Piotr LEWANDOWSKI, Anna FELKNER, Marek JANISZEWSKI

NASK – Research and Academic Computer Network

doi:10.15199/48.2019.08.29

Security analysis for authentication and authorisation in mobile phone

Streszczenie. *Artykuł zawiera analizę bezpieczeństwa wykorzystania telefonu komórkowego jako istotnego elementu procesu uwierzytelniania użytkownika w systemach teleinformatycznych (np. systemach SCADA). Analiza bezpieczeństwa obejmuje zarówno same metody uwierzytelniania jak i wykorzystanie telefonów komórkowych oraz sieci komórkowej w procesie uwierzytelniania. W podsumowaniu analizy bezpieczeństwa wskazujemy aplikację do generowania haseł jednorazowych jako rozwiązanie zarówno przyjazne dla użytkownika jak i bezpieczne.*
(Analiza bezpieczeństwa metod uwierzytelniania i autoryzacji z wykorzystaniem telefonu komórkowego).

Abstract*. In this paper we discuss some authentication and authorisation systems where mobile phone is a main or an important component to improve security. Some of the presented solutions are available for SCADA software. Based on our analysis we list and compare safety measures and threats in mobile phone's technologies. We also briefly analyse the security models of the most popular solutions. Results of our analysis point out that the application generating one-time passwords is both secure and convenient for the users.*

Słowa kluczowe: Bezpieczeństwo i ochrona prywatności, Kontrola dostępu, Mobilne systemy operacyjne, Sieci komórkowe
Keywords: Access control, Mobile communication systems, Security and Privacy Protection, Telephony

Introduction

Supervisory Control and Data Acquisition (SCADA) systems play a key role in monitoring industrial processes. Due to bidirectional connection with Programmable Logic Controllers (PLCs) and Remote Terminal Units (RTUs), SCADA systems are able to track process's parameters as well as alter required levels of parameters or even levels triggering alarms. This makes SCADA system a very tempting target for cyber attacks.

The most popular way of restricting access to systems or applications is the requirement to authenticate the user using login and password. To enhance the security of the user authentication process, more than one authentication factor can be used. To log in to the application, alongside something the user knows (login and password), the user must be in possession of something like a token or one-time password generator, and also has to prove his identity by using biometric methods. It seems that the mobile phone is a perfect choice for ”something that the user has” as it can be multi-factorial in many ways with additional layers of security, as well as many sensors embedded in nowadays smartphones (such as a camera, fingerprint reader, iris scanner, etc.)

To make the security solution complete, it should be convenient for the user as well as secure. To achieve the required level of convenience, SCADA systems could incorporate one of a Single Sign-On technology to make logging in process seamless with logging in to PC.

Possible implementations of Multi-Factor Authentication (MFA) as well as Single Sign-On (SSO) technologies will be described in the following sections.

Single Sign-On technologies

Single Sign-On is a technology of sharing authentication or authorisation token between applications. User has to log in once and then access to any connected application is granted automatically because one application can automatically send the appropriate authentication or authorisation token in the background. Because of that the whole process is seamless for a user. It reduces the number of logins and passwords to remember as well as reduces the risk of credentials breach. Another benefit of SSO can be the easier deployment of multi-factor authentication. As SCADA systems may be able to use SSO but not the Multi-Factor Authentication, the MFA service may be enabled on logging in to the application being the SSO's authorisation and authentication centre.

There are two major SSO standards: SAML (Security Assertion Markup Language) and OpenID with OAuth. These will be described in the subsequent sections.

Security Assertion Mark-up Language (SAML)

The first version of SAML has been published in 2002 [1]. Currently all major software vendors, like Microsoft or Google, are using second version of SAML from 2005 [2,3]. This means that SAML 2.0 is a well-known and well tested mature technology. Token is an XML document with the assertions about user's authentication, attributes and authorisations. Token is exchanged between applications with secure HTTPS communication and can be digitally signed to prove its authenticity and integrity. SAML 2.0 is best suited for web application while integration with mobile or desktop applications may be a hassle [4].

OpenID with OAuth

OpenID has been introduced in 2006 as the standard for sharing user authentication between web services with a user's Uniform Resource Identifier (URI) [5]. In 2010 the OAuth standard has been published [6]. OAuth gives an ability to share an authorisation token between services, so user can share resources or information gathered in one service with another without sharing credentials. In 2014 OpenID 2.0 was merged with OAuth 2.0 to create OpenID Connect 1.0 which combines abilities of these two standards so it can share authentication and authorisation [7]. OpenID Connect has been designed for mobile and desktop applications, which is why it is more versatile in integration than SAML. Moreover OpenID Connect allows the user to see which data will be shared and decide if one gives or rejects authorisation to this data [8].

