token assigned belong to legitimate users. Then, system admin would grant access to user to login to a user welcome interface.

Data Analysis

Data analysis was the process of systematically statistical and logical approach to evaluate data, to check results of implemented application. System admin or any executive team in the organization should recognize the considerations and/or issues in data analysis including concurrently selecting data collection methods and appropriate analysis, reliability and validity (Gotlschalk, 2003). Additional exploratory research of the proposed AMFA system would be useful in studying the entity relationships among events and objects of the programs.

Results

The user GUI of adaptive multi-factor authentication was designed to implement a basic authentication method based on one-time-password. With the security conditions applied to authenticate valid users, the A-MFA mechanism took place into the login system to be built a stimulate program to illustrate this secure method. A program of A-MFA was implemented by using Web Pages written in HTML, ASP.NET, CSS languages.

However, the prototype of Adaptive Multi Factor Authentication Controller would be designed in future. This research demonstrated that there were several areas involved in the procedures of defining the resiliency of authentication methods. In completing this research project, the study questions/hypothesis could be answered below:

***Question 1*:** Any threats or risks when the organization or individuals used online applications to access to the computer systems?

Answer: No threats or risks occurred because system administrator used multiple criteria decision-making method to define rules more strictly based on multiple selections criteria. And system admin would define weights and scores. Thus, it made the system to be harder for unauthorized users to get access to the system.

Question 2: Legitimate user could efficiently login to the system on a regular basis?

Answer: Valid user credentials could login to the system with adaptive multi factor authentication questions applied in the interface, thus user could answer security questions correctly.

Question 3: Where was the location of user's device to user login?

Answer: User's device location was recognized in the A-MFA Administrator Controller application. Based on MAC address of user device location, system admin would define weights and scores if the device was belonging to the organization or not belonging to the organization.

Question 4**:** Device belongs to the organization or not belong to the organization?

Answer: Some users used his/her own mobile device to log in to the online system. In this case, system admin would assign the scores of not provided to the organization associated with MAC address.

Question 5**:** Does authentication system recognize the location of the device in which the user attempted to login?

Answer: the authentication system could recognize the location of user's device via MAC address, because the authentication would be stored the MAC address whenever user logged in the system. Then system admin would define weights and scores to calculate the trustworthiness scores.

Question 6**:** How was the system detected unknown user? What would had happened and how it was occurred?

Answer: The system would detect unknown user by recognizing user enters invalid username and password. Or users could not answer correctly security questions, which was registered in the system.

The login page shown below was a simulation to implement a standard program to show how user could get access to the system – a Welcome Page.

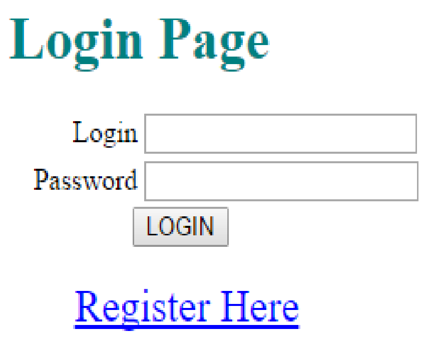


Figure 14: Login Page Implementation

Figure 14 shows the login page interface to allow current user to login to the system. If any new user attempted to login, the new user would register to the system and click on the link of “Register Here” to start registering username, password, email, phone number, security questions, etc.

In case of user had not been registered in the system. The registration page would be shown for user to enter the his/her credentials.

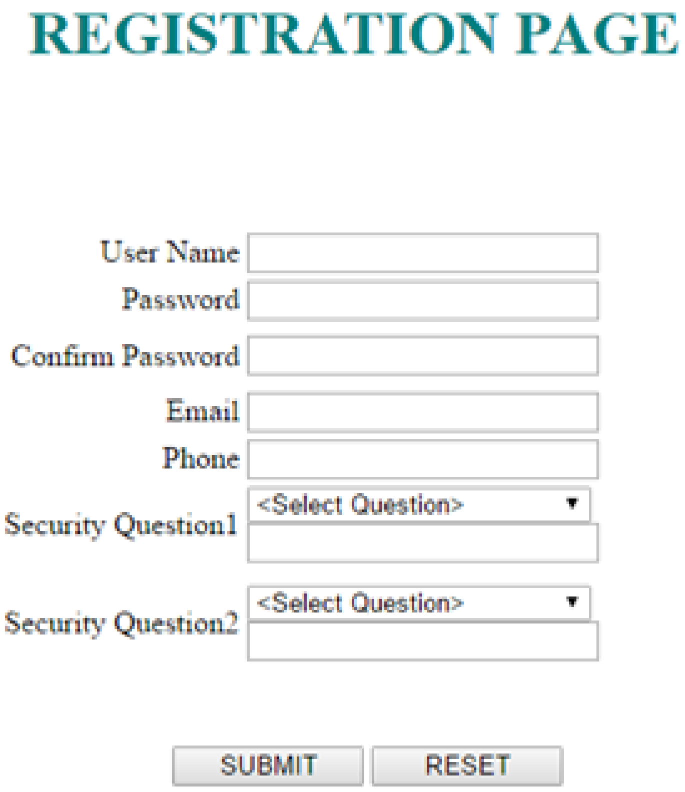
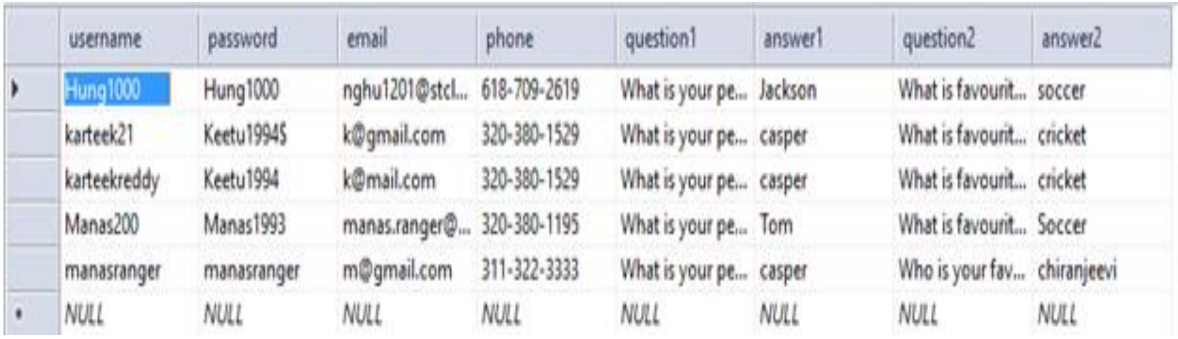


Figure 15: Registration Page

Figure 15 shows the Registration Page to allow users to register his/her credentials into the login system. The user credentials then were stored in the SQL Server database as shown below.

