ENHANCING USER AUTHENTICATION FOR CLOUD WEB-BASED APPLICATION

DETAR BEQO

**FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2018

ENHANCING USER AUTHENTICATION FOR CLOUD WEB-BASED APPLICATION

DETAR BEQO

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2018

UNIVERSITY OF MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Detar Beqo

Matric No: WGA150006

Name of Degree: MASTER OF COMPUTER SCIENCE

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"): ENHANCING USER AUTHENTICATION FOR CLOUD WEB-BASED APPLICATION

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

ENHANCING USER AUTHENTICATION FOR CLOUD WEB-BASED APPLICATION

ABSTRACT

Together with the fast growth of networks and mobile devices, cloud computing has become one of the top technologies that everyone has been talking about in the last decade. At the same time, it has become one of the most attractive and effective business solutions for many companies worldwide. Organizations are gradually migrating their employees' data into the cloud environments, due to flexibility and cost efficiency which the cloud systems offer. However, as organization are moving their data and employees' information into the cloud, it has become a great challenge to design a secure cloud system, as it strongly lies on the chosen authentication, as it is the one which provides authenticity and confidentially respectively. Due to virtualization and multi-tenancy of the cloud systems, the complexity of security issues has even increased compared to traditional data centers, and in many instances user accounts have been compromised. As a result of these incidents in recent years, there is a growing lack of trust in cloud infrastructures. This thesis present research on cloud security challenges and how they can be addressed by enhancing the current authentication mechanism. Security requirements of SaaS environments differs from traditional data centers. To address a specific cloud security challenges, an enhanced authentication method is developed during this research work. Motivated by a number of security experts in cloud computing, we proposed an innovative solution of authentication for cloud web-based applications. We aim to improve on passwords with respect to both usability as well as security. It uses an enhanced encryption algorithm, and the data is stored securely in the cloud systems. The proposed authentication method, uses an enhanced method where the credentials are encrypted through an algorithm. As a result, the user's information is more secured, and the risk of compromised accounts is less, compared with two factor authentication. We have developed a cloud-based application that adapts the enhanced authentication method, and its security measurement were evaluated using IBM Application Security on Cloud tool. Results of different security testings are then compared to validate the effectiveness of the proposed authentication method.

ENHANCING USER AUTHENTICATION FOR CLOUD WEB-BASED APPLICATION

ABSTRAK

Dengan pertumbuhan jaringan dan peranti mudah alih yang pesat, pengkomputeran awan telah menjadi topik perbualan teknologi yang utama dalam dekad yang lalu. Pada masa yang sama, ia telah menjadi salah satu penyelesaian perniagaan yang paling menarik dan berkesan untuk kebanyakan syarikat di seluruh dunia. Kebanyakan organisasi besar telah membuat pilihan untuk mengalihkan data pekerja mereka ke dalam persekitaran pengkomputeran awan atau lebih dikenali sebagai cloud computing oleh kerana kemudahan fleksibiliti dan kos yang ditawarkan oleh sistem ini. Walau bagaimanapun, sebagai organisasi yang memindahkan data dan maklumat pekerja mereka ke dalam sistem awan, ia menjadi satu cabaran besar untuk mereka membentuk sistem awan yang selamat, kerana ia bergantung pada cara pengesahan yang ditetapkan pengguna sistem awan. Disebabkan virtualisasi dan multi-penyewaan sistem awan, kerumitan masalah keselamatan telah pun meningkat berbanding dengan pusat data tradisional dan dalam banyak keadaan, akaun pengguna telah dikompromi sejak beberapa tahun yang lalu. Tesis ini mempersembahkan penyelidikan mengenai cabaran keselamatan awan dan bagaimana mereka dapat ditangani dengan meningkatkan mekanisme pengesahan semasa. Keperluan keselamatan persekitaran SaaS, "perisian sebagai satu servis" berbeza daripada pusat data tradisional. Untuk menangani cabaran keselamatan awan tertentu, satu kaedah pengesahan telah ditingkatkan semasa penyelidikan ini dijalankan. Dimotivasi oleh sekumpulan pakar keselamatan dalam pengkomputeran awan, kami mencadangkan penyelesaian inovatif perkhidmatan pengesahan dan kebenaran untuk sistem awan dan aplikasi web. Kami berhasrat untuk memperbaiki kata laluan berkenaan sistem keselamatan awan. Kaedah pengesahan yang dicadangkan, menggunakan kaedah yang lebih efektif di mana dienkripsi dilakukan melalui algoritma. Dengan ini, maklumat pengguna lebih terjamin, dan risiko akaun dikompromikan berkurang berbanding dengan dua faktor pengesahan. Kami telah mencipta aplikasi berasaskan awan yang menggunakan kaedah pengesahan yang ditingkatkan, dan tahap keselamatannya dinilai menggunakan alat Acunetix. Keputusan ujian keselamatan yang berbeza kemudiannya dibandingkan bagi mengesahkan keberkesanan kaedah pengesahan yang dicadangkan.

ACKNOWLEDGEMENTS

First of all, I am thankful to Almighty Allah who has given me the privilege and the ability to study. I would like to offer special thanks to my supervisors: Assoc. Prof. Dr. Rosli Bin Salleh for his invaluable guidance, supervision, and encouragement to me throughout this research work. He, not only provided helpful suggestions, but also accepted responsibility to oversee this research, and guided me to the successful completion of this thesis. This thesis would not have been produced without his invaluable advice, excellent knowledge, unceasing support and enormous patience.

I would like to express my sincerest gratitude and appreciation to my parents for their endless love and support during my life. Without their moral support, this dissertation would never have been completed. Additionally, I would like to express my deep appreciation to my colleagues, who provided me with so much support and encouragement throughout this research and studies process. I wish them all the best in their future undertaking.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AI	:	Artificial Intelligence
API	:	Application programming interface
APPS	:	Applications
CC	:	Cloud Computing
CSA	:	Cloud Security Alliance
ENISA	:	European Union Agency for Network and Information Security
NIST	:	National Institute of Standards and Technology

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

1.1 Background

Cloud Computing is among the most talked about technology trends of the last decade. Enterprises claim to be on the road to become cloud centric, as cloud computing plan is spoken about by many vendors, and industry analysts spend their time following the cloud computing revolution.

A Study conducted by Gartner (Mc Donald and Aron, 2013) revealed that in 2013 worldwide IT executives believed cloud computing technology to be among the top five most effective technologies. Enterprises are starting programs which would help them build on their previous apps and realize the benefits of cloud, or are gradually allowing Cloud Computing to percolate in their structure, processes and infrastructure. Additionally, it means that they're currently devoting budgets or increasing the expenditure on cloud-computing programs.

The Future of computing is supposed to lie in the cloud computing technology, by which the main principle and aim are to lower the price of IT operations, while increasing productivity, accessibility, reliability, and flexibility and reducing reaction times (Brian 2008).

Even though their data and companies are moving into the cloud, individuals raise more and more concerns about the security and privacy of cloud technologies. Securing clients' data and organizations in the cloud is vital to service providers and most cloud system developers. Our work in this research exercise aims to offer insights into the authentication and security methods used in cloud systems.

The term cloud computing has been defined by the National Institute of Standards and Technology (NIST) as “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction” (Mell & Grance, 2011).

Despite all the benefits which cloud computing can provided to all the enterprise companies, according to the surveys of (Khan, Mat Kiah, Khan, & Madani, 2013), 74 percent of IT executives are hesitant to migrate their applications or solutions to cloud systems due to security issues.

Furthermore, one of the services of cloud computing which is called "Platform as a service", has been designed specifically for developers who would like to create their cloud web applications, deploy and easily managing all of them in one platform. Creating and managing an application in the cloud space has its own security concerns for consumers. The concern is always about on how the application will meet all the security standards, and how the consumers are going to use it without compromising their privacy, and take full advantage of it. Some of the current existing security solutions are “single-factor authentication” and “multiple-factor” authentication schemes. The cloud web apps should always have the best security solution in place, or otherwise the whole system will be compromised. Data leakage and broken authentication mechanism are the top security issues for cloud applications (Stuttard & Pinto, 2011).

According to another study, the concern which the majority of the users complain most of the time about cloud technologies is about the security and data breaches (Cloud Security Alliance, 2016). According to (Xiao & Chen, 2015), broken authentication has been identified as the top ten risk for cloud web applications.

1.2 Problem Statement

According to the (Cloud Security Alliance, 2013), data breaches and enabling of attacks in cloud environments can occur because of a lack of scalable identity access management systems, failure to use multifactor authentication, weak password use, and a lack of ongoing automated rotation of cryptographic keys, passwords and certificates.

According to the Association of Corporate Counsel (Counsel, 2015), 87 % of IT professionals are concerned about the security of cloud web based application and the way data is encrypted. In another research survey conducted for Druva (Druva, 2015), it's easy to feel vulnerable in cloud environments, simply because the encryption mechanism opens the door to hackers.

The encryption mechanism and account management for cloud computing technologies are also not standardize according to a research made by (Xiao & Chen, 2015). There is a lack of management techniques whereby there is no standard mechanism to encrypt the information for cloud computing models and save them securely in the cloud.

Nearly 5 million records are stolen per day, yet only 40% of all data stored in the cloud is secured with encryption and key management solutions according to (Ponemon Institute, 2018). The top ten risks in the web applications have been identified by Open Web Application Security Project in 2013 to be the following: Broken Authentication and Session Management according.

The purpose of this study is to propose and compare the authentication mechanism and the security requirements of the existing methods regarding cloud web application.

1.3 Research Questions:

This research study will answer the following questions:

1) What are the authentication mechanisms required regarding cloud web application?

2) What is the performance of newly proposed method regarding cloud web application?

3) Are there any security improvements between our authentication method compared with the other existing methods?

4) What does the new authentication mechanism do in regards to strengthening the security level of cloud web application?

1.4 Research Objectives:

We aim to enhance the security of cloud web-based application. This solution will help enterprises to strengthen their security and take advantage of such a great technology. The following are the objectives of this research:

1) To propose a new method of authenticating user accounts in Cloud web application, which will strengthen authentication security in the cloud application.

2) To evaluate the proposed method using one of the best security tools.

3) To validate that our authentication mechanism is better compared with other cloud web applications.

4) To implement the proposed method for cloud web application.

1.5 Contributions:

The contributions achieved by this research are as follows:

Proposed authentication mechanism: A new encryption algorithm has been developed during our research work, which will help to strengthen the security level of the cloud-based application. The main aim of our research work is to enhance the authentication method particularly for cloud web apps and making it highly secure. We used an encryption algorithm for enhancing the security and encrypting all the user's log in credentials. Encryption helps by adding another layer of security and protects the system from any outside attacker.

The strength of the encryption depends on the type of algorithm used to encrypt the data, and also the key size of that algorithm. We designed our algorithm in such a way that it encrypts the data very securely, and the key size used in our algorithm is 1000 bits, which create secure password hash and generate a unique hash string. The main reason behind choosing the key size of 1000 bits for our algorithm, it is because it is much harder to crack and break through that systems compared with other standard algorithm which use the key size of 256 bits, even thought if a powerful machine is used.

Cloud-based application: To signify the logic of the proposed authentication method and implement it, a cloud web-based application has been developed. The application has been used to perform experiments to test the security strength of the proposed authentication method under different scenarios. We used Acunetix Vulnerability Management, which is an automated web application security testing tool that scans your web applications by checking for vulnerabilities like SQL Injection, Cross site scripting and other exploitable vulnerabilities.

Results and findings: After the experiments and testings that we conducted, the data that we gathered have shown that our application has a stronger security level in place. This is due to the new enhanced encryption algorithm that we developed, and as a result it encrypts the data and save them very securely. We used to run vulnerability testing with Acunetix tool, against the existing apps such as the Gmail, Hotmail, and Yahoo app, and the results showed that our implemented application was much stronger and the security vulnerabilities were much lower compared with the rest of the apps. Our contribution will help the other developers to create the next generation of the cloud web apps, to implement the authentication mechanism just like our approach, and also strengthen the security issues in cloud. Our work will also help the other vendors or researchers, who would like to implement it on their apps as well.

1.6 Thesis outline:

This thesis is composed of five chapters as shown in the figure below:

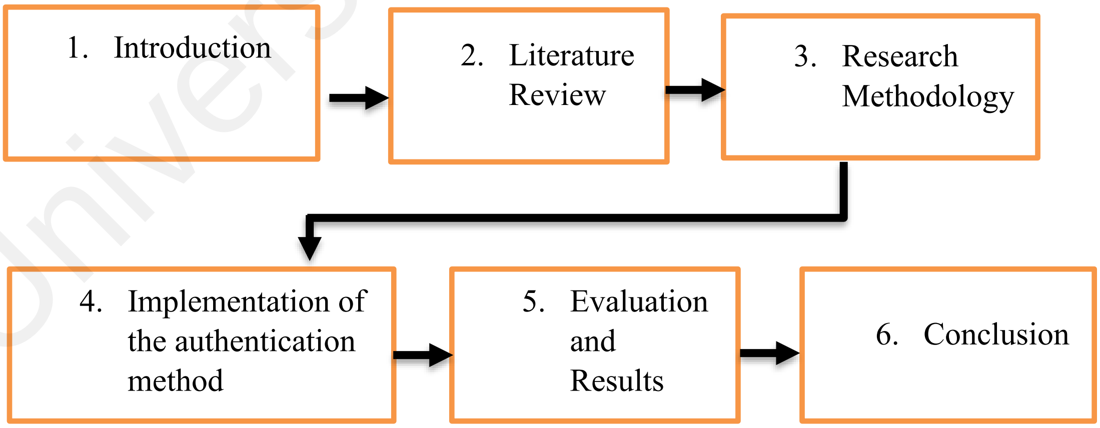

1. Introduction
2. Literature Review
3. Research Methodology
4. Implementation of the authentication method
5. Evaluation and Results
6. Conclusion

Figure 1.1: Thesis Structure

Chapter 2 presents the literature review of the existing authentication methods for cloud-based applications. It classifies the current authentication mechanisms based on the significant parameters, their challenges and the issues with them.

Chapter 3 discusses the researched methodology used in this research work as well as all the procedures taken to accomplish the desired results.

Chapter 4 proposes the authentication method for authenticating user accounts in a cloud environment. It describes the problem formulation, explains the architecture of the proposed authentication mechanism, and the assumptions of our research.

Chapter 5 discusses the significance of the proposed authentication method by comparing the results collected after the penetration testing, with other existing methods.

Chapter 6 concludes the thesis by explaining the findings of the research work, highlighting the significance of the proposed authentication method and discussing future directions of the research.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Cloud computing is changing the entire information technology system, and it represents one of the most significant changes that many of us will witness throughout our lifetimes (Cloud Security Alliance, 2010). This chapter will demonstrate the significance of cloud computing and its evolution, and that it is an important topic when we speak about information technology. First, we will discuss the service models in cloud computing as well as the deployment models, and together with its cloud reference structure formulate the technical basis of this research work. Then after that, we will discuss the current issues and the main concerns about cloud security, as well as the current method of authentication mechanism which cloud providers are currently using to secure their clients' information.

2.2 Evolution of Cloud Computing

Cloud computing is changing the consumption of computing and it often is referenced as the new model where many of the enterprise companies are leveraging computing resources. According to Nicolas Carr in his book "The Big Switch" (Lemke, Brenner, & Kirchner, 2017) he mentioned that cloud computing is a real revolution inside the information era and at the same time, it is an evolution of the industrial era. Cloud computing offers a variety of services and it is an evolution of computing that many of us used as a tool to share and collaborate on new ideas. Figure 2.1 depicts the development in the cloud computing environment, and how this has also affected the internet service providers (ISP). From the very first service providers (ISP 1.0), which was just providing just an internet access to the users, their service has been developing quickly, and many started supplying other services like email access as well as server access (ISP 2.0).

As this evolution was taking place, enterprises were making more demands and were asking for more services from their ISPs. Accordingly, the ISPs responded by offering dedicated data centers for hosting their customers' servers and applications. They also offered to those companies the infrastructure services which is required to run certain services, or in other words "co-location centers" ISP 3.0). Right after this phase, in the evolution of cloud computing turned towards the application of service providers (ASPs). which was not only offering computing resources, but, started to include customized applications for businesses (ISP 4.0).

