

**BEST PRACTICES IN ONLINE USER AUTHENTICATION: AN ANALYSIS  
AND SURVEY**

by

Fatema Bannat Wala

A thesis submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

Summer 2015

© 2015 Fatema Bannat Wala  
All Rights Reserved

ProQuest Number: 1602374

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



ProQuest 1602374

Published by ProQuest LLC (2015). Copyright of the Dissertation is held by the Author.

All rights reserved.

This work is protected against unauthorized copying under Title 17, United States Code  
Microform Edition © ProQuest LLC.

ProQuest LLC.  
789 East Eisenhower Parkway  
P.O. Box 1346  
Ann Arbor, MI 48106 - 1346

**BEST PRACTICES IN ONLINE USER AUTHENTICATION: AN ANALYSIS  
AND SURVEY**

by

Fatema Bannat Wala

Approved: \_\_\_\_\_  
Dr. Chase Cotton, Ph.D.  
Professor in charge of thesis on behalf of the Advisory Committee

Approved: \_\_\_\_\_  
Kenneth E. Barner, Ph.D.  
Chair of the Department of Electrical and Computer Engineering

Approved: \_\_\_\_\_  
Babatunde A. Ogunnaike, Ph.D.  
Dean of the College of Engineering

Approved: \_\_\_\_\_  
James G. Richards, Ph.D.  
Vice Provost for Graduate and Professional Education

## ACKNOWLEDGMENTS

To Chase Cotton: Dr. Cotton has given me the great opportunity to work in the cyber-security domain under his supervision, which is my field of interest. He has provided the right direction to my research and guided me throughout my master's. I feel lucky and, I am grateful to have him as my Advisor, He helped me to choose the courses relevant to my field of study and always provided answers to my queries. When I started, I had very little knowledge in cyber-security, but Dr. Cotton helped me to grow my knowledge base and because of his incredible experience in cyber-security domain, I have learned a lot from him down the road. I would not have been able to achieve what I have achieved without him. He is an amazing mentor and a great person.

To Fouad Kiamilev: Dr. Kiamilev always provided moral support and the boost required to achieve what I wanted to. The meetings with him were always focused on how to overcome weaknesses and improve the overall performance in the work environment. The weekly feedback always enhanced my quality of work, and the motivation provided always kept me going.

To Willett Kempton: I would like to thank him for giving me the opportunity to work in one of the brightest research groups in the department and to be a part of a group of great individuals, who go out of their way to help anyone out. He helped me to sharpen my programming skills in parallel with my course work. It was an unforgettable experience working with the V2G group.

This work was supported in part by NSF CNF FIA funding #1040614.

## TABLE OF CONTENTS

LIST OF TABLES .....	vii
LIST OF FIGURES.....	viii
ABSTRACT.....	ix

### Chapter

1	OVERVIEW.....	1
2	COMMON ATTACKS.....	7
2.1	Motivation .....	7
2.2	Common Attacks .....	9
2.2.1	Brute Force.....	10
2.2.2	Denial of Service (DOS).....	12
2.2.3	Phishing Attack .....	13
3	SECURITY BEST PRACTICES SURVEY .....	15
3.1	Category: Password Selection .....	15
3.1.1	Applicable To.....	15
3.1.1.1	Best Practice: Minimum Password Requirements.....	16
3.1.1.1.1	Minimum Length Restriction .....	16
3.1.1.1.2	Special Character Inclusion.....	17
3.1.1.1.3	Number Inclusion .....	19
3.1.1.1.4	Upper & Lowercase Letter Inclusion.....	21
3.1.1.2	Best Practice: Disallow Common String.....	24
3.1.1.3	Best Practice: Account Activation Check .....	27
3.1.1.4	Best Practice: 2FA For Login.....	30
3.1.1.5	Best Practice: CAPTCHA While AccountCreation.....	34
3.1.1.6	Best Practice: Show Password Strengthbar .....	36
3.2	Category: Bad logins Protection .....	38
3.2.1	Applicable To.....	38
3.2.1.1	Best Practice: Restrict Maximum Bad logins.....	39

3.2.1.2	Best Practice: Slowdown With CAPTCHA .....	41
3.2.1.3	Best Practice: Slowdown with timeout .....	44
3.2.1.4	Best Practice: Bad Login ErrorMessage .....	47
3.2.1.5	Best Practice: No Timing Difference Loading Pages ..	51
3.2.1.6	Best Practice: Notification Of Bad Logins .....	52
3.3	Category: Password Recovery .....	55
3.3.1	Applicable To .....	55
3.3.1.1	Best Practice: 2FA For Password Reset .....	55
3.3.1.2	Best Practice: Use Of Recovery Email Address .....	58
3.3.1.3	Best Practice: No Password Hint In Recovery Email ..	61
3.3.1.4	Best Practice: No Account Info In Recovery Email ....	64
3.3.1.5	Best Practice: No Personal Security Questions For Recovery .....	67
3.3.1.6	Best Practice: No Random emailId/Phone For Recovery .....	70
3.3.1.7	Best Practice: Timeout The Recovery URL .....	73
3.3.1.8	Best Practice: No Part Of Email/Phone Shown While Recovery .....	76
3.3.1.9	Best Practice: Password Reset Notification .....	79
3.3.1.10	Best Practice: Recovery EmailID/Phone Change Notification .....	80
3.4	Category: Service Authenticity .....	83
3.4.1	Applicable To .....	83
3.4.1.1	Best Practice: User Personal Security Image and Caption .....	83
3.5	Category: Suggested New Best Practices .....	85
3.5.1	Applicable To .....	85
3.5.1.1	Best Practice: Uncommon Username For Login .....	85
3.5.1.2	Best Practice: Mandatory Upgrade Of Password After New Password Policy .....	87
3.5.1.3	Best Practice: Mandatory Password Hash Salting .....	90
3.5.1.4	Best Practice: Periodic Password Reset .....	92
4	ONLINE SERVICE PROVIDERS – A SURVEY .....	94

4.1	The Comparison Table as of 1/20/15 .....	95
4.2	Discussion on findings .....	107
4.2.1	Online Cloud service providers .....	107
4.2.2	Social Media service providers .....	109
4.2.3	E-commerce service providers .....	111
4.3	A Discussion on Timeout Versus CAPTCHA (study from the survey) .....	113
4.4	A Discussion on 2-Factor Authentication (Dropbox case study) .....	115
5	CONCLUSION .....	117
5.1	Summary of Best Practices.....	119
6	FUTURE WORK .....	123
6.1	Tool for Automated Auditing .....	123
	BIBLIOGRAPHY .....	126
	Appendix	
	CURRENT PUBLISHED BEST PRACTICES .....	129

## LIST OF TABLES

4.1	Comparing online services with best practices: PasswordSelection .....	95
4.2	Comparing online services with best practices: Bad P/w Attempts .....	98
4.3	Comparing online services with best practices: Password Recovery .....	101



## LIST OF FIGURES

3.1	Minimum password requirements .....	23
3.2	Disallow common strings.....	27
3.3	Account activation check .....	30
3.4	CAPTCHA while account creation.....	36
3.5	Password strength bar .....	38
3.6	Slowdown with CAPTCHA .....	44
3.7	Slowdown with time-out .....	47
3.8	Incorrect username error message .....	50
3.9	Incorrect password error message.....	51
3.10	Bad login attempts notification.....	55
3.11	Use of recovery email address for password reset.....	61
3.12	No password reset hint in email.....	63
3.13	Password reset code in clear text in email.....	64
3.14	Account information in recovery email .....	67
3.15	Personal security questions while password recovery .....	70
3.16	Random email address for recovering the password .....	73
3.17	Recovery URL time-out (24 hours).....	76
3.18	Part of recovery email address shown.....	78
3.19	Notification for change in recovery phone number .....	82
4.1	Services using Timeout, CAPTCHA or neither. ....	115

## ABSTRACT

As the growth of technology has tremendously increased in recent decade, especially in the area of internet technology, majority of services used by the people in their day to day activities has now become automatic and therefore pushed to the internet to be more conveniently accessible to the users, anytime anywhere. However every innovation comes with its side effects, therefore talking about innovations in the area of internet technology, they also come with a price. Since now majority of services are offered online, they have become more vulnerable to the cyber-attacks. The number of online services offered has direct impact on internet cyber-attacks, i.e. more and more information/services go online, they become more prone to the Internet cyber-attacks [17]

This research was initially focused on a study of the encryption methods, such as public and symmetric key encryption, used by a new category of online service providers (e.g. Mega, Tresorit, Wuala, and SpiderOak) to offer end customer secure encrypted cloud data storage to end customers. To keep a users' data secure requires maintaining confidentiality of the data is both at rest and in transit. Some of these providers provided client-side encryption where the encryption is done in the end users machine, but most of them used server-side encryption to allow for faster, but less secure encryption.

The direction of this research changed due to two factors. First, as some of the providers security mechanisms were being investigated, it became clear that some of the basic account establishment and authentication schemes were quite weak and thus even the best data confidentiality mechanisms (e.g. encryption) would not ultimately have much effect securing the users' data. Second, a series of high profile public

breaches of other service providers' authentication and password recovery mechanisms were also found to be quite weak and subject to brute force[18] and other types of attacks. Examples include breaches of celebrity Twitter and YouTube accounts[19][20][21]. An expanded review of authentication mechanisms of other online service providers beyond secure cloud storage found a wide range of practices, from very secure to insecure. Hence, the target of this research was re-focused on better understanding the common user authentication processes of online service providers in order to better clarify which mechanisms might be considered best practice.

Many current online user authentication practices and mechanisms used by online service providers were surveyed. Providers included a wide range of services including social networking, email, and e-commerce sites. The research shows that even though it is obvious that some of the best practices should be incorporated in the user authentication process to shield the service from major cyber-attacks like brute force and Denial of Service (DOS), there are many online services which don't implement those, making their users more vulnerable to cyber-attacks. This research highlights precisely the best practices that should be used by online services to best secure their online user authentication. Additionally, several suggested new best practices are suggested which can provide additional security for users' accounts.

## **Chapter 1**

### **OVERVIEW**

One of the major challenges the internet is facing today is cyber-attacks. As discussed in previous section that majority of services are now available online to the users for their convenience, it is increasing the risk of cyber-attacks as well. Providers compete with each other to provide best experience to the users, thus increasingly moving online, but the dark side of this scenario is that as more and more services go online, they become more prone to the Internet cyber-attacks. For example, there was time when, if users want to back-up data or store data, they used various storage devices such as CDs, DVDs, or FLASH drives but the problem was that they can't access their data globally. And if they lost their physical storage device, and since there is no likely security on the storage device, whoever finds it can see the user's data residing on the device. These and other issues gave rise to the very popular online service known as cloud storage service, which is provided by many of large Internet service providers. Now the good news for the users is they can access their data in any part of the world – conditioned that the access Internet is available, but the bad news is - is my data safe in cloud? In addition to cloud storage services, there are services providing social networking online, banking services, online shopping services, e commerce and many more, which uses personal and confidential data that has to be protected when stored and used over Internet.

This research was initially focused on the encryption mechanisms used by various online services which store the private information of users and the data uploaded by those users into online storage containers (cloud storage). The way in which the complete confidentiality and security of the data is ensured involves the basic two steps of the encryption mechanism used as a general convention - protecting data when it travels from/to the user's computer (client) and protecting data when it has completed its way to the storage database and finally is going to reside in the service provider's database (server). The former step is known as encrypting data "in flight" and later is known as encrypting data "at rest". The security of data in flight is ensured by using TLS/SSL/HTTPS (Transport Layer Security/Secure Socket Layer/Hypertext Transfer Protocol Secure) [22][23][24] protocols which encrypts the transport layer/communication channel between the client and the server, hence the data is no longer available in clear text for the people who might be snooping on the wire. Similarly, the data at rest is secured by encrypting it with industry grade standards of encryption, usually a symmetric key encryption such as the Advanced Encryption Standard (AES) [25] or possibly a public key encryption algorithm, which encrypts the data resting on the server database, so that the user's data can neither be seen by the people working for that service provider nor available to the people (hackers) who might try to compromised that server. As the research progressed and these encryption schemes were explored, a more vulnerable aspect of these online service providers came into picture and that is - the initial or the first step of the service, user authentication. A superficial look at some providers practices revealed that the user authentication provided by various online service providers is not foolproof or highly secure and thus would allow unauthorized data access which

results in a data breach affecting the integrity and confidentiality of the users of that service, even if the service has the best mechanisms of encryption to protect the data. Hence, the target of this research was re-focused on better understanding the common user authentication processes of online service providers in order to better clarify which mechanisms might be considered best practice.

User Authentication [12] is the process of verifying the authenticity of the person who is trying to access an online service to know whether he/she is the legitimate user or not. This is the most crucial and important step to maintain the confidentiality, security and integrity of the account as well as the information it holds belonging to the legitimate account owner. It involves a series of online steps, designed and implemented by the service provider, which the user has to go through to make sure that the service accounts do not get compromised. The main step for the user authentication involves getting information from the owners that only they know such as usernames and passwords. Most of the online service providers rely only on this step to prove the identity of the user, which can be exploited by the malicious users (as we will see how in the following section) to hack into the account if proper best practices are not adopted for this step. For more stringent security, it is usually considered best practice to use Two Factor Authentication (2FA) [26] which involves the step of a further verification of the user identity by asking for the proof that only the legitimate user can own such as verification using devices such as cell phones or personal email addresses etc. This makes more difficult for the attacker to break into the accounts even if they have the user's credentials. The aim of this thesis is to focus on the externally visible aspects of the user authentication process, such as minimum number of password characters, common strings check on password etc., and not the

hidden aspects, such as whether passwords are salted, hashed etc., used by various online service providers, explore different kind of authentication techniques currently in use, and based on this survey, that suggesting some best practices that should be used to protect the integrity, confidentiality and security of the accounts (i.e. security concerns involved in online user authentication)

Vulnerability in a service is not just confined to user authentication using one or two factor authentication, but also how the overall service is protected from the third party intervention. More detailed research showed that there are three major areas where the fool proof security practices are needed in authentication. The first one is described in the introduction, is the process that the provider uses to help the user select a safe password. Two additional areas are introduced and detailed where the user's authenticity and the account can be further exploited, (1) practices to stop unlimited bad password attempts at guessing a user's password, also known as brute forcing, and (2) the method deployed for "password recovery", the process a user takes to reset their password if it is forgotten. These three activities are the doors, or what are called "attack vectors", used to break into user accounts, and hence securing all these three steps plays an important role in securing the confidentiality and integrity of the users of the online services. Different techniques are employed by different service providers to make their services secure, but unfortunately they still, in this day and do not always implement some of the best practices which can make these services much more secure. This thesis is based on survey research done on well-known online service providers such as cloud storage services, social networking online services and e-commerce services. Some of the vulnerabilities they have are

discussed which arise from not implementing best security practices seen across the industry.

In early 1995, there was a revolution in Internet security with the invention of SSL (Secure Socket Layer) protocol [23], which works between the Application layer and Transport layer of the Internet Protocol suite. It was invented by Netscape in early 1995 to provide confidentiality while data was in motion. After SSL was invented, couple of improved versions of SSL was released and then TLS (Transport Layer Security) protocol came into existence. Standardized by the IETF (Internet Engineering Task Force), it provided tougher security, authentication, privacy and integrity for client-server applications. SSL/TLS are the cipher-suites of protocols needed for the encryption of the data on wire. Encryption of data while it is in transit is equally important as the encryption of data at rest. This eliminates the risk of loss of integrity and privacy of data by the hackers snooping and/or modifying data the on wire. Especially for login purposes, where the usernames and passwords are sent over the wire to the server for the user authentication, SSL/TLS plays a vital role in preventing the attacker getting credentials from wire. These protocols have been in existence from 1995, but even today date there are many sites which don't implement these application security protocols, keeping the account holders security at risk. It seemed pretty obvious in this day and age that nobody uses a channel, not encrypted with SSL/TLS, for getting their users credentials and for other private data but consider the case of match.com, an America's online dating service that had implemented SSL/TLS but as some point allowed the site to drop the encryption requirements for several months [13]. Now, one could argue this is more an issue of systems configuration control [27], but even when implementing best practices it is



critical to continue to audit your implementation to ensure intended best practices continue to be used. With this example, it is worth mentioning that use of TLS/SSL for the security of data in transit should be incorporated as one of the best practices in the online user authentication mechanism and we strongly recommend to use TLS for encrypting the data in transit because no matter how perfect is the password selection or password recovery mechanism are of an online service, but if there's no encryption in transit (TLS), all the other best practices are of little use.

## **Chapter 2**

### **COMMON ATTACKS**

#### **2.1 Motivation**

Today, cyber security has become a prime focus of almost every nation and the reason is quite obvious. As the technology world is growing day by day, more data and services are pushed on to the Internet to reduce effort and to make the services provided more “user friendly”, and hence it should not be wrong to say that today the Internet has become “Internet of everything” [28]. But as these Internet services are growing, it is introducing risks into all kinds of data and the things which are available on the Internet. Since if the data is available to you that mean it is potentially available to everyone who uses Internet, but the thing which separates it to be used by only you (who is the owner or another authorized person) and not available to everybody on Internet is the user authentication process. Hence it is very crucial to implement the user authentication correctly and securely to make sure that the information should not land up in someone else’s hands that are not authorized to see or use it.

Cybersecurity has become such a vast topic that there are now almost unlimited areas of research in the field. The motivation behind this specific research was triggered by some recent activities that involved hacking of some celebrity Twitter and YouTube accounts [1]. In addition, there have been other attacks in the past which showed serious security flaws in the service providers making them vulnerable to the attacks [29]. Normally, that these flaws only become known come

after a successful attack which becomes known to the public or the media. Some real world incidents include the hacking of the website and Twitter page of WBOC [2], the hack and attempted extortion of Sony Pictures [3], the hack of the website of the Albuquerque Journal [4]. These data breaches again challenged the safety of some security practices which were used by the associated service providers and may still be in use by other service providers and forced a re-look at current security methods in use today for securing and protecting data.

One of the major focuses of the attacks were seen to be the user authentication mechanisms, so we started to look into the visible aspects of the user authentication process beginning with the account creation for the new user, which includes practices like checking of user selected password against common dictionary words, keyboard patterns, top 10,000 passwords etc., password makeup and strength, and so on. The other two most important visible aspects include password recovery steps used by the service, and the types of action taken by the service when user enters bad credentials more than maximum number of tries. And the overall authentication security of a given provider depends on the security of the specific practices employed in these three areas. There are another set of practices, those we call “invisible” aspects which are not visible to a user or external researcher. These include password hashing and salting mechanisms, and it is hard to know whether these methods are being used, or even how safe the specific practice is. This thesis illustrates the techniques that are currently in use and highlights the techniques that should be used to enforce strong security and privacy of user’s data.

This paper attempts to identify precisely the best currently in use practices that should be used by an online services to secure their online user authentication as well

as suggested several new best practices and illustrated those selections by showing several interesting vulnerabilities found in while surveying some of the major online service providers. The results showed that customer oriented services, which largely depend upon the number of subscribed customers don't want to lose their customers just by having really lengthy and time consuming enrollment as the customer just wants to create a new account, hence, often does not implementing some important authentication steps. This suggests that there's a tradeoffs between the customer retention policies versus security best practices implemented by the online service providers so that they do not lose their business by annoying customers by requiring them to go through a long series of steps that may discourage the customers to open a new account with the service and hence using that service, resulting less customers subscribed in their service. Unfortunately, to securing their customers' accounts they need to implement those missed steps sooner or later, and as digital crime is growing day by day. All the online services are at risk of an attack if they don't keep pace with current world's current best security practices.

## **2.2 Common Attacks**

When we talk about cybersecurity attacks, the list is never ending because the word cybersecurity in itself covers a vast technology space consisting of many specialized cyber security fields like forensics (packet, storage, memory, mobile device, etc.), web application security, network security, service authentication, cryptography, software security, and many more. It should be noted that the focus of this paper is only on one of the specified cybersecurity areas, that is, best practices for the user authentication in online service providers, hence the cyber-attacks described

in this section only covers the scope of the paper and are considered most common attacks for the online user authentication.

### **2.2.1 Brute Force**

A brute force attack is a type of cyber-attack in which attacker exhaustively searches or checks many or all possible combination of passwords or keys against the target, until the correct one is found. In general, in context of the paper, a brute force attack consist of a list of all combinations of a character set, or top common passwords which are checked against an online account that has weak password selection policies and hence can be easily compromised by using this kind of attack. The only way to make this attack more difficulty or near impossible to guess the correct password or key, the number of valid keys should be increased by making use of all the specifications included in section 4.1 of this paper. To illustrate the way how it works, let presume that an online service is using all the best practices described in section 4.1.1 Minimum Password Requirements.

Let's find out the number of combinations for generating a list of passwords for the brute force attack on this online service:

1. Minimum Length Restriction: 8 (at least)<sup>1</sup>
2. Special Character Inclusion: ~20 (total)
3. Number Inclusion: 10
4. Upper&Lowercase Letter Inclusion: 26,26

---

<sup>1</sup> As with all quantitative values in security, this value (8) will likely increase over time as computer power increases with time. The value 8 is a reasonable value at this point in time.