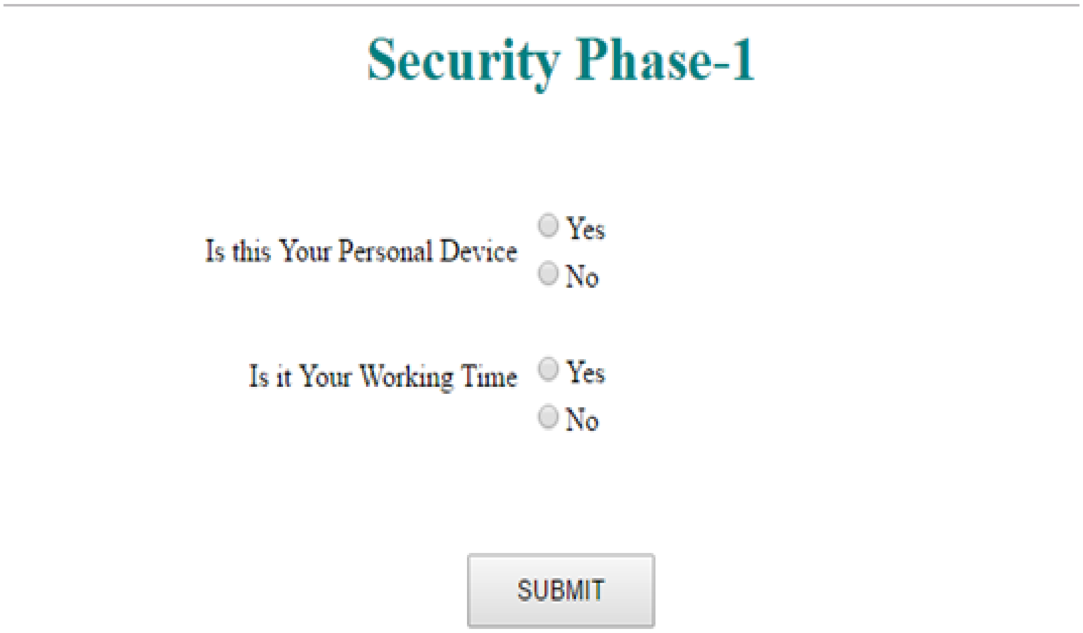


	username	password	email	phone	question1	answer1	question2	answer2
▸	Hung1000	Hung1000	nghu1201@stcl...	618-709-2619	What is your pe...	Jackson	What is favourit...	soccer
	karteek21	Keetu1994$	k@gmail.com	320-380-1529	What is your pe...	casper	What is favourit...	cricket
	karteekreddy	Keetu1994	k@mail.com	320-380-1529	What is your pe...	casper	What is favourit...	cricket
	Manas200	Manas1993	manas.ranger@...	320-380-1195	What is your pe...	Tom	What is favourit...	Soccer
	manasranger	manasranger	m@gmail.com	311-322-3333	What is your pe...	casper	Who is your fav...	chiranjeevi
•	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 16: SQL Server Database Entity

Figure 16 shows the user information stored in SQL server database.

To use adaptive MFA, the security questions phase 1 were applied for user to answer questions.



Security Phase-1

Is this Your Personal Device ○ Yes ○ No

Is it Your Working Time ○ Yes ○ No

Figure 17: Security Questions

Figure 17 shows a screenshot of authentication method for security questions.

In addition, security phase 2 was used to authenticate a legitimate user to log into the system, Adaptive MFA methods should be implemented based on the three following conditions:

- If both security questions “Is this Your Personal Device?” and “Is it Your Working Time?” were answered “Yes”. Access granted.
- If both questions were answered “No”. Access denied. The program would be redirected to the Login Page.
- If the first question “Is this Your Personal Device? was answered “Yes”. And the second question “Is it Your Working Time? was answered “No”

One-Time Password Authentication was then executed to generate a random number, the page shown below:

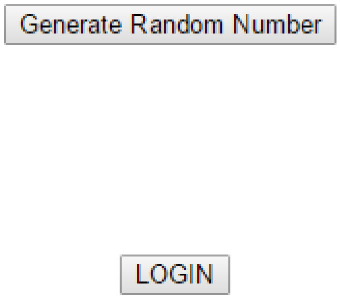
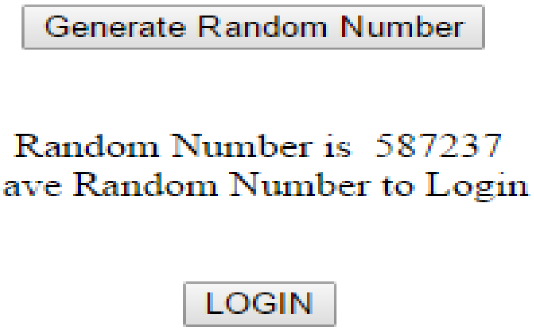


Figure 18: Random Number Generation

Figure 18 shows a screenshot of random number generation page of the program.

A random number was then generated in the field below, which would let user login with the random number confirmation.

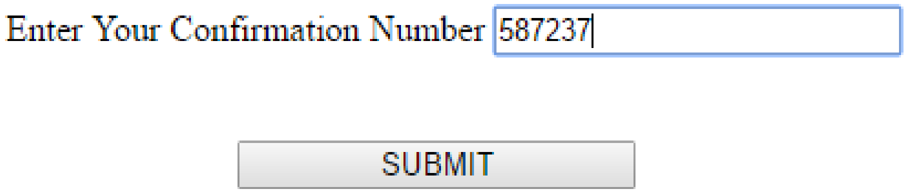


Random Number is 587237
Save Random Number to Login

Figure 19: Random Number Output

Figure 19 shows the output of random number generated via implemented authentication methods.

The generated random number was then entered in the field below. In doing so, the random number stored in the database for authenticating legitimate users to be granted to log into the system.



Enter Your Confirmation Number 587237

Figure 20: Confirmation Number

Figure 20 shows user entered the random number into the system, then clicked a submit button.