The main difference between ASPs and cloud:

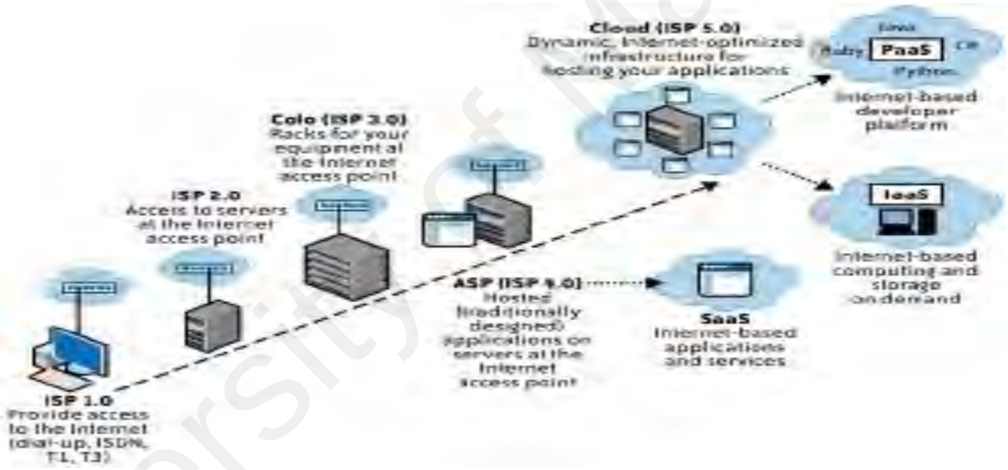
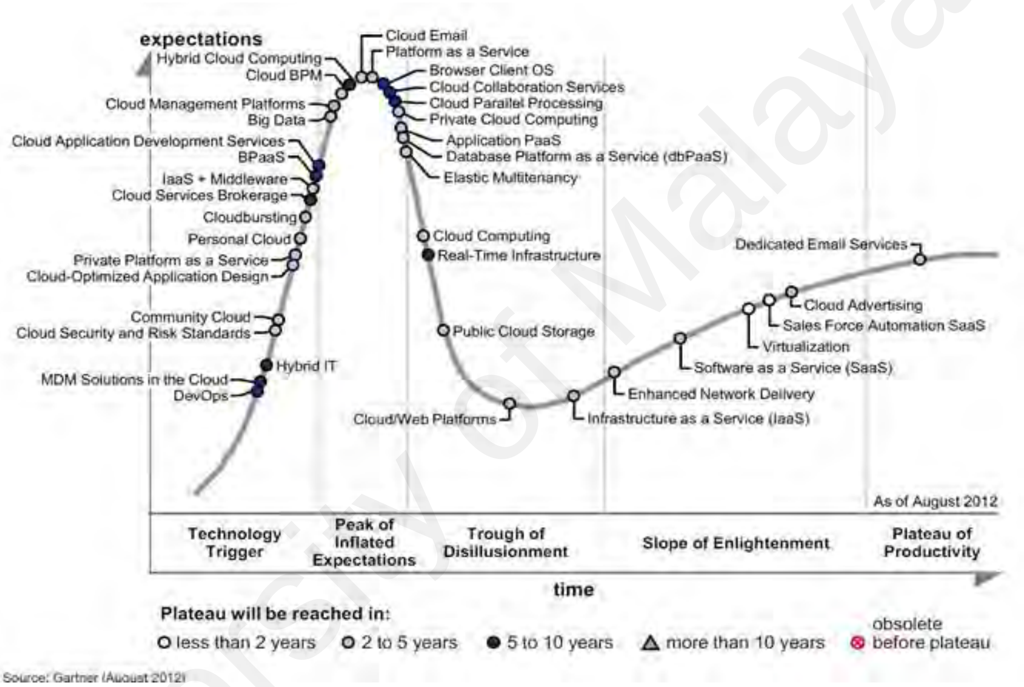

Cloud (ISP 5.0)
Dynamic, Internet-optimized infrastructure for hosting your applications
Java
Ruby
PaaS
C#
Python
Internet-based developer platform
Colo (ISP 3.0)
Racks for your equipment at the Internet access point
ISP 2.0
Access to servers at the Internet access point
IaaS
Internet-based computing and storage on demand
ASP (ISP 4.0)
Hosted (traditionally designed) applications on servers at the Internet access point
SaaS
Internet-based applications and services
ISP 1.0
Provide access to the Internet (dial-up, ISDN, T1, T3)

Figure 2.1: Evolution of cloud computing

The design of the underlying infrastructure is the key differentiator between the ASPs and the cloud service providers. Back in the early stages, the ASPs were offering their services to many clients, however, all these services were being provided through a dedicated infrastructure, and that means each clients had their own dedicated instances and no other clients could use the same computing resources. The Gartner Research group, is in the busines of predicting the hype of new technologies that will be used each and every year by many enterprises. They try to see what technologies are commercially viable in the market. While all the technologies of Cloud computing fall under the umbrella word "The Cloud", the Gartner Research Group devided them into three main categories, namly Cloud Computing, Cloud/Web Platforms and Personal Cloud Computing.

The Cloud technologies today are still categorized into three different entities, as they were discovered by the Gartner research group in the 2009 annual research for new technologies.



Source: Gartner (August 2012)

Figure 2.2: Technology hype cycle in 2013

As we can see from the picture above, the three main categories are Cloud Computing, Cloud Web Platform, and Personal Cloud Computing. According to Gartner back in 2013, cloud computing is more productive and it is not just a hype in the computing technology. Since 2011, Gartner has done more research in cloud computing compared with other fields in IT, and their studies have shown that cloud/web platforms are more productive for many users every year that passes on. Gartner also conducted some analyses in 2013, to see who the main influencers and the key decision makers were in the enterprise world, and whether have realistic plans for integrating this technology, due to the cost saving and other benefits that cloud providers offer (Stanley, Cradock, Bisset, McEntee and O'connell, 2016).

2.3 Cloud Deployment & Service Models

According to NIST, there are four deployment models when we talk about cloud computing (Mell & Grance, 2011). Figure 2.4 shows these typical cloud computing models: Private Cloud, Community Cloud, Public Cloud and Hybrid Cloud. The Private Cloud is a model in which the cloud is operated and managed by a third-party vendor, and the services are offered to a single tenant. Cloud services are offered over the Internet and are accessible through web applications. Security management is done by the vendor who is responsible for providing cloud services. Therefore, customers do not have a good insight into the physical and logical safety measures of this Private Cloud. A popular product of Private Cloud offer would be Elastic Cloud (EC2) from Amazon Web Services (AWS) (Varia & Mathew, 2017).

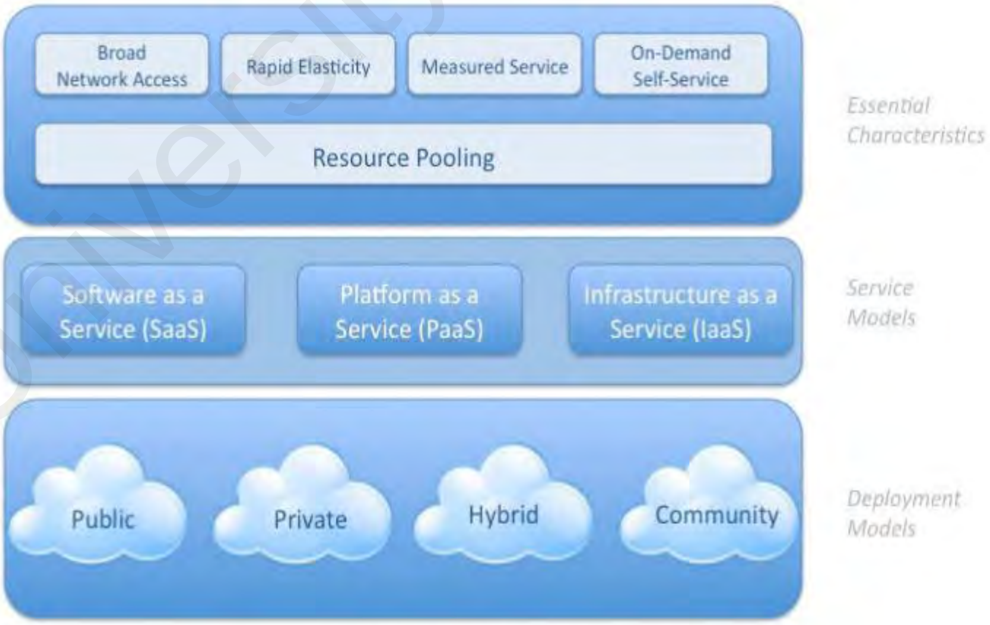


Figure 1—NIST Visual Model of Cloud Computing Definition[2]

Figure 2.3: Cloud deployment models and service models

1. Private Cloud. The cloud infrastructure is provisioned for private use by one organization comprising numerous customers (e.g. business units). It might be owned, managed, and run by the company, a third party, or any mixture of these, and it might exist on or off premises.

2. Community Cloud. The cloud infrastructure is provisioned for private use by a particular community of customers from organizations which have shared issues (e.g., assignment, security conditions, coverage, and compliance factors). It might be owned, managed, and managed with one or more of those associations locally, a third party, or any mixture of these, and it might exist on or off premises.

3. Public Cloud. The cloud infrastructure is provisioned for receptive use by the general public. It may be an organization, an enterprise or anyone who would like to use cloud services from any cloud vendor provider.

4. Hybrid Cloud. The cloud infrastructure is a composition of two or more different cloud infrastructures (such as private, community, or public infrastructures) that remained unique entities, but are bound together by proprietary or standardized technology that permits data and program reliability (e.g. cloud load balancing between clouds).

Finally, the three cloud service models which are defined by NIST (Mell & Grance, 2011) are as follows:

The three-service model, which contains three services known as the Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS), is described below. According to (Mell & Grance, 2011), NIST defines them as essential characteristics of cloud computing.

1. Software-as-a-Service (SaaS). The capacity given to the consumer would be to use the software while working in the provider's cloud infrastructure. The applications are available from several client devices through a thin client interface, like an Internet browser (e.g. online email), or even a program interface. The user does not control or manage the inherent cloud infrastructure such as servers, network, operating systems, storage, as well as individual program capacities, together with the possible exception of restricted user-specific application configuration preferences.

2. Platform-as-a-Service (PaaS). The capacity given to the consumer would be to deploy on the cloud infrastructure consumer-created or obtained software produced using programming languages, libraries, solutions, and tools supported by the provider. Again, the user does not control or manage the inherent cloud infrastructure such as servers, network, operating systems, or storage, but has command within the installed software and potentially configuration settings to the application-hosting environment.

3. Infrastructure-as-a-Service (IaaS). The capacity given to the consumer would be to supply storage, processing, networks, along with other basic computing tools where the user can deploy and run random software, which may include things like operating systems and software. The user does not control or manage the inherent cloud infrastructure but has control over operating systems, storage, and installed software; and maybe restricted control of selected networking elements (e.g. server firewalls).

Cloud computing isn't a single technology. It is better explained as a company growth, whose realization was enabled by numerous areas: computer architecture, operating systems, data communications, and operations and network management.

2.4 Security Concerns in Cloud Computing

All the enterprise analysts and researchers agree that cloud computing in the next innovation in IT, as it provides so many advantages. However, at the same time, it has also caused some security problems according to (Vaquero, Rodero-Merino, & Morán, 2011). In their research, they found some significant issues in cloud computing when they conducted a survey with a lot of CIOs and IT executives from the top companies in the US. These companies had been utilizing and taking advantage of cloud technologies for quite some time, and the security issues which they found are exemplified in Figure 2.6

Most of the enterprise companies use the private cloud solution for their IT infrastructure, as they are very much concerned about their information security and data integrity. Security in the cloud environment is more challenging compared with the traditional IT systems, because of some distinctive characteristics which those systems have. For instance, in cloud computing, most of the resources are shared and they can be accessed from anywhere in the world. Unlike the traditional IT system where a security incident can result in a specific issue, this risk is much larger in a cloud environment as it may also affect other enterprises which run their services inside the cloud and may affect their business operations.

Furthermore, the cloud providers also provide some best practices and recommendation such as ISO 27001 (Google, 2016), which explains in details how to go about protecting the data in a safe way, and also how to set privacy rules, so that the data never get compromised.

The following graph shows the security incidents in cloud computing in the past few years, where security remains the top main issue for the consumers:

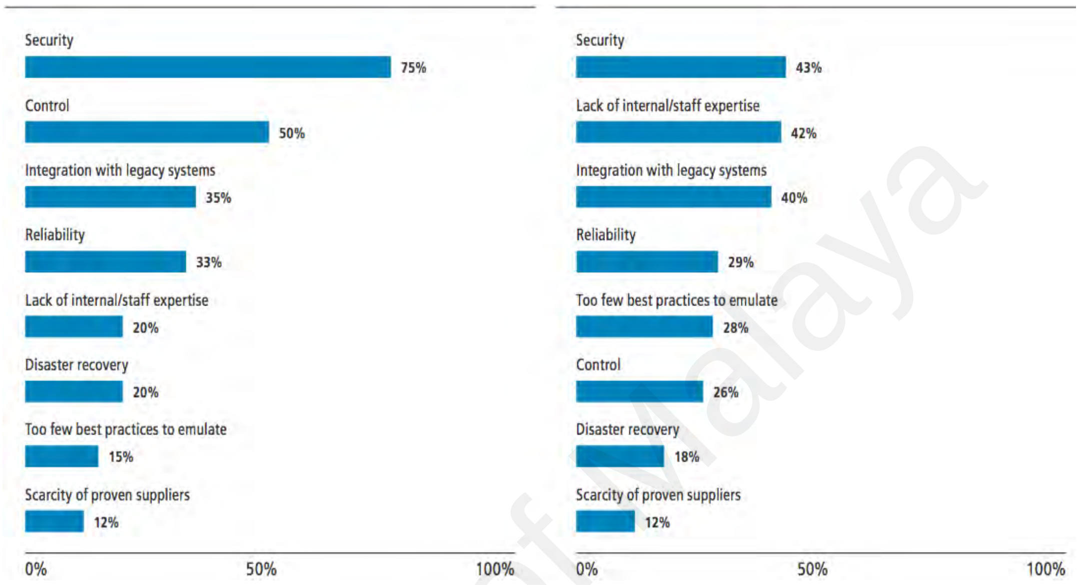


Figure 2.4: Top concerns on cloud computing

1. Amazon cloud services were not available for many consecutive days, and that resulted in data loss and a standstill for many business operations who were using AWS services (Matt Rosoff, 2011).
2. Microsoft accounts were also compromised due to some technical issues with the software Microsoft was using in their cloud (Naveen Thakur, 2012).

In April 2011, Amazon experienced an outage for their service of Elastic Compute or EC2, which caused many services like Reddit to be down for many days, and they were unable to serve their customers as well. This also shows that cloud computing operates very differently compared with IT traditional systems, and the damage it causes is very large as many enterprises share their resources under one cloud infrastructure (Matt Rosoff, 2011). While such an outage violates the quality of service and service level of agreement, which a cloud provider is supposed to deliver to their customer, it also created a loss of trust for the other enterprises who have not yet gotten on board with cloud services.

Since hardware sovereignty is given away in cloud computing security, health and information monitoring are critical to cloud users to build their services in an appropriate way regardless of which cloud model (public, hybrid, or private cloud) is used. This is already known from traditional IT outsourcing and providers try to gain the trust of customers by proving their compliance to IT security standards like ISO27001 (ISO/IEC 27001:2005, n.d.) or ISO9001 (ISO 9001:2008, n.d.). Amazon AWS so far seems to follow a contrary approach: although AWS provides status information about the cloud infrastructure at the Amazon Service Health Dashboard (Amazon Service Health Dashboard, n.d.), users can only see the service history for the last five weeks. Amazon's problems from April 2011 were not visible anymore to users by the end of May 2011. Maintaining consistent security across boundaries is complex and challenging for information security professionals (Mather, 2009). The Cloud Security Alliance defined a cloud model to consist of seven layers: facility, network, hardware, operating system, middleware technology, application, and user.

2.5 Related Work

When we talk about security in the cloud, there are many professional groups who publish about cloud computing. However, one of the most accredited groups is the Cloud Security Alliance (CSA). It is an organization which addresses many aspects of cloud computing in general, such as global security, compliance, and many other security-related legislation and regulation in cloud computing (Alliance, Simmonds, Rezek, Reed, & Alliance, 2011). There are many members around the world who are working very closely with the Cloud Security Alliance, to create better practices and improve the security of cloud services. It publishes the "Security Guidance for Critical Areas of Focus in Cloud Computing" (Alliance et al., 2011), which discusses the top security issues in the cloud environment and delivers a report of all the recent incidents of security. It provides an overview of all the important domains and also supplies appropriate recommendations accordingly. The latest guidance from the Cloud Security Alliance was released back in November 2011, which is also the current version, and it serves as a standard document for best security guidelines and practices in cloud computing.

Additionally, the Cloud Security Alliance also discussed the top threats in cloud computing and how to approach them in a professional manner, and in 2013 it released a report entitled the "Cloud Computing Top Threats in 2013" (Cloud Security Alliance, 2013). This report identifies the most urgent threats, which need critical attention and how to tackle them when using cloud services. Another organization which talks about the security and top threats to cloud computing is the "European Network and Information Security Agency" (ENISA). It discusses the current trends in cloud computing and what are some research areas that need to be addressed as cloud technologies are rapidly growing. It discusses the common architecture and security issues in the cloud environment, based on their research work and surveys. ENISA also provides some scenarios about the current vulnerabilities and risks of moving into the cloud especially for SMB companies.

"The German Federal Office for Information Security" is also an organization who does research in cloud computing and its current trends. It also publishes papers about "security recommendations for cloud computing provider" (Bsi, 2011). They discussed in detail some recommendations which any company should take into their consideration before making a transition into the cloud. They even discussed the type of questions which an enterprise should ask prior to choosing a cloud provider and see if they can meet all their business requirements such as security, data protection, and disaster recovery.

