International Journal of Multidisciplinary Research and Development
Online ISSN: 2349-4182 Print ISSN: 2349-5979, Impact Factor (RJIF 5.72)
www.allsubjectjournal.com
Volume 3; Issue 2; February 2016; Page No. 362-366

Security analysis of various login authentication techniques

Sonia Mahindru

Asstt. Prof., PG Deptt. Of Comp. Sc. & IT, HansRaj Mahila Maha Vidyalaya Jalandhar, India.

Abstract
With tremendous increase of number of users on internet. Online security has become a very important issue. Whether it is information related to an organization or an individual, if stolen it can lead to serious financial loses. The typical validation method used on the Internet is single-factor authentication, where users supply a username and password. This approach has significant drawbacks, particularly as cyber criminals become more organized and adept. Thus idea of using Multi-Factor Authentication has been introduced in the world of internet to harden the security of network and make it difficult for the attackers to crack systems. In this mechanism, users are required to provide some extra information along with their login Id and password. Most popular is using One-Time Passwords that are generated randomly and valid only for single login and even for short duration of time. In this paper review of various login authentication techniques has been performed, discussing pros/cons and attacks through which security of each can be compromised.

Keywords: One Time Password (OTP), 2-factor authentication, Short Message Service (SMS), Time based One Time Password (TOTP), Image based Authentication (IBA), Attacks, Trojans

Introduction

During a login process, an authentication factor is a requirement that is designed to verify the identity of an authorized user. In login security, there are three categories of authentication factors which are typically used to verify identity.

- Something that is known only by the user, such as a password or PIN
- Something that only the user possesses, such as a smartphone, smartcard, USB token, or other hardware key
- Something that is physically unique to the user such as a fingerprint or iris scan

Each category covers a range of potential requirements that can be used to verify identity and authenticate access to websites, applications, networks, systems, and other types of secured services. They can also be used electronically to approve transactions, Sign or approve documents, grant access rights to others, or establish a chain of administrative authority. In the wake of recent cyber-attacks, information security experts have universally called upon companies to implement, integrate, and enable two-factor authentication(2FA) to protect user accounts and access to their websites, applications, networks, servers, and systems. Two-factor authentication requires two authentication factors to verify identity, and it usually combines one factor from each of the categories discussed above. Thus, a password might be combined with physical possession of a smartphone, which is used to receive a one-time code via SMS text transmission and the user must enter this code as the final verification step during a login process. 2FA addresses the fundamental problem of cybersecurity, which is the continued use of traditional ID and password combinations for login security. 2FA helps avoid attacks by adding an additional layer of security that can prevent unauthorized access by requiring the user to verify identity through a separate method that is often inaccessible to attackers.

One-Time Passwords

A One-Time Password (OTP) is a password or code which is valid only for one login session or transaction on a computer system or any digital device. OTPs were introduced just to avoid the shortcomings that are associated with static passwords. Even they are valid for a small period of time and they automatically expires after the given time span. The most important advantage of OTPs, in contrast to static passwords, is that they are not vulnerable to replay attacks. It means that a potential intruder or attacker who manages to record an OTP that was already used by a user to login into the service will not be able to reuse it since it will be no longer valid. Also, OTPs are very difficult for humans to memorize. Another advantage is that a single OTP code cannot be used to login on multiple systems. Many Techniques have been introduced to generate and deliver these one-time passwords and most of them use Time-based One Time Passwords.

Login Authentication Techniques

SMS Based Onetime Passwords

Short Message Service (SMS) based One Time Passwords is the simplest, widely used mechanism for two factor authentication generating one time codes. In this method, the onetime code is generated on the server side and is being sent over the network to the registered mobile number via SMS. Authentication occurs when the server recognises that the user enters in the correct code for login. The phone number of the user must be registered with the service that provides SMS OTPs for authentication [1]. Whenever a user tries to login into his account with username and password, he will receive a unique code on his phone number and will enter this code to get access to his account. The user can receive the OTP either as a text message or via an automated call using text-to speech

conversion. Additionally, OTP is also restricted to a very short period of time and will expire automatically. There is no additional software or hardware requirement in authentication system that uses SMS based OTPs to authenticate or authorize a valid user which shows that this is the simplest way of delivering OTPs as SMS enabled devices are almost available with every person using internet these days.