So the total number of combinations for a password of 8 characters in length would be:

$$(8+20+10+26+26)^8 = 90^8 = 4 \times 10^{15} \text{ (approx.)}$$

This is a very long list of passwords. If it takes 1ms to check one password (best case scenario), then to check this complete list against an account would take 126,839 years for a single computer. Even if considering that the attacker has a botnet of 100 computers, then to try all these password combinations, it still would take 1268 years which is impractical for the human being to do that unless some advancements are made by the technology which could make the processors faster. This example shows how crucial it is for an online service provider to have a stringent password selection mechanism to ensure security towards this kind of attack.

Now a days to same time and resources, attackers avoid checking all the password combinations, and use intelligent password guessing, which means that rather checking all the combinations (including password combinations that don't make any sense to humans and hard to remember), they build a list which includes password combinations that could be more likely to be the right guess. Intelligent password guessing shrinks down the list of password combinations for an exhaustive brute force attack and more likely to work in majority of attacks. Therefore it is strongly recommended to have lengthy passwords with all minimum password requirements [3.1.1.1], so that even if the attacker is using intelligent password guessing, it should be hard enough for them to build the intelligent passwords list from the list of all the password combinations.

### **2.2.2 Denial of Service (DOS)**

Denial of Service, also known as DOS attack is normally thought of as an attack where a significant services overload is directed at a specific service in hopes of crashing the target servers or slowing them down so that a normal user cannot use the service. This kind of attack and can produce worst results for the online commercial services, dealing with online shopping where the provider could lose millions of dollars if their website goes down or users are not able to access their accounts at the peak hours of shopping.

A related DOS attack is one in which an attacker can prevent a specific user from accessing their account. In this case a poor implementation might result in the DOS attack on the users account preventing them from accessing their accounts. This is one of the versions of DOS where the service is no longer be accessed to a user via normal procedures of access and hence a temporary lock-out occurs on the users accounts, however it does not mean that the service is altogether down, the users who didn't fall prey to this attack can access their account by the normal procedure. It is not uncommon in the world of online service providers, where the users are intentionally locked out of their accounts and hence no longer can access their accounts. Online services can be DOS'ed, if they are using time-out mechanism on their users account when the service encounters the maximum number of tries available to the user to login. In that situation, to prevent malicious attacker brute forcing the account, the service temporarily suspends the user account to slow down the attacker. This is one of the practices implemented to secure the account. And we consider it as one of the best practices, but only when it is implemented correctly, leaving no scope of DOS. It is crucial that if the online service is implementing time-out, it should have a proper, sophisticated algorithm to perform time out, taking into

consideration all the aspects, hence allowing legitimate user to log into the account but keeps out the attacker. Again DOS attack can occur in many ways and can take many forms, but our discussion of DOS attack is limited to only online user authentication mechanism, which is the scope of our paper.

### **2.2.3 Phishing Attack**

Phishing attack is kind of cyber-attack in which user is convinced to provide the credentials to an attacker. Unlike a conventional cyber-attack which tries to take advantage of a vulnerability in software or hardware, this is an attack which attempts to exploit a person, and this is often called social engineering [30]. In this attack, the attacker works to gain the confidence of the victim and fools the victim into revealing the credentials or other information to the attacker. Phishing attacks can be executed in number of ways, but the most common way is spam emails. A spam email can be constructed with malicious web URLs in the body. The email could be constructed in such a way that it appears to come from a legitimate source and then user is either convinced or forced to click on the URL provided in the email body, as soon as user clicks the URL, it becomes the victim to the phishing attack. With respect to our interest in online user authentication, one of the phishing attacks could be where the attacker can pretend to be a legitimate service provider and fool the users into logging into their accounts, hence getting the credentials of the legitimate users of that online service. This is frequently seen with email service providers. It is not a difficult job for an attacker to build a web page with same look and feel as online service providers with login option [8]. One of the important thing to avoid these kind of attacks is to be alert, and the user should always check the URL of the web page in which they are trying to login. If something is not right or correct on the web page or the URL seems



not correct then user should refrain in providing any kind of information or clicking any hyper-link on that web page.

## **Chapter 3**

### **SECURITY BEST PRACTICES SURVEY**

As stated in prior sections that the main aim of this research is to identify and describe industry best practices for online user authentication, it is important to catalog and fully describe those best practices in some great detail. Comparing these practices against a provider's current practice should help further reduce the attack surface of online service's authentication scheme and make its users' accounts less likely to be breached. This section organizes and discusses these authentication best practices.

#### **3.1 Category: Password Selection**

##### **3.1.1 Applicable To:**

All online services who have a login account for their users to access their service, for example:

- e-mail providers
- social media service providers
- e-commerce websites
- online banking services
- online money transaction services
- cloud storage services
- other online service providers

### 3.1.1.1 **Best Practice:** Minimum Password Requirements

#### 3.1.1.1.1 Minimum Length Restriction:

- **Definition:** The passwords should be at least 8<sup>2</sup> characters or more in length.
- **Discussion:** This best practice should be adopted to protect the account of the a user from being compromised by an attacker trying to exhaustively guess the password of the account. The service should not allow a user to propose a password less than 8 characters in length. This is an important aspect for the security of the online service account's users because it significantly increases the number of combinations of all possible character strings to be tried by an attacker, hence increasing the time and effort needed of the attacker who is trying to guess the password with all possible combinations. This security practice combined with other best practices in this Password Selection will protect the account from being compromised by an attacker who is trying to brute force the account, and it will become near impossible to crack the user's passwords.
- **Implementation:** This best practice should be implemented at the time of account creation for the service. When a new user is trying to create an account for the online service, the service should not allow the user to propose a password less than 8 characters in length. So, if the password proposed by the user is less than 8 characters, the proposed password

---

<sup>2</sup> As with all quantitative values in security, this value (8) will likely increase over time as computer power increases with time. The value 8 is a reasonable value at this point in time.

should be rejected and the user should be asked to provide another password. This practice should not be implemented standalone but with all the other best practices mentioned in the Password Selection category.

- **Example:** For many years the top users passwords contained short strings like 123456, qwerty, mydog, love, brenda etc. which are very easy to guess either exhaustively or when trying a list of commonly used passwords.
- **Attacks:** Brute force

#### 3.1.1.1.2 Special Character Inclusion:

- **Definition:** The passwords should contain at least 1<sup>3</sup> special character<sup>4</sup>.
- **Discussion:** This best practice should be adopted to make the passwords too complex to be guessed or exhaustively tried. This reduces the number of possible top common plain string passwords which can be tried by an attacker to compromise the user's account. Now the attacker may have to build a new list of passwords satisfying this practice. Even the task of building a list of passwords, to be tried while exhaustively brute forcing the account, will get more difficult as now the attacker may have to replace the common characters in the top common passwords list with the special characters, like 'a' can be replaced by '@' or 's' can be replaced by '\$', but the attacker then has to build the list with all the combinations of a single simple string password with every special character that fits in, (like may

---

<sup>3</sup> As mentioned before for password length, like all quantitative values in security, this value (1) will likely increase over time as computer power increases with time. The value 1 is a reasonable value at this point in time.

<sup>4</sup> Here special character means one from the special character set:~!@\$%^&\*()-+=/\*.,;:'"?'/<>{}.

be for 'password' he has to try : p@ssword, pa\$\$word or p@\$sword) and this will significantly increase the number of combinations of all possible special character strings to be tried by an attacker, hence increasing the list in size of an attacker who is trying to guess the password with all possible combinations. It becomes even harder to guess the password when this best practice is used with all the other best practices listed in the Password Selection category.

- **Implementation:** This best practice should be implemented when the user is creating an account to access the online service. When a new user is trying to create an account for the online service, the service provider should not allow the user to select a new password which does not contain any special character in it. So, if the password proposed by the user does not have a special character, then, the proposed password should be rejected and the user should be asked to provide another password which should contain at least 1 special character in it.
- **Example:** Recent activities [7] reveal that even today most of the people use single English words as their passwords. A brute force attack on those kinds of accounts will result in more than 50% cracked passwords when using top common passwords list or dictionary list [30]. The top users passwords contained strings like password123, butterfly, qwerty etc. which do not contain any special character are very easy to guess when trying the top common passwords.
- **Attacks:** Brute force

### 3.1.1.1.3 Number Inclusion:

- **Definition:** The Passwords should contain  $1^5$  or more numeric characters (0-9).
- **Discussion:** The use of at least 1 numeric character increases the password character set by 10 using the numbers (0-9) to make the passwords less likely to be guessed. This best practice, even though not strong enough when implemented stand alone, but when used together with other best practices in the Password Selection category, can result in a strong password which can be way too hard to be enumerated or guessed. It is seen that users usually use the numbers which they can memorize easily, such as the day, month or year in which they have born, or just a consecutive sequence of numbers like 123 or 098, which either they append after their simple password or use them stand alone. Adopting this simple practice can easily allow their accounts to be hacked. For these kind of passwords, it is still not easy to guess the passwords when appending numbers, but for a specific target this kind of scheme may fall prey to the attacker. A strong suggestion would be to use random string of numbers with all other best practices in the Password Selection category, specially not including the date of birth and sequential strings, which results in more time for an attacker to guess the password if it contains a random string of numbers. If the number string which is included in the password contains 3 numbers, so for the single password there will be 1000 possible

---

<sup>5</sup> As mentioned before for password length, like all quantitative values in security, this value (1) will likely increase over time as computer power increases with time. The value 1 is a reasonable value at this point in time.

combinations of 3 digit number, hence increasing scope of the top common passwords tested against the account. If more digits are used then the size of top common passwords list will increase by a huge amount. Hence it is always a good practice to use all the best practices together to protect the users' accounts from any kind of online attack.

- **Implementation:** This best practice should be implemented when the user is creating an account to access the online service. When a new user is trying to create an account for the online service, the service provider should not allow the user to create the password without having at least 1 number in it. So, if the password proposed by the user does not contain at least 1 number (0-9) in it, then, the proposed password should be rejected and the user should be asked to provide another password which should contain at least 1 number (0-9) in it. Also the user should not be allowed to select a new password which only contains sequential numbers such as 12345678 or 1212. This practice should not be implemented standalone but with all the other best practices mentioned in the Password Selection category.
- **Example:** This is most common practice among others stated in the Password Selection category which is currently in use by many online services, but not all. The easy work around of this is users usually use their date of births or sequence of numbers in their passwords which is not hard to guess. Such poor passwords includes: passw0rd, 123456, 1212, 4444, abc123 etc. Only using this best practice will results in weaker and easily cracked passwords.

- **Attacks:** Brute force

#### 3.1.1.1.4 Upper & Lowercase Letter Inclusion:

- **Definition:** The password should contain at least 1<sup>6</sup> uppercase and 1<sup>7</sup> lowercase character.
- **Discussion:** Most of the online services assumes that when users create the passwords, they are going to use the alphabetic character set, and hence don't apply a restriction that the password should contain at least 1 or more upper and lower case characters which results in weak password and puts the account owner at the risk of the hacking of their account. Therefore it is necessary for the online services to strictly apply the rule of inclusion of alphabetic characters in the passwords. More specifically the best practice should be to force the users to include at least 1 uppercase character which increases the number of possible combinations by a factor of 26 (now the password will have at least 1 uppercase character). Further, by forcing users to include a lowercase character as well, will increase the length of the password created by the user as well as the combinations by again a multiplying factor of 26. The benefit that a service provider will have by adopting this best practice will be - now it will become very hard for an attacker to crack the password, since the list which he might be using may

---

<sup>6</sup> As mentioned before for password length, like all quantitative values in security, this value (1) will likely increase over time as computer power increases with time. The value 1 is a reasonable value at this point in time.

<sup>7</sup> As mentioned before for password length, like all quantitative values in security, this value (1) will likely increase over time as computer power increases with time. The value 1 is a reasonable value at this point in time.



not contain passwords satisfying the required conditions and hence it becomes difficult to recreate the complete list which includes the valid passwords for the online service accounts.

- **Implementation:** This best practice should be implemented when the user is creating a new account to access the online service. When the user is asked to provide a valid suitable password, he/she should not be allowed to create the account with a password without containing any characters of alphabets (both upper and lower character set). Hence, if the password proposed by the user does not contain at least one upper and one lower case alphabets (a-z, A-Z) in it, then, the proposed password should be rejected and the user should be asked to provide another password which satisfies the required condition to create a strong password. This practice should not be implemented standalone but with all the other best practices mentioned in the Password Selection category.
- **Example:** When there are no restrictions on password creation for including the alphabetic characters then the users end up generating the accounts with simple passwords such as: 666666, 070707 etc. which are easily guessable when tried against the top 1000 or the other top common passwords lists available online. And hence the accounts with such simple passwords get compromised.
- **Attacks:** Brute force.

To show the serious consequences of not implementing the best practices for guaranteeing strong passwords of users, following statement is a quote taken from *“The Wall Street Journal”* soon after the iCloud photos breach:

*“Apple could have done more to make people aware of the dangers of hackers trying to target their accounts or the importance of creating stronger and safer passwords.” - Tim Cook [Apple’s CEO]*

Unfortunately, Apple could have possibly avoided or reduced the numbers of breaches that they experienced if they had implemented best practices for creating strong passwords.

The following image shows the use most of these best password selection practices, which includes length restriction, number, special character and upper/lowercase character inclusion:

The image shows a web form for user authentication. At the top left is a 'Help' link and at the top right is the OMB No. 3206-0005. The main content area is a light blue box containing instructions: 'Select a username and password and enter them in the fields below, then click the "Submit" button to continue.' Below this, it states: 'Your username must be a minimum of six characters with no spaces or special characters. It may contain letters and/or numbers and is not case specific.' For the password, it says: 'Your password must be a minimum of eight characters and contain at least one character from three of the following four categories:'. A bulleted list follows: '• Uppercase letters (A-Z)', '• Lowercase letters (a-z)', '• Numbers (0-9)', and '• Special Characters (#, @, \$, %, &, +, =, \*, ?, {, }, [ , <, >, ;, ")'. Below the instructions are two groups of input fields. The first group has 'Username' and 'Confirm Username' labels with corresponding text boxes. The second group has 'Password' and 'Confirm Password' labels with corresponding text boxes. At the bottom left of the form is a 'Submit' button.

**Figure 3.1:** Minimum password requirements

### 3.1.1.2 **Best Practice:** Disallow Common String

- **Definition:** Online services should not allow common strings (e.g. English words) as passwords, but should allow common substrings as a part of the password.
- **Discussion:** This is one of the most important best practices discussed in the Password Selection category. As the definition suggests, there should be a regressive check against common strings for the password suggested by the user during account creation. This is important because the passwords that most users commonly use are either well known or common words which they can remember easily. This makes the attacker's job easier to crack the account's password, especially if the attacker knows enough about the person whose account he is trying to break into. A very good example to illustrate this would be: if I am a pet person (especially interested in dogs) and the attacker is following/stalking me on my social media sites and if he is gathering every minute detail about me then maybe he can easily guess my password by trying out every single English word that relates to me (which I have ever tweeted or included in my comments on Facebook etc.). For example: he could try out my pet's names, my favorite food or my favorite novel. But the point is that the attacker would likely be able to break into my account if I have used a simple password; more importantly if the online service has allowed me to create a password of my own choosing without checking it against the common English words, dictionary words and top common passwords (we will see the characterization of the lists in 'implementation' more specifically). This makes the clear that the online service provider

should check the passwords proposed by the users while creating an account against the top common strings or dictionary words to make sure that the users' accounts are secure enough against any this simple type of brute force attempt. On the contrary, the English words or common strings should be allowed as a part of a pass phrase; "dog" alone should not be allowed but "dog jumped off the wall" should be allowed as a password, because it is more difficult to guess the pass phrase containing multiple English words<sup>8</sup>.

- **Implementation:** When user tries to create an account for the online service provider, the proposed password should be checked against following checklist before approving the password for the users' accounts and if it does not satisfies any of the following conditions then the user should not be allowed to create the account with that password and forced to enter a new password.
  1. Recent compilations of both old and recent common passwords found in breach datasets. For example, top 1000 passwords [Burnett66].
  2. English and foreign words.
  3. Common given names and surnames (English and non-English).
  4. Any sub-string in the user's ID as well as the complete username.

---

<sup>8</sup> The total enumeration of a small number of common words used in a "pass phrase" might also be easily enumerated. A best practice look at pass phrases is needed.

5. Common keyboard sequences (e.g. "qwerty", "zaqxsdcde", etc.)

6. Old passwords of the users<sup>9</sup>.

- **Example:** For many years and even now many users' passwords contain simple English words such as butterfly, wizard, animal, football, etc. which are easy to guess and crack if lists of commonly used passwords are available. It is also easier to guess the passwords which contain their username in it because the valid usernames list for specific online service is easily available on the Internet, and when a particular user is targeted then it is not hard to find the valid username of that person. Hence it is strongly recommended for the online service providers to use this best practice to protect their customers' accounts from brute force attack.

Following image shows that the common strings are not allowed while creating the account. The image is taken from an online account creation screen. The account creation attempt was made with "qwerty" as the password:

---

<sup>9</sup> This check should never be implemented by storing old passwords in clear text. Saving old passwords as salted hashes would be an acceptable implementation.

The image shows a web form titled "Create your free account". It contains three input fields: a username field with "blah" entered, an email field with "blahblah@gmail.com" entered, and a password field with "....." entered. The password field is highlighted with a red border and contains a yellow warning triangle icon. Below the password field, the text "Strength: Very Weak" is displayed in red. A red-bordered box contains the following text: "Your password is easily guessed. Try making your password longer. Combine uppercase & lowercase letters. Add special characters. Do not use names or dictionary words".

**Figure 3.2:** Disallow common strings

- **Attacks:** Brute force

### 3.1.1.3 Best Practice: Account Activation Check

- **Definition:** Account should be activated only after the verification of the user's email address.
- **Discussion:** According to this best practice, the online services should not allow account activation unless it is confirmed that the email address used as login user ID at the time of account creation by the user, is valid. This means users should not be able to create a full-fledged working account with a non-existing email ID. It is specifically applicable to the online services that use a user's complete email address to login into the account.

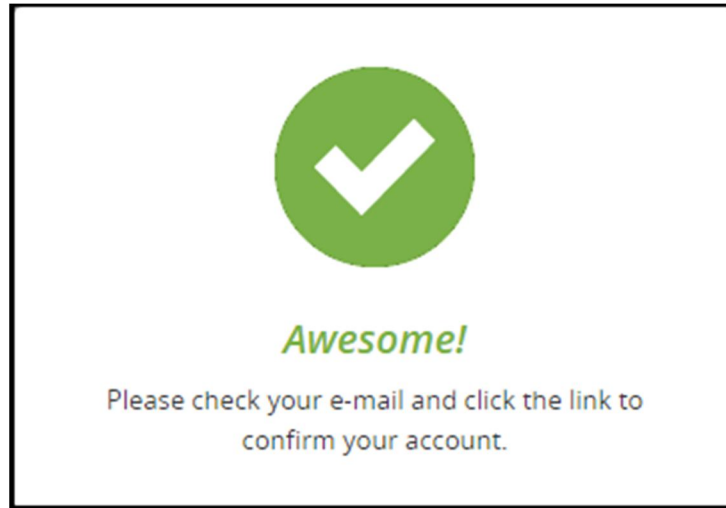
This includes some social media providers, for example, Facebook and Amazon, who have email address as the username for their users. Online service providers should check whether the given email addresses belong to a valid active account or it is just a dummy email address. This is important because it will prevent users to create fake accounts which are specifically used for some malicious activities. Also it reduces the number of fake accounts created, as now users have to get to their email address to activate the account. Furthermore, it protects the online services from the attackers using botnet and robots to creating large numbers of fake accounts (as potential DOS) because the account will not be activated unless the required activation steps, which are sent on the email account, has been followed. In these cases, account activation link is sent to the email address provided by the user as the username, and then user has to log into the email account and follow the account activation link to confirm that it is a human being and not a robot, who is trying to create fake accounts with fake, non-existing email addresses.

- **Implementation:** It is clear from the discussion that this best practice specially focuses on the online services that use email addresses of the users as their usernames. When implementing this best practice, when the user tries to create a new account for the online service provider, the service should verify whether the email address provided by the user is a valid account or not. To do this, the online service provider should send an account activation link to the provided email address of the user and should activate the account only when user logs in its email account and follow the

activation link. Also, user should be warned about the expiration of the activation link. Ideally the activation link sent to user should expire in few hours. Only after the verification of the user's email account, the online service account activation process should complete, otherwise if the activation link expires then the user should request another activation link to be sent by the online service provider.

- **Example:** This practice has been recently adopted by couple of online service providers. Because it is assumed not to have much impact on the security of the online service users, there are only a handful of online service providers using this best practice. We recommend using this best practice to cover every possible loophole which can prove to be a potential vulnerability in the future (precautions are often better than the cure). Following example image shows that the account has to be confirmed using the activation link sent to the email address provided at the time of account creation.