2.5.1 Authentication and security for Cloud Systems

Despite the attractive features provided by cloud technologies, according to the surveys of (Khan, Mat Kiah, Khan, & Madani, 2013), 74 percent of IT executives do not intend to migrate their infrastructures or solutions to cloud systems due to security issues. Customers will surely raise concerns regarding the privacy and data security of cloud systems, because their information and computation results are saved on shared cloud servers, and may be transmitted via open network connections (Ren & Wang, 2012). In order to construct a secure cloud system, like other information systems, it needs lots of important security properties such as:

Confidentiality. Implies that only the intended parties can read the secure details. Information leakage is a good example of confidentiality violation. Data stored in and sent to cloud systems might be encrypted to safeguard confidentiality.

Authenticity. Refers to the messages, transactions, and files that are assured to be genuine, i.e., made by the maintained parties and unaltered by somebody else. Please note that authenticity automatically implies integrity, where ethics means that data hasn't been altered in an unauthorized manner.

Availability. Means that data should be accessible when it's necessary. Availability is essential in cloud systems because the customers' companies may depend on the information stored on the external cloud servers. For example, denial-of-service (DoS) attacks specifically attempt to affect availability. This thesis concentrates on the confidentiality and authenticity of cloud systems, which are generally protected by encryption and authentication algorithms, respectively.

2.5.2 Cloud Web App Security

In the cloud computing world, they have a service especially dedicated for developers who would like to create their apps for the group, deploy and easily manage their applications, and they call it “Platform as a Service”. However, creating and deploying a cloud-based application is a security concern. What would be the best security solution, so that that the customers who are going to use it will have full trust in leveraging it. Some of the current existing security solutions are “single-factor authentication” and “multiple-factor” authentication schemes. The cloud apps should always have the best security solution in place, or otherwise the whole system will be compromised. Data leakage and broken authentication mechanism are the top security issues for cloud applications (Stuttard & Pinto, 2011).

A) Information leakage attack occurs when an application does not handle the requests properly, and it allows an attacker to break through the system.

B) Broken authentication attack occurs when an attacker pretends to be somebody else by using fake login credentials, and he/she manages to compromise the account.

As part of our research work, our aim is also to fix the issues with broken authentication. Therefore, we are looking at the current systems using different authentication methods, and after that, we will come up with our own proposal. Authentication is really important for web applications. As the number of users accessing them around the world increase, the number of attacks will also be higher. Since the applications are hosted in the cloud, if an attacker manages to break the authentication for an app, the chances are really high to break the authentication for the rest of the apps which are being hosted in the same cloud environments. Having said that, it is really crucial for the web developers to have a security solution for the web application where they can easily embed it into their source code.

2.5.3 Password-Based Authentication

The majority of the cloud web apps use password-based authentication, which is easy to implement and very practical to use. Typically, when using this type of apps, the user needs to provide the password prior to authenticating the accounts. When the user makes a request to access the application which is hosted with a cloud provider, they need to share the password for the cloud vendor to verify and grant access to the services. They will only allow access to those users who provide the correct information and matches the information stored in their databases. There are two basic concerns with this simple password-based authentication mechanism:

1) Users routinely pick low-entropy passwords that are particularly subject to dictionary attacks or brute-force search (McCarney, Barrera, Clark, Chiasson, & van Oorschot, 2012).

2) A device or server storing a high number of passwords is always a target for attackers, and the best way to keep passwords safe and minimize harms in the event the device or

server has been breached is non-trivial. As an effective countermeasure, all passwords should be obscured together with the user's specific high-entropy data (i.e., salts) by applying a computation/memory-heavy one-way function, namely password hashing, prior to storing them in the device or server. During the process of entity authentication, the password submitted by the user is processed by the same password hashing algorithm again and then the hashing outcome (i.e., hashtag) is compared with the one stored in the database. In this way, even if hashtags and salts are leaked to attackers, they cannot simply recover users' passwords by doing an exhaustive search or assessing pre-computed search tables. Additionally, the computation/memory-heavy process would make it more economically hard for attackers to construct efficient hardware for searching passwords and thus thwart brute-force strikes to a certain level. An example of a password hashing algorithm is the bcrypt (Sriramya & Karthika, 2015). Another important use case of password hashing algorithms is to serve as the key derivation function (KDF). KDFs are pseudorandom functions which are used to derive cryptographic keys out of a particular master or long-term qualifications such as passwords. Among the reasons why we want KDFs is that master credentials like passwords are usually alphanumeric combinations, but cryptographic keys require random, fix-length binary strings. The other explanation is that passwords could have reduced entropies and therefore are vulnerable to brute-force search. In the setting of cloud systems, clients may first create a secret key by applying a password hashing algorithm for their passwords, and then use the secret key to encrypt and authenticate their own data prior to sending it to cloud service providers.

2.5.4 Two-Factor Authentication

To further enhance the security of password-based authentication, a technology known as two-factor or multi-factor authentication, in which a user is required to provide additional authentication information besides passwords may be employed. The other piece of information may be what customers possess such USB tokens with built-in credentials or a particular information such as fingerprints or iris scans. The adoption of two-factor mechanisms makes it more challenging for attackers to bypass the cloud systems' entity authentication. Even if attackers could guess the password of a customer correctly, they still need to obtain the other part of information for authentication. Among the popular two-factor solutions in cloud security is to require a short passcode generated by a physical token such as the RSA SecurID or a software application such as Google Authenticator. The technical foundation of RSA SecurID and Google Authenticator is that the hardware token or software application shares a high-entropy credential with its corresponding authentication server, and a passcode is dynamically generated by applying a pseudorandom sequence generator to the credential together with certain time information, e.g. the passcode is regenerated each minute.

2.5.5 Three-Factor Authentication

"Three-factor authentication" uses the exact same mechanism for authenticating their users just like two-factor authentication, but with an extra layer of evidence identity. Besides requiring users to remember passwords which they need to key in from their devices, it uses biometric authentication as well. Biometrics identification includes identifications such as fingerprint or voice print (Jiang et al., 2016). Using this method tends to be more secure than the rest, as it requires unique attributes that every human being possesses. To implement a biometric authentication is as simple as adding another layer on top of the two-factor authentication in the authentication flow. However, the responsibility of handling biometric data in cloud environments is very sensitive because if the account by any chance gets compromised, the integrity of the users will be violated. Changing the attributes in biometric authentications is similar to the change of attributes which many users would make if they were to use only two-factor authentications. One thing which needs to be mentioned though is that biometrics validations require high computing resources and the cost is double when compared with two-factor authentications. Additionally, biometric attributes are very easy to replace, and it may result in security and data integrity violations of the users. In summary, adding another layer of authentication like biometric validations is quite pricey, and it does not protect from different types of attacks which two-factor authentications have already addressed (Huang, Xiang, Chonka, Zhou, & Deng, 2011).

2.6 Existing Authentication Solutions

As cloud computing is rapidly evolving each and every day, there are other solutions that are being proposed to enhance the security issue and better utilize all these services. As we are also approaching the phase of proposing our authentication solution for our research work, there is a need for us to be informed of the existing and planned solutions. Our solution must be demonstrably advantageous and provide some enhancement of the security issue in cloud computing. After an intensive research of previous solutions, a decision was made to come up with a new solution to strengthen the security of cloud web app applications. Our focus is to provide a more secure cloud web application where the user will have trust in using it, as we are working on the authentication mechanism for securing their services in the cloud environment.

2.6.1 SMS One-Time Password

Another method of authenticating users before using any of the cloud services is by using an SMS one-time password (OTP). Utilizing this mechanism requires the users to hold a device every time they authenticate their identity on a new device, and this adds another level of authentication (Sediyono, Santoso, & Suhartono, 2013). The users need to generate a passcode prior to accessing any application in the cloud environment. Once they enter the passcode, the system will send them an SMS code for one-time login to the application, and they need to enter that code into the system to gain access. A well-known example for two-factor authentication solution is Google's SMS one-time password. Basically, once the two-factor authentication has been enabled for a Google account, Google will send an SMS to the registered phone if the users need to authenticate the account on a new device. This passcode can only be used once, and it expires if not used within a limited period of time or user inactivity.

2.6.2 Device Generated One Time Password

Very similar to the other authentication method which we discussed earlier, the device-generated OTP solution generates a temporary code for a single login authentication. The only difference is that the user is required to install an application on their device first, and after that, the OTP generates a passcode from that software. An example would be the Google OTP solution as described by (Bo Zhu, Fan, & Gong, 2014), which they called Google Authenticator. The application generates a six-digit code every time the user makes a request for getting a new code. This solution again adds another layer for authenticating the user account prior to accessing the cloud services, where they need to first provide their login credentials for user identifications, and on successful authentication, a device-generated OTP passcode will be required. Google Authenticator requires one-time server authentication, and after that, all the passcode which the users are going to generate will be generated within the device itself, without having to connect over a public network.

2.6.3 Out-of-Band Authentication

Out-of-band authentication is a little bit different from the other authentication methods that we mentioned earlier, as the user receives a phone call for validating their account in this method. This approach is very simple, as the user tries to authenticate their identity on a web application, they receive a call immediately from the host server, which will provide the user with a passcode. The code is again a one-time login password and expires within a short period of time if the passcode is not used (Fujii & Tsuruoka, 2013). This alternative is quite similar to the SMS OTP solution as it contains a similar authentication flow to make sure that the user account is not compromised.

2.6.4 Image-Based Authentication

Image-based authentication is another method of authenticating users prior to allowing them access to certain services in the cloud environment. Basically, the user needs to select multiple pictures from a 3x3 picture matrix presented on the screen (Ritter, Schaub, Walch, & Weber, 2013). The images are presented to the end user in a random form, and all they need to do is select the correct images while authenticating their identity. This solution is encouraged by many technology companies, as based on some studies done, humans tend to remember much better by looking at images rather than using a password for account authentication. Additionally, this provides a highly secure mechanism as the user will only select the correct images presented on the computer screen and there will not be any fingerprints left. This solution is very difficult to attack, as even if attackers were to install a keylogger on the computer, they won’t be able to identify the pattern which was used during the image authentication.

2.6.5 Biometrics Authentication

Biometric authentication is trying to utilize the unique features that are exclusive and different to each human being. According to (Mudholkar, Shende, & Sarode, 2012), most utilizes features like the fingerprints which as we all know are different from one another. However, biometric authentication also includes other physiological and unique features like the tone of voice, eye iris or even face detection. An issue with biometric authentication is that not all devices support this solution for their customer, as this is quite challenging to implement on a device. Another fact about this authentication method is that it is very pricey to develop and requires a lot of computing resources to incorporate. Many customers also have a valid concern about this method, as the cloud providers may misuse their fingerprints for their own advantage.

2.6.6 Another Application for Authentication

This solution offers another authentication mechanism for allowing users to authenticate themselves to any cloud application. This is a great fit for those enterprise companies who are trusting the cloud vendors for providing them with an authentication solution, so they do not need to create their own authentication mechanism for authenticating their services in the cloud environments. According to (Chen et al., 2014), Google OAuth4 is a great example. When we talk about the other methods of authentication especially for cloud web app applications, whether the authentication is successful or not successful, in the end, all the requests are returned to the application itself. Google OAuth authentication uses a session token to authenticate any application that uses it. This method is widely used by developers who would like to use third-party API and provide more features to their clients by just embedding them into their application.

2.7 Research Gap:

In the previous section, we discussed some of the existing authentication methods and the existing issues. The systems descriptions were based on the authentication models in which they operate. Securing users' account in the cloud environments has always been a challenge for the vendor providers. When it comes to securing and authenticating user's accounts in the cloud environment, every vendor uses different approach. Authentication and key management for cloud computing paradigm is also not standardized. The absence of security and standard key management techniques for cloud does not allow a standard mechanism to scale well to the cloud computing model (Xiao & Chen, 2015).

Ref	Authentication Name	Issues Discussed	Proposed Solution	Strength/Benefits	Scope
(Mudholkar, Shende, & Sarode, 2012)	Biometrics Authentication	Tone of voice, eye iris, face detection	A solution to address authentication mechanism	Data Security and privacy will be ensured	Limited to addressing technical issues
(Ritter, Schaub, Walch, & Weber, 2013)	Image Based Authentication	Improvements in using image-based authentication	Correct images while authenticating	Security, privacy and compliance will be ensured	Limited to addressing identity and access control issue
(Fujii & Tsuruoka, 2013)	Out-of-Band Authentication	Using single Authentication identity	Phone call validation	Authentication and authorization will be ensured	Limited to addressing management and control issues
(Bo Zhu, Fan, & Gong, 2014)	Device Generated One Time Password	Code generation for a single log in authentication	OTP Solution	Authentication of accounts will be ensured	Limited to a single service provider
(Sediyono, Santoso, & Suhartono, 2013)	SMS one-time password	Client Compliance and trust	A solution to address identity management	Data security policy and procedures will be ensured	Scope to design standard SLAs

Table 2.1: Critical evaluation of security and privacy of cloud web apps

Based on the table above, even the tech companies including Google, Microsoft and Yahoo use the same methods as the once mentioned above, for authenticating their users prior to accessing their cloud services. Different authentication mechanisms have their pros and cons when it comes to cost, security and simplicity. However, the concern which the majority of the users complain most of the time about cloud web application is about the security and data breaches (Cloud Security Alliance, 2016). According to (Xiao & Chen, 2015), broken authentication has been identified as the top ten risk for web applications. Based on these finding we decided to develop a cloud web application which can resolve these authentication issues. Security has always been a concern for many users who would like to use these cloud technologies. Furthermore, the application provided to the end users by cloud providers is always hosted in the data center with users accessing it ubiquitously. One important characteristics of cloud applications is that they are not bonded with a specific user. One application may be accessed by many users at the same time. The cloud application inherits the same vulnerabilities as traditional web application and technologies. However, the traditional security solutions are not adequate for the cloud computing environment, because the vulnerabilities in a cloud web application can be way riskier than the traditional web application.

To address the above-mentioned issue, an application has been designed which incorporates some of the existing methods and enhance them even further. Enhancing the current methods will ensure the reliability and guarantee a better security method for cloud applications. The users will have a chance to use a much-secured cloud web application for authenticating their accounts, that is less costly and more reliable. In order to increase the security in cloud applications the priority should be given to an authentication mechanism which reduces the chances of compromising user accounts in the cloud space and protecting data integrity. Then, the detailed analysis comparison between the existing system and our method is explained in chapter five.

Finally, here is another overview comparison between the other existing methods which are now used by many tech companies such as Google, Microsoft and Yahoo. The graph shows their overall security level compared with our cloud web application. Based on our finding, we can see from the graph below the individual compliance for each application. The graph also shows that our application has the lowest overall security level, and this is as a result of our new implemented authentication method. Our proposed authentication which has been developed using an enhanced encryption algorithm strengthens the overall security of the application itself. Security is very crucial in the cloud environment, as that's where the trust between the services providers and the consumers is built. In addition, most of the cloud service providers offer an easy way to allow users to access all their services in the cloud, however, that does not mean that is the most secured way to leverage those services. Our approach aims to improve the security level and broken authentication issues of cloud web apps, by adding another security layer to the authentication mechanism, so that it strengthens the security level and the user's trust.

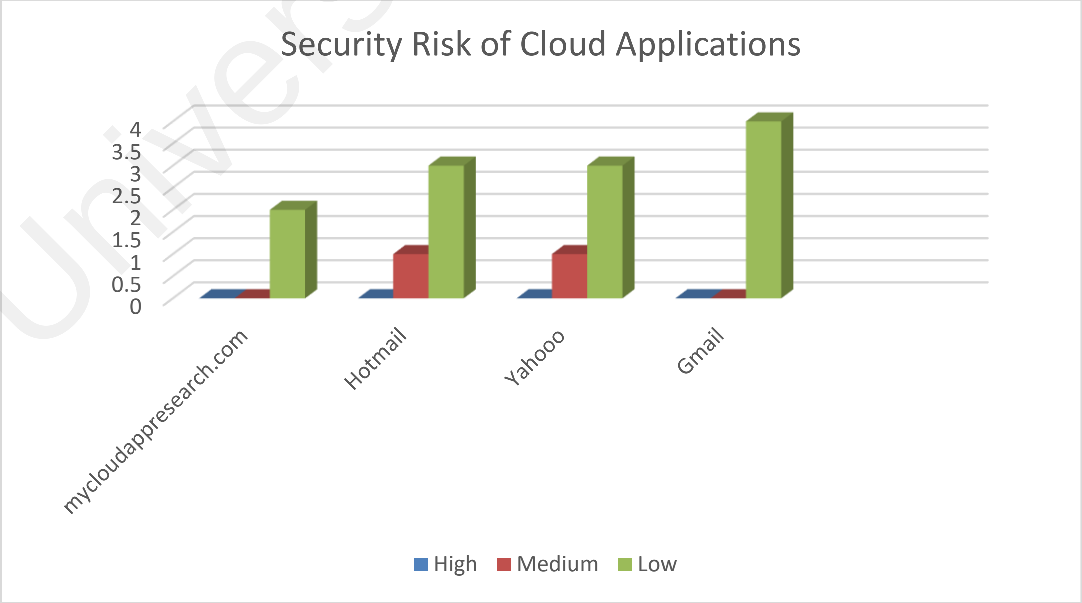


Figure 2.5: Security risk of cloud applications

2.8 Summary:

This chapter discusses the need for improving the security of cloud applications and enhancing their authentication method in a more effective way. It analyses the currently existing authentication methods for cloud application and highlights the commonalities and deviations of such methods on the basis of significant parameters. It also discusses the issues in the current authentication methods which are being used by the tech companies such as Google, Apple and Microsoft, and it highlights the importance of our authentication mechanism to overcome some of the challenges which exist today.