Single Sign-On security

Both presented solutions: SAML and OpenID Connect use open text to share authorisation and authentication tokens. Thus it is very important to properly secure the transportation layer using the SSL/TLS protocol. There are also other means of security to be considered during integration however it is out of the scope of this paper [8].

SAML as well as OpenID Connect are nowadays the standard used by tech companies like Google, Microsoft, Facebook, Twitter and so on to connect many applications from different vendors.

Mobile phone as second factor of authentication

As mentioned before, it is a good security practice to incorporate multi factor authentication. Mobile phones are equipped with many technologies that can serve as second factor authentication or at least as an additional communication channels. This makes the mobile phones a good choice to increase the security of the authentication or authorisation process.

From a functional and technical point of view there is a very small difference in the use of a mobile phone as the first or the second factor during authentication or as a tool for an authorisation of certain actions, which is why we will treat these cases indifferently. The general idea of using a mobile phone for authentication or authorisation is presented in Figure 1. The security analysis of such mechanisms has to be carried out on the basis of analysis of channels of communication and the mobile operating system's security.

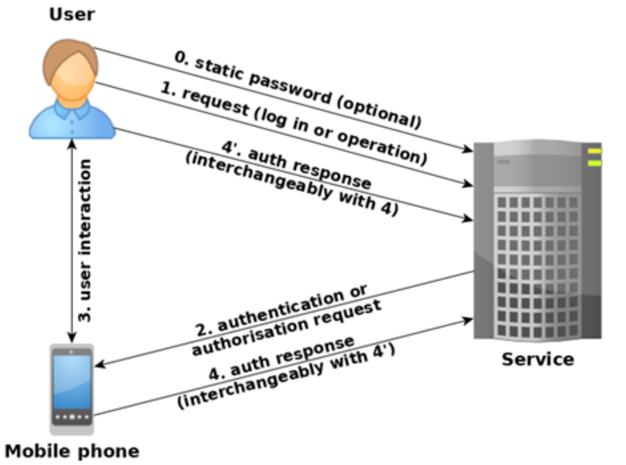
User
0. static password (optional)
1. request (log in or operation)
4'. auth response (interchangeably with 4)
3. user interaction
2. authentication or authorisation request
4. auth response (interchangeably with 4')
Service
Mobile phone

Fig. 1. General model of authentication or authorisation with mobile phone

Existing solutions for multi-factor authentication

There are many solutions (both commercial and open) that use the mobile phone as a second factor of authentication. We would like to present some of them in the following sections.

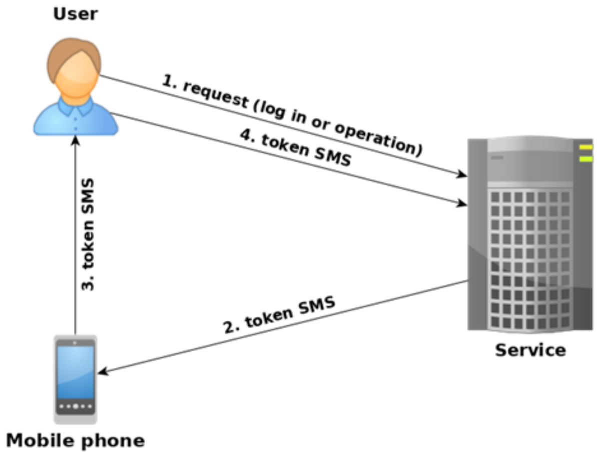

Fig. 2. Authentication or authorisation mechanism with one-time password sent via SMS

One-time password sent over SMS

The most popular solution is a one-time password sent via SMS. It is very popular among, for example, online banking services. The idea of this system is presented in Figure 2. The service sends an additional one-time passcode over a separated communication channel – an SMS to the mobile phone that the user ought to have.

This solution has a number of advantages. The user does not need to install any additional applications on one's mobile phone. Another benefit of this solution is the fact that it works on all types of mobile phones, not only smartphones. It is also important that the description of the on-going operation could be included in the SMS message, which is very important for the security reasons.

Unfortunately, it is not very convenient for the company to send text messages to employees during the logging in process, as it is an extra cost.

One-time password generated using the application

Currently, mobile applications for generating one-time passwords on the client's side are gaining in popularity solution. Examples of such applications are Google Authenticator, Microsoft Authenticator or Authy. Most of them use the TOTP (Timebased One-time Password) algorithm [9] or HOTP (HMACbased One-time Password) algorithm [10]. To generate tokens, the user has to gain a secret (string of random letters and numbers) from the service. After entering the secret to the application, the algorithm generates a new token periodically (most often every 30 seconds) or at will. This mechanism is presented in Figure 3. After the user has been authenticated with login and password, one has to generate a new token and provide it on the login page.