The key advantage of SMS based OTP system is that it is compatible with any SMS-enabled mobile phone. Since the only thing a SMS-based system needs to provide to the server is the user's phone number. Also, very few steps are involved in this technique and is the simplest way of generating one time passwords and even simpler in transmission of the unique codes. It also keeps the cost very low as a large customer already owns a mobile phone for purposes other than generating One Time Passwords. Because of these advantages most of the banking transactions like internet banking, Master/Visa credit or debit card transactions, enables an extra layer of security by providing an extra One Time Passwords SMS verification.

The problem with SMS-based OTP is that it is only as good as the mobile network of the user. If the network is slow, the user may be delayed from logging into account or even the

Received unique code may be expired [2]. Thus, a user would either be delayed to get access to the service or may request to send a new one time password. Also, several attacks against

GSM and 3G networks have shown that confidentiality for SMS messages cannot necessarily be provided [1]. The one time passwords sent via SMS are always transmitted in plaintext which is more vulnerable to man-in-the middle attack. A few attacks have been discussed in the following section.

Attacks

Physical Access to Phone

With physical access to the phone that receives the SMS messages, the attacker can easily extract the OTP. Of course, gaining physical access is hard, time consuming, and easily detected. While there are ready-made toolkits to extract data from mobile phones of most manufactures, this kind of attack is unlikely for fraud since it cannot be performed on a large scale.

Sim Swap Attack

The SIM Swap Attack [3] is a social engineering attack with the goal of acquiring a replacement SIM card for the victim's mobile phone number. The replacement SIM is linked to the victim's mobile phone number. Hence, the attacker will receive all SMS messages that are supposed to be read by the victim.

Wireless Interception

Cellular operators use the GSM, 3G, and CDMA technologies to provide mobile services such as SMS messages. However, GSM is insecure due to several vulnerabilities such as a lack of mutual authentication and weak encryption algorithms. In particular, there is no mutual authentication between mobile phones and base stations in GSM networks, hence fake base station attacks are possible. These are generally used to intercept mobile traffic (including SMS) of the end users. GSM uses different algorithms such as A5/1, A5/2, and A5/3 to encrypt wireless communication between mobile phones and base stations. The A5/0 algorithm means there is no encryption, A5/2 algorithm is weak and can be broken in a few seconds [4]. It is possible to capture GSM traffic using low cost devices and decrypt the traffic due to weak algorithms [5]

Mobile Phone Trojans

Mobile phone malware, and especially trojans that are specifically designed to intercept SMS messages containing OTPs, are a rising threat. This kind of malware is created by criminals directly for the purpose of making money. The ZITMO (Zeus in the MObile) [6] trojan for Symbian OS is the first known piece of malware that was specifically created for intercepting mTANs.

Also a ZeuS version for Windows Mobile was detected and named TrojanSpy.WinCE.Zbot.a [7]. The trojan contained the same basic functionality as ZITMO. Alike trojans also exist for android [8] and RIM's Black Berry [9]. Lately, a new variant of Android malware was discovered. It targets mobile banking users in Germany, the Netherlands, Portugal, and Spain [10].

