Effective Implementations of Multi-Factor Authentication

Jamie Chen[#1], Martin Ycaza[#2], Jishun Zhang[#3], Victor Zheng[#4], Zhong Zhu[#5]

Abstract— **Multi-factor authentication (MFA) is a method of providing security in a system by requiring two or more distinct factors for successful authentication. The most common form is two-factor authentication (2FA), which requires two different pieces of information from the user in order to verify an identity: something they know, something they have, or something they are. With its ease of implementation and widespread acceptance in most industries, MFA provides an extra layer of protection against social engineering attacks at an inexpensive cost.**

Keywords— **Multi-factor authentication, Two-Factor authentication, TOTP, SMS, Push**

I. INTRODUCTION

Social engineering is the most used form of attack by hackers to gain access to the private information of corporations, businesses, and the individual user. The growth of the usage of IoT devices in the past decade has only opened up more windows of opportunity for attackers to take advantage of. All it takes to gain access to private data is the compromise of one endpoint that belongs to the corporate network or the individual. For example, in the 2016 presidential election, over 150,000 emails containing information used by the Democratic National Conventions' twelve staffers were leaked to the public. Russian hackers were able to gain access to one of their email accounts through a spear-phishing email. The message appeared to be a legitimate Google email, warning the user about suspicious activity on the account, and asked for login credentials through a different link. In this scenario, the email account in question only had one layer of authentication being the password. The solution to such a problem would be to add more than one factor of authentication to the account. This form of authentication ensures that even if one factor is compromised, there are more factors that an attacker has to obtain, and therefore can prevent, if not, delay the social engineering attack.

In this paper, we discuss the present deployable forms of MFA and how their functionalities and efficiencies differ from one another. Future improvements on the security of MFA are also presented.

II. BACKGROUND

All online accounts employ single-factor authentication requiring a single factor, the most common one being a password. This single layer of defense is usually lacking since it is the most easily obtained piece of information through the use of social engineering. To further improve security, a 1984 US patent titled "Method and apparatus for positively identifying an individual" first introduced the idea of adding a second factor in the authentication process known as two-factor authentication. 2FA, a subset of MFA, requires two factors of authentication in order to confirm a user's claimed identity. The most commonly used factors are something you know and something you have. Most MFA schemes use a one time password (OTP) that changes with time (e.g. every 30 seconds). In the 1980s, Leslie Lamport et al. [1] proposed an early form of a OTP scheme called S/KEY, which generates a unique, single-use password for dumb terminals or public computers. The scheme takes the original password and combines it with a short set of characters to generate the OTP. A decrementing counter keeps track of when the set of characters should be changed. S/KEY has fallen out of use with the rise of cryptographic protocols such as SSH that secure the whole session rather than just the password.

The two most widely known implementations of the OTP algorithm used today are HMAC-based one-time password (HOTP) and time-based one-time password (TOTP). HOTP was introduced as the informational IETF RFC 4226 in December 2005. In this scheme, the unique password is generated using two factors: the secret key, or seed, known only by the token and validation server, and a changing factor, or event counter, stored on the token and the server. The counter on the token is implemented when its button is pressed while the counter on the server increments on successful validation. To get a OTP, the token uses the seed as the key and inputs its counter into the HMAC algorithm, which in turn uses SHA-1 hashing to produce a 160-bit value to be reduced into a 6 or 8 digit password for the token. TOTP is an extension of HOTP and uses time in its algorithm. The large validation window of HOTP allows a number of valid OTPs since they are valid until use. This means there are less resyncs with the server and therefore less inconvenience for the user. However, there is a disadvantage in that the larger window gives attackers a higher chance of brute-forcing a valid OTP. TOTP is more prefered since it is short-lived and has a wider variety of available tokens. Authentication software such as Google Authenticator utilize TOTP and is available across almost all mobile devices.

III. THREAT MODEL

We have developed an attack tree for social engineering attacks on IoTs in the former paper.