As the TOTP and HOTP algorithms are well described in RFC it is possible to implement them by the company or use one of the available free or proprietary libraries or mobile applications.

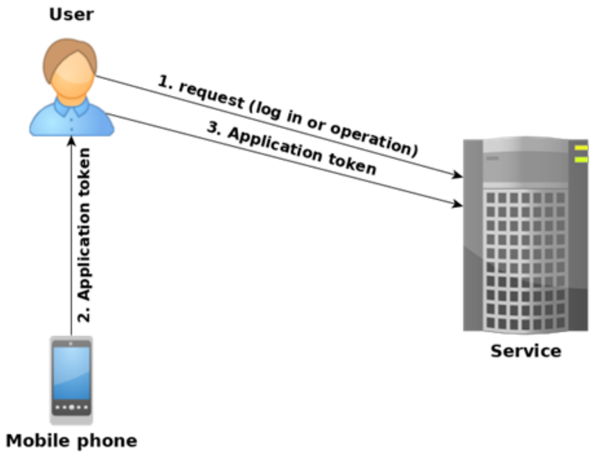

Fig. 3. Authentication or authorisation mechanism with one-time password generated by the application

Push notifications in the application

Another way to verify the user is to use a custom mobile security application paired with the user account. To authenticate or authorise, the user has to log in to the mobile application (with a PIN code or biometric) and accept the notification sent from the service over an encrypted HTTPS connection with the push mechanism. These steps are presented in Figure 4.

Some desktop applications can be paired with already available commercial solutions like Microsoft Authenticator or DUO [11,12]. Sometimes it is possible to develop your own dedicated web service and mobile application to be used with the company's system. It is a really convenient solution for users, as they only need to accept notification on their smartphones.

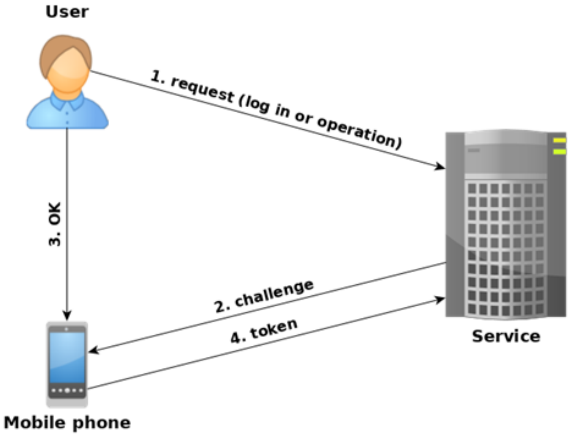

Fig. 4. Authentication or authorisation mechanism with token confirmed in application

Using SIM cards with cryptographic keys

Modern SIM cards are capable to store cryptographic keys that can be used by SIM applets (small applications installed on the SIM card) to perform cryptographic operations like encryption, decryption and signing. In the registration process the user gets a SIM card with a private key. To authenticate the user, the service uses the gateway to mobile networks to send notification over the USSD protocol (Unstructured Supplementary Service Data). Depending on the service requirements for security, the user has to accept the notification or enter the PIN code to make a SIM applet authorise the operation. This process is shown in Figure 5. As this method of authentication requires cooperation with mobile networks operators it is chosen rather by governments, for example Estonia, Finland or Azerbaijan.

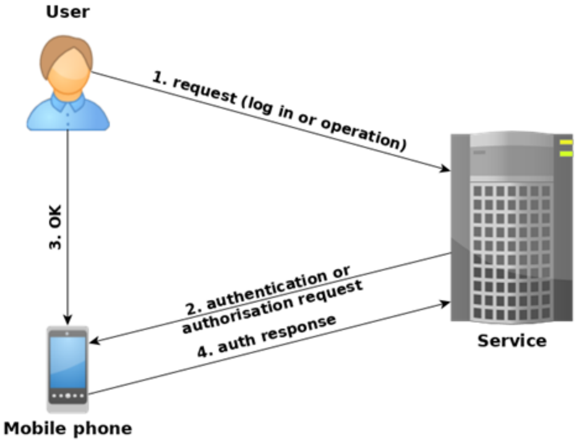
User
1. request (log in or operation)
3. OK
2. authentication or authorisation request
4. auth response
Service
Mobile phone

Fig. 5. Authentication or authorisation mechanism with token sent over USSD

Mobile security

Mobile phones seem to be a convenient additional factor of authentication, however, one must be aware that most of them have a fully functional operating system and they are continuously working in the network. These properties need an in-depth security analysis.