Chapter 3: Research Methodology

3.1 Introduction:

Selecting a suitable research approach is one of the most important aspects of a study. To identify which areas of cloud computing security needs more research, initially Cloud Computing (CC) challenges are found (this is done by searching the literature). Available methods for achieving this are literature review (LR) and systematic literature review (SLR). SLR is used to find all available data relevant to a particular research area (Kitchenham & Charters, 2007).

We study different cloud computing system and the current algorithm for encrypting user's information which they use to authenticate their user's account. To address this issue with the current methods in use, we then proposed another method with enhanced algorithm capabilities which adds another security layer to help the user and protect their data, when they are encrypted and saved in the cloud environment.

3.2 Research approach:

The analysis process for this research paper are from the problem statements to identifying the security issues for cloud web-based application and by studying literature. We also reviewed the existing systems, and we provided the research gaps, the pros and cons of the current systems. After our analysis, we decided to build a new cloud web application with a new authentication method in place, using the new enhanced encryption algorithm to protect the data in the cloud, which was produced during this research work. Then we launch the security testing for our application to measure the security level which we implemented. Both literature study and testing experiments are used for data collection. Afterwards we analyze this data for findings and suggestions.

Testing of a software is not merely a task under the software development life cycle process, it is the most important and a necessary activity. It ensures the required correctness, completeness, quality of the developed software, as per the customer's requirement (Aiya & Verma, 2015).

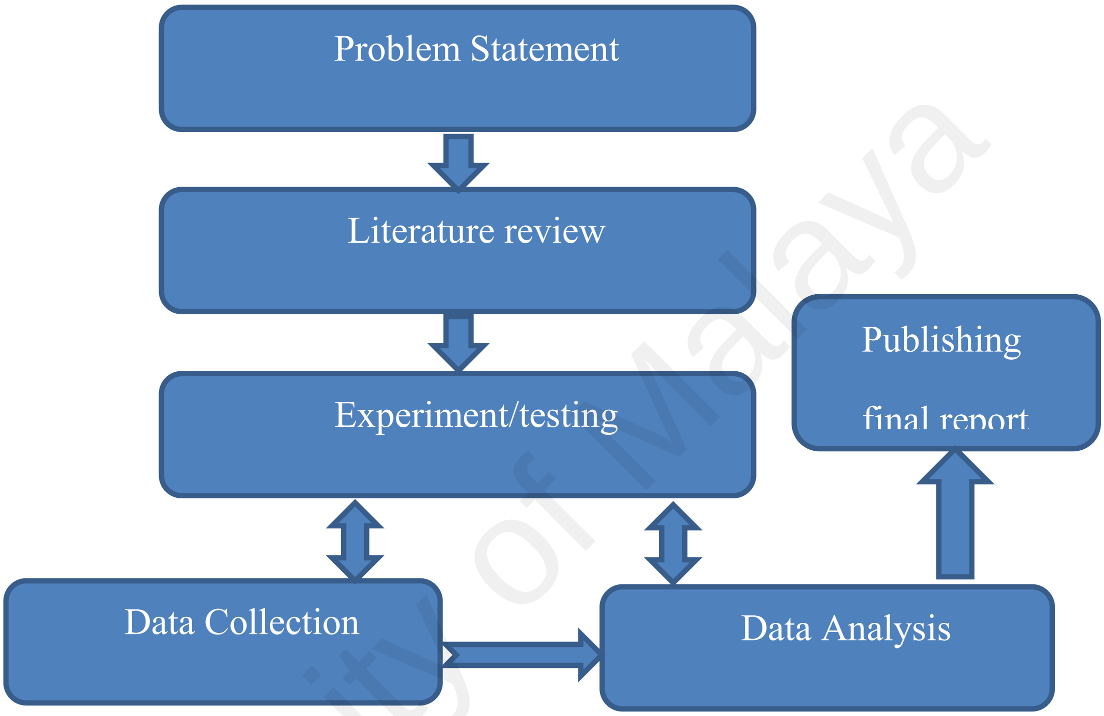


Figure 3.1: Research methodology procedures

3.3 Literature Review Approach

Selecting the right approach of reviewing the existing methods as well as the current issues in cloud web-based applications, is very important in the aspect of a research work. In that chapter, we discussed about the significance of cloud computing and its evolution, as well as the challenges which we identified. Then after that, we talked about the research gap, and we came up with a critical evaluation of security and privacy of cloud web apps.

Furthermore, after we discussed about our research gap and our analysis, we concluded that many cloud vendors use different authentication method prior to accessing their services, and there is not a standard mechanism for that. These different authentication mechanisms which we discussed about have their pros and cons, especially when it comes to security. At the same time, we also analyzed that majority of users complain most of the time is about security and data breaches, when they use cloud technologies (Cloud Security Alliance, 2016).

To address this issue, we designed and implemented a cloud web-based application and we enhanced its security by developing an algorithm. Enhancing the current methods definitely ensures reliability and it guarantees a better security method for cloud applications. Using our proposed method for cloud web applications, users will use a much-secure application for authenticating their accounts, that is less costly and more secure.

3.4 Experiment and testing:

Experimental results and testings for this research work, are collected by conducting testing in real-time environment. The proposed authentication method is evaluated by developing a real time cloud web-based application, and an enhanced encryption algorithm. The experimental testing is composed of GoDaddy as a remote server, to host the application on that server, and the Acuentix tool for launching the attack against the application and getting the results. The cloud web application is developed on Godaddy's infrastructure using PHP as the programming language, and at the same time it is connected to a cloud database. The application gathers information while users are signing up for creating their accounts. After that, using the developed encryption algorithm, the data is encrypted in a unique string form and stored on the database.

The cloud web application is developed for testing a new authentication method that is using an enhanced encryption algorithm to strengthen the security. The experiments are performed in two phases. In the first phase we talked about benchmarking our results. The results after we launched the security testing running the Acunetix tool (Pilli, 2016), are collected to check the security strength of the application. After that, the results that we collected are discussed to see the security level of the implemented application.

Furthermore, the second phase is about evaluation. Our testings and the results are compared against the security level of the other existing applications. We used a risk-based approach which is about focusing on identifying areas of highest risk, and then mitigate the risk. To mitigate the risk for the application after the testings, the results were collected using Acuentix tool. After all the testings, we discussed about the application vulnerabilities and the highest risk presented in a collaborative environment.

3.5 Data collection and data analysis:

The proposed cloud web application is tested with Acunetix tool (Pilli, 2016). Acunetix tool provides a single place in identifying security vulnerabilities (such as SQL Injection, Cross-Site Scripting, and Data Leakage) for a variety of applications. The service includes various types of application security scanning techniques, each of which identifies security issues in that application. A cloud web application is also developed during this research, together with an enhanced encryption algorithm, which is tested with different security scanning technique provided by Acunetix tool.

The Web Scanner launches an automatic security check of a website. A website security scan typically consists of two phases:

1. Crawling: Making use of Acunetix DeepScan, Acunetix automatically analyzes and crawls the website in order to build the site's structure. The crawling process enumerates all files, folders and inputs and is vital to ensure that all your website is scanned.

2. Scanning: Acunetix launches a series of web vulnerability checks against each component in your web application and in effect, it is emulating a hacker. The results of a scan include comprehensive details of all the vulnerabilities found within the website

The results of the application are analyzed for the security of authentication method, the security of user accounts in cloud environment, and also compromising accounts using the new method with the enhanced encryption algorithm. Acunetix tool also offers reports of the test results, which can then be compared with the existing system.

CHAPTER 4: USER AUTHENTICATION MODEL IN CLOUD COMPUTING ENVIRONMENTS

4.1 Introduction:

This chapter discusses the methods and procedures for solving the problem of user authentication in cloud computing. The chapter is organized into four sections. Section 4.2, discusses the problem formulation. Section 4.3 describes the user authentication of the proposed solution and its distinctive feature by highlighting the improvements of the authentication of the proposed solution. Section 4.4 depicts the conclusive remarks by highlighting the usefulness of the proposed framework.

4.2 Problem formulation

4.2.1 Introduction

As cloud computing is growing very rapidly, most of the enterprises nowadays have more than one cloud web application. An example of these application services could be e-mail servers and web servers. The main concern about cloud computing and especially cloud applications, is about confidentiality and security issues. Cloud applications need to have a strong authentication method to meet all the business requirements in today's world. Authentication is the process of having a strong user identity, presented electronically to a cloud application whenever a user makes a request to access certain services. (Burr et al., 2013). A large number of industries and businesses are gradually standardizing the authentication process which they use for sharing different resources using cloud technologies.

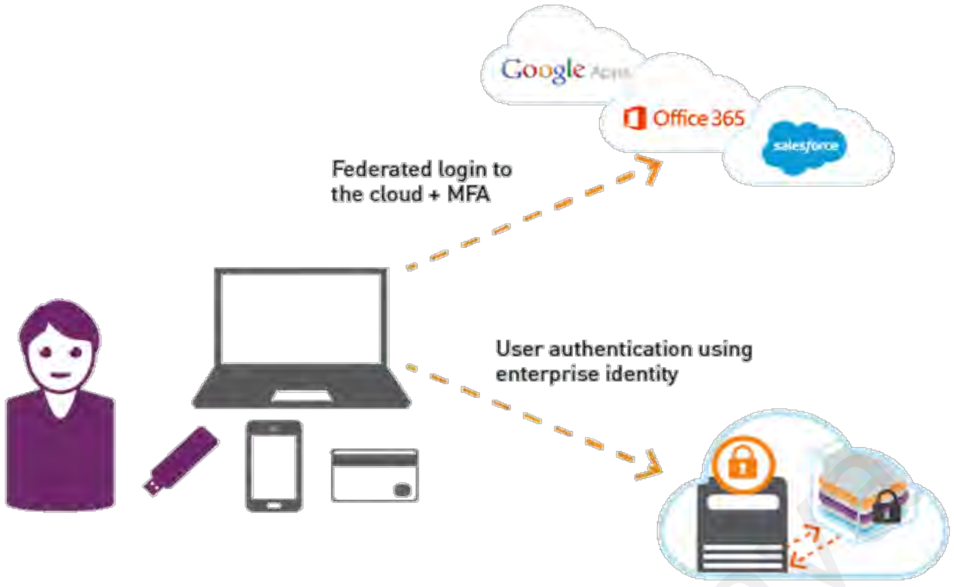


Figure 4.1: Models of user authentication in Cloud Computing

The authentication related terminologies as well as the authentication process is described in this area below:

According to (Hakobyan, 2012), the National Strategy for Trusted Identities in Cyberspace (NSTIC) in the US, which is a White House initiative, in order to support the enhancement of reliable, secured and interoperable identity solutions for cloud environment, they described the "Identity Ecosystem". As it has been explained in that strategy, enterprises and consumers can build a trusted relationship among themselves when using application hosted in the cloud, based on proper standard and a proper authentication method. "Identity Ecosystem" helps the consumers to manage multiple log in credentials for different online services. Individuals with a single digital identity credentials, can log in to different web-based applications, it is because the services providers trust a few third-party ident providers who manage individuals' authentication process.

Furthermore, the consumers during the online authentication procedures can control the revelation of their private credentials. The strategy highlighted by (NSTIC) describes four guiding principle about identity solutions for web application in order to have a great "Identity Ecosystem".

- Privacy-enhancing and voluntary identity solutions
- Secure and resilient identity solutions
- Interoperable identity solutions
- Cost-effective and easy to use

This type of "Identity Ecosystem" will be beneficial for enterprises using it in a corporate environment and for the consumers as well. The system will allow the individuals to perform a secured and reliable online authentication, without violating their privacy in any way. However, there are some disadvantage and vulnerability related issue to this system. It can represent a single point of failure when you have a single credential for authentication. In cases where the user happens to forget their credentials they will not be allowed to access any services through the web application, until and unless they recover their authentication credentials. Furthermore, if an attacker maliciously steals the credentials, it can lead them to access all the private information of the users which are saved in the cloud, until the issue with the compromised account is resolved.

Additionally, getting access to a huge amount of private information related to authentication credentials in identity databases will attract an attacker's attention, it is because the benefit is much more in case of hacking an identity database.

4.2.2 Registration and authentication phase

1. Registration Phase: Identity proofing is the first registration phase, when the user tries to identify themselves. Every user has unique attributes which will reveal their identity to the enrollment system. The registration authority is responsible to verify their identity based on the attributes which they provided previously. Usually, identity proofing works by collecting the user log in credentials such as a username and a password. Sometimes, on different occasions, the system may require the user to provide more information to prove their identity, and that may include the information on the user's national number or even a photograph. Registration authority then, presents all the collected results to the service provider, as soon as the identification of the user's confirmation is complete. In reality, cloud service providers allow users to register using their systems, and the users' information is stored in their databases, and if required a token is generated. This token then can be used for authentication purposes.

2. Authentication Phase: During the authentication phase, a token or a unique attribute needs to be presented to the verifier by the user. The user needs to make sure that that they have the right token and that they have control over that token. So, whenever a user needs to authenticate their account, they need to follow a certain communication protocol which will eventually allow them to present their token to the verifier. After that, the system needs to make a decision and works on the legitimacy of that request to check whether it is valid. Furthermore, the relying party will then allow access to their resources if the information were provided correctly, otherwise, access may be denied.

4.2.3 RSA SecurID

RSA SecurID is a well-known product, which allows users to authenticate their services in the cloud environment, and it uses the two-factor authentication method (B Zhu, 2015). It is just a hardware which has a small display showing a random pseudocode number changing every minute. The RSA SecurID is connected to a back-end server with which it shares a secret number. Every time an authentication code is submitted to the cloud provider by an individual, the service provider will check that code and will proceed to compute and compare it with their information which has been saved on their database. However, attackers may hack any pseudocode number which is generated randomly, if the servers are compromised. An incident of this kind happened back in 2011 (B Zhu, 2015), which caused RSA SecurID to be less effective despite using two-factor authentication.

4.2.4 Google Authenticator

Google has always been an innovator and amongst the pioneers in cloud computing. According to B Zhu (2015), Google Authenticator is a software that was developed as a result of the RSA SecurID incident discussed above. Its main goal is to provide users with another alternative, where they can easily authenticate their account to a service provider, in this case Google, without carrying a hardware device like RSA SecurID.

However, there are still some disadvantages when using Google Authenticator, as it needs to be connected to a back-end server first, and it requires the users to pair it manually to the service provider, which may not be a very user-friendly process.

4.2.5 Phone AUTH

Phone Auth is another authentication method and it uses a two-factor authentication mechanism as suggested by Czeskis, Dietz, Kohno, Wallach, & Balfanz (2012). First, the user's phone needs to be connected to a computer via Bluetooth, and the whole authentication process is assigned by the application itself. Customer's interaction is not required during this authentication mechanism as the application itself has been designed in such a way that it can authenticate itself.

However, when using the Phone Auth application in order to send the data over to the user's phone via Bluetooth, it requires an Internet browser, otherwise, the mechanism does not work. Additionally, in order for this method to work properly, the author of Czeskis et al. (2012) developed an extension and linked it to the Chromium web browser. The other web browsers which we normally use for surfing cannot support such capabilities.

4.2.6 Assumptions:

A web app based application is developed to implement our proposed adaptive user authentication model. As our application is a prototype application, which only tends towards addressing this research problem, naturally there are some assumptions. They are as follows:

(a) The web app application developed is a general-purpose application, which is using an encryption algorithm to strengthen the security of cloud apps.

(b) For the authentication process to be successful, the devices should be connected to the same cloud servers' network, so then it allows working in a distributed cloud network.

(c) The authentication request to be sent to cloud the device must have an internet connection either through a network cable or through a Wi-Fi connection.

4.3 The authentication mechanism

4.3.1 The overview of the proposed solution

According to (Burr et al., 2013) the security of cloud web apps is very important, as many users are accessing them every day. Thus, a strong authentication mechanism is required to meet user requirements. When we talk about authentication, all this information that are presented electronically, aim to build a strong user identity which does not violate the users' integrity. Again, only if the users provide their full identity and validate their data properly will they be given access to the cloud services.

There is no doubt that, cloud computing is the biggest thing in IT going on right now, where most of the IT innovations are taking place today. However, this has resulted in a few issues such as accounts being compromised and the integrity of users' security has been violated. This is because of lack of authentication mechanism which service providers use for the majority of their web application, and some users feel more comfortable using traditional IT systems rather than cloud systems. Encryption is a security strategy which different enterprises use in different ways to encrypt their data on premise. And cloud services operate very differently from traditional on-premise services. Encryption mechanism is very important as it adds another security layer for protecting the security of user data saved in the cloud. At the same time, it acts as a centralized point where access to data is enforced and can be audited. Encryption also provides a very important mechanism which ensures the confidentiality and integrity of the customer's data. It ensures that every information that is entered during the process of authenticating user's account in the cloud, the integrity of the data is not violated and it does not allow access of the data to unauthorized users.

