Assignment 3.2: Multi-Factor Authentication

Team 23: Harish Saravanakumar, Ikra Rashid, Haihua Wu, Tianshu Li

1 Abstract:

Multi-Factor authentication aims to strengthen regular password-based authentication by forcing users to provide one [or more] additional authentication factors. Examples of these factors include a one-time throwaway password to a range of different security tokens. However, it also introduces non-negligible costs for service providers, and increases the length of the authentication process for all users. In this paper, we will relay two main points. These include what Multi-Factor Authentication is and how it works, its challenges and benefits, and whether multi-factor follows different design principles from our Assignment 3.1 paper - authorized use of open ports, user authentication, consistent sub-application patching, and password safety. We find overall that multi-factor technologies are cited as the most common fix for the shortcomings of passwords, however only as a general extra security measure to passwords.

2 Introduction

Despite the numerous technologies available to secure accounts, most companies opt for a simple username-password combo as the main source of end-user authentication. Since passwords are so prevalent, almost every user is forced to "password recycle". Users reuse passwords across multiple platforms, which means if an attacker obtains a password through an insecure account, they might accidentally trip onto one's online bank account details. One way to mitigate the damage caused by these password breaches to add at least one authentication factor. This idea, named multi-factor authentication(MFA) is defined as the use of more than one factor from different types of authentication.

Authentication factors are usually separated into 3 different categories - *something you know*, *something you have*, and *something you are* [1]. **Something You Know** is anything the user can remember and then type, say do, perform, or otherwise recall when needed, for example, a password. **Something You Have** is anything the user can use to respond to a command issued by the server, for example, a smartphone or token device. **Something You Are** is anything that uses the human body which can be used for verification, for example, fingerprint scanning or facial recognition.

Multi-factor has been around for a while, mostly used by governmental and enterprise entities, where the high sensitivity of information has forced them to secure information in different ways. For example, most high-level corporations force users to add a numeric password to their mobile devices when attempting to link their work email to their devices. This, in addition to multi-factor authentication offers multiple layers of security, barring social engineering ploys and password breaches. More recently, a larger number of companies, especially social media, provide users with multi-factor authentication, most likely due to the increasing number of accounts hacked on a daily basis. The increase of MFA use shows the importance of dispelling its ubiquitous nature as an end-all be-all for authentication security. In this paper, we discuss the different kinds of multi-factor authentication used, the challenges and benefits with each type, and how we must design multi-factor in order to align with our threat model and various Internet of Things design principles.

3 Background

Multi-factor authentication has become increasingly popular, adopted by educational institutions and corporate organizations. There are several options such as *single factor authentication(1FA)*, *two factor authentication(2FA)* and *multi factor authentication(MFA)*.

The most commonly known login procedure is the username-password combination. Unfortunately, this method proves to be insufficient because hackers have multiple ways of accessing a user's information. An example of an attack is one where hackers try to login with passwords from a very long list of common ones. There are many different ways to improve the security of password storage. The user's password can be stored in multiple ways, ranging from plaintext, encryption through a hash function, or storage as a salted

password. Some entities have requirements for passwords, so that they are more secure and memorable. 2FA is more secure because the password is secured in conjunction with other authentication factors such as tokens, smart cards or biometrics.

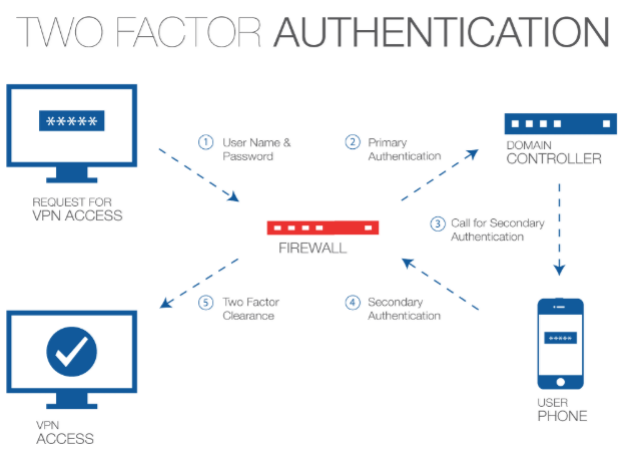

Figure 1: How Two Factor Authentication Works

A user requests access to a system. The system asks the user to authenticate through a firewall. At this point, the user will hit a domain controller which will deem whether the user is valid. It then sends a request to a cellphone or any other device. After approving the access request, the request is sent to the firewall, granting VPN permission access to the user.