In this section, we discuss the security mechanisms and present some of known weaknesses of mobile telephony in the context of the network as well as phones.

Mobile network security

The GSM (Global System for Mobile Communications) standard was introduced in 1987 [13]. With the development of mobile network coverage and the need for better security and new features, new standards like UMTS (Universal Mobile Telecommunications System) and LTE (Long Term Evolution) have emerged. In spite of all benefits of new standards it is impossible to update all networks so there are still places where users must use older standards for communication. Various mobile networks communication standards provide different security mechanisms, but each has some serious security flaws.

GSM (2G)

Due to the fact that GSM standard was developed during the Cold War, its cryptographic mechanisms constitute a compromise between security and the ability to get around it (for example by the intelligence agencies). GSM can work in four modes of encryption [13]:

- A5/0 – without encryption
- A5/1 – standard stream cipher used in GSM networks. The first version was flawed, because instead of using 64 bits keys as in the specification, the real key was 54 bits long (last 10 bits were zero). The second version uses real 64 bits keys.
- A5/2 – stream cipher developed in the late 1980s, weaker than A5/1 because it was invented for sale in Eastern Bloc countries.
- A5/3 (KASUMI) – the latest stream cipher used in GSM and UMTS networks. Based on the Mitsubishi MISTY1 algorithm.

UMTS (3G)

The 3G standard was developed in 2000 as the worldwide standard for wireless communication by the 3GPP (3rd Generation Partnership Project) convened by many organisations from Europe, USA, Japan, India, China and Korea. UMTS is based on the GSM concept but it is not backward compatible (there are differences in the used: frequencies, multiple access techniques, cipher encryption, methods of SIM card authentication and many others) [13]. Encryption of communication between the mobile device and the network can be done with one of two ciphers:

- KASUMI – described in the previous section
- SNOW 3G – this is a 128 bit stream cipher developed by the Security Algorithms Group of Experts (SAGE) based on SNOW 2.0 in 2006. It has been added to 3G specification to answer concerns about vulnerabilities in KASUMI cipher [14].

LTE (4G)

The LTE standard was approved by 3GPP in December 2008. This is the evolution of the 3G standard. It incorporates more security features like new ciphers, more secure SIM card authentication protocol and faster data transfer [15]. Communication in LTE networks can be encrypted with one of three ciphers:

- SNOW 3G – described in the previous section
- AES-128 – a well-known block cipher utilizing 128 bits keys
- ZUC – stream cipher developed in China using 128-bit keys

SS7 protocol

SS7 or Signalling System #7 is a set of protocols developed in 1980s and used for communication between mobile networks' operators infrastructure. One of the key feature of SS7 is roaming – the ability to conveniently handle users' transfers between Mobile Switching Centres (MSC) in one network or even between different networks (i.e. the ability to use the phone abroad, away from native network) [16].

Vulnerabilities of mobile networks

Even though every next generation of mobile network introduces new, stronger ciphers and other security means it is only a matter of time when we can hear about successful attacks on the confidentiality of communication in mobile networks.

GSM (2G)

All three ciphers (A5/1, A5/2 and A5/3) have been cracked. Communication encrypted with the A5/1 cipher can be decrypted with rainbow tables in the real time [17]. KASUMI (A5/3) can also be broken in a short time (around 2 hours with an Intel Core 2 DUO CPU), but it has not been tested on real world GSM communication [18].

UMTS (3G)

In 2010 two articles with two different attacks on SNOW 3G were presented [9] [10]. In spite of weaknesses of the SNOW 3G algorithm, there is no known attack in the real life scenario on the users of mobile phones.

LTE (4G)

According to [19,20] LTE networks are prone to attacks on communication's privacy and authenticity. Authors of [20] claim that most of attacks can be carried out using devices cheaper than 4000 US dollars. However, there is no direct evidence of such attacks on mobile phones' users.

SS7 protocol

SS7 is based on the assumption that every mobile network is trusted, so there is no authentication between networks. Access to one network can be used to access any other network. This leads to the ability of eavesdropping on users calls. Having an access to the network can also enable querying the network about the last known MSC for a particular user. This information can be used to track user's location, because of the fact that location of MSCs are known [17].

SIM card cloning

A SIM card is a kind of ID card for a mobile phone. It stores the International Mobile Subscriber Identity (IMSI) number and a secret key or keys to encrypt communication with the network and to authenticate that card (and of course the user). Having a clone of the SIM card gives attacker the opportunity to intercept all communications intended for the owner of the original one.

