CARD International Journal of Science and Advanced Innovative Research (IJSAIR)
ISSN: 2536-7315 (Print) 2536-7323 (Online)
Volume 2, Number 2, June 2017
http://www.casirmediapublishing.com

Comparative Analysis of Connected and Disconnected Tokens in Enhanced Multi-Factor Authentication

Obioha Iwuoha & Chidiebere C. Oparah

Department of Computer Science
Federal Polytechnic, Nekede, Owerri, Imo state
Email: ohaobi@yahoo.com, canonchychuks@yahoo.com
Corresponding Author: Obioha Iwuoha

ABSTRACT

The need for provision of security tokens as a possession factor in multi-factor authentication cannot be over-emphasized. But the fact remains that the most secure and adaptive type of token is yet to be known and strictly adhered to. The objective of this paper is to compare the two main types of tokens and determine which is best and most appropriate for defined situations. The data-flow modeling technique of the structured systems analysis and design methodology will be used for this research. It involves the process of identifying, modeling and documenting how data moves around an information system by examining processes, data stores, external entities and data flows of security tokens. The result of this paper is an educative insight and gross understanding of the types of security tokens required during possession factor presentation in enhanced multi-factor authentication.

Keywords: Tokens, disconnected tokens, hard tokens, authentication, key, encryption, user, authentication factors

INTRODUCTION

A token is an evidence or proof given or shown as a symbol or guarantee of authority. It is a sign of authenticity. In authentication, a token is called an access token. It contains the security credentials for a login session and identifies the user, the user's groups, the user's privileges, and, in some cases, a particular application.

An access token is an object encapsulating the security identity of a process or thread. It is used to make security decisions and to store tamper-proof information about some system entity (Access Tokens,

2007).Tokens can be duplicated without special privilege, for example to create a new token with lower levels of access rights to restrict the access of a launched application. An access token is used by Windows when a process or thread tries to interact with objects that have security descriptors that are securable objects. An access token is generated by the logon service when a user logs on to the system and the credentials provided by the user are authenticated against the authentication database. The authentication database contains credential information required to construct the initial token for the logon session, including its user id, primary group id, all other groups it is part of, and other information. The token is attached to the initial process created in the user session and inherited by subsequent processes created by the initial process. Whenever such a process opens a handle to any resource which has access control enabled, Windows reconciles the data in the target object's security descriptor with the contents of the current effective access token. The result of this access check evaluation is an indication of whether any access is allowed and, if so, what operations like read, write or modify, the calling application is allowed to perform.
There are two main types of tokens. They include;

1. Primary Token

Primary tokens can only be associated to processes, and they represent a process's security subject. The creation of primary tokens and their association to processes are both privileged operations, requiring two different privileges in the name of privilege separation - the typical scenario sees the authentication service creating the token, and a logon service associating it to the user's operating system shell ("American Express Introduces", 2014). Processes initially inherit a copy of the parent process's primary token. An example of a primary token is the bank automated teller machine - ATM card, where the bank creates the ATM card and a logon service verifies this card when presented at any ATM. The ATM card is a hard or connected token.

2. Impersonation Token

Impersonation is a security concept implemented in Windows NT that allows a server application to temporarily "be" the client in terms of access to secure objects. Impersonation has four possible levels: anonymous, giving the server the access of an anonymous/unidentified user, identification, letting the server inspect the client's identity but not use that identity to access objects, impersonation, letting the server act on behalf of the client, and delegation, same as impersonation but extended to remote systems to which the server connects (through the preservation of credentials). The client can choose the maximum impersonation level (if any) available to the server as a connection parameter ("American Express Introduces", 2014). Delegation and impersonation are privileged operations (impersonation initially was not, but historical carelessness in the implementation of client APIs failing to restrict the default level to "identification", letting an unprivileged server impersonate an unwilling privileged client, called for it). Impersonation tokens can only be associated to threads, and they represent a client process's security subject. Impersonation tokens are usually created and associated to the current thread implicitly, by IPC mechanisms such as DCE RPC, DDE and named pipes. An example of an impersonation token is the bank's one-time password – OTP generator device. It requires the user to press a button on the device for the device to display a set of random numbers which are soft or disconnected tokens.

A conventional token is made up of the following identifiers. An identifier is someone or something that establishes the sameness of something.

1. The identifier of the associated logon session. The session is maintained by the authentication service, and is populated by the authentication packages with a collection of all the information (credentials) the user provided when logging in. Credentials are used to access remote systems without the need for the user to re-authenticate, provided that

all the systems involved share an authentication authority for example a Kerberos ticket server.