After user entered the confirmation number into the system. A welcome page had shown to illustrate that user could get a successful login.

The proposed program contained login user interface to store user credentials, security question data, random number generated to authenticate user login. The user GUI was written in ASP.NET, C# and SQL server.

Data modeling was a software system using diagrams and symbols to represent communication of data. The Entity Relationship Diagram (ERD) was a graphical representation of data requirements for a database. ERD contained database values of all related entities. Entity Relationship Diagram was a type of structural diagram for use in database design. An ERD included entities, connector relationships between entities within the database system (“Data Modeling and Entity Relationship Diagram (ERD),” n.d.).

There are three components in ERD:

- *Entities*: the relations/tables need to keep data in database.
- *Attributes*: data or information such as property, facts to describe each entity or table.
- *Relationships*: connector to show how tables are linked together via primacy and foreign keys.

To design the ERDs, entity should be written in nouns to define classes, concepts, roles, events or things. For example: employees, users, students, courses, books, payment, projects. Relationships were the connectors between the entities, the relationship should be written in verbs to describe relationships between entities. In the research paper, the proposed user application had two entities called rand_num and registration. These two entities had one-to-many relationship associated with each other via primary and foreign keys called username.

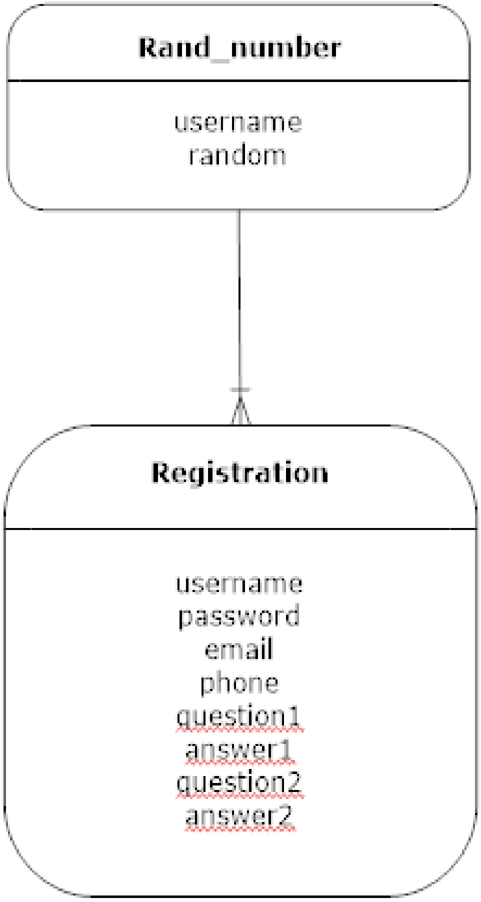


Figure 21: Entity Relationship Diagram (ERD).

Figure 21 shows the ERD of the user interface program. This ERD was designed as a basic program for user to login to the system. Continuous learning to develop this program would also encourage to get it done for improving process of future goals. As new technologies were continuous growing, the resiliency of adaptive MFA application was a good project to challenge programmer and system admin to implement the program.

Summary

Overall, this chapter had been covered the analytical algorithms, design of interface to understand how the prototypes and GUI were created to authenticate user credentials. With the motivation of new approach to implement the resiliency of A-MFA approach, system administrator would be able to weight the benefits and challenges of potential resiliency of A-MFA to select the best scenarios that would fit for the organization’s needs and requirements. The proposed application would allow system admin to validate user ID and passwords, to calculate trustworthiness scores, to assign authentication method rules to users for increasing or decreasing the user access to the system. The next chapter would depict the results, conclusions and recommendations.

Chapter V: Results, Conclusion, and Recommendations

Introduction

This chapter described the prototype of the application to increase the resiliency of A-MFA administration application. The prototype of the research application was to verify user credentials to determine user ID and password valid or invalid. System admins would define scores associated with user IP address between 0-10 based on conditions, which were belong to the organization, outside of the organization, and/or different country.

Also, scores associated with time login were defined between 0-10 based three cases, such as during daytime 8 AM – 5 PM, evening time after 5 PM, or different time zone or different state of country.

Time login scores should be defined by system admin between 0-10 based on the MAC address if the device was provided by the organization or outside of the organization.

Weights and scores of three triggering events – User IP Address, time log in, MAC address was determined by system administrator, then they would define the rules of authentication methods. The purpose of the authentication methods was to harden or soften the resiliency of adaptive MFA system. System administrator had authorities to define three authentication rules to grant access to user in three scenarios of authentication rules below:

- *Rule 1*: if trustworthiness >=9, access granted to the user.

- *Rule 2*: if trustworthiness <=8.9 and >=5, system admins send access code to the user and request user to return acknowledgement passcode to the authentication system.
- *Rule 3*: if trustworthiness score <= 4.9, user could not get access to the system.

Discussion

In the paper, the application was designed with the graphical user interface (GUI) for the user to login to the online system. This was a simulation of the program to implement the trustworthiness calculation based on different surrounding events based on the time frame of user login during working hours or outside of working hours, IP address to know where was the location of devices logged in, and MAC address to acknowledge that the user's credentials belong to the organization or outside of the organization.

According to Gottschalk, (2003), researchers should perform analysis on either qualitative or quantitative analysis to make sure the validity and reliability of a content analysis study corresponding to the results of the program.

By implementing the adaptive multi factor authentication approach, it would improve security to provide additional security to add protection in security network layers. The more secure layers in place, the more the risk of an intruder gaining access to critical systems. In addition, A-MFA could achieve the compliance, flexibility and productivity requirements to the organization (Carter, 2017).

Conclusion

The research highlighted the creation of analyzing and designing a robust and trustworthy framework to quantify different authentication methods in terms of selection of criteria (i.e. triggering events) to increase resiliency of scalable solutions for adaptive multifactor authentication modalities. The proposed trustworthiness model was computed the trustworthy values for different authentication factors by evaluating several probabilistic constraints of IP address, time login, and MAC address. The prototype of this proposal explored the applicability of the algorithmic approach to select multiple authentication modalities and their criteria. This research used comparisons among different devices, locations and time to identify sources of just-in-time login based on triggering criteria. The prototype also provided visualization of the authentication systems based on criteria, triggering events.