In paper [21] the authors show the ability of cloning some models of SIM cards working in 2G, 3G and even 4G networks. It takes 10 to 80 minutes to clone the SIM card so it is possible to perform such attack on an unattended mobile phone. Another approach is to try to possess a copy of the SIM card from the mobile network operator. Often it is just a matter of social engineering to convince the staff in the mobile network operator's store.

Fake base stations (IMSI catchers)

Mobile phones automatically switch to the nearest base station with the best signal range. This feature can be abused by an attacker with a fake base station. Running a fake base station (IMSI catcher) with a strong signal will cause all nearby mobile phones to connect to such a fake base station. Being in control of IMSI catcher allows forcing weak or no encryption of communication between mobile phones and a fake base station and for eavesdropping on voice calls, text messages and also data transferred without additional encryption (e.g. HTTPS). If the attacker connects the IMSI catcher with a real mobile network, one can obtain the key used for communication encryption and listen to the user's calls. This is done in the same way as standard roaming where new network must obtain encryption keys from the user's native network. An attacker can also impersonate any telephone number. This can be used for phishing [22].

Mobile phone's operating systems security

Smartphones started to gain popularity around 2005, for example BlackBerry and Nokia E series phones belong to this category. In 2007 Apple launched the first iPhone, and in 2008 the first Android phone was showed (HTC Dream also known as T-Mobile G1). Now over 90% of smartphones works with the Android or iOS operating system and therefore there is a need to analyse security solutions implemented in these two types of mobile operating systems.

Android

The first released version of Android was 1.5 ”Cupcake” in 2008, the latest version is 9.0 ”Pie”. Many security features have been added over the past 10 years and many vulnerabilities have been fixed.

Security features in Android

Android runs on a modified Linux kernel and incorporates some of Linux security mechanisms like:

- Application permissions and the isolation model based on unique identifiers,
- Security-Enhanced Linux (SELinux) – an additional access control mechanism for applications and services,
- Binder – secure inter-application communication interfaces.

Other security mechanisms in the Android system are:

- Application sandboxing – to separate the application process and prevent the application from being sniffed by another application,
- TEE (Trusted Execution Environment) – to provide a trusted environment, where the system and applications can store and use secret information like certificates, keys or biometric data e.g. fingerprints or iris patterns.
- Verified boot – to check if system files are intact between consequent boots of device [23].

iOS

iOS is a mobile operating system introduced in 2007 by Apple with the iPhone as the iPhone OS version 1.0. The latest iOS version is 12.

Security features in iOS

The main security features in iOS are:

- Secure boot – verifies the checksums of iPhone components like the bootloader, kernel and baseband software. This proves that there are no changes in these components.
- Secure enclave – this is the kind of TEE for iOS. It provides the implementation of cryptographic methods, storage of secret keys and certificates, and also handles fingerprints for the Touch ID or facial biometrics data for Face ID.
- Keychain – it is a database for secure storage of user's passwords and certificates.
- Applications verification – iOS allows only the installation of applications verified by Apple and signed with Apple certificates. There is also another way for companies that would like to have in-house applications available only to employers. The application must be signed with an enterprise certificate obtained from Apple.

- Application's entitlements (privileges) – each application has a file with its entitlements specified. This file is digitally singed so it cannot be altered after installation.
- Application sandboxing – the application runs in a sandbox that protects its resources from other applications [24].

Vulnerabilities of mobile phones' operating systems

In spite of many security mechanisms in both described mobile operating systems, they are still prone to attacks. The most common attack is malware installation on smartphone. Because the target of attackers may vary, we will focus on those attacks, which led to a compromise of communication's confidentiality or credentials theft.

Attacks on Android

The easiest way to install the application on Android smartphone is from the Google Play store. Google automatically scans applications in their store to filter out malware. There is also a possibility to install the application from any other source. To do this a smartphone user has to manually switch the appropriate option in the system settings. Since 2017 the Google Play store has a "Protect" function that offer scanning applications installed from sources other than the official store [25]. Unfortunately these scans fails on new or obfuscated malicious code. In addition, they cannot distinguish phishing applications.

One of the first well known attack on mobile phones targeting on intercepting SMS messages with one-time passwords was ZeuS in the Mobile campaign from 2010 [26]. The first part of the attack was malware installation on the user's PC. While the victim was using bank's online service in a PC web browser, malware was carrying out man-in-the-middle attack to steal the victim's credentials. Also, malware was serving phishing website to obtain a phone number associated with a bank account. Then, attackers could send phishing SMS message that contains a link to malware crafted for mobile phones to obtain one-time passwords sent over SMS.