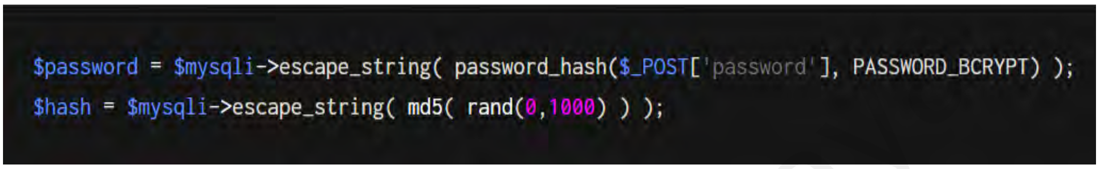
All the cloud providers use their own encryption algorithm when it comes to authenticating the users in their cloud environment. Additionally, they also have something called server-side encryption, and that means when they are authenticating the user information, they encrypt everything on behalf of their users. For instance, Google encrypts the information under the 256-bit Advanced Encryption Standard (AES-256), according to (Google, 2016a).

Furthermore, encryption is the process of replacing the data with unreadable code also known as ciphertext, and to decrypt it back to the original form, we need to use the key used in the algorithm which the information we encrypted in the first place. We designed a new encryption algorithm, and the key size that we used in our approach is different for the existing system, and it is stronger than the rest. We designed it in such a way that even if a powerful computer is going to be used, it is still going to be harder to break through that encryption algorithm. The key size used in our algorithm is 1000 bits, which create a secure password hash and generate a unique hash string.

Now, the encryption strength very much depends on a number of factors, such as how the key is created and managed as secured. At the same time, it also depends on the algorithm used and the key size of that algorithm. The reason we chose to use the key size in our algorithm as 1000 bits size, it is because the probability of cracking that algorithm is 3 times harder than cracking an algorithm with the key size of 256-bit which is the Advanced Encryption Standard which most of the service providers have in place.

4.3.2 The model for password algorithm

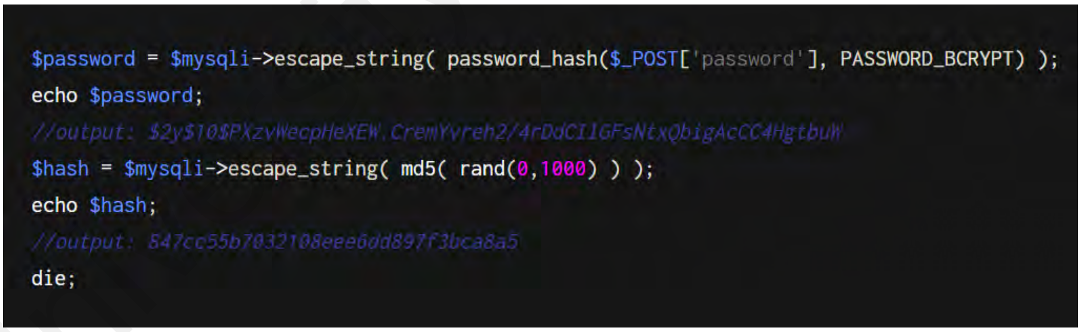
For the $password we have used the PHP function password hash() which takes in two parameters. The first is the raw password provided by the user and the second is the encryption algorithm, in this case, PASSWORD_BCRYPT, which creates a secure password hash and generates a unique hash string.



```
$password = $mysqli->escape_string( password_hash($_POST['password'], PASSWORD_BCRYPT) );
$hash = $mysqli->escape_string( md5( rand(0,1000) ) );
```

Figure 4.2: Password encryption

To generate a unique hash string, we simply use the rand() function which will generate a random number from 0 to 1000, and then we use the md5() function to generate a unique hash from the random number. If we printed out $password and $hash at this point, they would look something like this:



```
$password = $mysqli->escape_string( password_hash($_POST['password'], PASSWORD_BCRYPT) );
echo $password;
//output: $2y$10$PXzvWecpHeXEW.CremYvreh2/4rDdCIlGFsNtxQbigAcCC4HgtbuW
$hash = $mysqli->escape_string( md5( rand(0,1000) ) );
echo $hash;
//output: 847cc55b7032108eee6dd897f3bca8a5
die;
```

Figure 4.3: Generating a unique hash string

For security reason, the password cannot just be inserted in the MySQL database because that is not secure. So, we first must hash the password with a password underscore hash function which takes in two parameters, the raw password that the user entered being the first parameter. And then, the algorithm is going to use the so-called password decrypt which will basically create a random string out of the password. In the meantime, we also want to generate a new hash key which is going to take in a function called rand that generates a number between 0 and 1000. We then use md5 to generate a unique hash from that number.

4.3.3 Authentication Process

This section explains how the authentication mechanism works in our proposed method. First, when the user tries to access their services on the application server, the authentication process starts. Secondly, to explain how the whole authentication process works, consider the steps below:

1. The user initiates a request to authenticate their account using the login credentials.

2. The data will then be encrypted using either SSL/TLS encryption method depending on the user's browser capabilities.

3. The application's front end system directs the traffic to the application server.

4. Then, we hash the password with a password underscore hash function and takes in the raw password that the user entered.

5. The algorithm then uses the password decrypt to basically create a random string out of the password to secure it.

6. Additionally, we also generate a new hash key by using a function that is going to generate a number between 0 and 1000 and then generates a unique hash from that number.

7. On successful validation of the attributes, it will request for the user's information in the application server, and the servlet filter will take the control.

8. Then, the certificate credentials are extracted by the servlet filter to check for the particular user information in the LDAP user registry. The searching on the LDAP server must be done based on the unique certificate credentials, and the return should only be one entry.

9. The servlet filter stores locally all the LDAP entity, and on successful LDAP search, it returns the unique results.

10. After that, the cloud web server prompts the login page interface.

11. The web server is responsible to connect the login page to the web client.

12. Next, another servlet filter takes over the control. All the information which are stored locally in the LDAP entity will be compared by the servlet filter to see if it matches the login credentials which the user provided. The application server then will only take control if the credentials match with the attributes in the LDAP server.

13. The user will be prompted to answer three correct security questions, which are chosen randomly using a function method.

14. We also encrypt the security question by creating a random string out of the security question to secure it.

15. If the user is able to answers all three security questions correctly, then they will be granted access to their accounts, otherwise, the access will be denied even if the first part of the authentication of their attributes such as username and password was provided correctly.

```
    <?php
    class Encryption
    {
    public function encrypt_decrypt($action, $string)
       {  $output = false;
          $encrypt_method = "AES-256-CBC";
          $secret_key = '1234567890';
          $secret_iv = '1234567890';
          // hash
          $key = hash('sha1000', $secret_key);
          // iv - encrypt method AES-256-CBC
          $iv = substr(hash('sha1000', $secret_iv), 0, 16);
          if ($action == 'encrypt') {
             $output = openssl_encrypt($string, $encrypt_method, $key, 0, $iv);
             $output = base64_encode($output);
          } else if ($action == 'decrypt') {
             $output   =   openssl_decrypt(base64_decode($string),   $encrypt_method,
$key, 0, $iv);
          }
          return $output;
       }
    }
```

Figure 4.4: Encryption algorithm

4.4 Conclusion

We proposed an adaptive authentication method for the cloud-based application as a solution to improve security and simplicity. The existing authentication methods require additional devices such as a cellular device or a security key to authenticate user accounts in the cloud-based applications, besides their primary log in credentials.

There are also other ways of authenticating user accounts in cloud environments, some of which are costly and some others that can still be compromised. Our aim is to enhance the authentication mechanism in cloud environments and above all, to make it more secure. Our proposed approach consists of an enhanced encryption algorithm which will make the security of the cloud web applications much stronger compared with the existing systems.

Encryption adds another layer of security for the user's data, to make sure that their integrity is never violated. Encryption is a process of changing the data and transforming it to an unreadable format so that an attacker will not be able to change it back to the original format. The strength of the encryption depends on the type of the algorithm used, and also the key size of that algorithm. We designed the algorithm is such a way that the key size is different from the other algorithm, and the key size used in our algorithm is 1000 bits, which is very solid to make the data more secured. The reason we chose the key size of 1000 bits of our algorithm is because the probability to crack that is much harder, even if a powerful machine is used compared to the 256 key size which is used in standards algorithm.

CHAPTER 5: EVALUATION AND RESULTS

5.1 Introduction

This chapter reports on the data collection and the results for the evaluation of the proposed solution of the user authentication in cloud environments. It describes the tools used for testing the authentication method, the data collection technique and the data analyzing method used. This chapter is organized into four sections. Section 4.2 explains the experimental setup and the tools used for the implementation and the testing of the proposed authentication method. Section 4.3 presents the benchmarking technique and the description of the results for the evaluation of the authentication method. Finally, section 4.4 extracts conclusive remarks.

5.2 Evaluation of the authentication model

This section presents the methodology used for the evaluation of the adaptive authentication model. It discusses the experimental setup, prototype applications used for the evaluation, and the stats used for getting the results.

5.2.1 Experimental Setup

The proposed authentication scheme is evaluated by a real implementation of a web application. Experimental results are collected by testing the application in a real-time environment. The following section describes the experimental setup for the real implementation environment.

The experimental setup for testing the authentication method in a real environment is composed of GoDaddy as a remote server, the application itself running on that server, and the Acunetix tool for getting the result (Pilli, 2016). The cloud web application is developed on GoDaddy's infrastructure using PHP as the programming language. It is also connected to a MySQL database where the users' information is encrypted and stored. The application gathers information from the users while they are signing up for their accounts. Using an algorithm, the data is encrypted in a unique string form, and after that, stored on the database.

The cloud web application is developed for testing a new authentication method which is using a password algorithm to encrypt the user’s data before they are saved in the database. The experiments are performed in two phases. Benchmarking is done in the first phase. The results of the security report are collected by running the Acunetix tool (Pilli, 2016) to check the security strength of the application. Results are discussed to see the security level of the application itself.

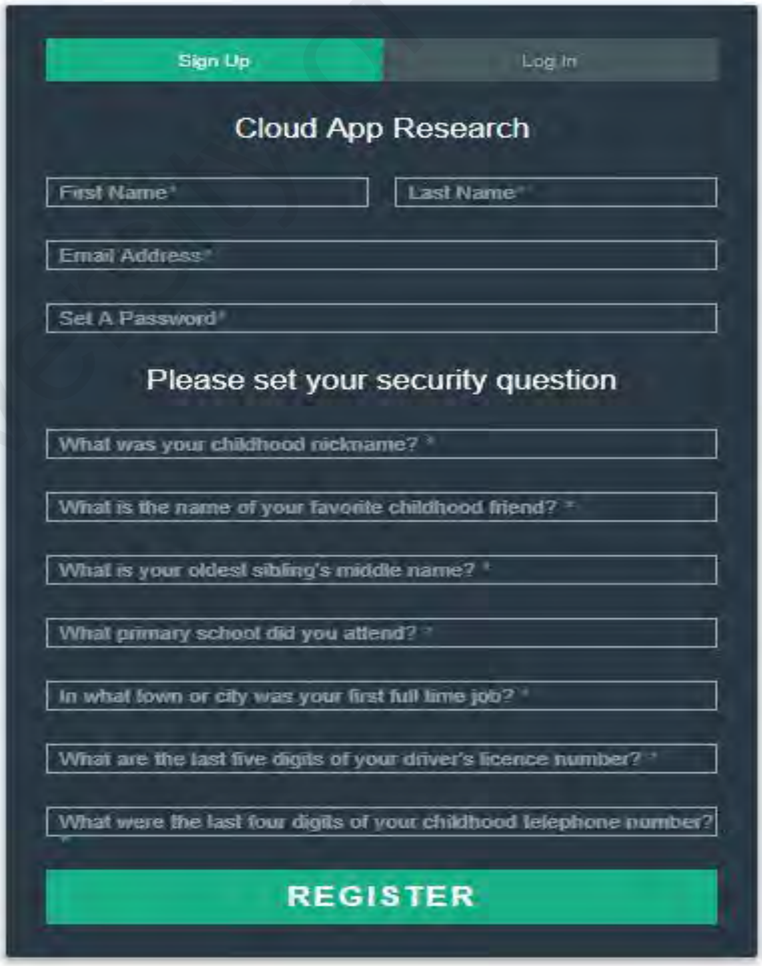
The second phase is the evaluation. These results are compared with other existing applications and their security levels. In this phase, we used a risk-based approach which is about focusing on identifying areas of highest risk, and then mitigate the risk. Addressing application security goes beyond scanning applications for vulnerabilities. The Acunetix tool (Pilli, 2016) will be running in the background to mitigate the risk for the application and the results are collected. Once the results were collected, we discuss the highest risk presented, and the application vulnerabilities in a collaborative environment.

5.2.2 The cloud web-based application

In order to test the security of our proposed authentication method, we also developed a cloud web app running on the GoDaddy’s servers. The cloud web-based application provides the users with a user interface where they can enter their information and create their account. We have implemented our proposed authentication method in this application. The application has been developed using PHP as the programing language, and the authentication mechanism includes the following components:

1) First, the application lets the user enter the information. Once their account has been created, it encrypts the information using the encryption algorithm, PASSWORD_BCRYPT, which generates a unique hash string and save the password securely in the database.

2) For authenticating the user accounts, there are two authentication phases. The primary phase uses an LDAP server to authenticate the user accounts based on their login credentials, and the second phase will prompt the user to answer three random security questions which are also encrypted using the encryption algorithm.



Cloud App Research

First Name*
Last Name*
Email Address*
Set A Password*

Please set your security question

What was your childhood nickname? *
What is the name of your favorite childhood friend? *
What is your oldest sibling's middle name? *
What primary school did you attend? *
In what town or city was your first full time job? *
What are the last five digits of your driver's licence number? *
What were the last four digits of your childhood telephone number? *

Figure 5.1: Graphic interface of the cloud application

5.2.3 Scope of research experiments

Technical security assessments:

The purpose of the technical security assessments is to determine the security posture of a system, network, or application. In addition, a successful technical security assessment identifies security gaps and recommends remediation steps to make the application more secure. Technical security assessments can be used to find security weaknesses and technical vulnerabilities, as well as determine compliance with internal and external security standards or benchmarks, such as PCI or ISO/IEC 27001 (Google, 2016b).

Technical security assessments are not a substitute for security. These assessments are conducted after security controls are implemented to measure the security status.

In addition, doing these assessments allows us to compare multiple results to identify trends and determine if security gaps are being remediated in a timely manner. In summary, a well-run security assessment is one of the best ways to understand the current state of an application's security.

Overview of technical security assessment techniques:

There are a wide variety of ways to find out how secure systems and networks are, and they fall into three general categories of techniques, review, target identification and analysis, and target vulnerability validation. Review techniques are often manual examinations of systems, applications, networks, policies, and procedures to ensure they meet the minimum-security requirements.

These techniques include reviewing system and network documentation, firewall and switch rulesets, and system configurations. Conducting network sniffing to examine the current state of the network and using file integrity checking to ensure that key files haven't been modified are also considered review techniques.

Target identification and analysis techniques are used to identify and analyze systems, networks, and security vulnerabilities which may be relevant to the assessment. These techniques are often performed using automated tools or systems which can conduct network discovery, network port and service identification, vulnerability scanning, and wireless network scanning.

Finally, target vulnerability validation techniques are used to confirm that any vulnerabilities identified in earlier testing are valid. Examples of these techniques include password cracking, penetration tests, and social engineering.

Selecting my testing viewpoint:

There are several different viewpoints one can take when conducting technical security assessments. The viewpoints we considered are external and internal. External security testing simulates what an attacker outside, from the Internet, could have been able to accomplish.

For external testing, we focused on security vulnerabilities which allowed us to penetrate the perimeter defenses and gain access to internal applications' data. For the external security assessments, we followed three phases, reconnaissance, enumeration and testing. **Reconnaissance techniques** include researching publicly available information about the target, such as domain registry information, to gather any relevant details which could help conduct the tests or even reveal potential vulnerabilities.

The next phase, **enumeration**, is the process of identifying the systems, services, ports etc., which are in scope for the assessment. Enumeration is accomplished by using network discovery and scanning tools. After reconnaissance and enumeration, **testing** techniques are performed. These typically include running automated vulnerability scanners, which look for thousands of vulnerabilities in application protocols.

Internal security testing is conducted from the viewpoint of either a malicious insider, someone who has trusted access to systems but may want to gain access to unauthorized data, or an attacker who has already breached the network perimeter. Because the network perimeter isn't a factor, internal security tests have much more latitude than external tests. For instance, additional techniques such as network sniffing are used in internal security tests. Like external security testing, the internal security test follows a similar reconnaissance, enumeration, and testing phases approach. It is started by granting the assessors regular user access and seeing if they're able to elevate their privileges and gain access to unauthorized data. This can provide significant findings in a security testing.

5.2.4 Strength of the research:

The main strength of our application is that our encryption algorithm is unique, and to test it we developed a web-based application and embedded the algorithm in it. The application can run on many operating systems today such as Windows, Mac OS, and Linux. The application is designed in such a way that it is compatible with most operating systems around the world.

The main purpose for choosing this approach is so that the user does not have to worry about the environment the application will run on. At the same time, it also makes it easy for us to conduct data collection. The application will be available on the Internet, so we can devise a testing approach for testing the security status of the application. The data collection will be commenced in a couple of hours after the testing is complete and we can easily run the security analysis report.