The proposal had been built a user login interface, that was a starting point of a program for user to get access to the online system. This program was used to implement a functionality of adaptive MFA to verify legitimate username and password or invalid, security questions, one-time-passcode via generated random number to authenticate a legitimate user efficiently.

In addition, the resiliency of Adaptive MFA System Administrator Controller application had not been built. It was a proposal as a prototype that would be implemented in the future work. The scope of the resiliency of A-MFA approach could be adaptively verified authenticated user’s credentials to log into the system, to calculate weights, scores, trustworthiness values. Authentication methods were based on the making decisions of system administrators.

Future Work

In future, additional study would be conducted with professional system administrators to test the trustworthiness framework in several scenarios as follows:

- *Various login time intervals from geographical zones*: A-MFA authentication system would use pairwise comparisons among various login time from different geographical zones to recognize if that user belongs to the organization or not. As a matter of fact, many users and/or contractors can login to the enterprise system remotely to work online applications, they can get access to the world-wide organization nowadays. In this case, if this valid user logs into the system from different geographical zone, this user's access role would be calculated as the best approach of trustworthiness scores, and would be granted the best authentication methods.
- *Various device used to login to finish one application:* if a user needs to complete the online financial system and other financial transaction, online medical records, online educational programs, etc. at a various time frame by using various device (MAC address), the authentication system will make it harder, more challenges to the user access to complete the whole financial transaction. By doing that, the adaptive selection schemes would be selected intelligent decisions and authentication factors to increase the performance, trustworthy scores of authentication methods, and to avoid illegitimate users attempt to login.

- *Trace the history of previous selection of criteria:* If the same individual gets into the system anytime and anywhere, the A-MFA would recognize that individual to be a legitimate user. The system administrator will use this approach to assign various authentication methods to that user. With this scenario, it will prevent any repeated selections of the same set of authentication factors in successive re-authentication attempts; thus, it would minimize the opportunity of gaining any recognizable login patterns. In addition, the authentication system would recognize the user’s credentials to be the same individuals to get access to system every time and every device, otherwise malicious hackers attempt to login.
- *Time-varying operating environments on daily basis*: this approach would calculate the trustworthy factors of triggering events in time-varying environments based on the access roles of user’s credentials like manager, system administrator, and executive members of the organization. For instance, the authentication system would design an urgent case option designed in the application. In case of any urgent case would occur like network attacks, application-layer attack, brute force attack, the executive members must monitor the network system to see if any an attacker who attempts to gain access to data, to decode a password or pin number, etc. Thus, the executive members would select the urgent case option that matches user’s credentials. This requires uniqueness and universal modalities could be incorporated with the existing set of constraints to calculate the trustworthy factors and a proper

scheme to get appropriate values for the new scenario of authentication methods. The resiliency of A-MFA framework would be extended to select user-roles, applications, operating environments, and user preference, which would be more benefits to implement more authentication modalities to verify user’s login into the system.

In this prototype proposal, the MFA Administration Controller GUI for System Administrator had not been coded completely to execute the prototype. This GUI for System Administrator had been designed as a prototype for future implementation.

Recommendation

In future, one user could register at least two devices for A-MFA, such as smart phone/cell phone and office or home phone. In case of a user might forget primary device at home, the user might need to get access to a protected application. In addition, users should change password frequently to protect data from hackers, the password would be setup as strong password including complicated words combination letters, special characters, and digits.

References

143 Million Equifax customers affected by data breach. Here’s what you should know. (n.d.). Retrieved October 20, 2017, from https://blog.dashlane.com/equifax-data-breach/

Adaptive MFA and Strong Authentication. (n.d.). Retrieved March 15, 2018, from https://www.centrify.com/products/endpoint-services/adaptive-mfa-and-strong-authentication/

Biometric Authentication: Security Benefits and Concerns. (2017). Retrieved November 1, 2017, from https://hscweb3.hsc.usf.edu/is/biometric-authentication-security-benefits-concerns/

Bolle, R. M., Nunes, S. L, Pankanti, S., Ratha, N. K., Smith, B. A., & Zimmerman, T. G. “Method for Biometric-based Authentication in Wireless Communication for Access Control”. (Publication: US 6819219 BI), 2004

Beal, V. (2004). MAC Address - Media Access Control Address. Retrieved November 5, 2017, from https://www.webopedia.com/TERM/M/MAC_address.html

Countering Advanced Persistent Threats with Cyber Forensics. (n.d.). Retrieved from https://www.guidancesoftware.com/docs/default-source/document-library/whitepaper/countering-advanced-persistent-threats-with-cyber-forensics.pdf?sfvrsn=1cc68bad_4

Cullen, T., & Abraham, E. (2016). Adapture. Ransomware Explained: The Business Decisions Required to Prevent and Survive an Attack. Retrieved from http://thinkforward.adapture.com/hubfs/Ransomware_2016_whitepaper_adapture.pdf?submissionGuid=64fc040d-e64b-47c9-af97-7a1af7c01a8e

Centrify. Multi-Factor Authentication Across Your Enterprise. Retrieved from https://www.centrify.com/solutions/why-multi-factor-authentication/

Carter, S. (2017). The Challenges and Benefits of Multi factor Authentication - MFA 101, Part 2. Retrieved from http://blog.identityautomation.com/the-challenges-and-benefits-of-multi-factor-authentication-mfa-101-part-2

Dasgupta, D., Roy, A., & Nag, A. (2017). Multi-Factor Authentication. *In Advances in User Authentication* (pp. 185–233). Springer, Cham. https://doi.org/10.1007/978-3-319-58808-7_5

Data Modeling and Entity Relationship Diagram (ERD). (n.d.). Retrieved March 15, 2018, from http://www.cs.uregina.ca/Links/class-info/215/erd/

Dyer, J. S., Fishburn, P. C., Steuer, R. E., Wallenius, J., & Zionts, S. (1992). Multiple Criteria Decision Making, Multiattribute Utility Theory: The Next Ten Years. *Management Science*, *38*(5), 645–654. https://doi.org/10.1287/mnsc.38.5.645

Feltner, S. (2016). Single factor Authentication vs. Multi factor Authentication. Retrieved December 15, 2017, from https://blog.centrify.com/sfa-mfa-difference/

Grembi, J. (2008). Secure Software Development: A Security Programmer’s Guide. *Design for Quality: The Big Picture.* Course Technology, Cengage Learning, pp. 134- 138.