**Figure 3.3:** Account activation check

- **Attacks:** DOS

#### **3.1.1.4 Best Practice: 2FA For Login**

- **Definition:** Ultimately a mandatory 2-Factor Authentication (2FA) for every time user logs in is far safer than a conventional user ID and password used alone.
- **Discussion:** As the best practice states that there should be mandatory 2-Factor Authentication for every login to the account that means the user should go through the 2-Factor Authentication steps every time the user logs into the account. This greatly impacts the time to brute force the account as now not only the attacker needs a correct password but also the answers to the questions of the second step of user Authentication which includes one or more following step for authenticating the user:
  1. A secret code sent to the mobile phone of the account owner.

2. A secret code sent to the secondary email address of the account owner,
3. \*Security questions\* asked at the time of account creation,

or any other method of authenticating users at second step. This is crucial step to protect the account from any involuntary activity. To illustrate this consider this example - an account got compromised by brute force attack on user's password, then after successfully cracking the password the attacker is now asked to enter a security code which was sent on the mobile phone of the legitimate account user, but now attacker is stuck because he can't access the account unless he has got access to that code or indirectly the account owners cell phone. Some would claim that if the attacker could steal the cell phone and then he can compromise the account, but two things have to be considered: 1. the attacker is slowed down at the very moment and 2. Majority of attacks happen remotely, i.e. involving countries borders separating the attacker from the owners vicinity and in that case it becomes impossible to get the access of the owner's cell phone. So in that case 2-Factor authentication with owners cell phone use will provide a defense to the users accounts. More stringent security is acquired by using a secondary email address of the account owner which can't be stolen by the attacker- which means if the attacker want to hack a particular account then he has to hack the secondary email account of the owner first which will delay his actual hack as well as will consume his resources (depending on the best practices implemented on the secondary email account). It is strongly suggested that there should be no use of any kind of

information related to the account in the security code message or in the security code email, so that even if the account of cell phone got compromised there should be no information of the account for which the code was sent. There is a complete new best practice for this kind of situation and is explained in more detail in Password Recovery category. Furthermore, while using ‘personal’ security questions, the online service should be aware that in this day and age nothing is ‘personal’ anymore, i.e. everything might be available over Internet for example on social networking sites, so if the online service considers using security questions as next step for user authentication, it is strongly recommended that the user should have freedom to enter its own security questions and answers so that the attacker should be unaware of the questions priority and try to gather the probable answers for that questions beforehand. Therefore the second step of authentication should be a mandatory step before user is actually logged into the account.

- **Implementation:** As discussed that 2FA should be mandatory on all online accounts to use online services. Many online services offer 2FA, but keeps it optional that means whenever users login to the account they don’t go through all the steps require for 2FA if it is not activated on the account, so that it is up to the user whether he/she wants to activate 2FA on the account or not. This kind of practice is strongly discouraged because when a user builds an account, the user hardly pays much attention to activate 2FA on the account, or even if noticed, the user does not activate it just to skip number of steps whenever it tries to login, or in many situations users just

don't know what's 2FA really is and what its used for? So the work around for this kind of situation, online services should not give the option of 2FA as an optional step, instead they should implement 2FA by default without any option of skipping it. To do this, it should be implemented when user creates a new account and after asking all the valid account creation parameters for a secure user authentication (username and password), the next step should be asking user the questions involved in 2FA such as phone number, or secondary email address etc. depending upon the choice of 2FA by the online service. This step should clearly state that this step is used every time the user tries to login, so that user refrains in providing any kind of fake data for this step because otherwise user won't be able to login to the account if it has provided non-existing mobile no. or any other fake information regarding 2FA. Only after this step user should be able to successfully access the account.

- **Example:** There are very few or almost none online services using this best practice, the reason is obvious, they don't want to annoy their customers every time the users try to login. This is fair enough, but they should consider the consequences of not implementing 2FA even though having it as an "option" to attract the users who are really paranoid of their online data security, doesn't really help to protect the security of the accounts. Presently there are very few online services having 2FA but none of them have implemented it as a mandatory step for user authentication while user logs into the account.

A very famous presentation made by a Russian security researcher [14] showed that Apple does not use the 2FA to protect the iCloud backup, Find My Phone data as well as documents stored in cloud, and by knowing only the Apple ID and password, one can perform remote data download without even notifying the actual user of the account. This shows the risk of security on the users account for not using 2FA at the time of login.

- **Attacks:** Brute force

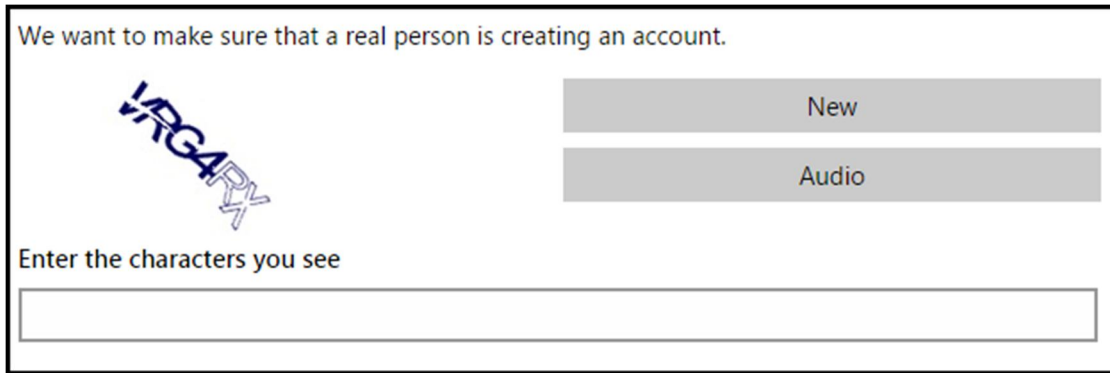
#### 3.1.1.5 **Best Practice:** CAPTCHA While Account Creation

- **Definition:** CAPTCHA [32] should be used for creating every new account.
- **Discussion:** CAPTCHA is an image of a string which may be pure numeric or alpha numeric letters generated by arbitrary selection of numbers and alphabets. Use of CAPTCHA ensures to some extent that the user trying to access a service or website is a real person not a robot. Still it is widely use to slow down the attacker trying to create new dummy accounts or just trying to brute force. Use of CAPTCHA is important while account creation because it protects the online service from infinite dummy accounts creation which utilizes the resources of the service exhaustively (DOS). On the contrary, many believe that CAPTCHA should no longer is a best practice [33][34] because there are automated tools which can be used to crack the CAPTCHA although consuming couple of seconds, but if the attacker is patient enough to create the accounts, then tools can be used to overcome the CAPTCHA wall protecting the service. But in a way

CAPTCHA is still considered a best practice by us, as according to our research which shows that some very complicated CAPTCHAs are hard to crack with the available techniques and serve as a vital role in protecting the account creation process. In this regard CAPTCHA is considered a better alternative to be used to provide at least first level of security to the service. We would still recommend using very complicated CAPTCHAs at the time of account creation by users and hence we consider it as one of the best practices for online user authentication.

- **Implementation:** CAPTCHA should be implemented at the time of account creation by the user. When a new user wants to use/access the online service and for that it has to create an online account then on account creation page, after user has selected a valid username and password and answered all the required fields asked by the service then before creating an account for the user, it should ask user to enter a CAPTCHA and after validating that the CAPTCHA provided by the user is correct then only an account should be created for that user. This process should be followed every time when a new user tries to create a new account.
- **Example:** For several years CAPTCHA was considered as a best practice for slowing down the attempts of account creation by an attacker. We still believe CAPTCHA is still a best practice and many online services has implemented it as a practice to verify that the new account for the service is created by a legitimate user or at least a human being. There are online services which don't implement CAPTCHA at the time of account creation

which might create a problem for the services in near future. Following image shows the use of CAPTCHA at the time of account creation:



The image shows a CAPTCHA interface for account creation. At the top, it says "We want to make sure that a real person is creating an account." Below this, there is a distorted image of the word "VIRGARY" in blue. To the right of the image are two buttons: "New" and "Audio". Below the image, it says "Enter the characters you see" and there is a text input field.

**Figure 3.4:** CAPTCHA while account creation

- **Attacks:** DOS []

#### 3.1.1.6 Best Practice: Show Password Strengthbar

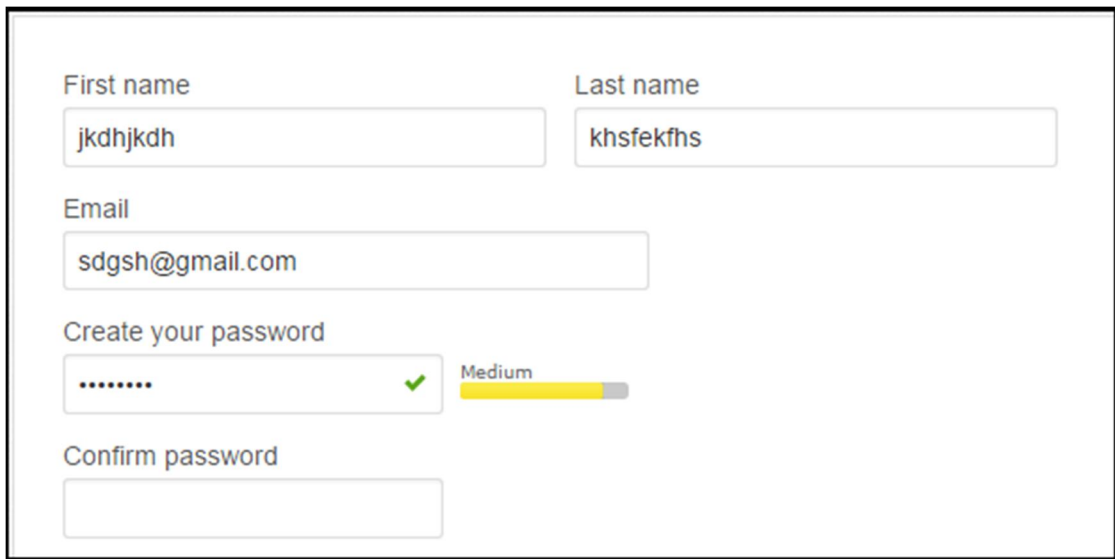
- **Definition:** Display a visible color strength bar showing strength of the password.
- **Discussion:** This practice illustrates that there should be a visible color bar showing the estimated strength of the password provided by the user at the time of account creation with the online service. For instance, bright red for the poor password and bright green for a good strong enough password. It is very clear that the use of this best practice will enhance the security much further because the visual interpretation gains a quick attraction of the user towards the alert it signifies and is very unlikely to be ignored especially red color, which has been a sign of alert or danger. This might

force the user to think again when creating a weak password. It should be noted that many online services use an optional message off to the side of the password field which contains a message saying that the password is weak or poor, but usually it is seen that people ignore any written warning on the form since they already have to go through all the annoying questions of asking the information regarding the account. Also there is so much text already present on the form that it is hard to get the user to focus on any written message, unless it is mandatory to correct all the errors on the form including that warning message. So we consider this as a best practice to show the visible strength bar for the password provided by the users.

- **Implementation:** This best practice has to be implemented at the time of creating an account for accessing the online service. While creating a new account for the service, there should be a visible color strength bar which shows the strength of the password entered by the user. Until the color bar shows a strong enough passwords proposed by the user, the service should not accept that password and ask the user to enter a new strong password.
- **Example:** Many online services don't have this practice implemented and as a result, most of the users don't realize, at the time of account creation, that the password provided by them does not satisfy specifications that a strong password should have and hence fell prey to the attackers. Also, more importantly the users don't understand that the password which they consider strong enough can actually be very poor and could be found easily in the top 10,000 passwords list. Hence, they need be get aware of the fact



that they should propose a password that is considered secure enough according to industry practice not according to their personal consent. For example: p@ssword, \*\*\*\*\* etc. are no longer considered safe. Following image shows the strength bar for the proposed password, the proposed password was “p@ssword”:



The image shows a registration form with the following fields and elements:

- First name:** jkdhjkdh
- Last name:** khsefkfhs
- Email:** sdgsh@gmail.com
- Create your password:** A text input field containing six dots (••••••) with a green checkmark to its right. To the right of the input field is a strength bar labeled "Medium". The bar is a horizontal line that is approximately 75% filled with yellow, indicating a medium level of security.
- Confirm password:** An empty text input field.

**Figure 3.5:** Password strength bar

- **Attacks:** Brute force

## **3.2 Category: Bad logins Protection**

### **3.2.1 Applicable To:**

All online services who have a login account for their users to access their service, for example:

- e-mail providers
- social media service providers
- e-commerce websites
- online banking services
- online money transaction services
- cloud storage services
- other online service providers

### **3.2.1.1 Best Practice: Restrict Maximum Bad logins**

- **Definition:** The maximum number of attempts to login should be restricted.
- **Discussion:** This best practice clearly illustrates that user should be restricted from attempting a large number of bad logins at the same time. This is very critical aspect to be taken care of while implementing a secure online service. It is not uncommon for attackers to exploit this vulnerability, as the very first thing that they do to exploit any online service is to see whether it can be brute forced or not. Brute force is the technique in which attacker vigorously tries to crack the password by attempting top 1000 or more simple passwords to compromise the accounts and thereby, the service. Hence, it is very important to stop the attacker trying a large number of password combinations to crack the password. This is done by restricting the maximum number of tries a user can have to successfully log into the account. To do this there should be a mechanism to stop or slow down the user from trying n number of passwords. Majority of services timeout the user and then suspends the account temporarily, so that if an attacker is trying to get into the account, he must have to wait till

the timeout expires and retry again. This kind of implementation, to protect the account being brute forced, and directly impacts the time needed to crack the password as we will see how in later best practices described in this category. The other important technique used to slow down the user is to use CAPTCHA. It is a random string of numbers, alphabets or combination of both which needs to be put in before attempting to go forward. The main purpose of using CAPTCHA was to distinguish between human user and botnets. Now days CAPTCHA is no longer considered to be safe by some standards [33][34], as there are many tools which can automate CAPTCHA for you easily available on Internet, but we recommend using complex CAPTCHAs to slow down the user. In any case, an online service should not give users any number of tries to log into the account even if they have good rules to create a hard password, if an attacker is really interested in hacking a particular account then sooner or later he will get through it if there are no restrictions on login attempts.

- **Implementation:** This best practice has to be implemented at the login time, when users already have created their accounts with the online service. At the time of login, users should get maximum 5 to 6 attempts to log into the account successfully and after that there should be a mechanism to prevent them to try to login into the account simultaneously to prevent account from being compromised.
- **Example:** Many times in the past [18], brute force attack took place on the accounts not secured against these kinds of attacks. And simple passwords without any kind of complexity becomes very easy to exploit. It makes the

task of the attacker easier to brute force the accounts because for a long period of time, people had passwords like password123, honeybee, etc. very easy passwords. And even the lists of top 1000 common passwords have been leaked from online service users and made available to the world. These lists are used by the attackers to brute force the accounts.

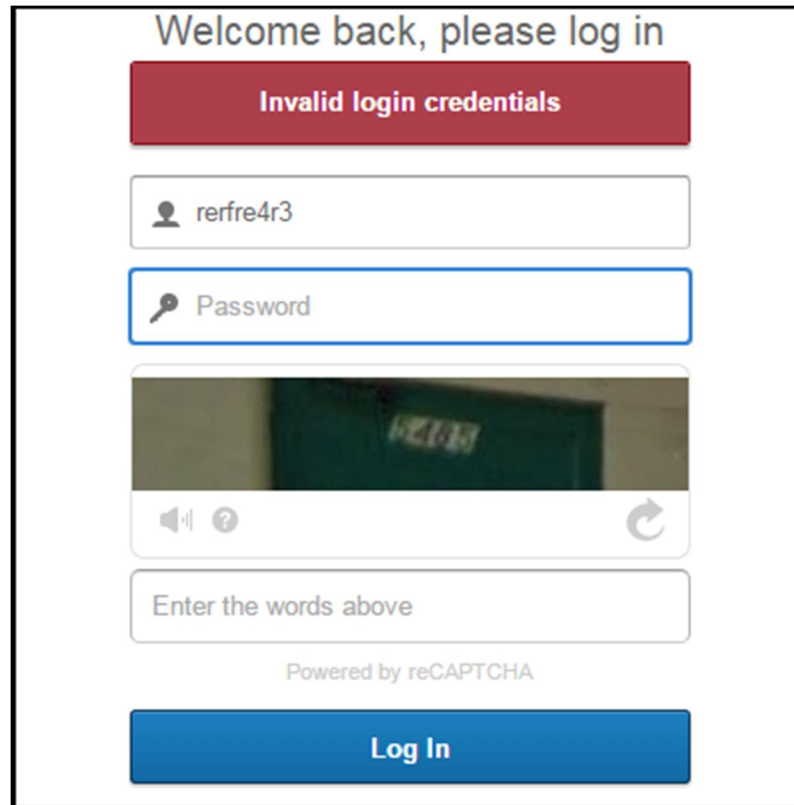
- **Attacks:** Brute force

### 3.2.1.2 **Best Practice:** Slowdown With CAPTCHA

- **Definition:** The user should be slowed down with the use of complex CAPTCHA, while exceeded the maximum attempts to login.
- **Discussion:** We recommend to use complex CAPTCHA as the first layer of protection from the brute force attack on the online services and it will become really valuable to use CAPTCHA, if the best practices of the Password Selection category are already employed, because then it will take years for an attacker to brute force any account. CAPTCHA can be implemented for various reasons, one of the reasons is already discussed in Password Selection CAPTCHA While AccountCreation, and here the implementation has the same purpose, as to distinguish between the legitimate users and the attacker. Here it is protecting the account from being compromised by the attacker who is trying to brute force it. The way it works is that when user exceeds the maximum number of tries to log into the account then the user is slowed down for every next attempt made and the user is forced to enter a CAPTCHA before trying to login. It slows down the user to attempt too many logins at the same time and prevents the account from being compromised.

- **Implementation:** This best practice should be implemented during the time of login. This is one of the mechanisms for preventing users to try multiple times to login with a bad password. When user tries to log into the account with a bad password, then after the user had exceeded the maximum number of tries, which should be typically 5 or 6, then user should get a CAPTCHA on the login screen which should ask user to enter the image of string displayed by CAPTCHA. The user should not be allowed to log in, even if the right password is provided, until he/she enters the correct CAPTCHA. The login button should be disabled until anything is provided in the CAPTCHA box. If the user entered the wrong CAPTCHA, then he/she should get the message that the provided CAPTCHA is not correct. The user should be allowed to log in only when the provided username, password and CAPTCHA combination is correct, if any of the fields is wrong then the user should again be forced to enter a new CAPTCHA with correct fields to log in successfully. It should be considered that CAPTCHA used by online service providers should be complex enough, so that it should be hard to automate the use of it.
- **Example:** In the past, CAPTCHA was considered a safe practice to keep attackers away from compromising the accounts, until the methods were developed for the automation of the CAPTCHAs, used by the online service providers. Even though these methods provide a solution for solving the CAPTCHAs, they are not very efficient and still cause delays in attack. Because of the automation of CAPTCHAs, it is no longer considered as a best practice by the industry to use CAPTCHA to slow

down the user [33][34]. But due to the fact that it is still hard to automate the use of strong and complex CAPTCHAs, many services still use CAPTCHA to slow down the attacker, and we strongly recommend the use of such type of CAPTCHAs to prevent attacker from brute forcing the accounts, since improper implementation of other mechanism to slow down the attacker such as timeout could result in another attack known as Denial of Service attack, which is discussed in more detail in next best practice under this category. So, best practices 3.2.1.1 and 3.2.1.2/3.2.1.3 go hand in hand, that means if the user is restricted to login after maximum number of allowed logins, then either 3.2.1.2 or 3.2.1.3 is used as next step to slow down the user. The following image shows the use of CAPTCHA to slow down the user after crossing the threshold of maximum number of login tries:



**Figure 3.6:** Slowdown with CAPTCHA

- **Attacks:** Brute force

### 3.2.1.3 Best Practice: Slowdown with timeout

- **Definition:** Suspend the account for some random amount of time.
- **Discussion:** This best practice states that when the user exceeds the maximum number of attempts allowed to login then the login should be suspended temporarily, that means the account should be timed out. Great precautions should be taken to avoid a Denial of Service attack, in which legitimate user is deprived of logging into the account even if the provided

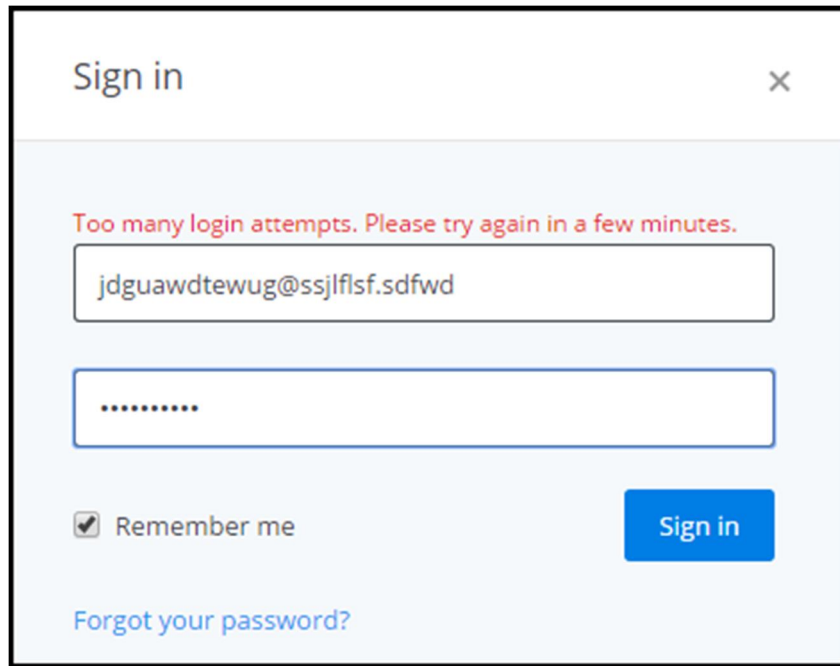
credentials are right. Complex algorithms were built to implement this practice, so that the correct implementation should be one in which if the attacker has blocked an account, and if the legitimate user access the account with correct credentials then it should provide the access to the user, but blocking any further attempts from attacker. This might involve certain factors to be taken into consideration such as monitoring traffic from a particular ip and blocking that ip for any suspicious activity for that particular account. Blocking the ip all together will result in DOS for the other users who will try to log into their accounts from that ip address. Hence majority of the services have algorithms which blocks a particular account, under suspicious activity, get accessed from a particular ip address and hence timeout the account to be used from the particular ip address. This best practice results in a huge lag for the attackers trying to brute force the accounts and hence if the timeout is large enough, say 60 minutes, and the maximum number of attempts is limited to 5-6 then it will take a week for an attacker to try top 1000 simple passwords (without any restrictions on length and special characters). Even if the length is at least 6 characters, without any restrictions on password (taking simplest case) then to try the total number of combinations ( $26^6$ ) with timeout of 60 minutes will take approx. 5000 years for an attacker. This is the simplest case considered; again if a botnet is used to attack a particular account then the time will vary depending upon the resources available to an attacker. But the bulk attack on various accounts would take forever to brute force since every account will require a botnet of many machines and to build a network of



huge amount of botnet machines to target few hundreds of accounts would be practically impossible. Hence, timeout is considered to be petty efficient while considering the security of the users accounts.