Google Authenticator

It is a 2-step authentication scheme introduced by Google for its Application users. After enabling this service user have to provide an extra verification code after logging into their Google accounts. This verification code could be received by a Short Message Service (SMS) text message or voice over text message, or even through a token or code generating application developed by Google. Google's 2-step verification requires something you have (like smart phone with Google authenticator installed to generate verification code) and something you know (that is the password of your Google account) that is required to access into your account [11]. The verification code could be retrieved via a token generator on a Smartphone. These token based verification codes are generated using a time-based algorithm. And application that performs this verification code generating is called as Google Authenticator. Google authenticator is a software-based OTP generation scheme based on Time-based One Time Passwords (TOTP). It implements TOTP; security token from RFC 6238 in mobile apps made by Google or may be referred to as 'Two-Step Verification'. Google Authenticator uses an offline scheme of TOTP, where it's user's device which generates one-time passwords for the user rather than the server. Authenticator provides a six to eight digit one time unique password which user must provide in contrast with username and password to get access or login to Google services or other sites. It is an open source project that is available for android, iOS and BlackBerry devices. The application generates one-time time based code using open-standards, including HMAC-based One-Time password (HOTP) algorithms and Time-based One- Time password (TOTP) algorithms. The generated Token or code is six digits in length and is valid for a 30 second timeframe [11].

Before the application could generate the unique tokens, it has to be linked with user's Google account either using Quick Response (QR) code which is created by Google and has to be scanned by the user Smartphone, or by using a secret-key provided by Google [11]. Once the account is linked to the device, the app can generate a token for 30 seconds time span, after which a new token will be generated. When a user has enabled his Google applications with Google's 2-step verification, his login process will be protected through an

extra layer of security. Firstly, the user will as usual has to enter his username or login id and password and then in second step, he will enter the 6-digit verification code generated by the application Google Authenticator installed on the users Smartphone [11]. But, before the application could generate verification codes, it has to be linked to users Google account. This can be done via two methods. Firstly, link can be made using Quick Response (QR) code which is generated by Google in the browser and has to be scanned by the Smartphone device. Another method is by using a secret key provided by Google. Once the account and Google Authenticator are linked to each other, the Application would generate security token or verification code that are valid not more than 30 seconds time span i.e. it will automatically expires after 30 seconds and a new code will be generated.

The major advantage of this system is that it generates tokens offline i.e. it can also generate verification codes even if there is no network connectivity. Also it ensures that only the rightful owner is given access to the account. The Time-based One Time Password (TOTP) generated verification codes based upon a synchronize time between the Google services

And the user's mobile device provides a robust login system that is not prone to attacks. But the major drawbacks of this system are seed transmission and seed storage [2]. When it comes to transfer the seed to the mobile phone, Google relies on QR codes in which the seed travels in plaintext during transmission. This is more vulnerable to be attacked by any intruder. Also secret seed and login credentials of user are stored on Android device in plaintext so can be accessible to anyone easily and can be used to enrol the same seed on multiple devices. Thus, retrieving the secret code to link the Google account with Google application or Authenticator and the verification codes can be easily done by performing Structured Query Language (SQL) query on the right databases. This would allow the user to use any of these devices to authenticate to same online account from various devices.

Attacks

Initialisation

TOTP needs to be initialised by a secret key in order to be able to generate valid OTPs. This secret key needs to be transported to the user's smartphone since it is generated by the authentication service.

In Google Authenticator the secret is displayed in the form of a QR-code in the user's browser during the setup phase. The idea is that the smartphone's camera should be used in conjuction with the Google Authenticator app to scan the code and thus retrieve the secret. The contents of a Google Authenticator QR-code which contains the secret are in clear text. Thus a PC residing malware could easily capture the QR-code from the setup page by taking a screenshot and send the contents to a C&C server. The attacker would however also have to know the username and password of the victim in order to exploit the QR-code contents. An augmentation of the malware to include key logging functionality would make it possible to also capture keystrokes and retrieve passwords. If this scheme is successful, the attacker would have access to an infinite amount of valid OTPs since they are all based on the secret seed.

Replay

A weak implementation of TOTP will not check whether a supplied OTP has already been used in an authentication session. An attacker that in some way has acquired a valid OTP could use it for logging in to a service, even if the victim had already done so by using the same OTP. This would however require that the attacker does so within the valid time period of the OTP, which is 30 seconds with the Google Authenticator app. However, since the Google Authenticator already provides protection against replay attacks by invalidating an OTP that has already been used, this vulnerability could only possibly exist in other implementations.