Gotlschalk, L. A. (2003). Responsible Conduct in Data Management. Retrieved February 25, 2018, from https://ori.hhs.gov/education/products/n_illinois_u/datamanagement/datopic.htm

Is Email Hacking Is a Serious Crime - Lawyers.com. (n.d.). Retrieved April 11, 2018, from https://www.lawyers.com/legal-info/communications-media/privacy-law/email-hacking-is-a-serious-crime.html

[Infographic] Is the internet getting safer? (n.d.). Retrieved March 15, 2018, from https://www.twilio.com/learn/account-security/is-the-internet-getting-safer

InWebo **|** Identity Protection & two factor authentication built to secure access to VPN, web sites, Mobile and Cloud Apps (2018). Retrieved March 15, 2018, from https://www.inwebo.com

Khandelwal, S. (2018). New 4G LTE Network Attacks Let Hackers Spy, Track, Spoof and Spam. Retrieved March 11, 2018, from https://thehackernews.com/2018/03/4g-lte-network-hacking.html

Khalig, A. (2013). Situational Awareness for Computer Network Security. Retrieved November 5, 2017, from https://www.slideshare.net/mmubashirkhan/situational-awareness-for-computer-network-security

Matyas Jr., et al. “Toward Reliable User Authentication through Biometrics”. Retrieved September 2, 2017, from https://pdfs.semanticscholar.org/979d/218385030c498416182a346cb5bef28a412b.pdf

Multiple Criteria Decision Analysis - Wikipedia. Retrieved November 2, 2017, from https://en.wikipedia.org/wiki/Multiple-criteria_decision_analysis

Nag, A. K., & Dasgupta, D. (2014). An Adaptive Approach for Continuous Multi-Factor Authentication in an Identify Eco-system. *In Proceedings of the 9th Annual Cyber* *and Information Security Research Conference* (pp. 65-68). New York, NY, USA: ACM. https://doi.org/10.1145/2602087.2602112

Nag, A. K., Roy, A., & Dasgupta, D. (2015). An Adaptive Approach Towards the Selection of Multi-Factor Authentication. In *2015 IEEE Symposium Series on Computational Intelligence* (pp. 463–472). https://doi.org/10.1109/SSCI.2015.75

Nicholas, M. (2017). Just discovered - over 560 million credentials added to list of stolen passwords used by criminals. Retrieved October 20, 2017, from https://blog.dashlane.com/dashlane-tech-check-may-19-2017/

One-Time Passwords. (n.d.). Retrieved November 1, 2017, from http://www.square-enix.com/na/account/otp/

O'Leary, R. (2017). It's here: The 2017 WhiteHat Security Application Security Statistics Report! - WhiteHat Security. Retrieved October 20, 2017, from https://www.whitehatsec.com/blog/application-security-statistics-report/

Pescatore, J. (2017). SAN Institute. Cyber Security Trends: Aiming Ahead of the Target to Increase Security in 2017. Retrieved from https://www.sans.org/reading-room/whitepapers/analyst/cyber-security-trends-aiming-target-increase-security-2017-37702

Recode Daily: Hackers got into three billion Yahoo accounts - Recode. (2017). Retrieved October 20, 2017, from https://www.recode.net/2017/10/4/16416444/yahoo-verizon-breach-uber-board-softbank-kalanick-twitter-dorsey-snap-google-pixel2-nobel

Rouse, M. (2015). Authentication. Retrieved October 20, 2017, from http://searchsecurity.techtarget.com/definition/authentication

Saha, S. K. (2015). Human-Cognition-Based CAPTCHAs. Retrieved from http://ieeexplore.ieee.org/abstract/document/7272742/

Vien, C. (2015). “As cyber breaches rise, consumers alter spending, browsing habits,” *Journal of Accountancy*. Retrieved Dec 16, 2017, from https://www.journalofaccountancy.com/news/2015/apr/information-security-breaches-201512179.html

Zindel, A. (2017). What is Adaptive Multi-Factor Authentication (MFA)? Retrieved March 15, 2018, from https://blog.centrify.com/adaptive-multi-factor-authentication-mfa-2/

Appendix: User Interface Programming Codes

The following section presents the user interface programming codes written in C#, ASP.NET, HTML, CSS to implement the user login interface. This interface was the first and standard application of this research paper prior to exploring the prototype of increasing the resiliency of A-MFA approach. The following script was used to log into the login system.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Login.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            text-align: center;
            font-family: "Times New Roman", Times, serif;
            font-weight: bold;
            font-size: xx-large;
            color: #008080;
        }
        .auto-style2 {
            width: 100%;
        }
        .auto-style3 {
            text-align: right;
            width: 531px;
        }
        .auto-style4 {
            text-align: center;
        }
        .auto-style5 {
            text-align: left;
        }
        .auto-style6 {
            font-size: x-large;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div class="auto-style4">
    <div class="auto-style1">
            Login Page</div>
                <table class="auto-style2">
                    <tr>
                        <td class="auto-style3">Login</td>
                        <td class="auto-style5">
```

```
                            <asp:TextBox ID="TextBox1" runat="server"
                             Width="157px"></asp:TextBox>
                            <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
                            runat="server" ControlToValidate="TextBox1"
                            ErrorMessage="Enter Login"></asp:RequiredFieldValidator>
                        </td>
                    </tr>
                    <tr>
                        <td class="auto-style3">Password</td>
                        <td class="auto-style5">
                            <asp:TextBox ID="TextBox2" runat="server" TextMode="Password"
                            Width="158px"></asp:TextBox>
                            <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
                            runat="server" ControlToValidate="TextBox2"
                            ErrorMessage="Enter Password"></asp:RequiredFieldValidator>
                        </td>
                    </tr>
                </table>
          <asp:Button ID="Button1" runat="server" Text="LOGIN" OnClick="Button1_Click" />
         <asp:HyperLink ID="HyperLink1" runat="server" CssClass="auto-style6"
         ForeColor="Blue" NavigateUrl="~/Registration.aspx">Register Here</asp:HyperLink>
     </div>
    </form>
  </body>
</html>
```

The following script was the code behind of the login user interface to execute the Webpage.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;

public partial class _Default: System.Web.UI.Page
{
    protected void Button1_Click(object sender, EventArgs e)
    {
        SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
        con.Open();
        string checkuser = "Select count(*) from registration where username='" +
TextBox1.Text + "'";
        SqlCommand check = new SqlCommand(checkuser, con);
        int temp = Convert.ToInt32(check.ExecuteScalar().ToString());
        con.Close();
        if(temp==1)
        {
            con.Open();
            string pwd= "select password from registration where username='" +
TextBox1.Text + "'";
            SqlCommand passwd = new SqlCommand(pwd, con);
            string password = passwd.ExecuteScalar().ToString();
            if(password==TextBox2.Text)
            {
                Session["new"] = TextBox1.Text;
                Response.Redirect("Security1.aspx");
            }
            else
            {
                Response.Write("Incorrect Password");
            }
        }
        else
        {
            Response.Write("Incorrect Username");
        }
    }
}
```