- **Implementation:** To implement timeout mechanism, precautions should be taken so that it should not result in DOS attack on the service. When user has lost the maximum number of attempts available to log into the account then the account should be temporarily suspended for definite amount of time. After the timeout has expired user should be again allowed to login into the account and given the maximum number of attempts. There are two ways to implement timeout on accounts. The first one is described above, to use a definite amount of time to suspend the account every time user exceeds the maximum number of attempts back to back. The other one is called exponential back-off, in which when the first time account gets suspended, the time interval of suspension is fixed, then if the user again exceeds the maximum login attempts after expiration of first time out, then the timeout interval increases exponentially and hence so on. This happens every time when user fails to login successfully into the account. Once the user successfully logs into the account, then the timeout is reset to the first timeout interval which is the minimum amount of time of account suspension.
- **Example:** Slowing down the user with timeout is considered to be so efficient than using CAPTCHA that majority of online service providers use this technique to shield the accounts from being compromised. The technique has its own advantages as well as disadvantages which are used

as a trade-off by online service providers whether to go with the implementation of the timeout or CAPTCHAs with complex strings. Following image shows the use of timeout to slow down the user after crossing the threshold of maximum number of login tries:



**Figure 3.7:** Slowdown with time-out

- **Attacks:** DOS

#### 3.2.1.4 **Best Practice:** Bad Login ErrorMessage

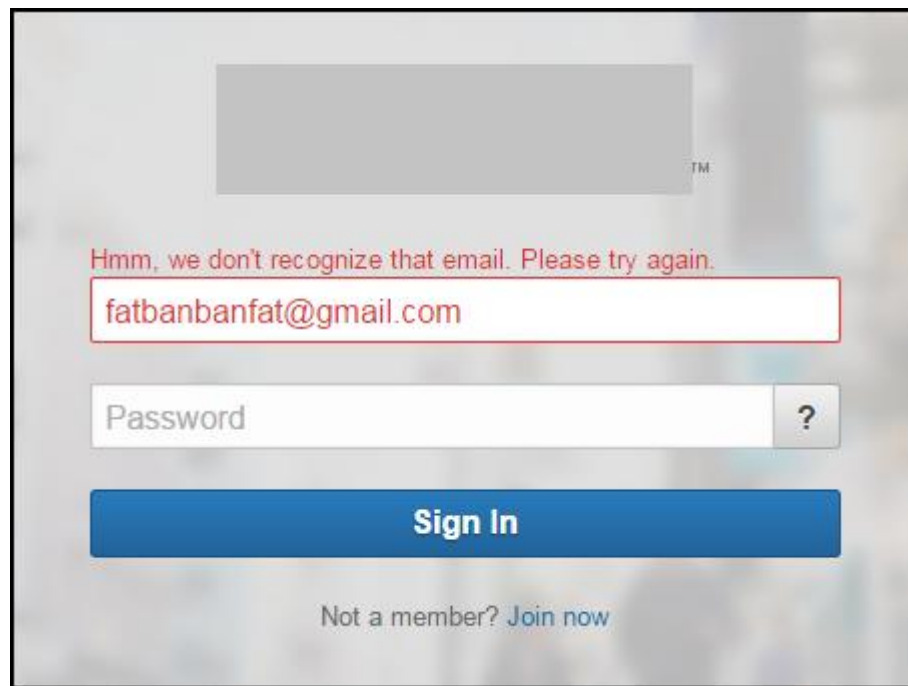
- **Definition:** Error message for bad login should not reveal whether username or password is incorrect.

- **Discussion:** This best practice illustrates the correct way to notify user about a bad login. When a bad login is encountered, the user is generally notified of what went wrong. In that case user might have either entered a wrong password or might have miss-spelled the username, in either ways the error notification message which is displayed after a bad login should never tell user what was entered wrong, username or password. This is crucial to hide the identity of the correct usernames of an online service. There are many ways how the leakage of correct usernames could harm any online service and especially the users. In case of email service providers, this information is sold to different promotional agencies/companies that then send promotional messages and spams to the usernames list of that email service. The worst case would be, these lists are then used to engage innocent users to phishing attacks or to infect their systems with viruses or Trojans, which can be easily done by a phishing email. These phishing emails are made with all the shiny and glittery offers which, at the first glance, appears to be attractive, and once the users are convinced to click some URL or download something from that phishing email, then they fell prey to the phishing attacks, which can take over their machines and could do all kind of weird stuff. Have you ever observed getting emails from unknown email ids that may contain promotional offers, some downloadable files containing virus or worms? Have you ever thought how they got to know your email address? Another example would be, the lists of legitimate usernames can be used to brute force the accounts, in which now attacker has to concentrate only on passwords,

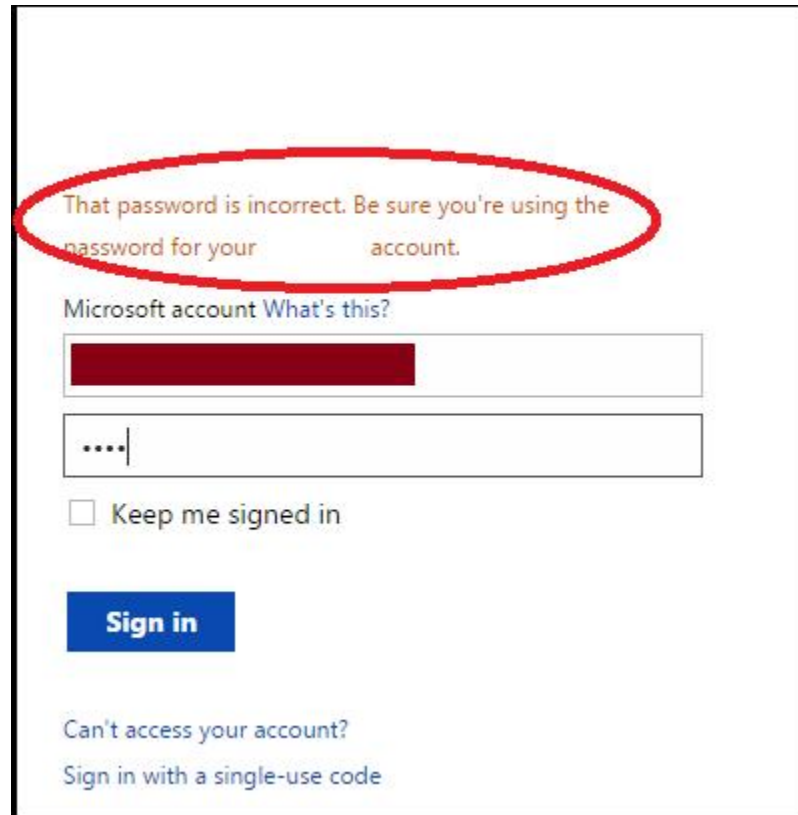
since he already has got the correct usernames of that online service users. Hence, the online services should be conscious about what information they should give out and to whom.

- **Implementation:** This best practice is implemented at the time of login. When the user provides the credentials to access the online service, then if the username/password combination doesn't match any valid record, then the error message provided should be a generic message saying: "username or password is incorrect" rather than saying "username is incorrect" or "password is incorrect". If the user is a legitimate user then he/she might figure out what's incorrect (since username is generally not hidden by asterisks) and may be able to login successfully in next attempt. The message should be clear enough and should not give any kind of hint whether the username or password was incorrect.
- **Example:** There are some online services that still use the error messages which are specific to the username and password. Although there are other ways too to determine whether a username is a valid username or not, but those ways takes time, such as while creating a new account with the service, if the proposed username already exists then the user is forced to provide other username. This way takes more time to determine a list of usernames that actually exists because for every username, the attacker has to go through the account creation process which takes time to answer all the questions on the page. But it takes less than 10 seconds to determine whether a username exists or not, if the service provides the specific error message at the time of bad login. Hence this best practice doesn't allow

attacker to build a list of valid usernames on the basis of bad login attempts. Following images shows that after a bad login, the error message appeared on screen reveals that either the email address (username) or the password is not registered with the online service and hence, it doesn't recognize it. This should be strongly avoided as per the above best practice.



**Figure 3.8:** Incorrect username error message



**Figure 3.9:** Incorrect password error message

- **Attacks:** Phishing attack, Brute force attack

### 3.2.1.5 **Best Practice:** No Timing Difference Loading Pages

- **Definition:** There should not be any timing difference while loading legitimate login page versus a bad login page.
- **Discussion:** The time to load a page after a successful login should not have a noticeable difference to the time to load a page after having encountered a bad login. It does have a significant impact on the attacker to get a hint whether the login was successful or not. By measuring the times

it takes to load different pages, one can predict the next pages after either a bad login or a successful login and can code accordingly to take actions based on timing differences to load pages. This best practice is not as critical as other mentioned in this category, but worth mentioning as one of the best practices to be implemented to make an online service fool proof from any kind of attacks.

- **Implementation:** This best practice has to be taken care of when implementing the pages involved in the user login process. When user logs in, the pages loaded after a successful login and after a bad login should not have considerable difference. Delay in loading of pages also can be caused by the factors like Internet connection of corresponding hosts or slow processing speed of the host's processors, but it definitely should not be because of poor implementation. Online services should make sure that it should not be properly implemented from their side.
- **Example:** There are not many incidences reported based on specifically exploiting this vulnerability. But a poor implementation might results in the exploitation, so it is better to take precautions.
- **Attacks:** Brute force

#### 3.2.1.6 **Best Practice:** Notification Of Bad Logins

- **Definition:** Email/text notification to the owner of the account about the bad login attempts.
- **Discussion:** This is one of the most important steps to be taken after experiencing exploitation on users account. The online service provider should notify users of each and every activity taking place on their

accounts. Even if it is annoying for the legitimate users to get hit by the email notifications every time they log into their account or every time they change their passwords or if they mistakenly type wrong passwords, in case they have forgotten their passwords, it is the responsibility of the online service provider to notify users as this could save them getting compromised. This best practice particularly focuses on the notification provided to the users when someone exceeds the maximum attempts of logins, to make the users aware that somebody is trying to log into their accounts, and if they didn't do that, then it is better they take actions like changing their accounts password to some strong and uncommon password, or just notify the security team of that particular online service provider to let them know somebody is trying to brute force their accounts, who then can take the required action. This can save the users information from being compromised and alerts the users as well as the service provider of the suspicious activity on the accounts. The majority of brute force attacks can be avoided by implementing proper mechanism to notify users of the suspicious activity and forcing them to create strong enough passwords which are hard to brute force.

- **Implementation:** This best practice should be implemented by the online service providers. We strongly recommend this best practice as a minimum requirement for an online service provider to avoid the impacts of attacks on user's accounts. This best practice should be implemented when somebody tries to login into the account and exceeded the maximum number of tries then an email or text notification should be sent to the

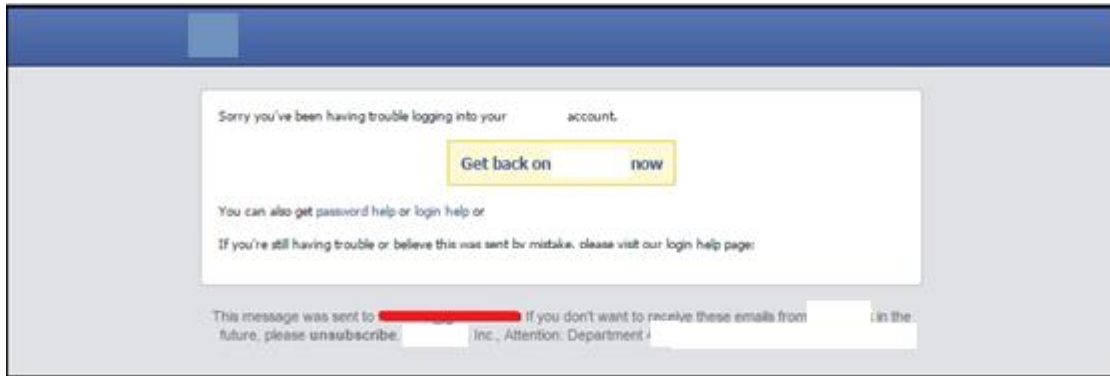


actual owner of the account notifying him about the bad login attempts, so that if the user wasn't trying to log in then he/she could take necessary steps to avoid any harm to its information and account. The notification should be sent every time the maximum number of login attempts is exceeded.

- **Example:** Again taking the example of Apple iCloud breach in which the photos of famous celebrities were leaked and made available on Internet, they lacked the proper notification implementation to the users, as seen in this statement by The Wall Street Journal [5]:

*“Apple said it plans to start sending the notifications in two weeks. It said the new system will allow users to take action immediately, including changing the password to retake control of the account, or alerting Apple's security team.”*

It is important for the online services to provide notifications on the activities taking place on users account including notifications for password reset, last activity of the account (when and where), and notification of cloud data download on the devices. The following image was a from the email sent for the bad login notification from an online service, even though the subject line of the email does not stated anything about the maximum login tries, it had “Getting back onto your account” as the subject line. It is noticed that this email was sent every time the login attempts exceeded the maximum allowed attempts, and account gets timed out:



**Figure 3.10:** Bad login attempts notification

- **Attacks:** Brute Force, [5]

### 3.3 Category: Password Recovery

#### 3.3.1 Applicable To:

All online services that have a login account for their users to access their service and provide the password recovery mechanism, mainly including:

- e-mail providers
- social media service providers
- e-commerce websites
- cloud storage services
- other commercial online service providers
- online banking service

##### 3.3.1.1 Best Practice: 2FA For Password Reset

- **Definition:** Mandatory use of 2 Factor Authentication while requesting a password reset.

- **Discussion:** Many online service providers have 2 Factor authentication, but how many of them make it mandatory to use 2FA while password recovery? This question is important, as how can you allow someone to reset the password of an account without verifying the true identity of the person? There are different mechanisms provided by online services to recover the password of the account. It is crucial to use 2FA to recover the password of an account, if users have forgotten it. Mandatory use of 2FA adds an extra layer of security while recovering the account's passwords. To show the importance of the 2FA, consider a scenario where the password recovery mechanism is based on some personal security questions, in that case if the targeted person is a famous or well-known person then probability of answering the security questions correctly by an attacker is quite high, since the basic information about the target can be easily found out via Internet. In this case if there is no shield of extra protection (such as 2FA) then attacker would easily get into the account. If 2FA have been used then in that situation, even after answering the security questions correctly, the attacker need to have the mobile phone access (if 2FA is based on text message on the mobile phone) to enter the security code sent to the registered mobile number on the account. Or the attacker has to get access to the email account on which the code has been sent, if email address was used for having 2FA. In either case, 2FA allows an extra security check on the user, who is trying to reset the account's password, by asking the security code which can only be provided by the legitimate user who owns the email account/phone. This could make it very difficult

to hack into the account if the user is using 2FA to protect the account. We all know the importance of having 2FA on the accounts, but we hardly pay attention while creating the accounts to activate it at the time of password recovery, or there are many services which uses 2FA only at the time of login but don't implement it at the time of account's password recovery.

- **Implementation:** This best practice is implemented while user asks for the password recovery. When user have forgotten the password, it goes to the password recovery site and asks for the new password, regardless of the method provided by the online service to reset the password, such as, security questions, email sent with recovery URL, recovery based on the personal information (last time of account access etc.) etc., the online service should use 2FA to reset the password. The first factor of authentication is the information provided by the user to any of the above mentioned methods and the second factor authentication should be either a security code sent to the registered mobile phone or the use of the secondary email address to send the security code to the user, or to use an authenticator. And after the user has verified itself by providing the security code using 2FA then only it should be allowed to change the password otherwise the user should not be allowed to change the password or should contact the security team of that online service provider, if it has lost access to its devices used for 2FA to recover the account.
- **Example:** Many online account hackings are caused because of not having proper implementation of 2FA on the user's accounts. It is easy to get personal information and answers to the security questions, but it is

difficult to circumvent the second layer of security which is 2FA. One famous example of such kind of breach was the Apple's iCloud breach in which the attackers managed to steal the accounts of the famous celebrities by answering the security questions correctly and resetting the accounts passwords. How hard is to guess the answers to the security questions such as "first school", "mother's maiden name", "favorite car" etc. of the celebrities whose information is available all over the Internet? An update made by the Apple's press contact - Natalie Kerris after the investigation of the Celebrity Photo Breach:

***"To protect against this type of attack, we advise all users to always use a strong password and enable two-step verification."*** - Apple Press Info [6]

The quote from *The Wall Street Journal*, in which it is so much clear that if Apple have used 2FA while password recovery then it would have saved them from the attack, even if attackers have managed to get all security questions correct.

***"Celebrities' iCloud accounts were compromised when hackers correctly answered security questions to obtain their passwords, or when they were victimized by a phishing scam to obtain user IDs and passwords."***-Tim Cook [CEO, Apple Inc.][5]

- **Attacks:** DOS

### 3.3.1.2 **Best Practice:** Use Of Recovery Email Address

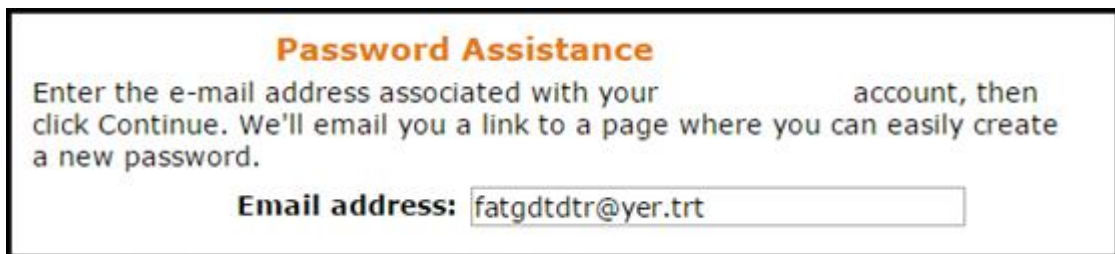
- **Definition:** Use of email address which owner of the account has provided while setting up the account.

- **Discussion:** This is a common practice to use the email address that the user has provided while creating the account with online service, to use for password recovery. This email address could be the username of that account, or if the username is some kind of string then this email address is the one which might have used for the verification and activation of the online account. It is a common practice by the online service providers to send an account activation link to the registered email address to verify the legitimacy of the user. In any case, the online services have the email addresses associated with the online accounts of the users. This best practice illustrates the use of these associated email addresses to recover the account's password. By doing this half of the burden of security is shed off to the email service provider, as it is the email service provider's responsibility to secure the accounts of their users to protect them from the attackers hacking into the accounts. Since almost every online service have email addresses of their users and uses it as the medium to authenticate the users, it is crucial for the email service providers to use proper mechanisms to secure the accounts of their users because if the email accounts get compromised then almost every online service account associated with that email address will be compromised. Because majority of online service providers send a password reset link to the associated email address, and it is not a hard task to find out the valid email addresses associated or having an account with the corresponding online service provider. Once the email account is compromised then almost all the online accounts of the victim with that email address get compromised. Hence it is very crucial for the

email service providers to have stringent security procedures to ensure the security of their accounts.