Hardware tokens are considered to be one of the older physical device methods of MFA. Its most common form is a key fob. The key fob has a "display showing time-based OTPs" [2]. Using hardware tokens, users face the risk of losing their physical token, an expensive commodity to replace. As companies upgrade their hardware, they are required to spend more with regards to compatibility. In cases where the token is not compatible with other devices, the user will have to buy adaptors, increasing their costs as well. A mobile token is considered to work like a hardware token because users utilize their phones, however it can pass off as a software token as well because a mobile app is being used. This is very convenient for a user because they will almost always have their smartphones. However, there are situations where one might not have a smartphone or not have access to it at that moment. In terms of security issues, if an attacker is able to gain access to a copy of the QR code, they will have a clone of the user's token. Smartphones can also be utilized to maintain an app that handles push-based authentication tokens. The user will send the request to login, and their app will receive an encrypted push message. The user can then accept the request, and in turn the token will generate an OTP internally on the user's phone and send the approval back for verification [2]. When using push-based authentication, there is a risk that an attacker might be able to mimic or spoof the push approval message.

Software tokens are programs that exist on systems and change over time. Some programs deal with one time passwords. There are three approaches to one time passwords. The first approach is when the user has a list of passwords that the application is aware of. Each of these passwords is used only one time. This is a difficult system to maintain. In another approach, the user signs in with one password, and then the system generates the next password to use for the next time the user logs in. The security concern with this approach is that an attacker can find all the passwords if they are to find the very first password that was used. Leslie Lamport devised an approach that utilized hash functions that would be used by the system to update a current password. The one time passwords can be delivered by SMS which rid the need for physical tokens. Although this seems effective, SMS messages are easy to hack because they are unencrypted; therefore users who use one time passwords are encouraged to seek different methods.

There are many benefits to using multi-factor authentication. Multi-factor authentication adds layers of protection for an application, which increases the challenges attackers would face when trying to surpass the number of security layers. Another benefit to MFA is that it allows organizations to implement compliance requirements that protect the company from potential files and better mitigate audit findings. While the usage of MFA depends on the company's implementation and usage, MFA can aid in increasing flexibility and productivity while reducing operational costs. Systems that require two-factor authentication for each login, help assure that users are no longer able to choose the "Keep Me Logged In" option. The problem with the "Keep Me Logged In" option is that the cookie that is saved on the computer can be discovered by the attacker [3]. Two-step authentication also helps to build secure brand experiences because the customer and their transaction is more secure. Overall, using multi-factor authentication helps users to prove their identity and securely use their applications.

There are also many challenges that experts must address to make sure users are well protected from attacks. One of the biggest challenges is MFA's cost. This is because MFA requires more layers of security, which calls for more resources. Multi-factor authentication proves to be a difficult discussion because there is no perfect solution as to how users can be protected the most. MFA's effectiveness depends not only on its innate security, but the user's as well. As mentioned earlier, there are situations where the physical token can be lost or stolen. In these situations, the user could easily be targeted while waiting for a replacement. The replacement would also cost both the user and the com-

pany. SMS messages may give users the impression that the method is safe, however they are easy to attack because SMS is not encrypted. Some methods, required by companies, might also just be difficult for employees to follow due to lack of resources. The cyber security field continues to search for solutions that will best benefit everyone.

4 Threat Model: *Refer to page 6*

4.1 Threats G6 and G12:

Both Phishing and Trojans can be easily avoided just through common sense, or by asking whether the source is familiar, if there are spelling errors, or whether the "offer" is too good to be true.

4.2 Threats G8, G14, and G11:

These all fall under the user not securely holding a part of their multi-factor authentication (password, device, or biometric). It is crucial to keep all parts safe, so the attacker is always in a position where they cannot pass the authentication.

4.3 Threats G10:

Many devices and accounts come with default passwords. By creating more secure passwords, the chance of one's devices being hijacked reduces greatly.