Stealing OTPS

With the release of Android 5.0, a new capability that allows for apps to take a screenshot of the current contents of the display has been introduced. Previously it was not possible for apps to gain the READ_FRAME_BUFFER permission that was required for accessing the display and to save the results as an image file. For this, a rooted device was required. Now with the Media Projection API it is possible. While this functionality certainly has some practical usage, it also opens up for malware to use this functionality to steal sensitive information.

Consider a malware that runs as a service in the background, taking screenshots of the contents of the display with the aid of the Media Projection API at regular intervals. The purpose of this malware would be to steal OTPs from the Google Authenticator app and send them to a C&C server. To determine whether a given screenshot has been taken of the actual Google Authenticator app, an OCR engine could be utilised to scan the specific area where the app title appears. If the results of the OCR scan matches ”Google Authenticator” the image could be compressed and sent over the network to the C&C server or be processed with the same OCR engine to extract the actual characters in order to save bandwidth when transmitting.

Image Based Authentication

The Image-based Authentication (IBA) is based on Recognition Technique. It is almost similar to text one time passwords as in this also the user is provided a shared secret as an evidence of his/her identity. However, text-based OTPs use alphanumeric characters to represent the secret and IBA uses visual information. When the user registers for the first time on the website, they are required to select a set of images that are easy to remember such as natural scenery, automobiles etc [12]. Every time a user login into the website or service, they are provided a grid of images randomly generated. Then, the user can identify the images previously selected by them. The user is authenticated by correctly identifying the password images. The category of images is stored by the authentication system on Image Identification Set. When a user login, the Image Identification Set for that user is only retrieved and is being used to authenticate that particular user. The human is more adept in retrieving or recalling a previously seen image rather than a previously seen text.

Major benefit of IBA is that it is more secure and requires less memory. Graphical passwords may be shared via taking photos, taking screen shots or even through drawing but they require more time than text passwords. Also, idea of using images as one time passwords makes it difficult for the attacker to intrude using Brute Force attack [12]. But this also facilitates to data manipulation and interpretation to a greater extent than the alphanumeric characters does. This complexity, however, makes IBA harder to implement and deploy, requiring environments with increased computational power and graphical capabilities. This prevents it to be used by most of the services of websites because of complexity.

The key drawback in case of security in Image based authentication is Hotspots. Hotspots are the specific areas in an image that have a higher probability of being selected by most of the users as a part of their passwords. If any attacker can accurately predict the hotspots in that image, a dictionary of images can be built basis on these hotspots. Thus, hotspots are meant to be problematic in Image-based authentication [12].

Attacks

Keystroke Logging

In password authentication methods, Eve can observe or log Alice's keystrokes and later authenticate herself as Alice. Just logging the mouse coordinates will not be helpful in the basic IBA system as the images are displayed at random locations each time. If the attacker is able to store the images in presentation order as well as log the keystrokes/mouse positions, then the system is compromised.

Shoulder Surfing

Modern displays including laptop screens have wide viewing angles. This makes over the shoulder peeping easier. To counter this image grid in our implementation is all grayed out. Depending on the mouse pointer location the image is sharpened. It is important that the attacker not be able to identify which of the images is selected, so the user interface must make this difficult.

Frequency Correlation Attack

Since Alice has to be authenticated based on her Individual image set, in any round one of the images from the image set will appear in the Presentation Sets. If an attacker collects the presentation sets over time, then the Individual image set may be deduced, depending on how the Presentation Sets are generated. These Presentation Sets may be collected by interception on an unencrypted channel, or by posing as a user and observing the Presentation Sets sent when trying to authenticate.

Intersection Attack

The Usable Image Space can be quite large compared to the Individual image set, and so it would be extremely unlikely that any of the random images will be repeated in multiple authentication attempts. Hence, Eve can isolate probable members of Alice's image set by identifying images that are repeated across the authentication attempts.