- **Implementation:** This best practice is implemented at the time of account recovery by the users. When user requests a password reset then after verifying the user with 2FA, the online service provider should send an email including the password reset instructions to the user's registered email address with the online service provider. This email might contain the password reset URL or some instructions how to reset the password, either ways the email should not reveal any kind of account information of the user or any clear text passwords or security code, these cases will be discussed in next best practices of this category in detail. After following the steps in the recovery mail, user should be allowed to change the password of the account.
- **Example:** This best practice is followed by majority of social media providers, e-commerce services and cloud services, who rely on the email service provider for the security of their users' accounts. Currently this best practice is implemented without any 2 Factor Authentication of the user and that means if the attacker has the valid email addresses of the online service users then it can request a password reset and the password reset email will be sent to the legitimate user. In this way the attacker can spam the legitimate users with a flood of password reset emails from the online service provider and might attack the users by phishing emails that appears to be from the online service provider and can fool the users to provide their credentials if they want to regain the access of their account. Or the

attacker might DOS the legitimate user from requesting a password reset if it has exceeded the limit of password resets allowed per day by the online service. The following image was from the Password recovery page of an online service. It uses the email address used at the time of logging into the account:



**Password Assistance**

Enter the e-mail address associated with your account, then click Continue. We'll email you a link to a page where you can easily create a new password.

**Email address:**

**Figure 3.11:** Use of recovery email address for password reset

- **Attacks:** DOS, Phishing attack.

### 3.3.1.3 **Best Practice:** No Password Hint In Recovery Email

- **Definition:** The password recovery email should not contain the old password, the new password or the password hint in clear text.
- **Discussion:** This best practice illustrates that the online services should not provide any kind of password related information within the email sent for the password recovery to the users. This is because if the attackers has any way get the access to the email accounts of the users then they should not get any kind of hint about the passwords the user usually uses for the online accounts (cloud or social media account). Because people usually use the same passwords for logging into the other online accounts as well



and any kind of hint or the old passwords can help attackers to predict and brute force the other online accounts of the user as well. Also if the attacker is snooping on the wire and the password hint is sent in the clear text then even though the communication is encrypted with SSL/TLS but majority of times the emails are sent unencrypted between different email servers and anybody snooping on the wire can actually get the password or password hint in clear text. Hence to be safe, online service providers should not send password related information such as old passwords, new passwords or password hints in clear text in emails.

- **Implementation:** The implementation of this best practice is pretty much similar to the implementation of Password Recovery->Use Of Recovery Email Address the only difference is while sending the recovery email to the user, after verifying the user, it should not contain old passwords, new passwords or any password hint in clear text.
- **Example:** As discussed, that many times the passwords of email accounts get leaked [7], and if the user has not changed the password after password leak and if the attacker is still be able to get into the account then it causes a chain reaction, in which the attacker might go through the recovery emails, which might be sent by the online services the user is using, and get the crucial information such as the password hint or old passwords which can make the attacker's task easy to hack into other online accounts of that user, even though the user is no longer using that email address for the password recovery of other online accounts. The attacker can predict and make combinations from the old passwords to crack the current passwords

of the online service accounts of the users. And in this way the attacker could be successful in compromising the other online accounts and DOSing the user from using those accounts by resetting the passwords of the accounts. The following image is a from an email sent to the user for resetting the password of its online service account. The email doesn't have any password reset code or hint, it just have a URL which does not reveal any kind of information regarding password hint or user account. On the contrary, in the next following image, the online service sends a clear text numeric code in the recovery email, which is not considered as a good practice:

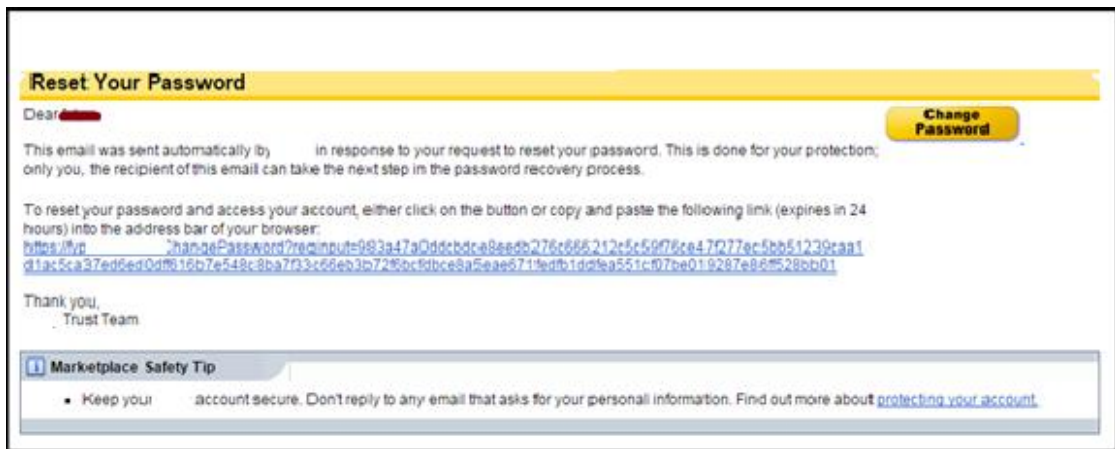


Figure 3.12: No password reset hint in email



**Figure 3.13:** Password reset code in clear text in email

- **Attacks:** Brute Force, DOS

#### 3.3.1.4 **Best Practice:** No Account Info In Recovery Email

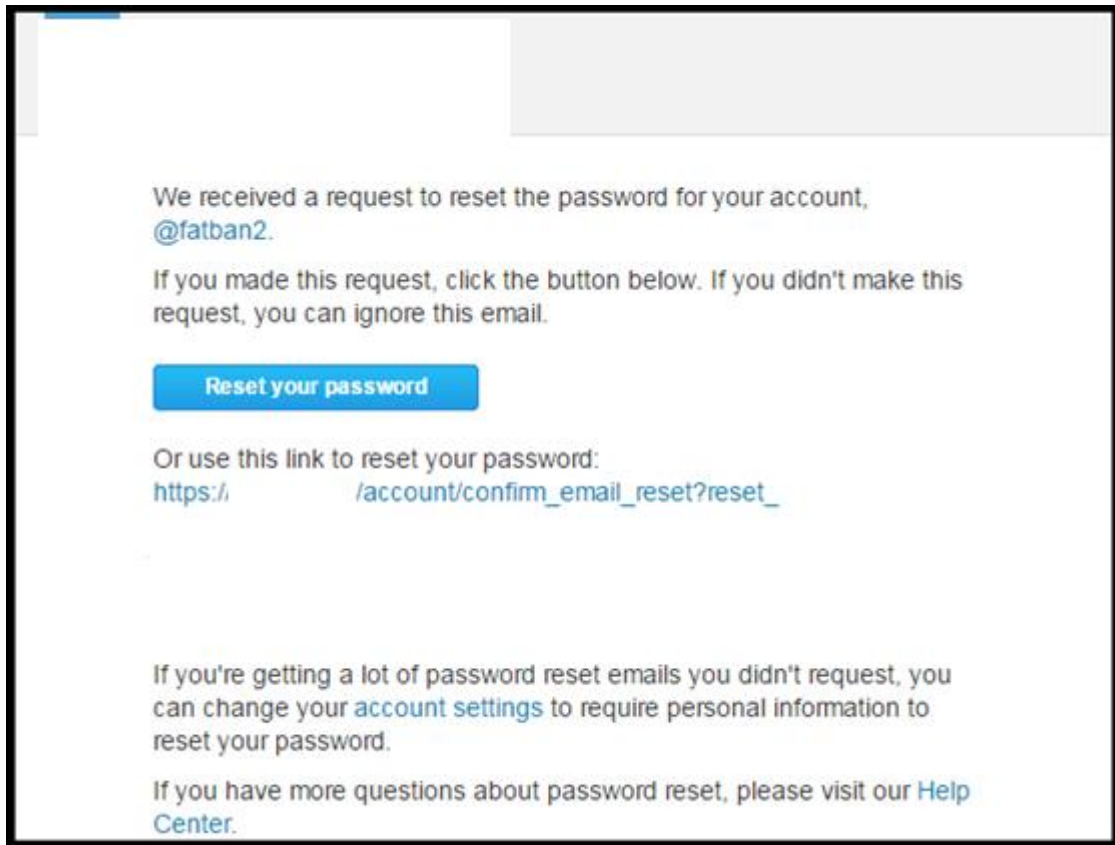
- **Definition:** The password recovery email should not contain any part of account information.
- **Discussion:** It is seen that majority of online service providers have a general mechanism of sending a recovery email to the registered email address (either the username or email associated with the account) which includes the information regarding password reset. This best practice illustrates that the recovery email should not reveal any part of account information, especially username or the part of username or any other information related to the account owner. The reason being pretty much the same as discussed in ‘No Password Hint In Recovery Email’ of this category. It is important because if the account gets compromised, this

information is directly available to the attacker and the most important thing is that people have same usernames or account information provided to the particular account as they used in other accounts because in this day and age single person have at least more than five accounts and it is difficult to remember five different sets of information for every account so people usually consider to have same or similar information shared between their accounts hence easy to remember. Therefore if some information of one account gets revealed, it then makes other accounts on stake as well. Hence it is important that online service providers should consider not sending any account information in the recovery email. Also it does not make users task difficult to guess which account's recovery email is this? because firstly if user have requested a password reset then he will be expecting an email from the service provider having password reset steps, also the users don't request password resets for many accounts simultaneously which can confuse them that which email belong to which service and secondly the service providers emails have service name mentioned either in the email address or in the subject line which makes it pretty clear that which service's recovery email it is.

- **Implementation:** As discussed in 'No Password Hint In Recovery Email' best practice's implementation, this best practice is also implemented in pretty much the same way. When user have forgotten the password and asks for the password reset, the service provider sends an email having either the steps of password reset or an URL for resetting the password. Either ways the email sent by the service provider should not contain any

username information or the account information in the email. When user gets the recovery email, then after changing the password, it is recommended to delete the email hence the account information should not get into the wrong hands.

- **Example:** In year 2014, 5 million google accounts usernames and passwords got leaked and posted on the Internet [7]. Even though the passwords were pretty old but if somebody might not have a new password and his username is on the list, and likely that the attacker can verify the email addresses with the passwords, then he might come across an active account and can steal as much information if he can get to find out the other accounts details and might try to exploit them as well. The following image is taken from the email sent by online service to the user. It clearly shows the user's account information in the email "@fatban2". Also it doesn't give the information about the time out of the URL. This is not considered as a best practice:



**Figure 3.14:** Account information in recovery email

- **Attacks:** Brute force

### 3.3.1.5 **Best Practice:** No Personal Security Questions For Recovery

- **Definition:** The password recovery should not be based on the personal security questions only.
- **Discussion:** As the definition suggests that the passwords should not be recovered just by answering some personal security questions. This best practice is very important and crucial to use, the major disadvantages of using security questions as the recovery mechanism are:

1. Only 10-12 security questions are implemented by the online service providers, and from them at least 4-5 should be answered correctly to get the password reset. Hence the combination set is limited and small.
2. The questions asked for the recovery are fixed every time user wants to reset the password, so once the attacker have questions, he can make a list of all probable answers.
3. It is not too difficult to get the security questions correct, because user answer these questions genuinely, as it helps them to remember the answers for the questions, and almost all the personal data is available on Internet (on social media sites).

Above brief points can give an outline of why security questions are considered insecure along while doing password recovery. Online services don't have a standalone security questions mechanism for password recovery, but they consider it as a worst case scenario, i.e. if user is not able to access the registered email account for some reason, or no longer have access to the registered phone number, and couldn't remember the password, then in that situation the service provide recovery by asking few questions related to the account, for example: "last time of accessing the account", "recent emails addresses used" or "when was the account created roughly", again these questions are not too hard to get through and besides some services directly ask the security questions selected and answered by the user at the time of account creation. In all the scenarios, it is not too difficult to hack into the account. Hence it is important that if an online

service still wants to use security questions as the “worst case recovery mechanism” then it is better to back it up by using another layer of security (might be 2FA) and if the user is unable to get through the another layer of security then they should have an only option of contacting the security team of that particular service for the account recovery.

- **Implementation:** This best practice is implemented at the time of account recovery, when user find no means of recovering their lost passwords, then the security questions come into picture. The password reset should not be allowed only on the basis of answering of the security questions, rather the service should use 2FA to authenticate the user, by using mobile phone or by using alternate email address, and if user is unable authenticate itself then the only option should be to contact the customer service- the security team- of that service provider to seek help.
- **Example:** The example of the breach happened by using security questions mechanism to recover the passwords of the accounts is explained in Password Recovery->2FA For Password Reset->Example. This example shows that it is not a hard task for the attackers to get the personal information, likes-dislikes and other relevant family information of famous people, who leaves the footprints all over the Internet. The following is an image taken from the page of password reset of an online service account, it asks for the basic account setup questions and if provided correctly then an email has been sent to the mentioned email address by the user at the time of account recovery. Not considered a good practice:



**Recover your account**

Help us to keep this account secure by verifying that it's yours. To get back into your account, enter as much info as you can.

**Account info**

First Name

Last Name

Birth date

Country/region

State

ZIP code

Other passwords you've used for this account (one per text box)

**Figure 3.15:** Personal security questions while password recovery

- **Attacks:** DOS and Phishing attack

**3.3.1.6 Best Practice:** No Random emailId/Phone For Recovery

- **Definition:** The password recovery should not be based on any random email address or phone number provided promptly at the time of recovery.
- **Discussion:** This best practice clearly states that there should not be any password recovery based on the email address/phone number provided at the time of account recovery. A few online services use this as their worst case situation, where user have forgotten the password, don't have access to the email account, and does not now other information of the account,

then online service allows to provide an alternate email address, which is not registered on their database and sends email for the further communication regarding account recovery. Even though they don't provide any steps of password recovery right away but uses this latest email address which is provided by the user at the time of recovery for communicating with the user/attacker. If attacker manages to convince the online service provider that he is the legitimate user who is locked out of the account then it will make the legitimate user's account vulnerable from being compromised. Hence, if ever the user is locked out of the account, the online service provider should not give any alternative of providing the email address in current use and ask the user to directly contact the security team of the service.

- **Implementation:** As many other best practices in this category, this best practice should be also implemented at the time of password recovery and its implementation is pretty much same as the one mentioned in Password Recovery->No Personal Security Questions For Recovery->Implementation. Instead in place of security questions, the account should not be able to get recovered using an email address provided at the time of recovery.
- **Example:** Many times people might have set up an email account decades ago, which they no longer use but have that email address as the account Id or username of other online services. In that situation whenever the user wants to reset the password, it faces the problem because he might no longer remember or use the email account provided as the username of that

service. In that situation they might want to use an alternate email address to recover the password, but the problem is that alternate email is not registered with the online service, so online service gives an option to provide an alternate email address at the time of recovery. It works if the user is legitimate, but if the user is not the authorized one that it might result in the account getting compromised and exploited by the attacker. Following is the image taken at the time of account recovery. Note that the contact email in the image is different from the one which is given at the creation of account, that means it is a random email address entered at the time of recovery when the user has lost the access to its actual registered account:

Recover your account

What account are you trying to get back into?

Email address

Note: If you've turned on two-step verification, you can't recover your account this way.

Where should we contact you?

Enter an email address that's different from the one you're trying to recover.

Contact email address

Enter an email address that's different from the one you're trying to recover. If you don't have another email address, [create a new one with](#)

Next

**Figure 3.16:** Random email address for recovering the password

- **Attacks:** DOS

### 3.3.1.7 **Best Practice:** Timeout The Recovery URL

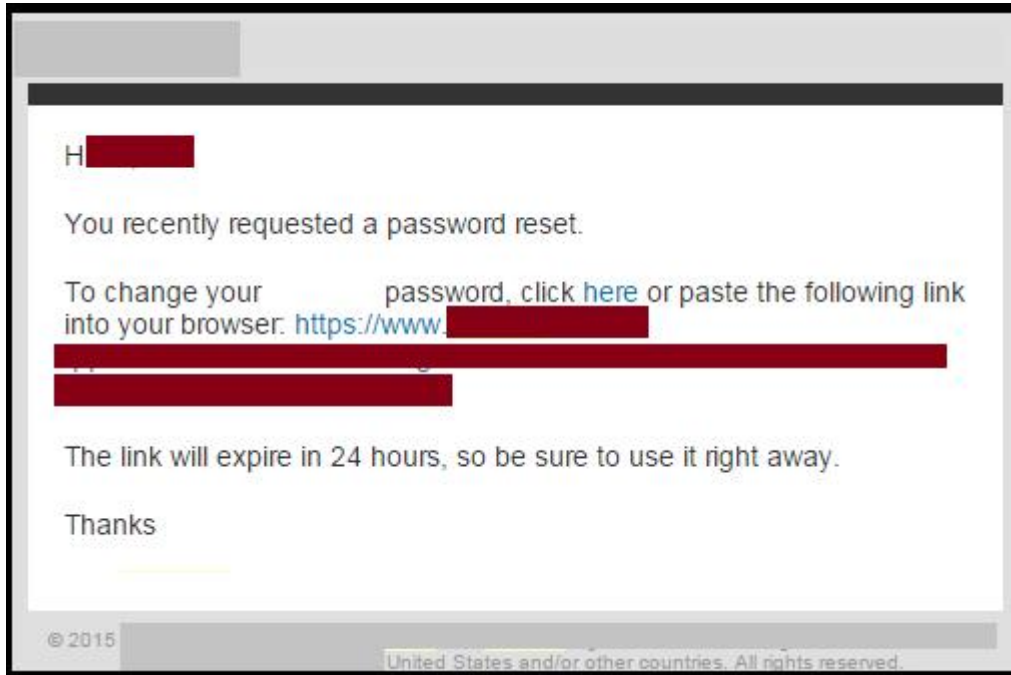
- **Definition:** The URL or the recovery steps sent to the user should timeout and should be invalidated after the timeout.
- **Discussion:** It is the most common practice to send the URL, steps to reset the password, or some kind of numeric/alphanumeric code in the recovery email by the online service provider to allow user to reset the password. It is pretty clear from this best practice that this email should not be valid forever because of some obvious reasons. Many reasons are already

covered in the previous best practices, such as when the email account gets compromised and if user is using that email account on other online services as well, then the other online services where this account is used becomes vulnerable to attack. And if there are password recovery emails resting in the inbox of the email account, then it even saves more time of the attacker to look and research for the online services which uses this email account. More importantly if the password reset emails were still active then the attacker can directly change the passwords and perform a Denial of Service from all the online services the user was using with that email address. This is the worst case which could happen to the user. Hence it is important that whenever the user requests the password reset for the online service accounts then, the service provider should make sure to time out the password reset process after certain amount of time from the time when email was sent. One common practice seen in online services is they typically have 1 hour as the time after which the password recovery email expires. This is fair enough time to allow user to reset the password and if for some reason user is unable to reset the password in the given amount of time then he can request the password reset again. Also it is highly unlikely that an attacker hacks into the account when the password recovery email is still valid and can use it to reset the account's password.

- **Implementation:** This best practice is implemented at the time when user requests for the password reset. Whenever the user requests the password reset an email is sent to users registered email address. The service provider should make sure to timeout the email after a particular amount of

time may be after an hour or thirty minutes from the time the email was sent. It should be mentioned in email that the password reset URL or code will expire in next 30 or 60 minutes so that the user should change the password before the email expires. Also it is recommended that after resetting the password, the user should delete the email forever (i.e. from the inbox as well as from the deleted messages list) to secure the online service account.