5.2.5 Limitation of the research:

Due to the prototype application-based research, there are some limitations towards our research as well. Our target, for now, is the development of a web-based application which will be hosted in the cloud environment. We have not considered developing an application for Android nor iOS platforms. In addition, we have only conducted this research on a limited number of devices due to the time and constraints of this research.

5.3 Results:

This section describes the results obtained from the experiments performed to evaluate our proposed authentication method. For our experiment, we used the Acunetix Web scanning tool (Pilli, 2016). The Acunetix Web vulnerability scanner is an application that automatically checks an entire website or web application for security vulnerabilities. It is capable of scanning the entire code and scripts for possible vulnerabilities that can be exploited. It also includes some penetration testing tools to automate the whole process. It creates fake attacks and checks the response of the website against the attack. After scanning, it displays a detailed report of what it has found and how to improve the security.

The Acunetix scanner divides the types of scanning according to the severity of the types of web attack. It has four levels of severity, high, medium, low and informational. Acunetix is used to detect various types of web vulnerabilities and some of them are listed as below:

i) Authentication: Brute Force.

ii) Insufficient Authentication.

iii) Insufficient Authorization.

iv) Information Leakage.

v) SQL Injection.

SQL Injection and Brute Force attack scans fall under the high severity type as they are considered the most dangerous attacks in the web security. Other attacks are also categorized according to their severity on the web services.

5.3.1 Benchmarking experiments:

This section is a detailed report that explains each vulnerability found according to the individual compliance categories.

Authentication: Brute Force attack

A brute force attack is an automated procedure of learning from errors used to detect somebody's username, password, credit card variety or critical cryptographic material. We utilized Acunetix’s expert brute-force authentication strategies based on HTML authentication or Basic authentication, based on the research findings of Wasc (2017).

Table 5.1: Brute Force attack

Vulnerability type:	Number of detected vulnerabilities:
Brute Force attack	No alerts in this category

Insufficient Authentication:

According to Wasc (2017), Insufficient Authentication occurs every time a website enables an individual to gain access to sensitive content or functionality without needing to properly authenticate. Web-based management programs are a fantastic example of the websites providing access to sensitive functionality. Depending upon the particular online source, these apps must not be directly reachable without requiring the consumers to correctly verify their identity. To get around setting up the authentication, a few resources are safeguarded by "concealing" the specific location and not linking the location into the main website or other public places. It is necessary to understand that even when a resource is not known to an attacker, it remains accessible specifically via the URL. The specific URL can be detected by way of a brute-force recruitment for common files and directory places (such as /admin), mistake messages, referrer logs, or perhaps even documented in help files. These resources, if they are content or operation driven, should be satisfactorily shielded.

Table 5.2: Insufficient Authentication

Vulnerability type:	Number of detected vulnerabilities:
Insufficient Authentication	No alerts in this category

Insufficient Authorization:

Insufficient Authorization is when a web app enables access to sensitive content or functionality that should require increased access control constraints. It does not mean that a user needs to get access and should really only be granted when an individual is authenticated into a website as described by Wasc (2017).

Authorization procedures are conducted utilizing what a user, service or application has been allowed to perform. Special actions should be governed by restrictions that have been thought in line with the policy. Sensitive parts of an app may need to be strictly forbidden to other users, allowing only an administrator.

Table 5.3: Insufficient Authorization

Vulnerability type:	Number of detected vulnerabilities:
Insufficient Authorization	No alerts in this category

Information Leakage

Data Leakage happens every time a website displays sensitive information, for instance, programmer remarks or error messages, which may support an attacker at getting through the system. Sensitive information might be present inside HTML codes, error messages, or simply in plain sight. There are various ways an app might be coaxed into showing this type of information. While data leakage does not necessarily represent a breach of safety, it will give an attacker a useful insight for prospective abuse. Leakage of sensitive information can carry many levels of danger and should be limited (Wasc, 2017).

Table 5.4: Information Leakage

Vulnerability type:	Number of detected vulnerabilities:
Information Leakage	No alerts in this category.

SQL Injection

Based on the same report (Wasc, 2017), SQL Injection is an attack technique used to exploit web apps that construct SQL statements. It is feasible for an attacker to alter the construction of back-end SQL statements when a web application does not properly handle the inputs. The process will likely run using exactly the same permissions as the element that executed the command, as soon as an attacker has the capability to modify a SQL statement (e.g., Database server, Internet application, Web host). The effects of this attack can allow attackers to execute orders in the system or gain total control of the database.

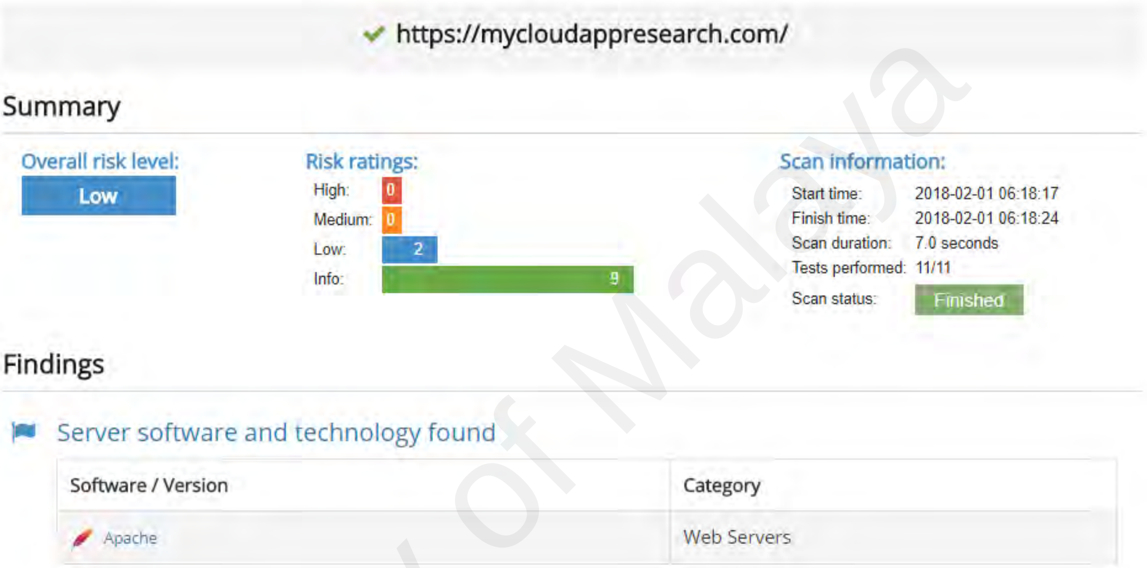
Table 5.5: SQL Injection

Vulnerability type:	Number of detected vulnerabilities:
SQL Injection	No alerts in this category.

5.3.2 Evaluation of our authentication mechanism

The authentication of our application happens in two phases. The first one happens when the user tries to log in using their login credentials such as username and password, and the second one happens when the user is prompted to answer three security questions. All the inserts in the database are encrypted using a password algorithm, then a unique hash string with a unique value is generated and after that, saved in the database. We also use a function which will generate a random number from 0 to 1000, and afterward use the md5() function to generate a unique hash from the random number. When the authentication happens, it has to match with the exact unique hash key, otherwise, the authentication will fail and the user will not be able to log in to their account. Having two steps of authentication increases the security level and strengthens the web application's authentication mechanism. As we can see from the figure below, based on the scanning results, there are only a few risks and their levels are very low. The most important part is the authentication, and in this case, there is only one low-level risk about the web application's authentication mechanism. Additionally, there is also another low-level risk of stealing customer session cookies which might result in allowing the hackers to view user records. Again, the risk level is very low.

The picture below shows the overall risk level of our application after the penetration testing using Acunetix tool. As we can see from the results, there are only two low-risk findings. One is about the authentication mechanism which may be possible to bypass, and the other one is about cookies where it may be possible to manipulate the customer session.



✓ https://mycloudappresearch.com/

Summary

Overall risk level: Low

Risk ratings:

High:	0
Medium:	0
Low:	2
Info:	9

Scan information:

Start time:	2018-02-01 06:18:17
Finish time:	2018-02-01 06:18:24
Scan duration:	7.0 seconds
Tests performed:	11/11
Scan status:	Finished

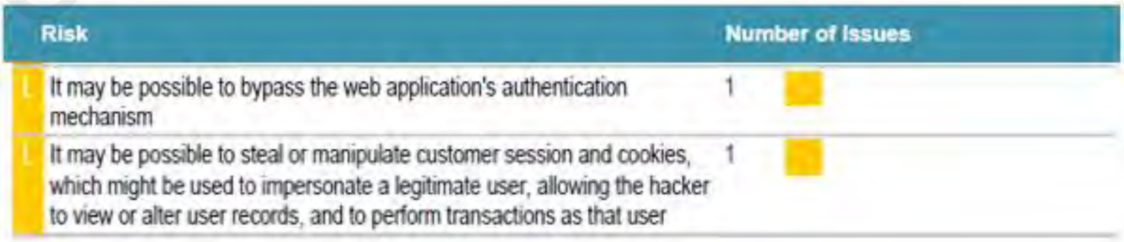
Findings

Server software and technology found

Software / Version	Category
Apache	Web Servers

Figure 5.2: Overall risk level after the pen testing (Low)

The table below shows the two risks found from our tool after the scanning is complete. It also provides a small description of what the risk is all about together with the number of issues.



	Risk	Number of Issues
L	It may be possible to bypass the web application's authentication mechanism	1
L	It may be possible to steal or manipulate customer session and cookies, which might be used to impersonate a legitimate user, allowing the hacker to view or alter user records, and to perform transactions as that user	1

Figure 5.3: Security risk results after the penetration testing

5.3.3.1 Comparison with the other existing models.

Figure 4.4 below shows the security risk between our application and the other existing applications which are in use today. From the figure, it can be understood that the security risk for our application is lower compared with other major and modern application such as Gmail, Outlook, iCloud, Hotmail and Yahoo. Our application security is slightly better against all these applications due to the authentication mechanism that we used in our proposed method, and also due to the authentication algorithm that we used to encrypt all the information that comes to our application. The security level for some cloud-based applications such as Yahoo, Hotmail, and Outlook is even Medium level, which is not good for their users, some who use them for personal purposes and some for business purposes. So, comparing our application with these apps, the results are showing that our application is performing much better in terms of security with the proposed authentication mechanism. We understand that all these applications use different authentication methods as discussed in Chapter 2, as there are no standards just yet when it comes to using a specific authentication method for security purposes. However, security is very important in today's world and the users must have a world-class application which will keep their data very confidential, highly available and above all with great integrity without changing anything in it.

The other two applications, Gmail and iCloud, seem to have a strong security level in place, and this is as a result of their authentication mechanism, and also the way the data is being transmitted through their network. However, our application's security is slightly better compared with those as well, even though their security level result is low according to the penetration testing. The security level has been measured in the same way for all the applications that we have been testing in our research work.

As mentioned earlier, we launched fake attacks such as insufficient authentication attack, brute force attack, and login page password attack. As the results showed, our proposed authentication method resulted in a better security level.

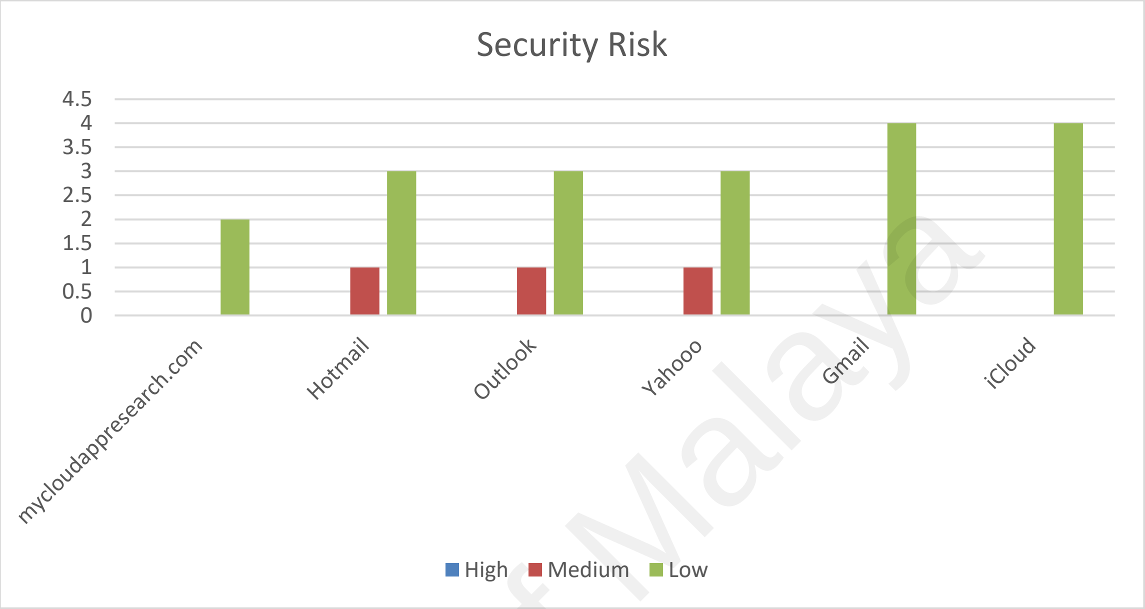
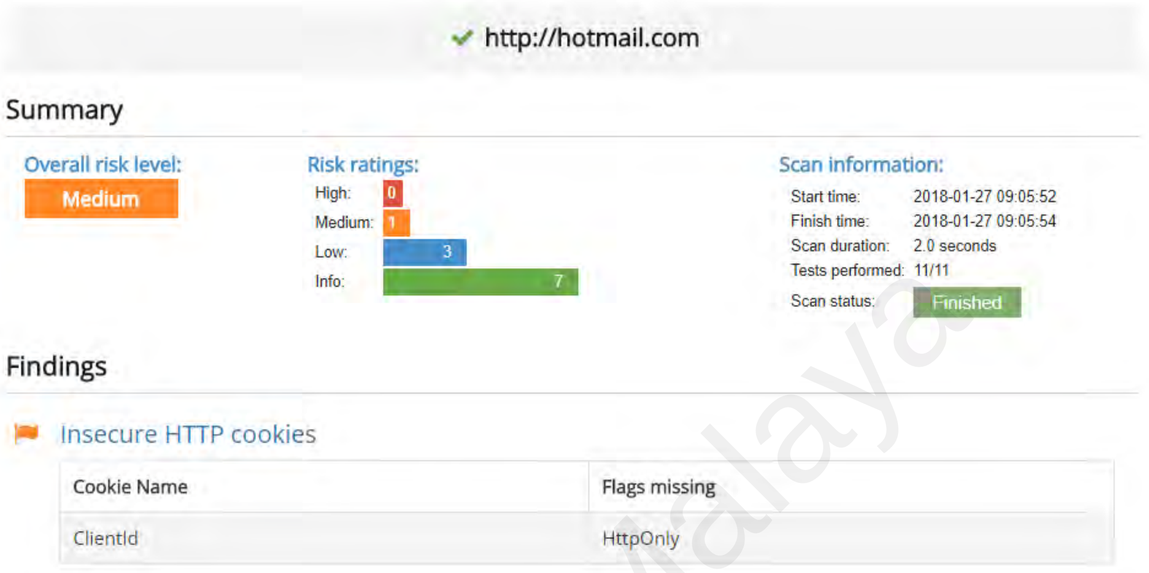


Figure 5.4: Comparison with the other existing application using Acunetix tool

5.3.3.2 Comparison with Hotmail application

After implementing our authentication method, we launched fake attacks to our application and also against the Hotmail application to compare the security levels. Our application contains two authentication phases, which we call primary authentication and secondary authentication. The primary authentication has been designed in such a way that is initiated to the application server. The application sends an authentication request to the authentication proxy. The primary authentication uses an Active Directory or an LDAP server to authenticate the user accounts. If the primary authentication is accepted, the user is prompted for the secondary authentication model. In the secondary authentication phase, the user is prompted to answer three randomly selected security questions. It only authenticates the user if all three answers are correct, otherwise, the authentication will fail. The Hotmail application, on the other hand, is authenticating the user accounts based on their login credentials, and additionally, the users have the option to set a two-step verification using their mobile phones, as discussed in Chapter 2 above. As can be seen from the picture below, our application has only two security issues and they are low. However, Hotmail application has one medium issue, which, based on the findings is about insecure HTTP cookies. Moreover, as we can see from the results, the Hotmail application also has another three low-security risks, and their overall risk level is Medium.

Here are the results that we gained for the Hotmail application after the penetration testing was complete, and the results are shown in the summary report presented below:



✓ http://hotmail.com

Summary

Overall risk level: Medium

Risk ratings:

High:	0
Medium:	1
Low:	3
Info:	7

Scan information:

Start time:	2018-01-27 09:05:52
Finish time:	2018-01-27 09:05:54
Scan duration:	2.0 seconds
Tests performed:	11/11
Scan status:	Finished

Findings

Insecure HTTP cookies

Cookie Name	Flags missing
ClientId	HttpOnly

Figure 5.5: Overall security risk level for Hotmail application (Medium)

The graph below shows the comparison of our application's security risk and the Hotmail application's.