Logic Attack

A smaller Image Space, or a user-specific subset of the Image Space may be used on every authentication attempt, so that the Presentation Set never changes over time. However, if the attacker knows that exactly one of the images in each Presentation Sets is a key image, then using logic, within a small number of authentication attempts the attacker can narrow down the Individual Image Set to one or a few subsets from the Presentation Set.

Conclusion

In this paper three different multifactor authentication techniques like one time passwords via SMS, Time based onetime password (TOTP) technique like Google Authenticator and Image based authentication are discussed, focussing on their functionality, limitations and certain issues related to their security. One Time Passwords are an efficient technique to generate passwords randomly each time for user. OTP prevent users from replay or eavesdropping attacks. SMS based OTP have come under heavy attack, especially by mobile phone trojans that are specifically designed to intercept and forward OTP authentication and authorization credentials to criminals. Google authenticator if implemented correctly, it is resistant against replay- and brute force attacks. IBA is a more user-friendly, secure technique that helps to increase the password quality tremendously compared to a text-based approach. Almost every kind of authentication system discussed above is widely used today to provide security to the users.

References

1. Muliner C, Borgaonkar R, Stewin P, Seifert J. SMS-based One-Time Passwords: Attacks and Defense, Springer-Verlag Berlin Heidelberg, 2013; 7967:150-159.
2. Uymatiao, Mariano Luis T, William Emmanuel S Yu. Time-based OTP Authentication via Secure Tunnel (TOAST): A mobile TOTP scheme using TLS seed exchange and encrypted offline keystroke. 4th IEEE International Conference on Information Science and Technology(ICIST), IEEE, 2014, 225-229.
3. icici Bank. What is SIM-Swap fraud? http://www.icicibank.com/online-safe-banking/simswap.html.
4. Barkan E, Biham E, Keller N. Instant ciphertext-only cryptanalysis of gsm encrypted communication. Springer-Verlag, 2003, 600-616.
5. Osmocom SDR. Inexpensive SDR (Software Defined Radio) project. http://sdr.osmocom.org/trac/wiki/rtl-sdr, 2011.
6. Apvrille A. Zeus in the Mobile (Zitmo): Online Banking's Two Factor Authentication Defeated. http://blog.fortinet.com/zeus-in-the-mobile-zitmo-online-bankings-two-factor-authentication-defeated/, 2010.
7. Maslennikov D. ZeuS in the Mobile is back. http://www.securelist.com/en/blog/11169/Zeus, 2011.
8. Secure F. Threat Description: Trojan: Android / Crusewind A. http://www.f-secure.com/v-descs/ trojanandroid crusewind a.shtml, 2011.
9. Fisher D. Zeus Comes to the BlackBerry. http://threatpost.com/enus/blogs/ zeus-comes-blackberry-080712, 2012.
10. Klein A. The Song Remains the Same: Man in the Mobile Attacks Single out Android. http://www.trusteer.com/blog/ song-remains-same-man-mobile-attacks-single-out-android, 2012.

11. Appelman M, Scheelen Y. Analysis of Google's 2step Authentication, University of Amsterdam, May, www.scribd.com/doc/95267199/ Analysis-ofGoogle-s-2-StepVerification#scribd, 2012.
12. Parmar H, Nainan N, Thaseen S. Generation of Secure One time passwords based on Image Authentication System, © CS & IT-CSCP, 2012, 195-206.
13. Kaur N, Mandeep D. A Comparative Analysis of Various Multistep Login Authentication mechanisms, International Journal of Computer Applications, 2015; (9):127.
14. Newman R, Harsh P, Jayaraman P. Security Analysis of and proposal for image based authentication" www.cise.ufl.edu/~nemo/papers/Carnahan,pdf, 2005.
15. Ericson C. Two-factor Authentication in Smartphones: Implementations and Attacks lup.lub.lu.se/student-papers/record/7792889/file/7792890.pdf.