Another example can be the BankBot. The first BankBot version was found in 2014 and it is still active. It is distributed as a popular applications installation files (.apk) outside the Google Play store. It can intercept and send SMS messages, steal personal data from a smartphone and detects banks' and social medias' applications installed on your smartphone to obtain user credentials. This is done by serving phishing overlays on top of legitimate applications during the authentication or authorisation process. Knowing the credentials and having access to a second factor like SMS messages or emails it can individually authenticate and authorise actions such as, for example, money transfers [27].

Attacks on iOS

On contrary to Android most applications available on the Apple App Store are manually verified by Apple before being published. There is no easy way to install application from outside the store. The user has to break security system of iPhone (also known as "jailbreak") or install an application signed with enterprise certificate available only to companies developing in-house applications. In spite of such a strict security policy there are successful attacks on iOS.

AceDeceiver (FairPlay Man-In-The-Middle) was two-step attack on iPhone users who use iTunes on a PC with Microsoft Windows. This attack was spotted in 2016 [28]. The first step was to install the malware on PC. This malware was able to automatically install a malicious application on the iPhone connected to infected PC. Malware could bypass iOS security by exploiting the way iTunes installs applications on a connected iPhone. AceDeceiver was able to obtain user's Apple ID and password by phishing.

YiSpectre has been spotted in 2015. It was offered on some suspicious websites as a mobile video player with an adult content. It used the enterprise certificate to be installed on the iPhone. In contrary to applications from App Store, applications signed using an enterprise certificate can use the private API to access user's sensitive data [29].

Pegasus is one of the most sophisticated malware. It can be silently installed on iPhone after visiting malicious website. To achieve this, Pegasus uses a set of not publicly know vulnerabilities. It was developed by the Israeli company NSO Group and can be bought by governmental agencies. It can completely spy on its victim (text messages, voice calls, mails, taping the microphone, taking pictures and so on) [30].

Security analysis

The security of the second factor based on mobile phones depends on the security of mobile networks (as communication channels providers) and mobile phones' operating systems (as an environment where security tokens are processed).

It is worth noting that due to existing attacks (for example with SIM card duplicate or on signalling layer in mobile networks), a one-time password sent over SMS is no longer secure. Also an application functioning on the mobile phone, which has sufficient privileges can read such text message and send it to the attacker. From a functional point of view, this is not a very convenient method, because the user has to manually rewrite the token.

Methods based on applications have many advantages, for example they can implement various security methods and authorisation mechanisms by themselves (such as PIN code or biometrics). On the contrary, the main disadvantage is that the possible malware installed on the mobile phone can sniff and redirect the token to the attacker. From a security point of view, the following threats are worth considering:

- if the authorisation system sends a token to a mobile phone, it can be sniffed and/or redirect to the attacker. Because of that, if the token is sent via native communication protocols in mobile networks (for example via SMS or USSD) it is important to evaluate the security of this native mechanism. On the contrary, when the token is sent by common Internet protocols, it is very important to properly authenticate the origin of such traffic and use strong encryption mechanisms;
- if the token is sent from the mobile phone to the authorisation system, it can be sniffed or blocked by the attacker in a very similar way;
- if the token is generated on the mobile phone, it is important that the cryptographic parameters used to generate the token could not be easily stolen and replicated in all cases. The security of the mobile phone itself is very important, because unauthorised access of an attacker to such device can compromise the entire security model;
- in all cases, it is very important that the user has clear information about all the details of the currently authorising action.

Examples of SCADA software with Single Sign-On or Multi-Factor Authentication

There are a lot of SCADA software available, so it is impossible to mention all of them. In this paragraph, we would like to present only some of them that integrates SSO or MFA technology to improve security.

Ignite 8 from Inductive Automation can be integrated with Microsoft Active Directory Federation Service for SSO as well as with DUO for MFA [31]. DUO is an commercial solution offering application to authenticate user with push notifications sent to smartphone. VTScada developed by Trihedral can be connected to Microsoft Active Directory with LDAP protocol to authenticate users with their MS Windows accounts [32]. VTScada can also use OpenID Connect for authentication to their VTScada Anywhere Client [33]. Aveva (part of Schneider Electric) develops SCADA software, which is capable to use OpenID Connect as a Single Sign-On standard [34].

Conclusion

Our security analysis leads to the conclusion that every MFA method has some flaws. However, it looks that the least prone to attacks is a solution based on some kind of one-time password generating application that takes an additional argument (also known as a "challenge") to generate a new token.

1. This method is immune to eavesdropping on communication channels (SMS, USSD, internet protocols) as there is no communication between the application and the service.
2. It is also resistant to SIM cloning because the application is paired with the service through a shared secret.
3. The user has to manually initiate the process of generating a new token. This makes it harder to convince victim to forward a one-time password to the attacker when the attacker tries to initiate the action himself.
4. As another security mean the user ought to input a challenge to generate a new one-time password. If the challenge contains some details of the action which is going to be authorised, for example a part of the recipient's bank account number and some random number generated on the service side, it is resistant to man-in-the-middle attacks which alters web site's content to mislead victim.