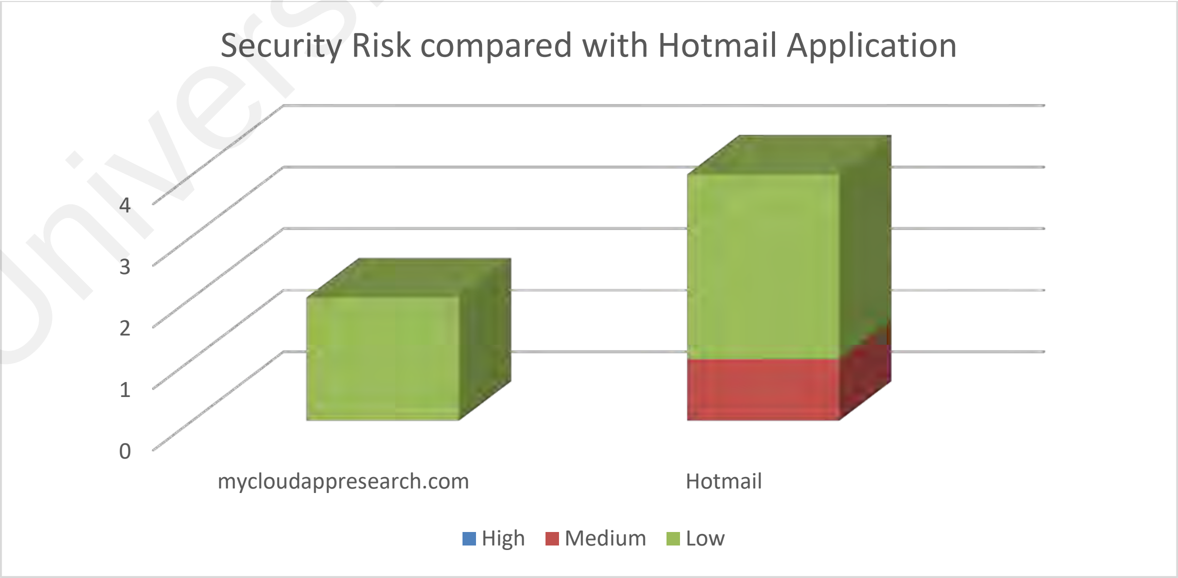
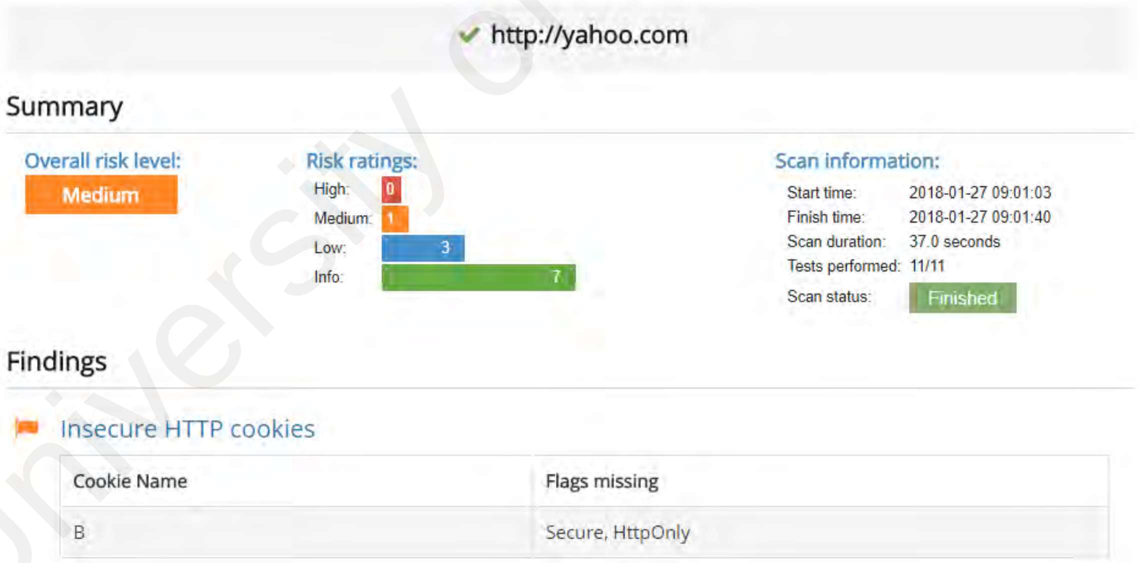


Figure 5.6: Security risk compared with Hotmail application

5.3.3.3 Comparison with Yahoo application

The Yahoo application is a well-known app and it has been around for many years. Just like the other experiments, we also launched fake attacks to their online application to test their security level, and after that, compare it with our application. Based on the findings of the test, we found that it also contains insecure HTTP cookies. That means that some information is not being encrypted through an HTTPS request. In an XSS breach case, an attacker could inject some JavaScript and potentially access the cookies, which often contain sensitive information.

After the penetration testing was complete, here are the results that we gain, and it shows the overall security risk level of Yahoo application, which is Medium.



✓ http://yahoo.com

Summary

Overall risk level: Medium

Risk ratings:

High:	0
Medium:	1
Low:	3
Info:	7

Scan information:

Start time:	2018-01-27 09:01:03
Finish time:	2018-01-27 09:01:40
Scan duration:	37.0 seconds
Tests performed:	11/11
Scan status:	Finished

Findings

Insecure HTTP cookies

Cookie Name	Flags missing
B	Secure, HttpOnly

Figure 5.7: Overall security risk level for Yahoo application (Medium)

The figure below shows the security risk of our application compared with the Yahoo application.

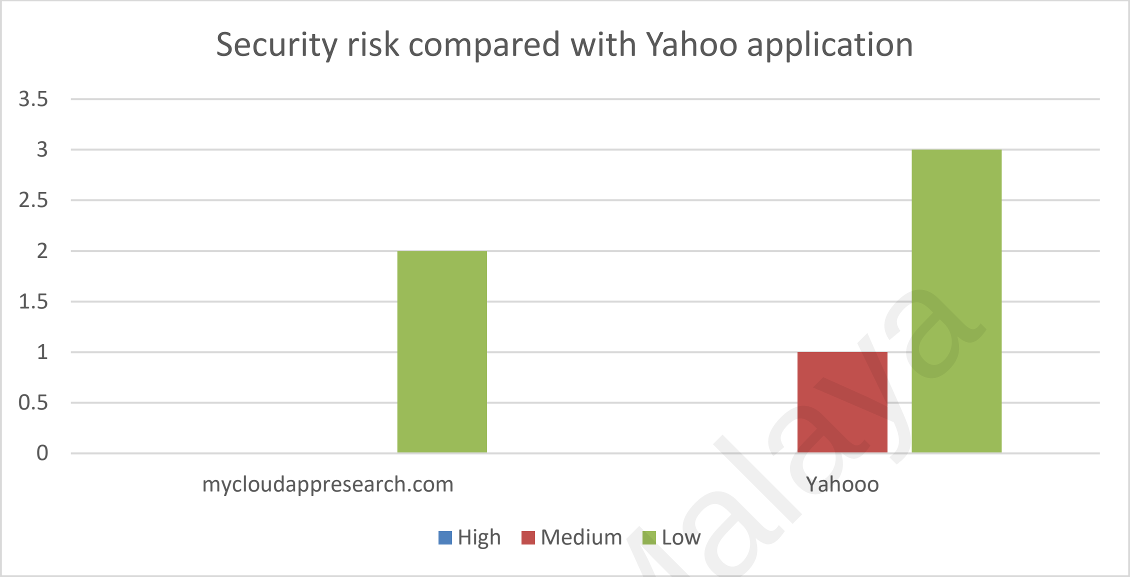
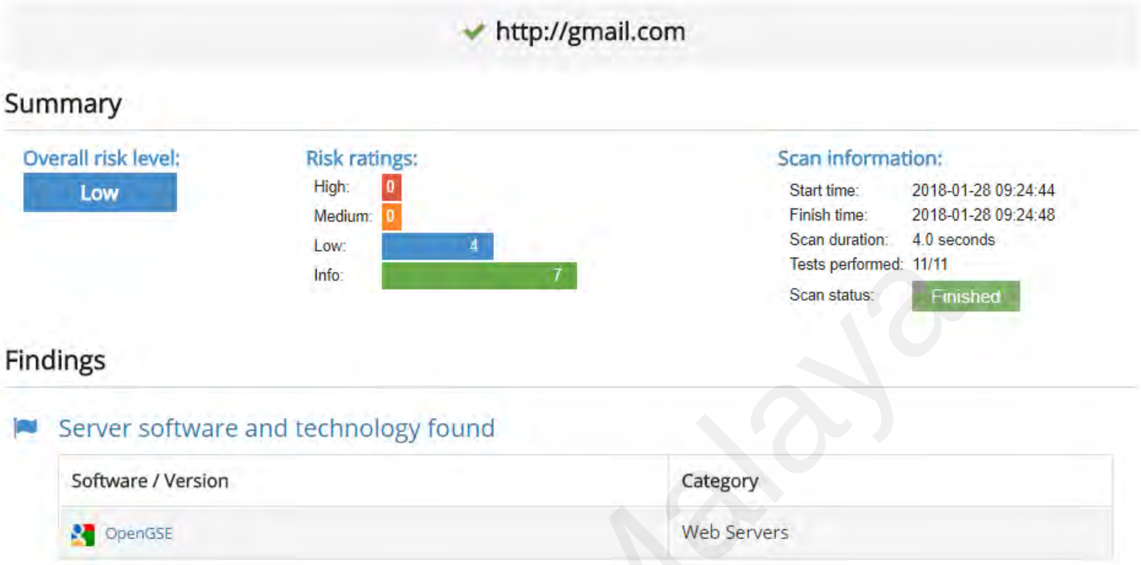


Figure 5.8: Security risk compared with Yahoo application

5.3.3.4 Comparison with Gmail application

Google has its own two-step verification system. However, there are some drawbacks when using this system. Google will send a verification code to the phone number registered with the customers' account every time they sign in to a new device. But the issue with this is, there would be some situations where the phone is unavailable, and at that time the customers will be locked out of their account. Additionally, using a phone number to receive a code may expose it to be intercepted and hacked, and the attacker may be able to receive that code and compromise the account. We applied the same authentication method to our application, which is easy to use and does not involve any other third-party devices. From the results we can see that Gmail's overall risk level is low, however, again they are having four security risks, compared to our application which only has two low-security risks.

Again, as we can see from the results below, our application is slightly better compared with the Gmail application as well.



✓ http://gmail.com

Summary

Overall risk level: Low

Risk ratings:
High: 0
Medium: 0
Low: 4
Info: 7

Scan information:
Start time: 2018-01-28 09:24:44
Finish time: 2018-01-28 09:24:48
Scan duration: 4.0 seconds
Tests performed: 11/11
Scan status: Finished

Findings

Server software and technology found

Software / Version	Category
OpenGSE	Web Servers

Figure 5.9: overall security risk level for Gmail application (Low)

The figure below shows the security risk of our application compared with the Gmail application, which has been developed by Google.

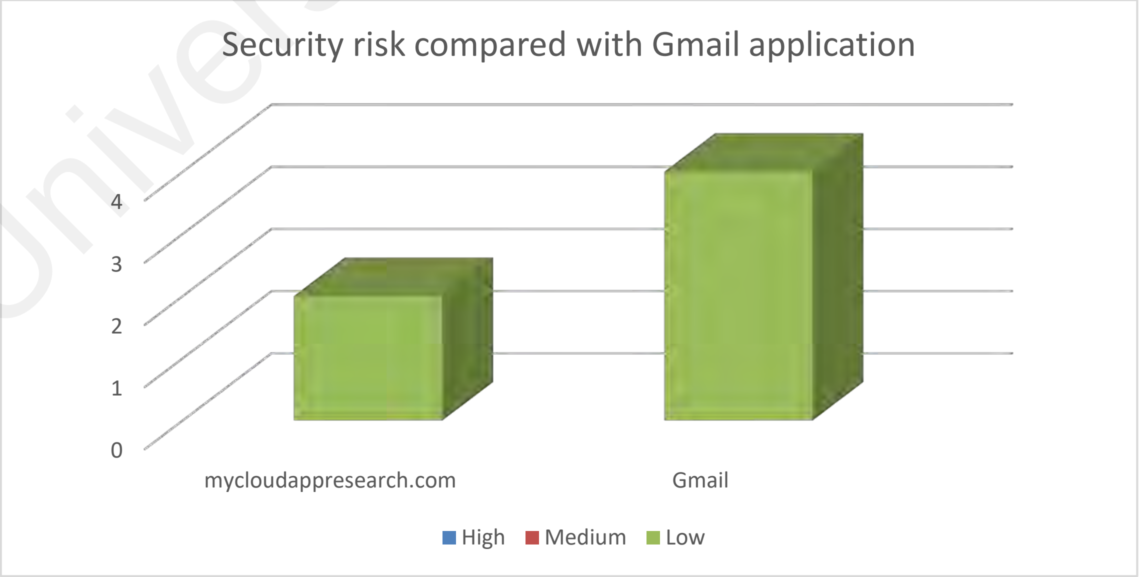
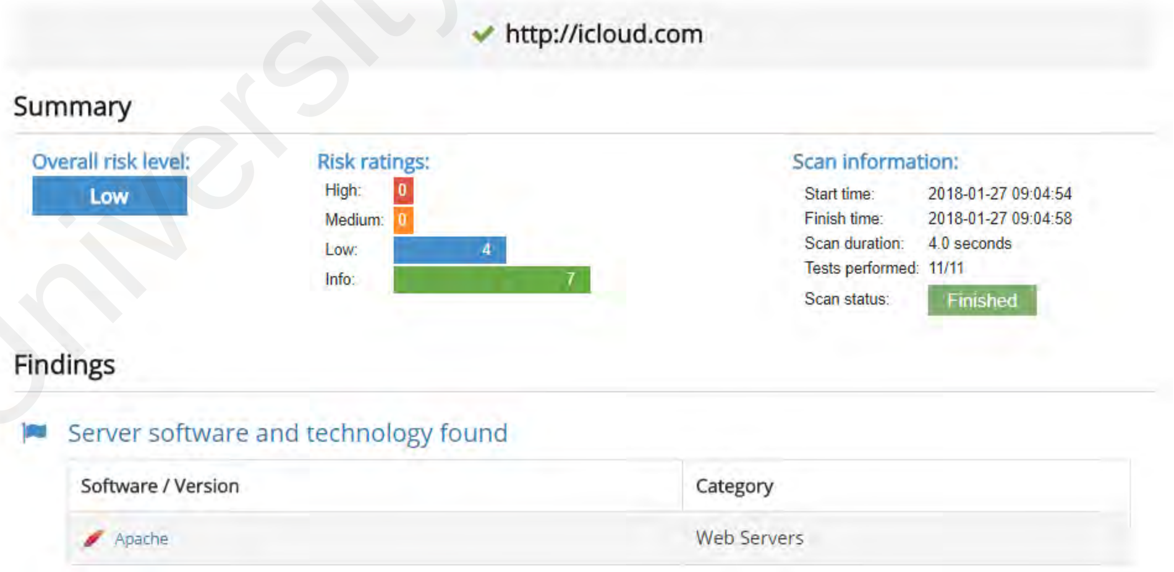


Figure 5.10: Security risk compared with Gmail application

5.3.3.5 Comparison with iCloud Application

Apple is also using two-factor authentications for Apple ID. When customers want to sign in for the first time to a new device, they are required to provide two pieces of information, the login credentials and also a six-digit verification code which will be sent to their trusted devices. Apple claims that, because the password alone is no longer enough to access the account, two-factor authentication dramatically improves the security of Apple ID and all the personal information stored with Apple. On the other hand, the authentication method which we proposed also asks for two pieces of information from the user, the first one is the login credentials and the second one is the random security questions. Furthermore, the results from the penetration testing have shown that our proposed authentication method has improved security quite significantly. The picture below shows the overall security risk of iCloud application with the results of the penetration testing.



✓ http://icloud.com

Summary

Overall risk level: Low

Risk ratings:

High:	0
Medium:	0
Low:	4
Info:	7

Scan information:

Start time:	2018-01-27 09:04:54
Finish time:	2018-01-27 09:04:58
Scan duration:	4.0 seconds
Tests performed:	11/11
Scan status:	Finished

Findings

Server software and technology found

Software / Version	Category
Apache	Web Servers

Figure 5.11: Overall security risk level for iCloud application (Low)

The figure below shows the security risk of our application compared with the iCloud application, which has been developed by Apple.