The following script was used for registration user interface.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Registration.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            text-align: center;
            font-family: "Times New Roman", Times, serif;
            font-size: xx-large;
            color: #008080;
        }
        .auto-style2 {
            width: 100%;
        }
        .auto-style3 {
            width: 472px;
            text-align: right;
        }
        .auto-style4 {
            margin-left: 440px;
        }
        .auto-style5 {
            width: 472px;
            text-align: right;
            height: 56px;
        }
        .auto-style6 {
            height: 56px;
        }
        .auto-style7 {
            width: 100px;
        }
        .auto-style8 {
            width: 472px;
            text-align: right;
            height: 30px;
        }
        .auto-style9 {
            height: 30px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div class="auto-style1">
            <strong>REGISTRATION PAGE</strong></div>
            <table class="auto-style2">
                <tr>
                    <td class="auto-style3">User Name</td>
                    <td>
                        <asp:TextBox ID="TextBox_UN" runat="server"
                        Width="180px"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server" ErrorMessage="*Please Enter a User Name " ForeColor="Red"
ControlToValidate="TextBox_UN"></asp:RequiredFieldValidator>
                        <asp:RegularExpressionValidator ID="RegularExpressionValidator4"
runat="server" ControlToValidate="TextBox_UN" ErrorMessage="Minumu 8 characters"
ForeColor="#FF3300" ValidationExpression="^[a-zA-Z0-
9']{8,15}$"></asp:RegularExpressionValidator>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style3">Password</td>
                    <td>
                        <asp:TextBox ID="TextBox_pwd" runat="server" Width="180px"
                         TextMode="Password"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ErrorMessage="*Please Enter Password" ControlToValidate="TextBox_pwd"
ForeColor="Red"></asp:RequiredFieldValidator>
                        <asp:RegularExpressionValidator ID="RegularExpressionValidator3"
runat="server" ControlToValidate="TextBox_pwd" ErrorMessage="Minimum 8 characters"
ForeColor="#FF3300" ValidationExpression="^[a-zA-Z0-
9'@&amp;#.\s]{8,15}$"></asp:RegularExpressionValidator>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style8">Confirm Password</td>
                    <td class="auto-style9">
                        <asp:TextBox ID="TextBox_pwd2" runat="server" Width="180px"
                        TextMode="Password"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server" ErrorMessage="*Please Re-Enter Password" ControlToValidate="TextBox_pwd2"
ForeColor="#FF3300"></asp:RequiredFieldValidator>
                        <asp:CompareValidator ID="CompareValidator1" runat="server"
ErrorMessage="Password does not Match" ControlToCompare="TextBox_pwd"
ControlToValidate="TextBox_pwd2" ForeColor="#FF3300"></asp:CompareValidator>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style3">Email</td>
                    <td>
                        <asp:TextBox ID="TextBox_email" runat="server"
                         Width="180px"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server" ErrorMessage="*Please Enter your Email " ControlToValidate="TextBox_email"
ForeColor="#FF3300"></asp:RequiredFieldValidator>
                        <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ErrorMessage="*Enter Valid Email" ValidationExpression="\w+([-
+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" ControlToValidate="TextBox_email"
ForeColor="#FF3300" Display="Dynamic"></asp:RegularExpressionValidator>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style3">Phone</td>
                    <td>
                        <asp:TextBox ID="TextBox_phone" runat="server" TextMode="Phone"
Width="180px" MaxLength="13"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server" ErrorMessage="*Enter Phone Number" ControlToValidate="TextBox_phone"
ForeColor="#FF3300"></asp:RequiredFieldValidator>
                        <asp:RegularExpressionValidator ID="RegularExpressionValidator2"
runat="server" ControlToValidate="TextBox_phone" ErrorMessage="xxx-xxx-xxxx"
ForeColor="#FF3300" ValidationExpression="((\(\d{3}\) ?)|(\d{3}-))?\d{3}-
\d{4}"></asp:RegularExpressionValidator>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style3">Security Question1</td>
                    <td>
                        <asp:DropDownList ID="DropDown_q1" runat="server" Width="180px" >
                            <asp:ListItem>&lt;Select Question&gt;</asp:ListItem>
                            <asp:ListItem>What is your pet name</asp:ListItem>
                            <asp:ListItem>What is your favourite color</asp:ListItem>
                            <asp:ListItem>What is your first car</asp:ListItem>
                        </asp:DropDownList>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator8"
runat="server" ErrorMessage="*Select a Question" ControlToValidate="DropDown_q1"
ForeColor="#FF3300" InitialValue="&lt;Select Question&gt;"></asp:RequiredFieldValidator>
                        <asp:TextBox ID="TextBox_q1" runat="server"
Width="180px"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator6"
runat="server" ErrorMessage="*Enter Text" ControlToValidate="TextBox_q1"
ForeColor="#FF3300"></asp:RequiredFieldValidator>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style5">Security Question2</td>
                    <td class="auto-style6">
                        <asp:DropDownList ID="DropDown_q2" runat="server" Width="180px">
                            <asp:ListItem>&lt;Select Question&gt;</asp:ListItem>
                            <asp:ListItem>What is favourite sport</asp:ListItem>
                            <asp:ListItem>Who is your favourite Actor</asp:ListItem>
                        <asp:ListItem>What is favourite Subject</asp:ListItem>
                        </asp:DropDownList>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator9" 
runat="server" ErrorMessage="*Select a Question" ControlToValidate="DropDown_q2" 
ForeColor="#FF3300" InitialValue="&lt;Select Question&gt;"></asp:RequiredFieldValidator>
                        <asp:TextBox ID="TextBox_q2" runat="server" 
                         Width="180px"></asp:TextBox>
                        <asp:RequiredFieldValidator ID="RequiredFieldValidator7" 
runat="server" ErrorMessage="*Enter Text" ControlToValidate="TextBox_q2" 
ForeColor="#FF3300"></asp:RequiredFieldValidator>
                    </td>
                </tr>
            </table>
            <p class="auto-style4">
            <asp:Button ID="Button1" runat="server" Text="SUBMIT" Width="100px" 
             OnClick="Button1_Click" />
            <input id="Reset1" class="auto-style7" type="reset" value="RESET" /></p>
    </form>
  </body>
</html>
```

The following script was used for registration webpage to login system.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if(IsPostBack)
        {
            SqlConnection con = new 
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
            con.Open();
            string checkuser="Select count(*) from registration where username='" 
+TextBox_UN.Text +"'";
            SqlCommand check = new SqlCommand(checkuser, con);
            int temp = Convert.ToInt32(check.ExecuteScalar().ToString());
            if(temp==1)
            { Response.Write("Username not Available"); }
            con.Close();
        }
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        try
        {
            SqlConnection con = new 
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
            con.Open();
            string insert_data = "insert into registration values 
            (@username,@password,@email,@phone,@question1,@answer1,@question2,@answer2)";
            SqlCommand check = new SqlCommand(insert_data, con);
            check.Parameters.AddWithValue("@username", TextBox_UN.Text);
            check.Parameters.AddWithValue("@password", TextBox_pwd.Text);
            check.Parameters.AddWithValue("@email", TextBox_email.Text);
            check.Parameters.AddWithValue("@phone", TextBox_phone.Text);
            check.Parameters.AddWithValue("@question1", 
            DropDown_q1.SelectedItem.ToString());
            check.Parameters.AddWithValue("@answer1", TextBox_q1.Text);
            check.Parameters.AddWithValue("@question2", 
            DropDown_q2.SelectedItem.ToString());
            check.Parameters.AddWithValue("@answer2", TextBox_q2.Text);
            check.ExecuteNonQuery();
            Response.Write("Registration Successful");
            System.Threading.Thread.Sleep(4);
            Response.Redirect("Login.aspx");
            }
        catch(Exception ex)
        {
            Response.Write("Error" );
        }
    }
}
```