The only possible attack on this kind of second factor is the compromise of the shared secret used to generate a new onetime password so that the attacker can generate the tokens himself.

On the other hand, it may be inconvenient for the user to unlock the phone, enter the challenge into the OTP generator and then enter the pass into SCADA application. A more suitable solution here seems to be an application (like for example DUO or Microsoft Authenticator) in which the user only has to accept the heads-up notification after unlocking the phone.

The choice of the second factor for such critical systems like SCADA, should be done on the basis of risk analysis for certain application and their role in business.

Authors: *Piotr Lewandowski, NASK - Research and Academic Computer Network, R&D Division, Information Security Methods Team, ul. Kolska 12, 01-045 Warsaw, E-mail: piotr.lewandowski@nask.pl; Dr.Eng. Anna Felkner , NASK - Research and Academic Computer Network, R&D Division, Information Security Methods Team, ul. Kolska 12, 01-045 Warsaw, E-mail: anna.felkner@nask.pl; Marek Janiszewski, NASK - Research and Academic Computer Network, R&D Division, Information Security Methods Team, ul. Kolska 12, 01-045 Warsaw, E-mail: marek.janiszewski@nask.pl.*

REFERENCES

[1] Standards | OASIS Available online: https://www.oasis-open.org/standards (accessed on Jan 3, 2019).

[2] Konfigurowanie własnej aplikacji SAML - Administrator G Suite - Pomoc Available online: https://support.google.com/a/answer/6087519 (accessed on Jan 3, 2019).

[3] Azure AD SAML Protocol Reference | Microsoft Docs Available online: https://docs.microsoft.com/en-us/azure/active-directory/develop/active-directory-saml-protocol-reference (accessed on Jan 3, 2019).

[4] Dennis, Z. Choosing an SSO Strategy: SAML vs OAuth2 | Mutually Human Available online: https://www.mutuallyhuman.com/blog/2013/05/09/choosing-an-sso-strategy-saml-vs-oauth2/ (accessed on Jan 3, 2019).

[5] Recordon, D., Fitzpatrick, B. OpenID Authentication 1.1 Available online: https://openid.net/specs/openid-authentication-1_1.html (accessed on Jan 3, 2019).

[6] Hammer-Lahav, E. The OAuth 1.0 Protocol, Request for Comments, RFC Editor, (2010)

[7] Sakimura, N., Bradley, J., Jones, M.B., Medeiros, B. de, Mortimore, C. OpenID Connect Core 1.0 incorporating errata set 1 Available online: https://openid.net/specs/openid-connect-core-1_0.html (accessed on Jan 3, 2019).

[8] Fronczak, M. Zarządzanie tożsamością w chmurze i standardy SAML, OpenID, OAuth Available online: https://zaufanatrzeciastrona.pl/post/zarzadzanie-tozsamoscia-w-chmurze-oraz-porownanie-standardow-saml-openid-oauth/ (accessed on Jan 8, 2019).

[9] M'Raihi, D., Machani, S., Pei, M., Rydell, J. TOTP: Time-Based One-Time Password Algorithm, (2011)

[10] M'Raihi, D., Bellare, M., Hoornaert, F., Naccache, D., Ranen, O. HOTP: An HMAC-Based One-Time Password Algorithm, (2005)

[11] Duo Mobile: Duo Security Available online: https://duo.com/product/trusted-users/two-factor-authentication/duo-mobile (accessed on Jan 8, 2019).

[12] How to use the Microsoft Authenticator app Available online: https://support.microsoft.com/en-us/help/4026727/microsoft-account-how-to-use-the-microsoft-authenticator-app (accessed on Jan 8, 2019).

[13] Sébire, G. GSM Standarization History. In *GSM/EDGE: Evolution and Performance*, Säily, M., Sébire, G., Riddington, E., Eds., John Wiley & Sons, (2011)

[14] ETSI/SAGE, Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 5: Design and Evaluation Report. Version 1.1, (2006)

[15] Alt, S., Fouque, P.-A., Macario-rat, G., Onete, C., Richard, B. A Cryptographic Analysis of UMTS/LTE AKA. In *Applied Cryptography and Network Security*, Manulis, M., Sadeghi, A.-R., Schneider, S., Eds., Lecture Notes in Computer Science, (2016), Vol. 9696, 18–35.