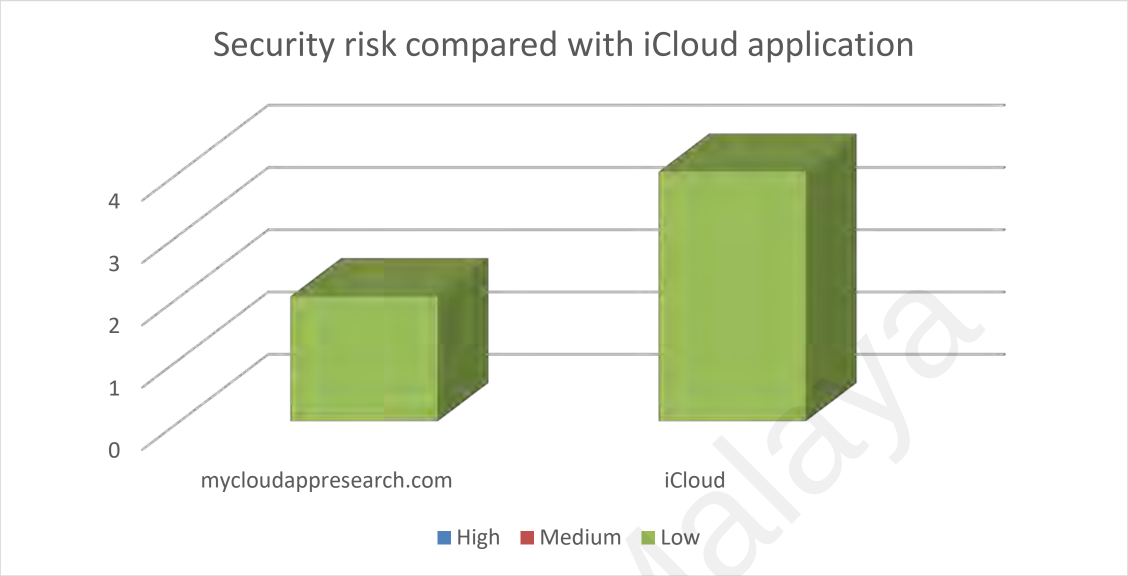


Figure 5.12: Security risk compared with iCloud application

Finally, here is another final overview of our comparison with all the other existing cloud web apps. The graph below depicts the overall security level of all the apps. Based on the results that we found, the picture shows the vulnerability risk according to the individual compliance categories. We can clearly see that our application has the lowest overall security risk and this is as a result of our proposed and implemented authentication method. Our method is very simple to implement, and it also strengthens the overall security of the applications. Security is very crucial in the cloud environment, as it creates that trust between the vendor providers and the customers. When using the Software as a Service (SaaS) model, most cloud vendors offer an easy way to access their service, with additional security steps which need to be implemented by the consumers when required, such as the two-factor authentication or the security key. Our approach aims to improve the security level on the application itself, by adding another security layer to the authentication mechanism, so that it strengthens the security level. And the results have shown that our approach has improved that security level significantly.

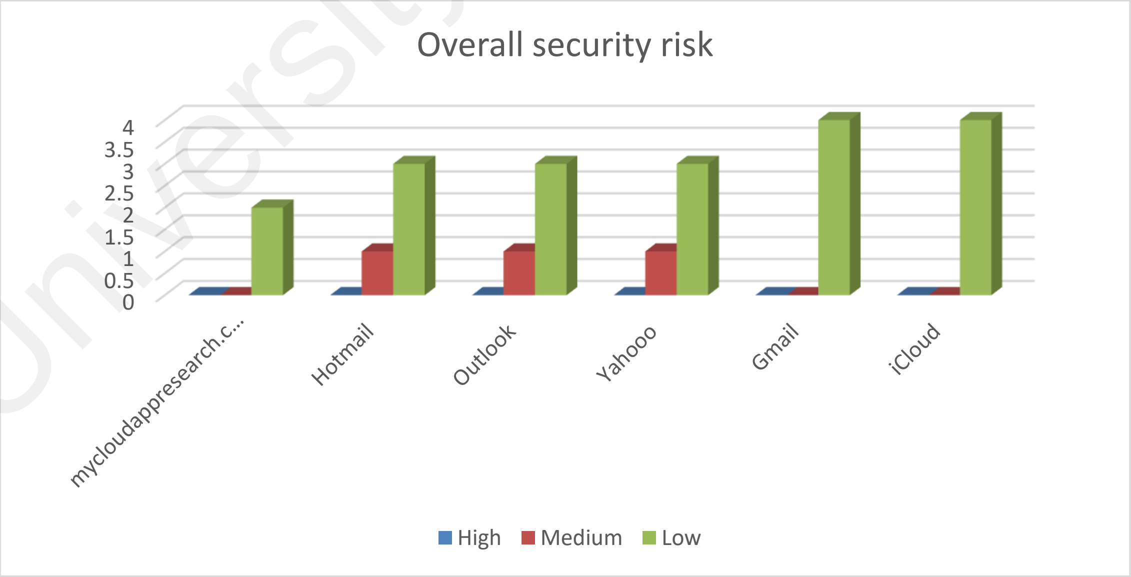


Figure 5.13: Overall security risk compared with all the existing apps.

5.4 Summary

Our authentication mechanism is designed for web-based applications and the benchmarking is done by evaluating the authentication method on a web server. Experiments and testings are done using the Acunetix tool (Pilli, 2016). Based on the results that we gathered, we can strongly conclude that the evaluation of the proposed authentication method indicates that the authentication can be improved without using any other device. The main aim of our research work is to enhance the authentication method for web applications running in the cloud environment, and to make it highly secure.

Encryption adds another layer of security and protects the system from an outside attacker. Encryption also helps the integrity of the user's data, and it only allows access to authorized personnel. Encryption is the process of protecting the data by changing or transforming it in a ciphertext format, so that an attacker may not be able to read the data and decrypt it into the original format using a security key. The strength of the encryption depends on the key size of that algorithm, and also on the type of the algorithm used to encrypt the data. We designed our algorithm in a way that it encrypts the data in longer bits so that the data will be stored securely.

The authentication mechanism has been designed in such a way that is initiated by the application server. The application server submits an authentication request to the authentication proxy. Then, the authentication proxy uses an Active Directory or an LDAP server to authenticate the user accounts if the login credentials are correct. If this authentication is accepted, then the user is prompted for the secondary authentication model. In the secondary authentication phase, the clients are required to answer three random security questions, and it only authenticates the client only if all three answers were entered correctly, otherwise, the authentication will fail.

Furthermore, the overall security level result of our application is low, compared to Hotmail, Yahoo, and Outlook applications. Our application is also slightly better compared with the applications of the two tech giant companies, namely Google and Apple. So, comparing our application with all these apps, the results are showing that our application is performing much better in terms of security having implemented our proposed authentication mechanism in place.

CHAPTER 6: CONCLUSON

6.1 Introduction

This chapter concludes our research work by summarizing what we have done during this research and emphasizing the impact and contribution of our research in section 6.2. It then presents and identifies the areas of further research, in section 6.3.

6.2 Achievements

This research addresses the problem of security issues such as broken authentication for cloud web-based application. It discusses the current methods for authenticating user accounts in the cloud environments, and after that it proposes a new authentication method. The objectives of this research are stated in chapter 1. Below is a brief summary of the achievement of the objectives.

A detailed review of existing authentication method for web application was produced in the literature review. After that we also created a summary table of all the existing authentication methods which are available now, and we gave our insights about the disadvantages of these methods. We performed the review about the security issues and challenges of cloud web applications and selected the most relevant and those with the strongest authentication mechanism. These authentication methods were reviewed based on their challenges and the issues. Finally, the research gap was stated in chapter 2.

A new authentication method was proposed as a solution for strengthening the security issues in cloud web application. Different components of the authentication mechanism were discussed in chapter 3. The main aim of our research work is to enhance the authentication method for web apps and to make it highly secured. To do that, we used an encryption algorithm for encrypting the passwords, and enhanced the security key for storing the data much more securely. Encryption is the process of protecting the data by changing or transforming it in a cyphertext format, so that any attacker may not be able to read the data and decrypt it into the original format using a security key. Encryption helps on adding another layer of security and protects the system from being compromised. Encryption also helps the integrity of the user's data, and it acts as a checkpoint where it only allows access to authorized users. The strength of the encryption depends on the type of algorithm used to encrypt the data, and also the key size of that algorithm. We designed our algorithm is a such a way that it encrypts the data very securely, and the key size used in our algorithm is 1000 bits, which create secure password hash and generate a unique hash string.

To implement the new authentication method and to evaluate its performance, a cloud web-based application was developed using Godaddy's server and hosting it in real time. The application was integrated with the new authenticated method which we proposed. Then, we conducted a series of experiments to evaluate our proposed authentication method. Experiments and testings were done using Acunetix tool (Pilli, 2016). Based on the results that we gathered, we can strongly conclude that the evaluation of the proposed authentication is much stronger and more secured compared with other existing system. Our authentication method was compared with the other existing method from well-known application such as Gmail, Yahoo, Hotmail and iCloud. Results, were compared with each and every one of these applications to show the feasibility of our idea in chapter 4. The results showed that, the overall security of our application which we developed and integrated with our authentication mechanism, is much stronger compared with the rest of the applications.

The contributions of this thesis can be summarized as:

The contribution of our work is that, we managed to create a better way to authenticate user accounts using cloud web apps with the new enhanced encryption algorithm. We embedded the new encryption algorithm to our application that we developed, and after the experiments and testing that we conducted, the data that we gathered have shown that our application has stronger security level in place. This is as a result of the newly enhanced encryption algorithm that we developed, as it encrypts the data and save them very securely. Our work has been proven to have a better authentication mechanism as we used to run vulnerability testing with Acunetix tool, against the existing apps such as the Gmail, Hotmail, and Yahoo app, and the results showed once again that our developed application was much stronger, and the security vulnerabilities were much lower compared with the rest of the existing apps. As the new technology is growing very rapidly, our contribution will help the other developers to create the next generation of the cloud web apps, to implement the authentication mechanism just like our approach. Security is main concern for cloud technologies, and our work has addressed and fixed some of the security issues in cloud environments. Our work will also help the other vendors or researchers, who would like to address the security issues in their apps and implement a new authentication mechanism just like our approach. To conclude our research work, we would like to mention these points below which were achieved as the result of our work:

1. An enhanced encryption algorithm has been developed for cloud web apps.

2. A cloud web-based application with new enhanced algorithm embedded has been developed to strengthen the authentication mechanism.

3. Our authentication method has been tested to signify the logic of our proposed authentication for cloud application, and the results indicated that our approach is much better.

6.3 Future Work

"The best way to predict the future is to invent it." (Alan Kay, Computer Scientist).

Cloud computing will continuously grow, and it is changing the way we utilize IT resources. It leads to a more flexible way of accessing and using all the computing resources in a truly on-demand fashion. However, the way we take advantage of this technology also determines the security level that we set to protect our information. Security has been an issue for IT applications that we use on-premises, and it will continue to be an issue in the cloud environments.

In our proposed work, we introduced a new method to strengthen some of the security issues. Our proposed work was initiated for cloud web-based applications, and one possible thing which can be done is to create a mobile application and implement the same method. We all know that the mobile sector is the medium where a majority of the people interacts with when they are on the move, or away from their workplace. In addition, we used the Acunetix tool to scan for vulnerabilities and security issues in our application, and the results showed a better performance than the other existing applications. The report of our data can also be used as a case study for future work in the field of security for web applications.

In conclusion, security will become more important and will be a decision criterion for the majority of the people who would like to use cloud computing technologies, especially in corporate environments. The ability to have a clear understanding of cloud applications and the security state of the surrounding infrastructure will be fundamental for the end users. The future of IT is in the cloud, and that is where the innovation and most advanced technologies such as AI (artificial intelligence) is taking place.

REFERENCES

Aiya, K. V, & Verma, H. (2015). Keyword driven automated testing framework for web application. *9th International Conference on Industrial and Information Systems, ICIIS 2014*.

Alliance, C. secyrity, Simmonds, P., Rezek, C., Reed, A., & Alliance, C. secyrity. (2016). Security guidance for critical areas of focus in cloud computing v3. 0. *Cloud Security Alliance*, 176.

Amazon Service Health Dashboard. (n.d.). AWS Service Health Dashboard - Oct 1, 2017 PDT. Retrieved October 1, 2017, from http://status.aws.amazon.com/

Bsi. (2011). Security Recommendations for Cloud Computing Providers. *Federal Office for Information Security*, 24; 28; 52.

Burr, W. E., Dodson, D. F., Newton, E. M., Perlner, R. A., Polk, W. T., Gupta, S., & Nabbus, E. A. (2013). Electronic Authentication Guideline, *2*.

Chen, E. Y., Pei, Y., Chen, S., Tian, Y., Kotcher, R., & Tague, P. (2014). OAuth Demystified for Mobile Application Developers. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, (1), 892–903.

Cloud Security Alliance. (2010). Top Threats to Cloud Computing. *Security*, (March), 1–14.

Cloud Security Alliance. (2013). The Notorious Nine. Cloud Computing Top Threats in 2013. *Security*, 1–14.

Cloud Security Alliance. (2016). The Treacherous 12 Cloud Computing Top Threats in 2016. *Security*, 1–34.

Counsel, A. of corporate. (2015). One-Third of In-house Counsel Have Experienced a Corporate Data Breach, ACC Foundation: The State of Cybersecurity Report Finds - Association of Corporate Counsel (ACC).

Czeskis, A., Dietz, M., Kohno, T., Wallach, D., & Balfanz, D. (2012). Strengthening user authentication through opportunistic cryptographic identity assertions. *Proceedings of the 2012 ACM Conference on Computer and Communications Security - CCS '12*, 404.

Druva. (2015). The State of Data Privacy in 2015 - Druva. Retrieved September 9, 2017, from https://www.druva.com/resources/analyst-reports/the-state-of-data-privacy-dimensional-research-report/

Fujii, H., & Tsuruoka, Y. (2013). SV-2FA: Two-factor user authentication with SMS and voiceprint challenge response. *2013 8th International Conference for Internet Technology and Secured Transactions, ICITST 2013*, 283–287.

Google. (2016, June 13). Encryption at Rest in Google Cloud Platform, 17. Retrieved from https://cloud.google.com/security/encryption-at-rest/

Google. (2016, June). Google for Work Security and Compliance Whitepaper, 27.

Hakobyan, D. (2012). Authentication and Authorization Systems in Cloud Environments.

Huang, X., Xiang, Y., Chonka, A., Zhou, J., & Deng, R. H. (2011). A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, *22*(8), 1390–1397.

ISO/IEC 27001:2005. (n.d.). ISO/IEC 27001:2005 - Information technology -- Security techniques -- Information security management systems -- Requirements. Retrieved October 1, 2017, from https://www.iso.org/standard/42103.html

ISO 9001:2008. (n.d.). ISO 9001:2008 - Quality management systems -- Requirements. Retrieved October 1, 2017, from https://www.iso.org/standard/46486.html

Jiang, Q., Wei, F., Fu, S., Ma, J., Li, G., & Alelaiwi, A. (2016). Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. *Nonlinear Dynamics*, *83*(4), 2085–2101.

Khan, A. N., Mat Kiah, M. L., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems*, *29*(5), 1278–1299.

Kitchenham, B., & Charters, S. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering. *Engineering*, *2*, 1051.

Lemke, C., Brenner, W., & Kirchner, K. (2017). *Einführung in die Wirtschaftsinformatik*.

Mather, T. (2009). *Praise for Cloud Security and Privacy*.

Matt Rosoff. (n.d.). Amazon: Outage Enters Second Day -- What Went Wrong? - Business Insider. Retrieved October 1, 2017, from http://www.businessinsider.com/amazon-outage-enters-its-second-day-lots-of-sites-still-down-2011-4/?IR=T

McCarney, D., Barrera, D., Clark, J., Chiasson, S., & van Oorschot, P. C. (2012). Tapas: Design, Implementation, and Usability Evaluation of a Password Manager. *Proceedings of the 28th Annual Computer Security Applications Conference on - ACSAC '12*, 89–98.

Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. *Nist Special Publication*, *145*, 7.

Mudholkar, S. S., Shende, P. M., & Sarode, M. V. (2012). Biometrics Authentication Technique for Intrusion Detection Systems Using Fingerprint Recognition. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, *2*(1), 57–65.

Naveen Thakur. (n.d.). New Hotmail Exploit Can Get any Hotmail Email Account Hacked for just 20$.

Pilli, E. S. (2016). *Fundamentals of Network Forensics*.

Ponemon Institute. (2018). The 2018 Global Cloud Data Security Study, (January).

Ren, K., & Wang, C. (2012). Security Challenges for the Public Cloud, 69–73.

Ritter, D., Schaub, F., Walch, M., & Weber, M. (2013). MIBA: Multitouch Image-Based Authentication on Smartphones. *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, 787–792.

Sediyono, E., Santoso, K. I., & Suhartono. (2013). Secure login by using One-time Password authentication based on MD5 Hash encrypted SMS. *Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013*, 1604–1608.

Sriramya, P., & Karthika, R. A. (2015). Providing password security by salted password hashing using Bcrypt algorithm. *ARPN Journal of Engineering and Applied Sciences*, *10*(13), 5551–5556.

Stuttard, D., & Pinto, M. (2011). *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws* (Vol. 7).

Vaquero, L. M., Rodero-Merino, L., & Morán, D. (2011). Locking the sky: A survey on IaaS cloud security. *Computing (Vienna/New York)*, *91*(1), 93–118.

Varia, J., & Mathew, S. (January). Overview of Amazon Web Services. *Amazon Web Services*, (January), 22.

Wasc. (2017). Web Application Security Consortium, (February).

Xiao, Z., & Chen, J. (2015, January 24). Cloud Computing Security Issues and Countermeasures. *Proceedings of the 4th International Conference on …*, *4*(5), 82–93.

Zhu, B. (January). Analysis and Design of Authentication and Encryption Algorithms for Secure Cloud Systems.

Zhu, B., Fan, X., & Gong, G. (2014). Loxin - A solution to password-less universal login. *Proceedings - IEEE INFOCOM*, 488–493.