The following script was used for Security1 user interface to authentication user's rights. The list of codes written in ASP.NET, HTML, CSS to display the webpage of Security questions.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Security1.aspx.cs" 
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            text-align: center;
            font-family: "Times New Roman", Times, serif;
            font-weight: bold;
            font-size: xx-large;
            color: #008080;
        }
        .auto-style2 {
            width: 100%;
        }
        .auto-style3 {
            text-align: right;
            width: 506px;
            height: 47px;
        }
        .auto-style4 {
            height: 47px;
        }
        .auto-style5 {
            margin-left: 480px;
        }
        .auto-style6 {
            margin-left: 0px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
    <div class="auto-style1">
        Security Phase-1</div>
        <table class="auto-style2">
                <tr>
                    <td class="auto-style3">Is this Tour Personal Device </td>
                    <td class="auto-style4">
                        <asp:RadioButtonList ID="RadioButtonList1" runat="server"
                         AutoPostBack="True">
                         <asp:ListItem>Yes</asp:ListItem>
                         <asp:ListItem>No</asp:ListItem>
                        </asp:RadioButtonList>
                    </td>
                </tr>
                <tr>
                    <td class="auto-style3">Is it Your Working Time </td>
                    <td class="auto-style4">
                        <asp:RadioButtonList ID="RadioButtonList2" runat="server"
                        AutoPostBack="True">
                            <asp:ListItem>Yes</asp:ListItem>
                            <asp:ListItem>No</asp:ListItem>
                        </asp:RadioButtonList>
                    </td>
                </tr>
            </table>
        <div class="auto-style5">
             <asp:Button ID="Button1" runat="server" CssClass="auto-style6" Height="37px"
             OnClick="Button1_Click" Text="SUBMIT" Width="109px" />
        </div>
     </form>
   </body>
</html>
```

Below was the list of code behind the Security1 webpage.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

```
public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if(Session["new"]==null)
        {
            Response.Redirect("Login.aspx");
        }
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        if (RadioButtonList1.SelectedIndex == 0 && RadioButtonList2.SelectedIndex == 0)
        {
            Response.Redirect("Welcome.aspx");
        }
        else if (RadioButtonList1.SelectedIndex == 0 && RadioButtonList2.SelectedIndex ==
1)
        {
            Response.Redirect("Security2a_1.aspx");
        }
        else if (RadioButtonList1.SelectedIndex == 1 && RadioButtonList2.SelectedIndex ==
0)
        {
            Response.Redirect("Security2b.aspx");
        }
        else if (RadioButtonList1.SelectedIndex == 1 && RadioButtonList2.SelectedIndex ==
1)
        {
            Session["new"] = null;
            Response.Redirect("Login.aspx");
        }
    }
  }
```

Below was a list of code of Security2a written in ASP.NET, HTML, CSS for user enters a confirmation number.

```
<!D<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Security2a.aspx.cs"
Inherits="Security2a" %>
OCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            width: 100%;
        }
        .auto-style2 {
            width: 469px;
            text-align: right;
        }
        .auto-style3 {
            margin-left: 360px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
           <table class="auto-style1">
            <tr>
                <td class="auto-style2">Enter Your Confirmation Number</td>
                <td>
                <asp:TextBox ID="TextBox1" runat="server" Width="180px"></asp:TextBox>
                </td>
            </tr>
        </table>
        <asp:Button ID="Button1" runat="server" Text="SUBMIT" Width="180px"
         OnClick="Button1_Click" />
    </form>
  </body>
</html>
```

Below was a list of Security2a code written in C# to execute the confirmation number webpage.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;

public partial class Security2a : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if(Session["new"]==null)
        {
            Response.Redirect("Login.aspx");
        }
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        SqlConnection con=new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
        con.Open();
        string num="select random from rand_num where username='" +Session["new"]+"'";
        SqlCommand check = new SqlCommand(num, con);
        string rnum=check.ExecuteScalar().ToString();
        if(rnum==TextBox1.Text)
        {
            Response.Redirect("Welcome.aspx");
        }
        else { Response.Write("Incorrect Confirmation Number");
```

```
            System.Threading.Thread.Sleep(4);
            }
    }
}
```