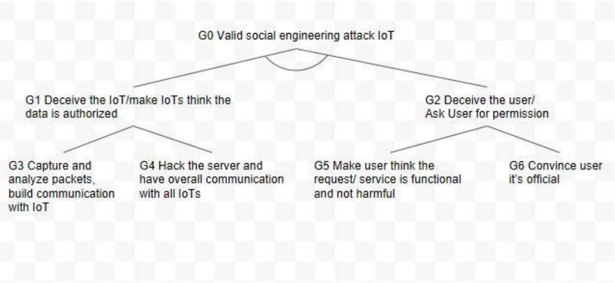

G0 Valid social engineering attack IoT
G1 Deceive the IoT/make IoTs think the data is authorized
G2 Deceive the user/ Ask User for permission
G3 Capture and analyze packets, build communication with IoT
G4 Hack the server and have overall communication with all IoTs
G5 Make user think the request/ service is functional and not harmful
G6 Convince user it's official

Figure 1. Attack Tree

The attacker targets the two most important points of IoT system - the devices interface and the users. Generally, users that are unaware of potential security threats have always been the weakest security point of the whole system. Thus, in this way it is unrealistic to assume the users have enough knowledge to prevent the social engineering attacks. So, one possible solution is to enhance the device's security through MFA.

IV. DESIGN

Some forms of MFA schemes include: Time-based One-Time Password (TOTP), Short Message Service (SMS), Electronic Mail (Email), and Push Notifications. These schemes are explained at a high level in an article from Auth0, written by Prosper et al [4] and are hereafter discussed:

For TOTP, when the user logs in with the correct username and password, they are prompted with a two-factor authentication which will require them to enter a shared-key that is sent to the user via an app (e.g. Google Authenticator, Auth0 Guardian). After entering the key can the user then successfully login.

For SMS, when the user logs in with a correct username and password, they are prompted to enter a valid phone number (one that is associated to the account). A unique one-time code is generated on the server and sent to the phone number. Upon entering the code into the MFA, the user will be authenticated and allowed access.

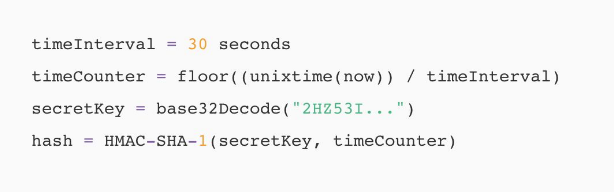
For Email, when the user logs in with a correct username and password, an email will be sent to their registered email account which contains a one-time code. The user can then enter that code to the app and if it is valid, they will be authenticated and allowed access.

For Push Notification, upon entering valid login information, the user is sent a push notification from another app (e.g. Duo Mobile, Auth0). When opening the notiaction, the user can then see a login attempt along with login details (e.g. app name, OS, browser, location, and date of request), which they can confirm or decline with a tap of a button. Upon pushing accept, the user will be logged in successfully.

V. IMPLEMENTATION

TOTP can generate a one-time password using a shared secret key and a current timestamp. The RFC 6238 paper [2] defines the TOTP encryption scheme. In the algorithm, $TOTP = HOTP(K, T)$, where T is an integer representing number of time steps between initial time T0 and the current Unix time. $T = (Unix\ time - T0) / X$, where X is the time step in seconds (default is 30s). Note that to make T an integer, it is rounded down by the floor function. Also note that since TOTP depends on time, the input for T gets larger as time progresses, being larger than a 32-bit integer after the year 2038 [2]. HOTP is defined as:
HOTP(K,T) = Truncate(HMAC-SHA-1(K,T)). Details for this algorithm is in RFC 4226 [3].

In an SMS authentication system, a unique identification code can be generated without the need for a one-time password to be produced locally on the user's phone [6]. In order for the server to authenticate the user, it will send a "unique" (usually 6-digits long) code to the user account's associated phone number. The user will then be prompted to enter this code into the device he is trying to login with. There is a set amount of time before the information is invalid, since codes are calculated for one-time use[9]. Generation algorithms can vary among companies; usually, the algorithm looks like this [9]:



```
timeInterval = 30 seconds
timeCounter = floor((unixtime(now)) / timeInterval)
secretKey = base32Decode("2HZ53I...")
hash = HMAC-SHA-1(secretKey, timeCounter)
```

The secretKey variable is the shared secret key between the server and user. The resulting hash is 20 bytes long; if the SMS code is 6 digits long, then a random 4-byte chunk is selected, treated as an integer, modified modulo 10^6 , and then zero-padded to give the 6-digit token. Data charge rates will apply since the method is through SMS messaging.

Email MFA methods are usually a 2-step verification process. The user will enter his account information on a machine to login. The server then sends a confirmation email to the account's registered email account with a code, link, etc. [7]. Once the user confirms through email, the server grants the login request. Most of the time, the server allows the user to request it to remember the machine so that email MFA is not required every login attempt. The algorithm will be similar to SMS authentication, if not the same [9].