4.4 Threats G15 and G16:

To stop attackers from doing this, many mobile carriers allow the user to add a PIN to their smartphone account, creating a tight third layer of protection.

4.5 Threat G3

The best long-term solution would be to hope companies send out patches quickly, and to turn on auto-updates whenever possible.

5 Design:

The proposed solution should not only attempt to resolve the threats laid out in the threat model, but also follow the Internet of Things design principles laid out in the previous paper:

- authorized use of open ports
- user authentication
- consistent sub-application patching
- password safety

Since Multi-factor authentication systems are more reliable yet more expensive than ones with single-factor authentication, we must balance the amount of security with the amount of money we are willing to pay. Although some more expensive systems that use biometrics as an authentication method are harder to attack, we still need to discuss the threats described above.

6 Implementation:

In order to increase the security of MFA, we must attempt to integrate not only the IoT design principles discussed in the previous paper, but also solutions to threats from our threat model. The threats discussed are partly due to user error securing different authentication factors [ex. passwords, devices]. As discussed before, attackers will always pick the weakest link in an authentication system to attack. Since many of the threats to MFA are in conjunction with our IoT design principles, they will be discussed first.

- **Authorized Use of Open Ports:** There is no real problem here, since unlike Assignment 3.1, MFA is not based on a Linux kernel system. Every use of a port must travel through a firewall before being granted VPN permission access.
- **User Authentication:** This is the core of many problems, since there are a plethora of different types of authentication (password, app, and biometric) with their own security pitfalls.
- **Consistent Application Patching:** Although the design principle has been slightly altered, the idea still stands: apps must be consistently patched in order to secure from already "solved" threats. Although this is not a feature attached to MFA, this is an important factor regarding the security of the authenticating device.
- **Password Safety:** Password safety is mostly under the user's control; i.e. a more secure password, the better. Depending on the situation, storing passwords in physical space might not be the best idea.

We can divide the threats discussed above into two major categories: **User Dependence** and **MFA Dependence**.

6.1 User Dependence:

Many threats to MFA are based on how securely the user keeps their information. Users must be careful of phishing and trojan attacks, as they are the most common way attackers bypass MFA, along with keylogging.

Phishing is when an attacker attempts to get a user to perform an action for them. This can be logging into a fake webpage by making it look like the real webpage. This can also be asking someone to transfer money by making them believe you are someone you aren't. A Trojan often takes the form of a piece of software or an attachment in an email, and once you give it permission to install on your machine, it can gain complete control of your PC.

Mobile carrier hijacking is a common way for attackers to bypass two-factor authentication, where the authenticating device is one's mobile phone. They don't steal the phone, but rather just hijack the phone number. By doing this, they are able to intercept one-time verification codes sent to the mobile number using text, phone call, etc. By using a user's available personal information [information such as date of birth or last 4 digits of SSN is readily available through the dark web], attackers can impersonate the user to a mobile carrier. From here, attackers transfer their target's phone number onto a new device that they have access to and can control. This not only affects availability, but allows attackers to control one part of the multi-factor authentication process. As discussed before, the user can easily secure their mobile carrier account to prevent attackers from stealing their devices by adding a PIN to their smartphone account, which in turn, creates a tight third layer of protection.

Users should also be on guard regarding any instances of social engineering or security of physical copies of authentication factors [passwords, tokens, etc], since these are practically impossible to secure otherwise. Finally, the first authentication factor, the password is one created by the user. As a result, users are advised to change defaults and secure their accounts with more complex and uncommon passwords.

6.2 MFA Dependence:

The main threat discussed in this section is Inconsistent Application Patching of MFA Apps. Although the chances of an attacker attacking a third-party app is fairly small, if a breach is found, attackers can breach the users that have not patched their applications. For example, popular password manager LastPass's 2FA secret keys could be accessed without a fingerprint, password, or other security measures. By accessing the LastPass Authenticator app's settings activity, one could enter the settings pane for the app without any checks. From there, an attacker could press back once to access all of the authentication codes. However, it took LastPass seven months to patch this for its application [4]. We discussed the long-term solution would be to hope companies send out patches quickly and to turn on auto-updates whenever possible. However, during this time, the multi-factor authorizer, if applicable, should disable all devices affected from the breach from being used as authentication methods and allow every user to use their backup code [similar to Google Authenticator, which comes with the creation of the account] for users to use while the application is patched.