- **Example:** Many times the email accounts get compromised and information gets leaked, that makes all the other accounts linked with that email address vulnerable to attack. And most importantly if the online service does not timeout the recovery email and if in case the user have ever requested for one, then it allows the attacker to have easy access to the online service by resetting the password again and locking out the legitimate user from that online service. The example for this best practice is similar to the one illustrated in Password Recovery->No Password Hint In Recovery Email. The following image is taken from the email sent to the user. It clearly shows that the URL is only valid for 24 hours. Most of the URLs are timed out after certain period of time. Considered as one of the best practices:



**Figure 3.17:** Recovery URL time-out (24 hours)

- **Attacks:** DOS

### 3.3.1.8 **Best Practice:** No Part Of Email/Phone Shown While Recovery

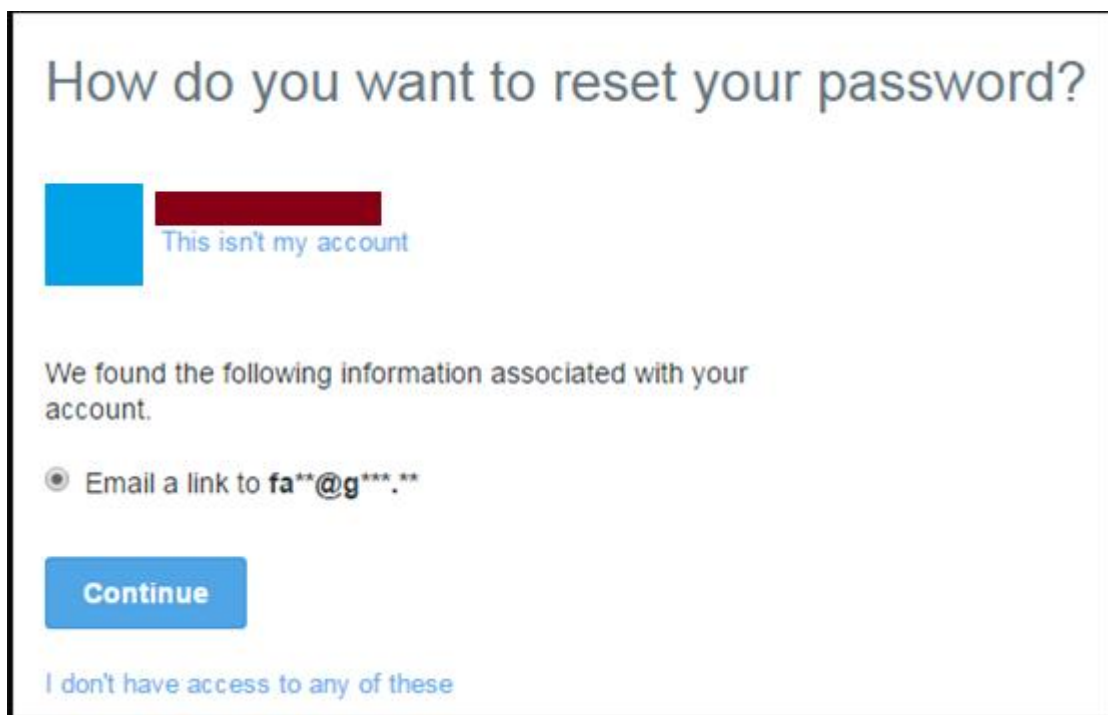
- **Definition:** While the password recovery process, no part of recovery email/phone number should be revealed.
- **Discussion:** According to this best practice, while recovering the password for an account, the service should not reveal any part of email address or phone number which will be used to send the recovery email or code. Generally while walking through the steps of password recovery, online service shows a part of email address, which user has registered with the service, for example “jon\*\*\*\*\*@gmail.com” or something of similar sort, if the user has registered the phone number then it shows like: “302-123-

\*\*\*\*\*” which is considered a bad practice. It will give a hint of the email address/phone number of the user and if attacker knows the username then it will not be hard to find out the complete email address or phone number of the user. At the first glance it might look small and trivial, but these small things make a big difference. It is said in security industry is 100% security is a myth, but we can make systems and services hard enough to break into only if we consider each and every loop hole, no matter how small it is. Same is true with online services as well; they should minimize the risk of attack by reducing as much leak of information from their services as they could.

- **Implementation:** This best practice is implemented when user wants to reset the password of the account. When user goes through the steps of whatever password recovery mechanism is implemented by the service, the service should make sure not to reveal any part of email address or phone number information associated with the user account. Online service should control on the amount of information available to the user while password recovery, so that it should not make online service accounts vulnerable to the attacks.
- **Example:** It is not under the control of the online service providers that how attacker gathers relevant information related to the online accounts of a particular online service before attacking it. But what definitely can be controlled by the service provider, is what information it makes available to the public, especially at the time of account login and password recovery. Online services should pay great deal of attention to control the



leakage of information from their side to protect the users account from getting hacked. The following image is taken from the screen that appears when the user requests the password reset for its online account. It clearly illustrates that the online service is not using the above mentioned best practice, revealing the part of email account user has registered with the service, which is used for password recovery:



**Figure 3.18:** Part of recovery email address shown

- **Attacks:** DOS and Brute force

### 3.3.1.9 **Best Practice:** Password Reset Notification

- **Definition:** The online service should send user a notification every time a password reset takes place on the account.
- **Discussion:** This is one of the most important best practices which need to be implemented to secure the users accounts. According to the best practice, the user should be notified whenever the password for their online accounts changes or resets. This is important because if the user is not the one who has requested the password reset or changed the password of the account, then the user can take immediate actions to retake the control of the account, or immediately contact the security team of the online service provider so that it should suspend all the activities occurring on the account. Also notification helps the user to keep track of the activities happening on account and if the user has not initiated any of the actions for which it is getting notifications then immediate actions can be taken to lessen the impact of the breach. Unfortunately majority of online service providers don't have notification service active to notify user about the password change on the account. Because of this, neither the user nor the online service provider knows that there was a breach on the account, until the attacker reveals, or posts the data or account specific data over Internet. When the prevention measures are taken, it is already become too late to recover from the breach.
- **Implementation:** This best practice is implemented when the user's requests for the password reset, a new email is sent to their accounts, or whatever password reset mechanism is offered by the service. Whenever a password reset happens on the user's account, then the online service

should send an acknowledge message to the registered email address or the phone number of the user notifying about the password change occurred on the account with the date and time of the event. Also, for more precise identification of the computer from where the password change happened, the online service could provide the IP address and the location of that computer as well.

- **Example:** There are many examples of this kind of attacks where the attack take over the account without intimating the user that its account got already hacked. One of those kinds of attack is Apple's iCloud breach where the attackers took over the accounts by resetting the password without the users having any information of that. It has already have been discussed in Password Recovery->2FA For Password Reset, where celebrities accounts got hacked and no notifications were sent to the users.
- **Attacks:** DOS

#### 3.3.1.10 Best Practice: Recovery EmailID/Phone Change Notification

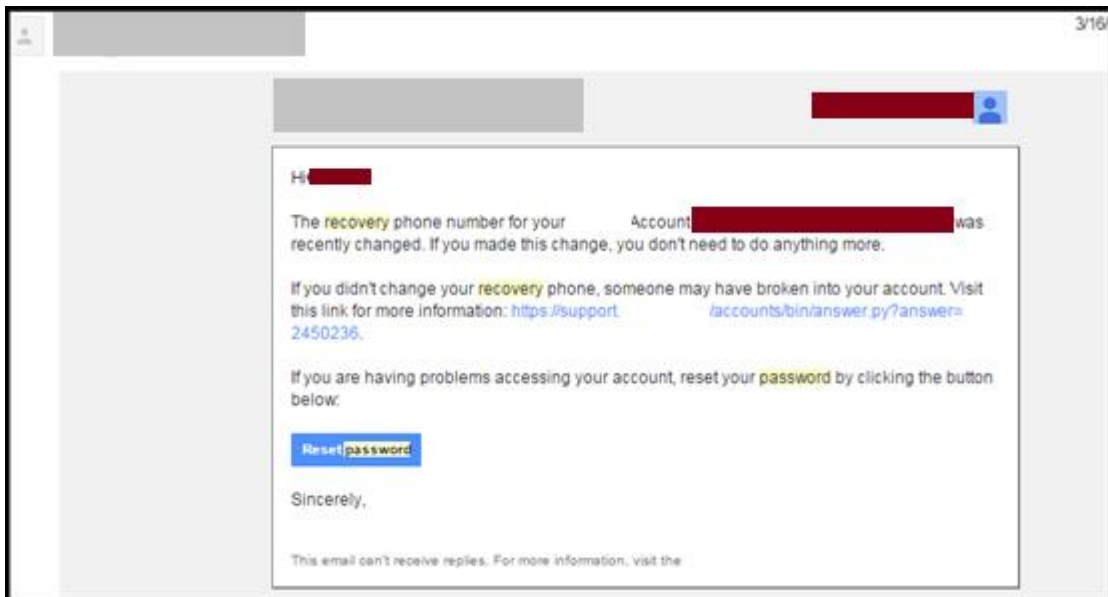
- **Definition:** Notify the user whenever the recovery emails address/phone number changes.
- **Discussion:** This is also another type of notification that an online service provider should consider to send the users whenever the information like password recovery email address or phone number changes. As discussed previously that many online services don't have email address as their username, instead they have the general alphabetic or alphanumeric strings as their usernames and saves the email addresses/phone numbers of the users for the password recovery mechanism. It happens many time that

user doesn't remember which email address or phone number they have provided to the online service at the time of account creation and generally asks for the new email address or phone number for sending the recovery email or code at the time of account recovery. Since many online service providers do provide this type of service, where user can change the email address or phone number for the password recovery (although not considered as a best practice, see Password Recovery->No Random emailId/contact For Recovery), it becomes important for the online service provider to notify user whenever such type of change happen, because as discussed in Password Recovery->No Random emailId/contact For Recovery best practice the disadvantage of using such practice, the account becomes vulnerable to attacks. Hence if the online service providers practice such type of password recovery mechanism where user can change the email address/phone number for password recovery, then the online service should send an email notification to the registered email address so that if an attacker is trying to reset the password then the user should be able to access the registered email account and should get the notification of email/phone change done by someone.

- **Implementation:** This best practice is implemented for the online services that use email addresses or phone numbers for the password recovery. Whenever user changes any of those, either at the time of password recovery or in general when changing the settings of the account while logged in, the user should get a notification of the changes no the registered

email address or phone number. So that if the user hasn't made those changes then immediate prevention actions can be taken by the user.

- **Example:** Many services allow users to change the recovery email address or phone number at the time of password recovery. This best practice should be implemented to make sure that user is notified of any changes in the password recovery method to maintain the security, integrity and privacy of the account. The following image was a from the email sent for the change in the recovery phone number notification from online service provider:



**Figure 3.19:** Notification for change in recovery phone number

- **Attacks:** DOS

### 3.4 Category: Service Authenticity

#### 3.4.1 Applicable To:

Especially for the services who have their usernames other than the general first and last name combination of users (in other words, services whose usernames are hard to guess and generally not available to public):

- Other commercial online service providers, especially dealing with money transactions.
- Online banking service

##### 3.4.1.1 Best Practice: User Personal Security Image and Caption

- **Definition:** Mandatory use of Personal Security Image and Caption for users accounts to verify the service's authenticity.
- **Discussion:** This best practice verifies the authenticity of the online service providers. It is important to use personal security image and caption so that the users should not fall prey to the phishing attacks, in which the user is fooled by the convincing and legitimate look and feel of the web page created by the attacker to steal the credentials of the users. If the online service has set up the personal security image and caption corresponding to every user account then, generally at the time of login, the caption and image are sent back to the user from the service corresponding to the valid username entered by the user. This image and the caption makes the user to make sure that it came from the legitimate service as they think it is, because if the attacker was trying to steal credentials, then he would not be able to get the image and caption corresponding to the user and user after getting the security image and caption can verify that the service provider

is legitimate or not. And only when the user verifies the security image and caption, he provides the password and logs in. This saves the users to provide the credentials to the attacker trying to fake the service provider. Hence this best practice is used to verify the authenticity of the service provider and prevents the users being falling prey to phishing attacks.

- **Implementation:** To implement this best practice, online service has to make sure that the usernames for the online accounts should not be available easily to the public via any source or applicable to the online services that do not use the common username for the login purpose, especially banking institutions. This best practice is implemented at the time of login. The login page should first ask for the username, after user has entered the correct username, a personal security image and caption is pulled corresponding to the user account and shown to the user with the password field. Only after user verifies the personal caption and image associated to its account by checking the check box, the login button should get enabled and user should be allowed to login with correct password provided. If the security image and caption does not match the security image and caption user have corresponding to the account, then user can back off from providing password and hence will be saved from any kind of phishing attack hacker might have planned.
- **Example:** Many banks implement this best practice to make sure that user's credentials are secured and users should get alarmed if some site claiming their bank asking for username and password does not show the correct personal security image and caption. This best practice works for

the banks because the usernames on bank accounts are pretty uncommon and rigid, also they make sure that there is no way to guess the correct username of an account because after 3-4 tries the account gets locked out, hence attacker can't guess the username by such few tries, also the process becomes harder because of rigid rules on the usernames imposed by the banks.

- **Attack:** Phishing Attack

### **3.5 Category: Suggested New Best Practices**

#### **3.5.1 Applicable To:**

All online services who have a login account for their users to access their service, for example:

- e-mail providers
- social media service providers
- e-commerce websites
- online banking services
- online money transaction services
- cloud storage services
- other online service providers

##### **3.5.1.1 Best Practice: Uncommon Username For Login**

- **Definition:** The usernames used for the online accounts should be uncommon or should be hard to guess.
- **Discussion:** This best practice suggests that online service providers should have mechanisms to impose some specifications for the username



selection, just similar to the password selection, also giving users an option to use their common usernames to be displayed on their account whenever any authorized friend or person looks or visits the online account (in case of social online service providers). In this way online service obscure the username used for the login purpose, just like passwords are never exposed to others who look or visit the online accounts. This would be a great disadvantage at the part of attacker, because now he has to guess the correct username as well as the correct password, since now the usernames are not available and it is pretty hard to guess the usernames because of the username selection mechanism. Hence doubling the task of the attacker for both brute force attack as well as DOS attack.

- **Implementation:** The implementation of this best practice should be done in same way as done for the implementation of a secure password selection mechanism. The only difference should be that there should not be many restrictions on username such as all the subcategories of Password Selection->Minimum Password Requirements. The restrictions should be sufficient enough to give some randomness to username so that even the attacker knows the full name of victim; it should be hard for him to guess the username used for the login to victim's account. For example, Number Inclusion and Upper & Lowercase Letter Inclusion should suffice. The user should be given freedom to choose a common name to tie up with the account and use for display purposes on the account, but that name should be different than the one used for the login. This will slow down the attacker for having a bulk brute force attack on the accounts of an online

service, because now he has to do the same exercise to guess the correct usernames of the accounts as he does for guessing the passwords.

- **Example:** The very first step to perform a brute force attack on the online service provider is to build the list of legitimate usernames using that account and after that the next step is to guess the password for every username in the list. Currently the first step is considered as not much time consuming and easy to do because majority of the online services either use email addresses of the users as their login username, or uses the first name, last name or combination of both as the username. Both of this information is easily available on Internet. And to make a list of usernames which are the combination of first and last name, attacker can take top 1000 common first names and common last names and can make all the random combinations like johnsmith, andy, etc. to make the list of the legitimate usernames for the online service. But this process can be made difficult if usernames are now harder to guess and to guess a single username, it requires all the combinations of the restrictions imposed on the username (uppercase lowercase and numbers) and this could consume a lot of time of attacker and make the attacks like brute force, DOS harder on hacker's side.
- **Attack:** Brute Force, DOS.

#### **3.5.1.2 Best Practice:** Mandatory Upgrade Of Password After New Password Policy

- **Definition:** When an online service makes any changes in the password policy, users should be forced to comply with the new policy by changing their current passwords.

- **Discussion:** This is very important aspect, which should be taken into consideration while online services make any updates on their password creation policies for the online accounts accessing the service. In the past, when the cybercrime was not very prominent, the online services especially email service providers used to have very few or almost no password restrictions on the accounts of the users, hence users can have any password they want without having thought about security. Maybe at that point in time, it wasn't as crucial as it is today to have a strong passwords for the online accounts. These email services are still being used by the users, new as well as the old users who created their account almost a decade ago. Now they have upgraded their password selection policies and force users to have good passwords while creating the account and the user's accounts in the current day and age may be secure enough. But the question is what about the security of the accounts created over a decade ago? Do the old users have changed their passwords to comply with new password policies? It is difficult to answer these questions but it is easy to mitigate the cause of these questions, by forcing all the users, who have created their accounts before the new password policies were enforced, to change their passwords for better security of their accounts. This avoids the risk of the old accounts getting hacked by the attacker, brute forcing the accounts. Hence, the online services should make sure that whenever they upgrade to a more secure password selection mechanism, they should make sure that all the users should mandatorily change the password to comply with the new system of password selection.

- **Implementation:** This best practice should be implemented at the time when there are changes in the password policy of an online service provider. When an online service provider decides to upgrade its password selection mechanism, including some new rules imposed on the passwords for the new accounts, and then the service should make sure that all the account holders prior to the password policy upgrade should change/upgrade their passwords according to the new password policy. To make sure, user changes the password, the online service should give the users a timeline till which they should change the password and after that the users should not be allowed to login with their old passwords, when the service detects that the old password has not been changed for the particular user then the service should force the user to reset the password by following password reset mechanism of the online service provider. After successful reset the user then should be allowed to login.
- **Example:** There is couple of examples in past which signifies the importance of this best practice. One of the recent activities involving the hack and leakage of around 5 million Gmail passwords with the corresponding usernames is an example where they claimed that majority of the accounts were old accounts with their old passwords, and said that even though the usernames are legitimate usernames, the passwords are the old passwords and users have changed their passwords corresponding to their accounts [7]. But there might be some users who might not have changed their passwords since creation of their accounts and could have easily become vulnerable to the attack because Google did not have

mandatory upgrade of old accounts passwords after they changed their password policy. This shows that if this best practice could have implemented then no accounts could have been on risk, if the accounts were old enough.

- **Attack:** Brute Force.

### 3.5.1.3 **Best Practice:** Mandatory Password Hash Salting.

- **Definition:** Passwords should be secured with cryptographic hashing, using a salt at least as large as the number of bits used for hash.
- **Discussion:** Now days it is a common practice to encrypt the passwords with a hashing algorithm before storing them on server. Hashing, while secure, is vulnerable to pre-computation of large dictionaries of possible passwords. To prevent this, it is recommended to use a large salt, a random number generated for every password hash, for more stringent security, so that it becomes near impossible for the attacker to hash every potential password with every potential salt value. This best practice recommends the use of at least the number of bits of hash algorithm used as a salt with the hashing algorithm to encrypt the passwords because large salt increases the number of combinations for every password and hence it becomes time consuming for an attacker to produce all the combinations of the salt with the corresponding hash to crack the password. And the procedure is repeated for all the possible combinations of the passwords available to the attacker, or for the top 1000 common passwords. With the current state of the technology, it has become easy to compute the hashes but it is still difficult to compute the hashes salted with number of bits as large as the

hash. Hence we strongly recommend using salt of at least 64 bits in length while hashing the passwords.

- **Implementation:** This best practice is implemented at the server side of the best practice and hence, it is hard to know that which online service uses password hashes with salt and how long the salt is. This best practice makes sure that whenever an account is created for the online service, the passwords submitted by the users at the time of account creation are hashed with the salt bits as large as the hash and then stored on the server side of the online service provider.
- **Example:** Salting is important because it makes the task of pre-computing the passwords more difficult. Consider a situation, for example, that the attacker has access to a password file containing all the usernames and password hashes. Without a salt, attacker could take top common passwords hash them and then compare each of them with the hashes available in the password file. It is pretty likely to find a match if the passwords are easy to guess. But if a salt is added to the hash of the passwords then for every password, the attacker has to compute all the possible combinations of salt bits with the hashes to guess the right password. This makes the cracking task more time consuming. The length of the salt added to the hash is critical because if the salt is small enough, then it would not have a large impact on the cracking time and resources used, although it would definitely be better than no salt, but it doesn't solve the problem of easy password cracking. If the salt is large enough such as 64 bits then the possible combinations for the salt would be  $2^{64}$  which is

pretty huge number and it would be not considered worthy to compute all these combinations for a single password and especially when you don't know whether that password would produce a match or not. Hence it is strongly recommended to use large salts while hashing the passwords, at least as large as the hash.

- **Attack:** Brute Force.

#### 3.5.1.4 **Best Practice:** Periodic Password Reset

- **Definition:** Users mandatorily changing the passwords periodically
- **Discussion:** This best practice suggests that the online service providers should ask their users to change their passwords periodically, may be every six months or annually. This could be annoying to the users and may have a bad impact on the online service provider's market. But this is important and taken into consideration because due to advancement in the processing speed of the processors and ever growing technology, the passwords considered safe today might not be secure enough a year later. Hence password strength regulation is important and it makes the accounts secure by the fact that if an attacker is collecting a database of the passwords from years, they might not be effective at the current point in time, if the online services have a periodic password reset policy. Hence it is recommended that users should change their passwords periodically and to make sure it happens, online service provider should implement this best practice.
- **Implementation:** This best practice makes it mandatory that the users change their passwords after certain amount of time. Hence as soon as this best practice is implemented, after the first password reset, the online

service should remind the users couple of days ahead to reset the passwords after certain amount of time has passed (may be half yearly or annually. When the time comes for the password reset, i.e. one year after the previous password reset, the user should get warning message to reset the password whenever user logs into the account.

- **Example:** Many major industries implement this best practice to secure their employees login credentials, by annually or half yearly forcing all the employees of the organization to change their business account passwords so that they are secured and nobody can hack into organizations network with an old password. This best practice could be used in the same way with the public online service providers as well to make sure that the accounts should not get compromised, if the attacker is collecting the information for past years.
- **Attack:** Brute Force.



## **Chapter 4**

### **ONLINE SERVICE PROVIDERS – A SURVEY**

The list of best authentication practices summarized in Chapter 3 were gathered by documenting the various authentication practices and mechanisms used by a wide range of online services providers in the Spring of 2015. These providers included social networking, email, and e-commerce services. The survey revealed a significant range of different levels of care used to protect users' accounts. Some appear to be very sound and secure and others will be less effective against different forms of attack such as brute force password attacks and password recovery compromises. Discussions with at least one provider indicated that a provider may have one level of security for free accounts and implement more stringent methods for accounts in which the customer is paying a monthly fee. In section 4.1, several tables now follow detailing the various practices found to be in use. Section 4.2 is a discussion of the survey findings, and Section 4.3 discusses best practices for slowing down an attacker.