Push notification MFA requires users to download an app, such as Duo Mobile, to their smartphones. Users must register this device to the app so that the server recognizes the devices as registered to the user account [8]. Thus, when the server recognizes a login attempt associated with the user's account, it can send a push notification to the user's associated device. Push notification relies on the user authorizing certain devices; the server will not and cannot send push notifications to unassociated devices. If authentication is received, then the request is granted. If not, then the request is not granted.

VI. EVALUATION

Multi-factor authentication is an essential step to protecting private user information, whether this step is small and simple, or long and complex. Even something as small as a push notification can increase user security greatly, since an attacker must now go through another step. With the dependency on time, it is difficult to attack encryption methods. Valid social engineering attacks can bypass MFA methods, however this requires much more effort (and especially time) for the attacker than simply revealing the user's password. MFA might become vulnerable if a user has been compromised on multiple avenues, such as having his phone stolen and email account hacked. However, because MFA requires more than one security measure to be broken, it guarantees much more security than not having MFA implemented. Therefore, because both bypassing MFA and breaking user passwords is a lot less practical than simply breaking the password, MFA is a good way to secure user accounts.

VII. RELATED SOLUTIONS

A. Biometric Data

Other, more recent, solutions to combat social engineering is to authenticate using not something that you know or have, but what you are. For example voice recognition, facial features, and fingerprints. This technology has become common with cellular phones and other smart devices, but it has come along with other potential security and privacy risks. Memon et al. [10] found that there was cause for concern for some fingerprint-based authenticators due to a potential vulnerability of a MasterPrint being generated which can then be used to impersonate the user. However, these authenticators often come equipped with near unbreakable encryption as seen in the 2016 FBI-Apple Encryption Dispute [11][12]. Unless an attacker somehow obtains a copy of the fingerprint, iris scan, or voice, there are minimal options for an attacker to obtain a digital copy of this biometric data.

B. Microchip Implants

While not so popular in America, “Biohacking” has become a trend in Sweden. Thousands of Swedes have implanted tiny microchips under their skins which can eliminate the need for credit cards, keys, tickets, and can even be used for authentication [12]. For example the media company Mindshare allows employees to simply enter their passcode and wave their wrist where the chip is embedded to unlock the doors of the building. This of course, like all technology, is hackable. However due to it literally being a part of the user, it is still harder for someone to access than most authentication method.

VIII. CONCLUSION

Although the concept has been around for decades it has just recently become widespread. Organizations have even started requiring MFA for all administrative accounts, and other companies providing incentives for users to opt-in. Granted MFA is not the end-all solution to security, there is no question that adding this additional layer of protection can mean the difference between a vulnerable user account and an ironclad security strategy.

REFERENCES

[1] RFC 1760 https://tools.ietf.org/html/rfc1760

[2] RFC 6238 https://tools.ietf.org/html/rfc6238

[3] RFC 4226 https://tools.ietf.org/html/rfc4226

[4] Prosper Otemuyiwa (Nov 2016). “What are the different ways to implement Multifactor Authentication?” https://auth0.com/blog/different-ways-to-implement-multifactor/

[5] Ometov, Aleksandr & Bezzateev, Sergey & Mäkitalo, Niko & Andreev, Sergey & Mikkonen, Tommi & Koucheryavy, Yevgeni. (2018). Multi-Factor Authentication: A Survey. Cryptography.

https://www.researchgate.net/publication/322288752_Multi-Factor_Authentication_A_Survey

[6] Aloul, Fadi, et al. “Multi Factor Authentication Using Mobile Phones.” https://pdfs.semanticscholar.org/2599/ad2d3b40a47b7d7816b28f2791d4edb95109.pdf

[7] Google 2-Step Verification https://www.google.com/landing/2step/index.html#tab=how-it-works

[8] Kathleen Garska. (2018). “Two-Factor Authentication (2FA) Explained: Push Notifications” http://blog.identityautomation.com/two-factor-authentication-2fa-explained-push-notifications

[9] Chris Birchall. “How Virtual MFA Tokens Work”. https://tech.ovoenergy.com/mfa-tokens/

[10] Roy, Aditi & Memon, Nasir & Ross, Arun. (2017). MasterPrint: Exploring the Vulnerability of Partial Fingerprint-Based Authentication Systems. IEEE Transactions on Information Forensics and Security. PP. 1-1. 10.1109/TIFS.2017.2691658.

[11] Pym, Sheri (February 16, 2016). "Order Compelling Apple, Inc. to Assist Agents in Search" (PDF). United States District Court for the Central District of California.

[12] “Why Thousands of People in This Country Got Microchip Implants.” *South China Morning Post*, South China Morning Post, 13 May 2018