7 Evaluation:

We quickly realize that the problem with multi-factor authentication systems revolves around how insecure each factor can be in certain circumstances. To a non-tech savvy user, it may be even more difficult to uphold privacy and integrity because of attacks such as phishing and trojans. Ideally, we want the most security possible, since no matter what level we are at, there is no such thing as a completely safe authentication method. With biometrics, it becomes increasingly harder for attackers to impersonate the user, but this method has an extremely high cost of implementation. However, even biometrics are not completely secure, since in some circumstances, they can be bypassed.

As shown by the threat model, the majority of the threats to MFA are things that can only be prevented through user dependence. Many solutions require the user to install extra security measures such as mobile carrier PIN codes. This not only requires the user to be tech savvy, but forces the user to spend extra time fortifying the various authentication methods. After analyzing the implementation of these solutions to threats in an MFA context, we have come to the conclusion that although MFA is very reliable as a vessel for security, it does not guarantee it.

8 Related Solutions:

Companies aim to increase employee productivity while also maintaining high levels of security when discussing ways to authenticate their employees. One possible solution is to restrict access to connection types such as wi-fi and VPN. This way employees are only able to login based on their locations or specific times throughout the day. Using this method of safety would make things difficult for the attacker because even if they knew the user's username and password combination, they would not be able to gain access and the administrators would receive notifications of these attempts [5].

Another alternative solution is one where physiological biometrics are taken into consideration.Vivek Khandelwal, VP Business Development at Delta ID, states that using both a smart phone and passive physiological biometrics will help increase user's security against attackers. When a user logs in with both a password and username combination with a fingerprint, there is a very small chance that an attacker can gain access to unauthorized data. Khandelwal mentions that there is a "false accept error rate of 1 in

tens of thousands"[6]. He then states that using one's iris to log in is the equivalent of having a 6-digit passcode. Though one might wonder as to why the 6-digit passcode isn't favored since they have the same modality, it is easier for the user to not have to remember their password. Others will not be able to use social engineering, eavesdropping, or overseeing to figure out the user's password if the iris is used since it is a biometric. Although this appears to be a good alternative, the technology has not been advanced enough to where society can completely stop using the SMS authentication. The costs for companies to implement biometrics in their everyday systems are extreme as well.

9 Conclusion:

In this paper, we discussed several variations within options that users have to securely log in to their applications. The simple one-factor authentication is insufficient because attackers only have one layer of security to go through. As more layers of security are added on, along with increased protective measures for storing passwords, the attacker will face more obstacles. Hardware tokens and biometrics cost companies heavily so the advancements in that field have slowed down. Software tokens can be easier to use but difficult to maintain (especially one time password algorithms) and promise protection from attackers. Multi-factor authentication addresses issues such as phishing, trojan attacks, intercepting push notifications, social engineering and using default passwords. The threats to multi-factor authentication prove that although there are ways that technology can help to overcome some vulnerabilities, it is also up to users to educate themselves on technology to help protect themselves from attacks, given how quickly technology is advancing.

10 References:

1. "Pearson IT Certification." 7 Popular Layer 2 Attacks Pearson IT Certification, www.pearsonitcertification.com /articles/article.aspx?p=1718488.

2. Cagnoni, Alexandre. "Not All Multifactor Authentication Is Created Equal." Dark Reading, 11 Oct. 2018, www.darkreading.com/endpoint/authentication/not-all-multifactor-authentication-is-created-equal/a/d-id/1332991.

3. French, Steve. "2-Factor Authentication: What It Is & Benefits." OpenMarket, 17 Jan. 2014, www.openmarket.com/press/enterprise-mobility-protecting-business-employees-customers-two-factor-authentication/.

4. Price, Dan. "It's Time to Stop Using SMS and 2FA Apps for Two-Factor Authentication." MakeUseOf, 15 Feb. 2018, www.makeuseof.com/tag/two-factor-authentication-sms-apps/.

5. Amigorena, François. "It's Time to Find an Alternative to Multi-Factor Authentication." SC Media, 6 July 2018, www.scmagazine.com/home/opinions/its-time-to-find-an-alternative-to-multi-factor-authentication/.