#### 4.1 The Comparison Table as of 1/20/15:

**Table 4.1:** Comparing online services with best practices: Password Selection

Category	Best Practices	CLOUDS						
		Google Drive	LiveDrive	Just Cloud	Mimedia	Dropbox	box	Microsoft oned
Account creation- PASSWORD SELECTION	Minimum length restriction.	Y(8)	Y(6)	Y(3)	Y(6)	Y(6)	Y(6)	Y(8)
	Special character restriction	N	N	N	N	N	N	N
	Number(0-9) restriction.	N	N	N	N	N	N	should contain at least two of these.
	Uppercase letter restriction.	N	N	N	N	N	N	N
	Lowercase letter restriction.	N	N	N	N	N	N	N
	Check for common strings.	Y	N	N	N	N	N	N
	email the account activation link and account should only be activated by clicking that link.	N	N	N	N	N	N	Y
	Password should not contain any part of username.	N	N	N	N	N	N	Y
	Is 2-FA provided by the service?	Y	N	N	N	N	N	Y
	Should use CAPTCHA while creating the account.	Y	N	N	N	N	N	Y
Passwords should be salted hash.	Don't Know, server side attribute.							
Strength bar showing strength of password.	N	N	N	N	Y	Y	N	

Category	Best Practices	CLOUDS							
		Amazon	Mega	Wuala	Tresorit	Mediafire	SpiderOak	Synplicity	SugarSync
Account creation-PASSWORD SELECTION	Minimum length restriction.	Y(6)	Y(5 maybe)	Y(6)	Y(8)	Y(6)	N	Y(8)	Y(6-24)
	Special character restriction	N	N	N	N	N	N	should contain at least one of these.	N
	Number(0-9) restriction.	N	N	N	Y	N	N		N
	Uppercase letter restriction.	N	N	N	Y	N	N	Y	N
	Lowercase letter restriction.	N	N	N	Y	N	N	Y	N
	Check for common strings.	N	Y	N	Y	N	N	N	N
	email the account activation link and account should only be activated by clicking that link.	N	Y	N	Y	N	N	Y	N
	Password should not contain any part of username.	N	N	N	N	N	N	N	N
	Is 2-FA provided by the service?	Y	Y	N	N		N	Y	N
	Should use CAPTCHA while creating the account.	N	N	N	N	N	N	N	N
Passwords should be salted hash.	Don't Know, server side attribute.								
Strength bar showing strength of password.	N	Y	N	N	Y	N	N	N	N

Category	Best Practices	Social Media						e-commerce	
		LinkedIn	Facebook	Yahoo	Gmail	Twitter	PayPal	eBay	
Account creation-PASSWORD SELECTION	Minimum length restriction.	Y(6)	Y(6)	Y(7)	Y(8)	Y(6)	Y(8)	Y(6)	Y(6)
	Special character restriction	N	N	N	N	N	N	should contain at least one of	should contain at least one of
	Number(0-9) restriction.	N	N	Y	N	N	N	should contain at least one of	should contain at least one of
	Uppercase letter restriction.	N	N	Y	N	N	N	N	N
	Lowercase letter restriction.	N	N	Y	N	N	N	N	N
	Check for common strings.	N	Y	N	Y	N	N	N	N
	email the account activation link and account should only be activated by clicking that link.	N	N	N	N	N	Y	N	N
	Password should not contain any part of username.	N	N	Y	N	N	N	N	N
	Is 2-FA provided by the service?	Y	Y	Y	Y	Y			N
	Should use CAPTCHA while creating the account.	N	N	N	Y	N	N	N	Y
Passwords should be salted hash.	Don't Know Server side attribute.								
Strength bar showing strength of password.	N	N	Y	N	N	N	N	N	Y

**Table 4.2:** Comparing online services with best practices: Bad P/w Attempts

Category	Best Practices	CLOUDS						
		Google Drive	LiveDrive	Just Cloud	Mimedia	Dropbox	box	Microsof onedrive
<b>BAD PASSWORD ATTEMPTS AND NOTIFICAT IONS</b>	Restrict maximum tries for password.	Y	N	N	Y	Y(15)	Y	Y
	Slow down by using CAPTCHA.	Y	N	N	Y	N	Y	N
	Suspend the account(time out) for some random amount of time.	N	N	N	N	Y	N	Y
	Should not reveal whether the username or the password is incorrect while encountered a bad login.	Y	Y	Y	Y	Y	Y	N
	should not have timing difference to load legitimate login versus bad login pages.	N	N	N	N	N	N	N
	Notify by email/text the owner of account about the bad password login attempts.	N	N	N	N	N	N	N

Category	Best Practices	CLOUDS							
		Amazon	Mega	Wuala	Tresorit	Mediafire	SpiderOak	Syneplicity	SugarSync
BAD PASSWORD ATTEMPTS AND NOTIFICAT IONS	Restrict maximum tries for password.	Y	N	N	Y	Y	N	Y	N
	Slow down by using CAPTCHA.	Y	N	N	N	N	N	N	N
	Suspend the account(time out) for some random amount of time.	N	N	N	Y (ONLY 10 MIN)	Y (ONLY 1 MIN)	N	Y(60 mins)	N
	Should not reveal whether the username or the password is incorrect while encountered a bad login.	Y	Y	Y	Y	Y	Y	Y	Y
	should not have timing difference to load legitimate login versus bad login pages.	N	N	N	N	N	N	N	N
	Notify by email/text the owner of account about the bad password login attempts.	N	N	N	N	N	N	Y	N

Category	Best Practices	Social Media					e-commerce	
		LinkedIn	Facebook	Yahoo	Gmail	Twitter	PayPal	eBay
<b>BAD PASSWORD ATTEMPTS AND NOTIFICAT IONS</b>	Restrict maximum tries for password.	Y	Y(19)	N	Y	Y(14)	Y	Y(25)
	Slow down by using CAPTCHA.	Y	N	N	Y	N	Y	Y
	Suspend the account(time out) for some random amount of time.	N	Y(60 mins)	N	N	Y(60 mins)	N	N
	Should not reveal whether the username or the password is incorrect while encountered a bad login.	N	Y	N	N(udel. edu)	Y	Y	Y
	should not have timing difference to load legitimate login versus bad login pages.	N	N	N	N	N	N	N
	Notify by email/text the owner of account about the bad password login attempts.	N	Y	N	N	N	N	N

**Table 4.3:** Comparing online services with best practices: Password Recovery

Category	Best Practices	CLOUDS						
		Google Drive	LiveDrive	Just Cloud	Mimedia	Dropbox	box	Microsoft onedrive
PASSWORD RECOVERY	<b>Mandatory</b> use of 2 FA so that the person can be verified at the time of recovery.	N	N	N	N	N	N	N
	Use of email address which owner of the account has provided while setting up the account.	Y	Y	Y	Y	Y	Y	Y
	The recovery email should not contain password/password hint in clear text.	Y		Y	Y	Y	Y	N
	recovery email/URL should not reveal any part of account info.Hash in the URL sent for recovery cant be reverse engineered(use of salt prevents that).	N						N/A
	Time out the recovery URL/code sent to owner.	Y		Y	Y	N	N	N



Category	Best Practices	CLOUDS							
		Amazon	Mega	Wuala	Tresorit	Mediafire	SpiderOak	Syncplicity	SugarSync
PASSWORD RECOVERY	<b>Mandatory</b> use of 2 FA so that the person can be verified at the time of recovery.	N	N	N	No Password recovery.	N	N	N	N
	Use of email address which owner of the account has provided while setting up the account.	Y	Y	Y		Y	Y	Y	Y
	The recovery email should not contain password/password hint in clear text.	Y	Y	N		Y	Y	Y	
	recovery email/URL should not reveal any part of account info.Hash in the URL sent for recovery cant be reverse engineered(use of salt prevents that).	Y	Y	N/A			N	Y	
	Time out the recovery URL/code sent to owner.	Y	N/A	N/A		Y	N	Y	

Category	Best Practices	Social Media					e-commerce	
		LinkedIn	Facebook	Yahoo	Gmail	Twitter	PayPal	eBay
PASSWORD RECOVERY	Mandatory use of 2 FA so that the person can be verified at the time of recovery.	N	N	N	N	N	N	N
	Use of email address which owner of the account has provided while setting up the account.	Y	Y	N	Y	Y	N	Y
	The recovery email should not contain password/password hint in clear text.	Y	N	N/A	Y	Y	N/A	Y
	recovery email/URL should not reveal any part of account info.Hash in the URL sent for recovery cant be reverse engineered(use of salt prevents that).	N	N	N/A	N	N	N/A	N
	Time out the recovery URL/code sent to owner.	Y	N	N/A	Y	N	N/A	Y

Category	Best Practices	CLOUDS						
		Google Drive	LiveDrive	Just Cloud	Mimedia	Dropbox	ibox	Microsoft onedrive
PASSWORD RECOVERY	Recovery should not be on the basis of personal information.	N	Y	Y	Y	Y	Y	N
	Recovery should not be on the basis of any random contact email address/phone no. provided at the time of recovery.	N	Y	Y	Y	Y	Y	N
	Should not reveal any part of information while recovery, ie, part of phone no. or part of recovery email address.	N	Y	Y	Y	Y	Y	N
	Notify by text/email when password reset happens.	Y		N	N	N	Y	Y
	Notify by email/text when the recovery email address/phone no. has been changed.	Y	N/A	N/A	N/A	N/A	N/A	N

Category	Best Practices	CLOUDS							
		Amazon	Mega	Wuala	Tresorit	Mediafire	SpiderOak	Synapticity	SugarSync
PASSWORD RECOVERY	Recovery should not be on the basis of personal information.	Y	Y	Y	No Password recovery.	Y	Y	Y	Y
	Recovery should not be on the basis of any random contact email address/phone no. provided at the time of recovery.	Y	Y	Y		Y	Y	Y	Y
PASSWORD RECOVERY	Should not reveal any part of information while recovery, ie, part of phone no. or part of recovery email address.	Y	Y	Y	No Password recovery.	Y	Y	Y	Y
	Notify by text/email when password reset happens.	Y	N/A	N		N/A	N	N/A	N
PASSWORD RECOVERY	Notify by email/text when the recovery email address/phone no. has been changed.	N/A	N/A	N/A	No Password recovery.	N/A	N/A	N/A	N/A
		N/A	N/A	N/A		N/A	N/A	N/A	N/A

Category	Best Practices	Social Media					e-commerce	
		LinkedIn	Facebook	Yahoo	Gmail	Twitter	PayPal	eBay
PASSWORD RECOVERY	Recovery should not be on the basis of personal information.	Y	Y	Y	N	N	Y	Y
	Recovery should not be on the basis of any random contact email address/phone no. provided at the time of recovery.	Y	Y	N	N	N	Y	Y
	Should not reveal any part of information while recovery, ie, part of phone no. or part of recovery email address.	Y	N	N	N	N	N	N
	Notify by text/email when password reset happens.	Y	Y	N	Y	Y	Y	Y
	Notify by email/text when the recovery email address/phone no. has been changed.	N/A	N/A	N	Y	N	N	N/A

## **4.2 Discussion on findings:**

The survey was done on major cloud service providers, social media providers and e-commerce service providers. The following sub-sections provides a discussion on the findings from the survey and tradeoffs of using particular best practices by the service providers.

### **4.2.1 Online Cloud service providers:**

The survey includes the analysis of the implementation of online user authentication in the major cloud service providers. The categorized descriptions of the results found are discussed below:

- **Password Selection:** The best practices described in this category in section 3.1.1.1 should be incorporated for the best password selection mechanism for the online service providers. This tremendously reduces the risk of brute force attack, provided all the best practices in this category are implemented. When surveyed some major cloud storage service providers, the results varied in a wide range, having services, like SpiderOak, implementing none of the best practices of password selection, to the services like Tresorit and Syncplicity, implementing almost all the mentioned best practices for password selection in section 3.1.1.1. It is surprising that the most popular and widely used cloud services such as Google Drive, Amazon, Dropbox and Mega don't have any password restrictions, except for the minimum length, which means users can have easy passwords and hence could be easily cracked by the brute force attack. That could mean that they don't want to impose the difficulty on their customers, trying to create a new account, because simple passwords are easy to remember. Also nobody wants to spend half an hour deciding and creating a password which they will definitively forget if

they don't access the service frequently. Because of fear of losing potential customers, the services don't force user to have strict passwords. But now people are aware of risk of having simple and weak passwords because of password hacking problem faced by the online services. So, services should try to make the users passwords hard enough so that they couldn't be easily guessable. It can be seen from the results that besides Tresorit, Syncplicity and Microsoft, no other cloud service has any kind of restriction on password selection, except minimum length. If the mechanism of restricting bad login attempts is not properly chosen and implemented, these services might be on the risk of brute force attack on their accounts.

- **Bad login attempts:** There are two major implementations seen for restricting the user to try too many bad passwords at the time of login: slowing down using CAPTCHA and a login timeout on the user. It can be seen by the results of the survey that a majority of online cloud service providers (6) uses CAPTCHA as primary step for the protection against bad login attempts and 5 other providers use a timeout mechanism for slowing an attacker down. Mega, Wuala, SugarSync and SpiderOak don't use any of the mechanisms to slow down the user, hence keeping them at high risk of brute force attack and except Mega, none of them have rigid password selection rules, which may exclude Mega from being brute forced, but no mechanism for protection from bad logins will keep it under the attack zone as well. Also, considering the bad login notification, no one except Syncplicity notifies the user about bad login attempts, which makes Syncplicity stand out of the crowd with majority of the best practices implemented by the service, with the only concern being a DOS attack because Syncplicity implements a timeout

mechanism to slow down the user and if not properly implemented, can allow the target user's account to be DOS'ed, making the account unavailable to the user.

- **Password Recovery:** For password recovery, all the cloud service providers surveyed use the email account used as the account login username, except Tresorit which offers no password reset.

#### 4.2.2 Social Media service providers:

The survey was done on the major social media providers such as social networking sites like Facebook, Twitter etc. for their implementation of online user authentication and following is the description of results found from survey for each category:

- **Password Selection:** The survey shows that the password selection mechanism for social networking services such as Facebook, Twitter, and LinkedIn is directly proportional to the ease of remembering the passwords for the accounts, that's why these services don't impose rigid password selection rules while creating the accounts. The reasons for doing this are pretty straight forward, these services are used very frequently by the users, maybe daily or even many time a day, hence the services don't want to annoy the users by requiring hard passwords which are most likely to be forgotten by the users, so that users can easily login and access their accounts whenever they want. The business success of these services is directly proportional to the number of users of the service; therefore they don't want to give their new users a hard time while creating the account by asking to include all the randomness in their password. Hence easy passwords are preferred so that the services should not lose their customers which will have direct impact on their business. Also, as seen from the results the minimum length for password is 6



characters, hence from the security perspective, this could result in brute force attack on the user accounts, if proper security mechanism is not employed for protecting the accounts.

On the other hand, the email service providers, Yahoo and Google, have a good password selection mechanism as compared to those of social networking services; the minimum lengths for the password are 7 and 8 characters, with Yahoo implementing rigid rules for selecting the passwords. It should be noted that none of the social media providers uses CAPTCHA at the time of account creation, except Gmail, which means that an attacker can create thousands or millions of accounts thus consuming provider service resources.

- **Bad login attempts:** As seen from the results, Yahoo doesn't implement any of the best practices for slowing or restricting the user, who is attempting to login with bad passwords. The reason is pretty obvious, as discussed above that Yahoo has rigid password selection policies and does not allow users to have weak passwords hence it is difficult to build a list of all combinations of the passwords to brute force the account (although not impractical, and may be in future when the processors get better, this can be easily done, hence vulnerable to brute force). They could have employed strong intrusion detection system to log all the bad password attempts and may initiate some prevention steps. Gmail and LinkedIn uses CAPTCHA to slow down the user, hence excluding them from the risk of being DOS'ed by the attacker. Conversely Facebook and Twitter uses timeout mechanism to slow down their users, putting them on the risk of being getting DOS'ed, if the timeout mechanism is not properly implemented. Focusing on one of the most important practice that should be implemented, bad login attempts

notification, only Facebook implement the notification practice, this could be critical because, if ever the accounts get brute forced, the users will never get notified and hence would not be able to take any preventive measures until the accounts get compromised.

- **Password Recovery:** For password recovery process, it is important to verify the authenticity of the person doing the password reset and the method to do this, which is considered most reliable at present, is 2-Factor Authentication (2FA). Unfortunately, none of the social media providers, analyzed in the survey, use the 2FA to authenticate the person as a legitimate user, at the time of account recovery. Except Yahoo, other social media providers use the email address associated with the account to send the recovery URL/code. It is important to timeout this recovery URL, Facebook and Twitter don't implement that, resulting the recovery code might be reused by the attacker if the email accounts get compromised. Most importantly the users should be always notified whenever the password reset happens on their accounts, so that if the user has not initiated it, the prevention measures could be taken on time to prevent further damage to the account. Unfortunately, none of the social media service providers surveyed implements this.

#### **4.2.3 E-commerce service providers:**

Finally the survey included the analysis of the implementation of user authentication on several e-commerce services. A discussion on their implementation of user authentication follows.

- **Password Selection:** The e-commerce services surveyed include PayPal and eBay. Both the services provide some complexity by forcing users to have at least

a number or a special character in their passwords. Both have different mechanism to slow down the attacker from creating number of fake accounts. PayPal uses email account for sending the activation link and the account only gets activated by verification of the email account associated with the PayPal account of the user. eBay uses CAPTCHA to slow down the number of attempts for account creation. These practices show that they are concerned about the password complexity for preventing a brute force attack, but at the same time, they don't want to make the login process too tough a process to create an account for the service. Also, their tradeoff between the password complexity and number of user customers is similar to the social media providers, because their businesses depend on the number of customers they have.

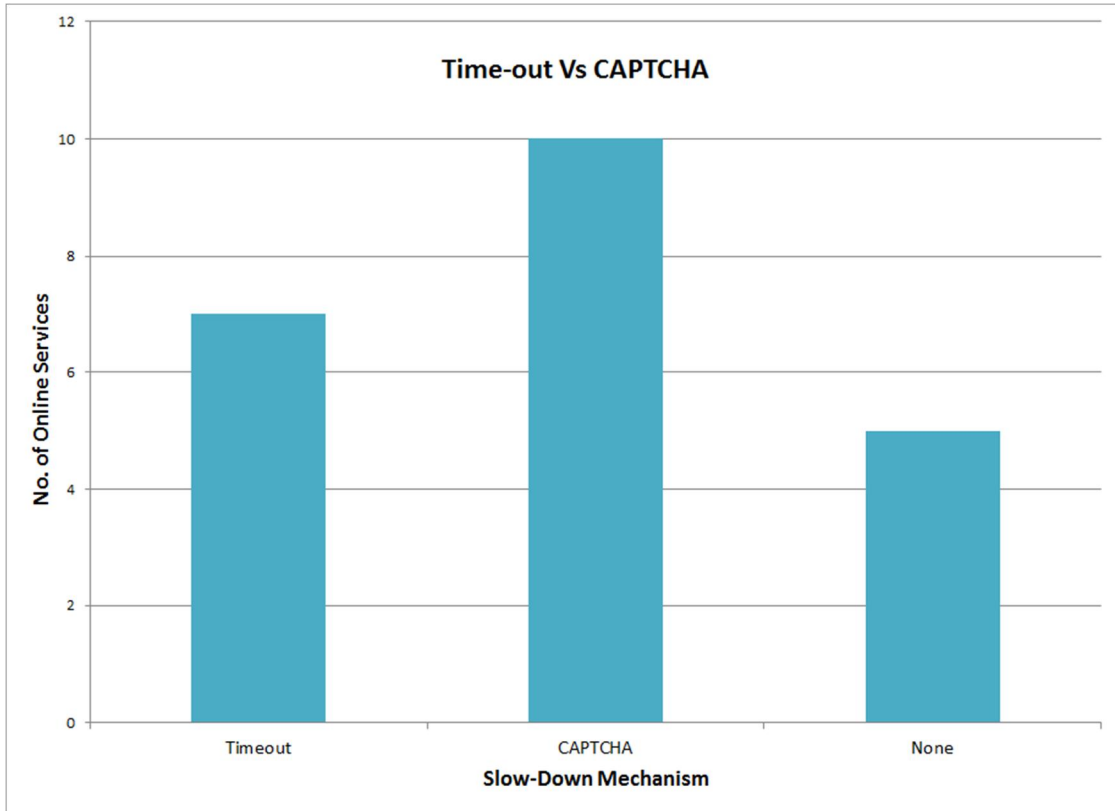
- **Bad login attempts:** The result of the survey shows that both of providers, PayPal and eBay, don't use the timeout mechanism to slow down the customer, instead they use CAPTCHA. This is a critical choice because both the services deal with online money transactions and they know that they could suffer a loss in their business, in terms of money, if they lock out the potential user who may want to buy something online or want to make a money transaction. Their profit directly depends on the time users spend online using their services, either shopping or transacting money, and hence they don't want to lock out potential customers of their service even for couple of minutes. This could impact their annual profit and hence impact their business goals. This could be a potential reason of preference given to CAPTCHA than to timeout. This shows that CAPTCHA still belongs to be in the list as one of the best practices.