Below was a list of Security2a codes for user enters confirmation number and save to the system.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Security2a_1.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            text-align: center;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="auto-style1">
        <div class="auto-style1">
        <asp:Button ID="Button1" runat="server" OnClick="Button1_Click" Text="Generate
         Random Number" />
        <asp:Label ID="Label1" runat="server" Text="Random Number is"
         Visible="False"></asp:Label>
        <asp:Label ID="Label3" runat="server" Visible="False"></asp:Label>
        </div>
            <asp:Label ID="Label2" runat="server" Text="Save Random Number to Login"
             Visible="False"></asp:Label>
        <asp:Button ID="Button2" runat="server" OnClick="Button2_Click" Text="LOGIN" />
      </div>
    </form>
  </body>
</html>
```

Below was a list of Security2a_1 codes written in C# to execute the generate random number webpage.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;

public partial class _Default: System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["new"] == null) { Response.Redirect("Login.aspx"); }
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        Label1.Visible = true;
        Random rand = new Random();
        for (int i=0;i<2;i++)
        { Label3.Text = (Convert.ToString(rand.Next(111111, 999999))); }
        Label3.Visible = true;
        Label2.Visible = true;

        SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
        con.Open();
        string checkuser = "Select count(*) from rand_num where username='" +
Session["new"] + "'";
        SqlCommand check = new SqlCommand(checkuser, con);
        int temp = Convert.ToInt32(check.ExecuteScalar().ToString());
        if (temp == 0)
        {
            string num = "Insert into rand_num values (@uname,@rnum)";
            SqlCommand check1 = new SqlCommand(num, con);
            check1.Parameters.AddWithValue("@uname", Session["new"]);
            check1.Parameters.AddWithValue("@rnum", Label3.Text);
            check1.ExecuteNonQuery();
        }
        else
        {
            string num1="Update rand_num set random= '" + Label3.Text +"' where
username='"+Session["new"]+"'";
            SqlCommand check2 = new SqlCommand(num1, con);
            check2.ExecuteNonQuery();
        }
            con.Close();
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        Response.Redirect("Security2a.aspx");
    }
}
```

Below was a list of code Security2b written in ASP.NET, HTML, CSS to display the webpage.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Security2b.aspx.cs"
Inherits="Security2b" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <style type="text/css">
```

```
        .auto-style1 {
            width: 100%;
        }
        .auto-style2 {
            width: 497px;
            text-align: right;
        }
        .auto-style3 {
            margin-left: 440px;
        }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <table class="auto-style1">
            <tr>
                <td class="auto-style2">
                    <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
                     runat="server" ControlToValidate="TextBox1" ErrorMessage="*"
                     ForeColor="#FF3300"></asp:RequiredFieldValidator>
                </td>
            </tr>
            <tr>
                <td class="auto-style2">
                    <asp:Label ID="Label2" runat="server" Text="Label"></asp:Label>
                </td>
                <td>
                    <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
                    <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
                     runat="server" ControlToValidate="TextBox2" ErrorMessage="*"
                     ForeColor="#FF3300"></asp:RequiredFieldValidator>
                </td>
            </tr>
        </table>
        <div class="auto-style3">
        <asp:Button ID="Button2" runat="server" Text="SUBMIT" Width="180px"
         OnClick="Button2_Click" />
        </div>
     </form>
  </body>
</html>
```

Below was a list of Security2b code written in C# for user to answer security two questions for secure verification.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data.SqlClient;
using System.Configuration;

public partial class Security2b : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if(Session["new"]==null)
        {
            Response.Redirect("Login.aspx");
        }

        if (Session["new"]!=null)
        {
            SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
            con.Open();
            string value1="select question1 from registration where username='" +
Session["new"].ToString() +"'";
            string value2= "select question2 from registration where username='" +
Session["new"].ToString() + "'";
            SqlCommand q1 = new SqlCommand(value1, con);
            SqlCommand q2 = new SqlCommand(value2, con);
            string l1_text = q1.ExecuteScalar().ToString();
            string l2_text = q2.ExecuteScalar().ToString();
            Label1.Text = l1_text.ToString();
            Label2.Text = l2_text.ToString();
            con.Close();
        }
    }

    protected void Button2_Click(object sender, EventArgs e)
    {
        SqlConnection con = new
SqlConnection(ConfigurationManager.ConnectionStrings["ConnectionString"].ConnectionString
);
        con.Open();
        string value1 = "select answer1 from registration where username='" +
Session["new"].ToString() + "'";
        string value2 = "select answer2 from registration where username='" +
Session["new"].ToString() + "'";
        SqlCommand q1 = new SqlCommand(value1, con);
        SqlCommand q2 = new SqlCommand(value2, con);
        string l1_text = q1.ExecuteScalar().ToString();
        string l2_text = q2.ExecuteScalar().ToString();
        if(TextBox1.Text!=l1_text.ToString() || TextBox2.Text!=l2_text.ToString())
        {
            Response.Write("Incorrect Answer");
        }
        else if(TextBox1.Text == l1_text.ToString() && TextBox2.Text ==
l2_text.ToString())
        {
            Response.Redirect("Welcome.aspx");
        }
    }
}
```

Below was a list of welcome user interface code to display the Welcome page.

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Welcome.aspx.cs"
Inherits="_Default" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title></title>
    <style type="text/css">
        .auto-style1 {
            text-align: center;
        }
        .auto-style2 {
            font-size: x-large;
        }
        .auto-style3 {
            font-size: xx-large;
        }
        .auto-style4 {
            text-align: center;
            height: 50px;
            width: 1052px;
        }
        .auto-style5 {
            text-align: center;
            height: 104px;
        }
        .auto-style6 {
            text-align: right;
        }
        </style>
</head>
<body>
    <form id="form1" runat="server">
        <div class="auto-style1">
        <div class="auto-style5">
        <div class="auto-style4">
        <asp:Label ID="Label1" runat="server" CssClass="auto-style2" Text="Hello
"></asp:Label</div>
            <div class="auto-style6">
                <asp:Button ID="Button1" runat="server" Font-Bold="True" Font-
            Underline="False" ForeColor="Black" OnClick="Button1_Click" Text="LOGOUT" />
                </div>
                <asp:Label ID="Label2" runat="server" CssClass="auto-style3"
                Text="Welcome Page."></asp:Label>
                </div>
           </div>
       </form>
   </body>
</html>
```