6. Townsend, Kevin. "NIST Denounces SMS 2FA - What Are the Alternatives?" Information Security News, IT Security News and Cybersecurity Insights: SecurityWeek, 17 Aug. 2016, www.securityweek.com/nist-denounces-sms-2fa-what-are-alternatives.

7. Townsend, Kevin. "NIST Denounces SMS 2FA - What Are the Alternatives?" Information Security News, IT Security News and Cybersecurity Insights: SecurityWeek, 17 Aug. 2016, www.securityweek.com/nist-denounces-sms-2fa-what-are-alternatives.

8. Carter, Samuel. "Multi Factor Authentication Terms and Factor Types - MFA 101, Part 1." Identity Automation Blog, 22 May 2017, blog.identityautomation.com/multi-factor-authentication-terms-and-factor-types-mfa-101-part-1.

9. Cristofaro, Emiliano De, et al. "A Comparative Usability Study of Two-Factor Authentication." Proceedings 2014 Workshop on Usable Security, 31 Jan. 2014, doi:10.14722/usec.2014.23025.

10. Villadiego, Ricardo. "The Future Of Authentication Is Here." Forbes, Forbes Magazine, 4 June 2018, www.forbes.com/sites/forbestechcouncil/2018/06/04/the-future-of-authentication-is-here/#58c6d4aa432e.

11. Schneier, Bruce. "Schneier on Security." Blog, 1 Apr. 2005, www.schneier.com/essays/archives/2005/04/two-factor_authentic.html.

12. Brandom, Russell. "Two-Factor Authentication Is a Mess." The Verge, The Verge, 10 July 2017, www.theverge.com/2017/7/10/15946642/two-factor-authentication-online-security-mess.

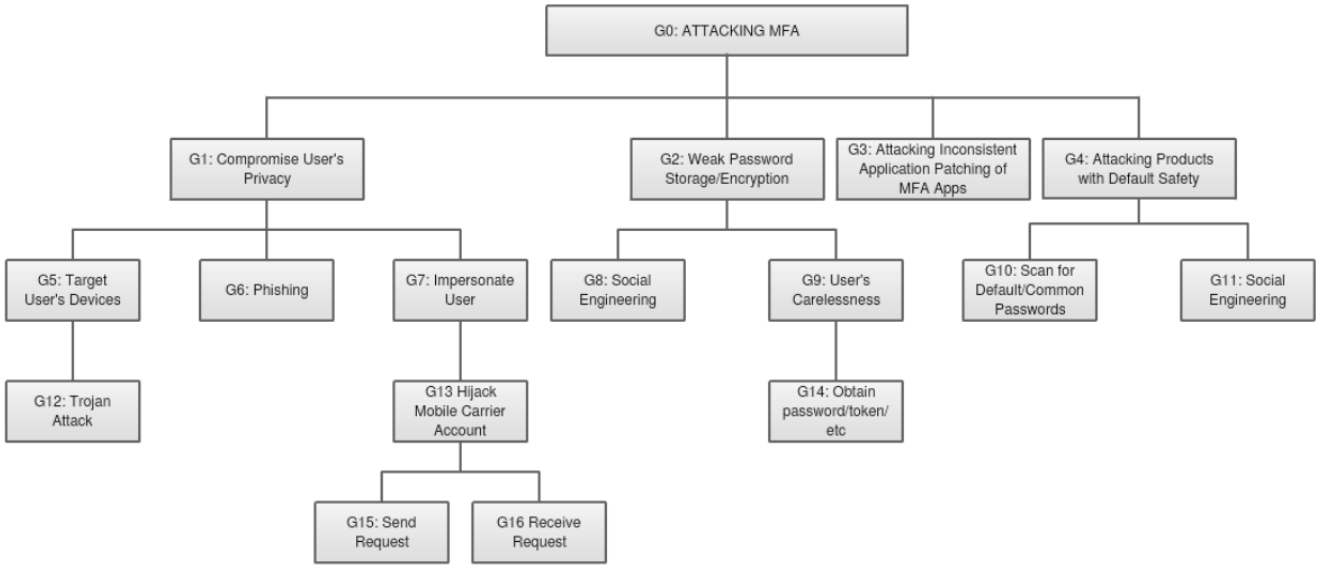

Figure 2: MFA Threat Model