- **Password Recovery:** Password recovery of PayPal account is based on the personal information, such as use of the pre-registered cell phone number. They send a verification code on the registered mobile phone and then the user has to enter the verification code to reset the password, and after entering the verification code the user is able to select the new password for the account. This kind of implementation is vulnerable to social engineering attacks. The user does not even have to unlock the phone to see the code in the message headline. This preview display, which is usually displayed when the phones are locked, contains the code as very first word of the message. This method is useful only when after entering the code, an email has been sent to the email account to reset the password, and this provides the 2FA needed for the password recovery. eBay does not uses 2FA for password reset, but uses the email account associated to the eBay account of the user to send a password recovery email. Both of the services do not implement password reset notification practice which means that the password could be reset and go unnoticed by the user, if attacker resets the password.

#### **4.3 A Discussion on Timeout Versus CAPTCHA (study from the survey):**

Earlier, the use of CAPTCHA for the purposes of slowing down the attacker/user was considered efficient and probably the best practice. Although recently, after the automation of solving a CAPTCHA challenge, many people suggest that it should now be considered a bad practice to use CAPTCHAs for slowdown purposes [33][34] and that the trend is the industry might be now moving away from the use of CAPTCHA and back toward timeout. We can look at the survey for insights. The results of the survey shows, surprisingly out of 22 online service providers only 7 service providers (31.8%) use timeout mechanism to slow down the

users, 10 service providers (45.4%) still use CAPTCHA, and 5 service providers (22.7%) don't use any kind of mechanism to slow down their users. Out of these 5 services, Wuala, SugarSync and SpiderOak are at high risk of getting brute forced, since they don't have any restrictions on password as well as do not implement any mechanism to slow down the attacker (neither timeout nor CAPTCHA). These results show that CAPTCHA is still very much in use in the online industry and still considered as one of the best practices as a first step of protection from brute force attacks. Finally, the use of CAPTCHA reduces the risk of both brute force and DOS attacks, hence a properly implemented CAPTCHA could lower the risk of both types of attacks, whereas timeout only reduces the risk of brute force attack and adds a risk of a DOS attack on the user's account. As such, we recommend the use of CAPTCHAs as one of the best practices. The following chart illustrates the findings:



**Figure 4.1:** Services using Timeout, CAPTCHA or neither.

#### 4.4 A Discussion on 2-Factor Authentication (Dropbox case study):

As discussed previously in the section of 3.1.1.4, it is a best practice that a service should use mandatory 2FA at the time of login for all kind of accounts. Here it is important to note that 2FA, even if not mandatory, should also be made available to every account of the service to allow users who are really security conscious to improve their attack resistance. In discussion this researcher had with Dropbox they revealed that they are using mandatory 2FA for their non-free accounts, i.e. business accounts and paid accounts, but for a free user, 2FA is not made mandatory. This shows a tradeoff between the kinds of accounts a user creates versus the security of

that account. There is no harm in paying extra attention towards the security of the important accounts (paid and businesses accounts), but it should be kept in mind that if the majority of the accounts an online service provider has are the free customers' accounts, and if the security of all those accounts get jeopardized, then the service might lose the goodwill in the market even though all the paid accounts were protected. Also, one of the factors, noting the popularity or the service and getting more customers, free or paid, is the number of customers a service has. And in majority of cases the big chunk of these customers are free account users, not the paid ones, hence the service should also consider to safeguard the free accounts as well with 2FA, because as discussed the breach of these accounts might result in the service loosing goodwill and position in the market. Hence it is strongly encouraged to use 2FA for all types of accounts to better protect the online service accounts to get hacked.

## **Chapter 5**

### **CONCLUSION**

We have surveyed the current user authentication practices and mechanisms of many online service providers such as cloud storage services, social networking, email, and e-commerce services. There are various authentication methods currently in use and what mechanism provides the best security is in constant flux. One example of such a change would be the use of CAPTCHA, used to slow down the an attacker from trying to guess a user's password as fast as possible, but people have found the work around for it, as many online tools are available for automatically solving a CAPTCHA challenge, it is assumed that it is no longer a best practice to use for slowing down the attacker [33][34]. But it is still worthwhile to use some very complicated CAPTCHAs as the solution to slow down the attacker and prevent a direct brute force attack. Other more common techniques currently used is normal or exponential time-out, which temporarily locks out the users from logging into the account if they exceed a maximum number of password tries. The critical problem with these techniques is, if not properly implemented; it can result into Denial of Service (DOS) attack on the user's specific account (e.g. Dropbox) and thus makes the service vulnerable to DOS for high profile clients (e.g. celebrities). Hence in that case CAPTCHA still may be preferred over timeout time out because timeout may have more serious consequences if not implemented correctly.



This research shows that even though it is obvious that some of the best practices should be incorporated into the user authentication process to shield the service from major cyber-attacks, there are many cloud storage services [Table 4.1, 4.2, 4.3], who are not using them, possibly making them more vulnerable to the attacks like brute force. The reason could be business related, in the sense that they might assume that they may not get hit by the attacks because of their business size or business model (e.g. a free service which needs many millions of users), but sooner or later they may need to adopt more of these or other best practices to at least defend against the most common cyber-attacks such as a brute force attack. Also, the research revealed that the way different service providers implement security practices depends on the nature of the service provided, for example social networking services or e-commerce service providers may want it to be easy for new customers to get enrolled in the service, so rather than asking them to follow all security best practices mentioned in the ‘Account creation - Password selection’ section of the table, they allow the user to choose any kind of password they want, and then adopt a more stringent process for other authentication steps - Password recovery and Bad Password Attempts. On the other hand services like online banking have tight secure mechanisms for the complete user authentication process, since online banking is available only to the account holders of that bank and their business model does not depend on the number of online users of that service. This illustrates that there is always a trade-off between the type of business model of the service and security practices adopted for the user authentication in that service.

The current published NIST best practices [9], at least in the lower impact systems,

appear weaker than the current required industry practices we have cataloged. And they are not sufficiently detailed to guide the implementer how to incorporate them in their services. We present a concise selection to follow of what we believe are current best practices and we recommending several new practices. The recommended best practices in this paper are defined with respect to current security vulnerabilities and what we believe are best current practices, but since new cyber-attacks are coming into picture on a continuing basis, it should be said that what is safe today may not be safe tomorrow. But it can be seen clearly that a majority of online service providers still have some gaps with the best current security best practices described here. It is often said that 100% security can never be achieved in cybersecurity, the only possible way to achieve it is to be cut off from the Internet (e.g. air-gap) which can't always be fully achievable, but we can strive for our systems to become less vulnerable and protect ourselves from attacks by practicing security best practices.

### **5.1 Summary of Best Practices:**

In closing, this section provides a concise summary of all the best practices mentioned in this thesis as a quick reference, which should be currently implemented to provide strong security against the common cyber-attacks made on the online service accounts.

1. Password should include:
  - At least 8 characters or more<sup>10</sup>.

---

<sup>10</sup> As with all quantitative values in security, this value (8) will likely increase over time as computer power increases with time. The value 8 is a reasonable value at this point in time.

- Special characters.
  - Numbers.
  - Uppercase as well as lowercase characters.
2. Password should be checked for the common strings, parts of user names and top common passwords.
  3. Email an account activation link to verify the user before activating the account.
  4. Mandatory 2FA to verify the user's identity at the time of login.
  5. Use CAPTCHA during new account creation.
  6. A Password strength bar should be shown to the user while creating a password.
  7. Restrict the maximum number of incorrect password attempts.
  8. Slow repetitive incorrect login attempts by using a complex CAPTCHA challenge.
  9. Slow repetitive incorrect login attempts by using a sophisticated time-out mechanism.
  10. Bad login error messages should not reveal whether a username or password is incorrect.
  11. No timing difference between loading a legitimate login page versus a bad login page.
  12. Send email/text notification to the owner of the account after a number of bad login attempts.
  13. Mandatory use of 2FA for verifying user while requesting a password reset.

14. Use a user's email address for password reset.
15. The password recovery email should not contain the old password, new password, or the password hint in clear text.
16. The password recovery email should not contain any part of account information.
17. The password recovery mechanism should not be based on the personal security questions only.
18. The password recovery should not be based on a random email address or phone number provided promptly at the time of recovery.
19. The recovery URL should have a timeout and should be invalidated after the timeout.
20. While recovering the password, no part of recovery email or phone number should be revealed.
21. Notify the user every time a password reset takes place on the account.
22. Notify the user whenever the recovery emails address or phone number changes.
23. Use a user selected and user specific personal security image and caption phrase to verify the service's authenticity.
24. The user names used for the online accounts should be uncommon or should be hard to guess.
25. In case of changes in the password policy, all users should be required to comply with the new policy by updating their current passwords.
26. Passwords should be secured with hashing, using a salt at least as large as the number of bits used for hash.

27. Users should be required to change the passwords periodically.

## **Chapter 6**

### **FUTURE WORK**

The information incorporated in this survey and thesis was done manually by researching the current observed best practices and industry standard best practices available to the online service providers. The survey was made by researching every online service listed in the table manually, by testing the online service against every best practice and getting the result whether the online service uses the best practice or not. This testing process is time consuming and might be automated by developing an audit tool which then can be used to perform the tests on the online service, testing that service against the best practices listed in the table and providing the results based on some weighted sophisticated algorithm to conclude how vulnerable the online service is towards the common cyber-attacks like brute force, DOS etc. Such a tool would also allow us to quickly re-audit providers on a periodic basis and get a measure of whether their security stance is improving.

#### **6.1 Tool for Automated Auditing:**

As we see, whenever a new vulnerability is discovered, developers build tools to detect the vulnerability and also, in some cases, suggest a work around. For example, there are tools that automate the process for vulnerability assessments like OpenVAS, Nessus, and there are tools to detect viruses and malwares like McAfee, Malwarebytes and all the other anti-virus software products. So the motivation behind building up an automated audit tool, which can scan an online service and detect the

vulnerabilities with respect to user authentication by checking the service against all the best practices, comes from that idea where it might be useful to have an automated tool to assess the online service so as to know the weaknesses in the user authentication implementation, and possibly recommend better mechanisms or a work around.

The main challenge in building an automated tool of that kind is that every online service has their own set of methods to implement a secure online user authentication, including their own set of practices implemented for the different categories illustrated in section 4. Hence the tool should be built in such a way that it should allow for customized auditing corresponding to the online service which is using it. To describe the challenges associated with the development of the tool, let's take an example. Say there is an online service A which uses CAPTCHA to slow down the attacker while attempting bad passwords, and a service B which uses time-out to slow down the attacker. To perform an audit on the service A, a customized set of best practices has to be used, and to perform an audit on service B other set of customized best practices has to be used. Similarly, the behavior of the tool also depends on the methodology implemented for the password recovery. There are many ways to implement the password recovery mechanism, as described in section 4 category Password Recovery, hence there should be a way in which the tool could be tuned or customized for the different mechanisms for password recovery by the online service to perform audit on the password recovery method.

In summary, there are various ways to implement the mechanism for the password selection, bad logins protection and password recovery, and all are mentioned in section 4 in each corresponding category. And all these different

mechanisms have to be taken into consideration when building the automated audit tool so that it can be used widely across various online service providers to perform an audit on their service. Finally, the audit tool should show the level of seriousness of the vulnerability and should suggest a better solution or workaround to that problem.



## BIBLIOGRAPHY

1. ISIS HACK OF PENTAGON TWITTER ACCOUNTS.  
<http://www.infowars.com/isis-hack-of-pentagon-twitter-accounts-traced-back-to-maryland/>
2. WBOC Website and Twitter Page Hit by Cyber Attacks.  
<http://www.wboc.com/story/27773584/wbocom-wboc-twitter-page-hit>
3. Timeline of the Sony Pictures Entertainment Hack.  
<http://www.aol.com/article/2014/12/18/timeline-of-the-sony-pictures-entertainment-hack/21118609/>
4. Hackers target Albuquerque Journal website.  
<http://krqe.com/2014/12/24/hackers-target-albuquerque-journal-website/>
5. Tim Cook Says Apple to Add Security Alerts for iCloud Users.  
<http://www.wsj.com/articles/tim-cook-says-apple-to-add-security-alerts-for-icloud-users-1409880977>
6. Update to Celebrity Photo Investigation.  
<http://www.apple.com/pr/library/2014/09/02Apple-Media-Advisory.html>
7. PASSWORDS FOR 5 MILLION GOOGLE ACCOUNTS LEAKED, Once again: This is a good time to change your passwords.  
<http://www.fastcompany.com/3035558/fast-feed/passwords-for-5-million-google-accounts-leaked>
8. Tabnabbing: A New Type of Phishing Attack.  
<http://www.azarask.in/blog/post/a-new-type-of-phishing-attack/>
9. NIST: Special publications (800 series).  
<http://csrc.nist.gov/publications/PubsSPs.html>
10. Karen Scarfone and Murugiah Souppaya. *NIST 800-118: Guide to enterprise password management*, April 2009.  
<http://csrc.nist.gov/publications/drafts/800-118/draft-sp800-118.pdf>
11. *NIST 800-53r4: Security and privacy controls for federal information systems and organizations*, April 2013.  
<http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf>

12. William E. Burr, Donna F. Dodson, Elaine M. Newton, Ray A. Perln, W. Timothy Polk, Sarbari Gupta and Emad A. Nabbus. *NIST SP 800-63-2: Electronic authentication guideline*, August 2013.  
<http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf>
13. Match.com's HTTP-only login page puts millions of passwords at risk.  
<http://arstechnica.com/security/2015/04/match-coms-http-only-login-page-puts-millions-of-passwords-at-risk/>
14. Apple's iCloud cracked. <http://www.zdnet.com/article/apples-icloud-cracked-lack-of-two-factor-authentication-allows-remote-data-download/>
15. CENTCOM's Twitter, YouTube accounts hacked.  
<http://www.usatoday.com/story/news/nation/2015/01/12/twitter-centcom-isis/21640577/>
16. Dropbox confirms security breach. <http://www.information-age.com/technology/security/2114488/dropbox-confirms-security-breach>
17. World's Biggest Data Breaches.  
<http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/>
18. Brute-force attack. [https://en.wikipedia.org/wiki/Brute-force\\_attack](https://en.wikipedia.org/wiki/Brute-force_attack)
19. Taylor Swift's Twitter and Instagram Accounts Hacked.  
<https://celebrity.yahoo.com/blogs/celeb-news/taylor-swift-s-twitter-and-instagram-accounts-hacked-183300894.html>
20. New York Post Adds to List of Famous Twitter Hacks.  
<http://www.bloomberg.com/news/articles/2015-01-16/new-york-post-adds-to-list-of-famous-twitter-hacks-timeline>
21. YouTube hack 'threatened' Justin Bieber videos.  
<http://www.bbc.com/news/technology-32173657>
22. T. Dierks and E. Rescorla. *RFC 5246: Transport Layer Security*, August 2008. <https://tools.ietf.org/html/rfc5246>
23. A. Freier, P. Karlton and P. Kocher. *RFC 6101: Secure Socket Layer*, August 2011. <https://tools.ietf.org/html/rfc6101>
24. E. Rescorla. *RFC 2818: HTTP over TLS*, May 2000.  
<http://tools.ietf.org/html/rfc2818>

25. FIPS 197: Announcing the ADVANCE ENCRYPTION STANDARD (AES), November 26, 2001.  
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf>
26. Two-factor authentication. [https://en.wikipedia.org/wiki/Two-factor\\_authentication](https://en.wikipedia.org/wiki/Two-factor_authentication)
27. System configuration. [https://en.wikipedia.org/wiki/System\\_configuration](https://en.wikipedia.org/wiki/System_configuration)
28. THE INTERNET OF EVERYTHING: 2015  
<http://www.businessinsider.com/internet-of-everything-2015-bi-2014-12>
29. 9 Recent Cyber-attacks Against Big Businesses.  
<http://www.nytimes.com/interactive/2015/02/05/technology/recent-cyberattacks.html? r=0>
30. Social engineering (security). [https://en.wikipedia.org/wiki/Social\\_engineering\\_\(security\)](https://en.wikipedia.org/wiki/Social_engineering_(security))
31. Statistics Will Crack Your password.  
<https://www.praetorian.com/blog/statistics-will-crack-your-password-mask-structure>
32. L. Von Ahn, M. Blum, N. J. Hopper, & J. Langford. CAPTCHA: Using hard AI problems for security. *Advances in Cryptology—EUROCRYPT 2003* (pp. 294-311), Springer Berlin Heidelberg, 2003.
33. E. Bursztein, M. Martin, & J. Mitchell. Text-based CAPTCHA strengths and weaknesses. *Proceedings of the 18th ACM conference on Computer and communications security* (pp. 125-138), ACM, October 2011.
34. J. Yan, & A. S. El Ahmad. A Low-cost Attack on a Microsoft CAPTCHA. *Proceedings of the 15th ACM conference on Computer and communications security* (pp. 543-554), ACM, October 2008.

## Appendix

### CURRENT PUBLISHED BEST PRACTICES

There are many articles on the Internet that describe the best practices that should be implemented by the online services to make their online user authentication process secure. But the most reliable source for this kind of information is considered to be the documents published by governments to tighten the computer security and other domains of security. One of the most widely accepted set of documents are published by NIST, the National Institute of Standards and Technology for the United States. NIST has a special publication series for computer security community [9]. It includes the published documents for almost all areas of computer security where the guidelines specify recommended mechanisms to maintain rigid control over the security and privacy of the service. Since the scope of this thesis is limited to online user authentication, we researched the NIST documents establishing the guidelines in this area of computer security. We have analyzed the three NIST publications: Guide to Enterprise Password Management [10] (NIST.SP.800-118), Security and Privacy Controls for Federal Information Systems and Organizations [11] (NIST.SP.800-53r4) and Electronic Authentication Guideline [12] (NIST.SP.800-63-2). The best practices listed in these documents were compared with the best practices in this thesis document. A brief discussion of some of the comparisons follow.

- When analyzing the NIST document NIST.SP.800-53r4, we found that the document does not discuss about the use of CAPTCHA as one of the practices

anywhere in the document, neither at the time of account creation nor at the time of unsuccessful login attempts. We recommend use of complex CAPTCHAs as the first layer of protection against the brute force attack as well as at the time of account creation, so that the server resources should not get overwhelmed with millions of fake accounts. Also, the NIST.SP.800-53r4 document does recommend automatic lockouts of the accounts while encountered unsuccessful attempts, but doesn't say about how long it should be. Furthermore, the document does not recommend mandatory salting of the passwords, stating- "*organizations **may** also consider salting passwords*", also without recommending a minimum length of the salt that should be used while hashing the password because as already have been discussed in the thesis that salts with 8-16 bits are not that complex and easily crack able. The document should strongly recommend the use of salt while password hashing which should be as large as the hash.

- When analyzed the NIST document NIST.SP.800-63-2 for the guidelines for the electronic authentication, we found that the document does recommend the use of CAPTCHA for protecting the account from attacker trying to authenticate. Also, the document does recommend the time-out of the accounts for the short period of time followed by an unsuccessful attempt of authentication. Also, as seen in the NIST.SP.800-53r4 document, this document also does not strongly recommend the use of salt for hashing the passwords before storing them, stating "Passwords may be concatenated to a variable salt". We strongly recommend the use of a large enough salt, at least as large as the hash, to protect the passwords against the dictionary attacks of password cracking. These documents don't clearly state the practices and how to implement them in real world scenario.

- The other document which was analyzed was the NIST sp800-118 document, which briefly illustrates the guidelines for enterprise password management. This document does not state anything about the use of CAPTCHA for slowing down the attacker at the time of password guessing attempts. Also, the document does state the use of salt for securing the passwords from getting easily cracked using the dictionary attack, but does not recommend the bare minimum length of salt used for the hashing of the passwords. Finally the document also mentions of locking of the user account after consecutive failed authentication attempts in a row. But the document states that time-out should be 15 minutes after 50 consecutive tries, to protect the legitimate user from locking out of the account for a long time. The maximum number of tries threshold is quite high (50) and the timeout period is pretty low, so if the accounts have easy passwords then even though having this kind of timeout would not solve the problem of brute force attack (top 1000 passwords can be tried in 5 hours from a single computer, this could take even less time when used botnet, more than one computer, which is pretty common practice to brute force the accounts). Also, improper implementation of the timeout technique could result in the DOS attack on the user accounts, and the document does not state this warning, resulting in some misleading information which could have serious impact on the online services security and privacy.

While NIST does often mention many of the best practices across the 150 or more SP 800 documents, those practices are not always found uniformly across the documents. An implementer, just using a recent document, might leave out a key best

practice found in an older document or a document seemingly unrelated to the implementation at hand.