[16] Engel, T. SS7: Locate. Track. Manipulate., 31. Chaos Communication Congress, 2014, Available online: https://media.ccc.de/v/31c3_-_6249_-_en_-_saal_1_-_201412271715_-_ss7_locate_track_manipulate_-_tobias_engel (accessed on Jan 8, 2019).

[17] Nohl, K. Attacking Phone Privacy., Black Hat, 2010. Available online: https://media.blackhat.com/bh-us-10/whitepapers/Nohl/BlackHat-USA-2010-Nohl-Attacking.Phone.Privacy-wp.pdf (accessed on Jan 8, 2019).

[18] Dunkelman, O., Keller, N., Shamir, A. A Practical-Time Related-Key Attack on the KASUMI Cryptosystem Used in GSM and 3G Telephony. In *Advances in Cryptology – CRYPTO 2010*, Lecture Notes in Computer Science, (2010), Vol. 6223, 393–410.

[19] Ghanim, A. Overview of ZUC Algorithm and its Contributions on the Security Success and Vulnerabilities of 4G Mobile Communication. *International Journal of Computer Applications* (2017), n. 168, 34–38.

[20] Hussain, S.R., Chowdhury, O., Mehnaz, S., Bertino, E. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *Proceedings 2018 Network and Distributed System Security Symposium*, (2018).

[21] Liu, J., Yu, Y., Standaert, F.-X., Guo, Z., Gu, D., Sun, W., Ge, Y., Xie, X. Small Tweaks Do Not Help: Differential Power Analysis of MILENAGE Implementations in 3G/4G USIM Cards. In *Computer Security -- ESORICS 2015*, Lecture Notes in Computer Science, (2015), Vol. 9326, 468–480.

[22] Dubey, A., Vohra, D., Vachhani, K., Rao, A. Demonstration of vulnerabilities in GSM security with USRP B200 and open-source penetration tools. In *2016 22nd Asia-Pacific Conference on Communications (APCC)*, (2016), 496–501.

[23] Elenkov, N. Android’s Security Model. In *Android Security Internals. An In-Depth Guide to Android’s Security Architecture*, No Starch Press, Inc., (2014), 1–19.

[24] Apple *iOS Security - iOS 12*, Apple Inc., 2018, Available online: https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf (accessed on Jan 8, 2019).

[25] Cunningham, E. Keeping you safe with Google Play Protect Available online: https://www.blog.google/products/android/google-play-protect/ (accessed on Jan 10, 2019).

[26] Maslennikov, D. ZeuS in the Mobile is back Available online: https://securelist.com/zeus-in-the-mobile-is-back/29830/ (accessed on Jan 10, 2019).

[27] Android.BankBot.149.origin — Dr.Web — innovation anti-virus security technologies. Comprehensive protection from Internet threats. Available online: https://vms.drweb-av.pl/virus/?_is=2&i=14895561 (accessed on Jan 10, 2019).

[28] Xiao, C. AceDeceiver: First iOS Trojan Exploiting Apple DRM Design Flaws to Infect Any iOS Device - Palo Alto Networks Blog Available online: https://researchcenter.paloaltonetworks.com/2016/03/acedeceiver-first-ios-trojan-exploiting-apple-drm-design-flaws-to-infect-any-ios-device/ (accessed on Jan 10, 2019).

[29] Xiao, C. YiSpecter: First iOS Malware That Attacks Non-jailbroken Apple iOS Devices by Abusing Private APIs - Palo Alto Networks Blog Available online: https://researchcenter.paloaltonetworks.com/2015/10/yispecter-first-ios-malware-attacks-non-jailbroken-ios-devices-by-abusing-private-apis/ (accessed on Jan 10, 2019).

[30] Bazaliy, M., Flossman, M., Blaich, A., Hardy, S., Edwards, K., Murray, M. Technical Analysis of Pegasus Spyware. An Investigation Into Highly Sophisticated Espionage Software Available online: https://info.lookout.com/rs/051-ESQ-475/images/lookout-pegasus-technical-analysis.pdf (accessed on Jan 10, 2019).

[31] Ignition 8 | Inductive Automation Available online: https://inductiveautomation.com/ignition/whatsnew (accessed on Jan 11, 2019).

[32] Windows Security Integration Available online: https://www.trihedral.com/help/Content/D_Customize/Dev_WinAuthIntro.htm (accessed on Jan 11, 2019).

[33] OpenID Connect Authentication Available online: https://www.trihedral.com/help/Content/D_Customize/Dev_OpenIDConfig.htm (accessed on Jan 11, 2019).

[34] Security Statement | Trust | AVEVA | Insight powered by Wonderware Online Available online: https://sw.aveva.com/trust/security (accessed on Jan 11, 2019).