2. The user identifier. This field is the most important and it's strictly read-only.

3. The identifiers of groups the user (or, more precisely, the subject) is part of. Group identifiers cannot be deleted, but they can be disabled or made "deny-only". At most one of the groups is designated as the session id, a volatile group representing the logon session, allowing access to volatile objects associated to the session, such as the display.

4. The restricting group identifiers (optional). This additional set of groups doesn't grant additional access, but further restricts it: access to an object is only allowed if it's allowed also to one of these groups. Restricting groups cannot be deleted nor disabled. Restricting groups are a recent addition, and they are used in the implementation of sandboxes.

5. The privilege that is special capabilities the user has. Most privileges are disabled by default, to prevent damage from non-security-conscious programs. Starting in Windows XP Service Pack 2 and Windows Server 2003 privileges can be permanently removed from a token by a call to Adjust Token Privileges() with the SE_PRIVILEGE_REMOVED attribute (How Access Tokens Work, 2014).

6. The default owner, primary group and ACL for objects created by the subject associated to the token.

A connected token is also called a hard token. It is a hardware security device that is used to authorize a user. A common example of a hard token is a security card that gives a user access to different areas of building or allows him to log in to a bank computer system. Some hard tokens are used in combination with other security measures to further

enhance security (Biba, 2005). For example, a username and password or a fingerprint scan may be required along with the hard token to gain access to a secure system. The connected token is the tried and tested token used by countless organizations over the years. Connected tokens are tokens that must be physically connected to the computer with which the user is authenticating. Tokens in this category automatically transmit the authentication information to the client computer once a physical connection is made, eliminating the need for the user to manually enter the authentication information. However, in order to use a connected token, the appropriate input device must be installed. The most common types of physical tokens are smart cards and USB tokens, which require a smart card reader and a USB port respectively (Biometric token, 2013). Other examples of connected tokens include; older PC card tokens are made to work primarily with laptops, the audio jack port is a relatively practical method to establish connection between mobile devices, such as iPhone, iPad and Android, and other accessories, and the credit card reader for iPhone and Android. Many connected tokens use smart card technology. Smart cards can be very cheap to buy and contain proven security mechanisms as used by financial institutions, like ATM cards. However, computational performance of smart cards is often rather limited because of extreme low power consumption and ultra-thin form-factor requirements. Smart-card-based USB tokens which contain a smart card chip inside provide the functionality of both USB tokens and smart cards. They enable a broad range of security solutions and provide the abilities and security of a traditional smart card without requiring a unique input device. From the computer operating system's point of view such a token is a USB-connected smart card reader with one non-removable smart card present. Some types of connected token like the single sign-on token uses its token to store software that allows for seamless authentication and password filling. As the passwords are stored on the token, users need not remember their passwords and therefore can select more secure passwords, or have more secure passwords assigned. Usually most

tokens store a cryptographic hash of the password so that if the token is compromised, the password is still protected. A good example is a registered MTN SIM card.

Disconnected tokens also known as soft tokens, have neither a physical nor logical connection to the client computer. They typically do not require a special input device, and instead use a built-in screen to display the generated authentication data, which the user enters manually themselves via a keyboard or keypad. Disconnected tokens are the most common type of security token used, usually in combination with a password, in two-factor authentication for online identification.

Disconnected tokens have many types. They include contactless tokens, Bluetooth tokens and mobile phone tokens. Unlike connected tokens, contactless tokens as type of disconnected token, form a logical connection to the client computer but do not require a physical connection. The absence of the need for physical contact makes them more convenient connected tokens (DeBorde, 2007). As a result, contactless tokens are a popular choice for keyless entry systems and electronic payment solutions such as Mobil Speed pass, which uses RFID to transmit authentication info from a keychain token. However, there have been various security concerns raised about RFID tokens after researchers at Johns Hopkins University and RSA Laboratories discovered that RFID tags could be easily cracked and cloned. Another downside is that contactless tokens have relatively short battery lives; usually only 5–6 years, which is low compared to USB tokens which may last more than 10 years. Though some tokens do allow the batteries to be changed, thus reducing costs. The Bluetooth token is another type of disconnected token (How Access Tokens Work, 2014). Its low Energy protocols serve for long lasting battery life cycle of wireless transmission. Mobile device tokens involve the use of a mobile computing device such as a smart phone or tablet computer as the authentication device. This provides secure two-factor authentication

that does not require the user to carry around an additional physical device. Some vendors offer a mobile device authentication solution that uses a cryptographic key for user authentication. This provides a high level of security protection including protection from a Man-in-the-middle attack, which can occur from a rogue Hotspot (Wi-Fi).A user wishing to access a protected resource, such as a VPN or internet banking site, uses the Mobile Token App to generate a One-Time Password. The application can be PIN protected. It is licensed per user, and licenses can be used across multiple personal mobile devices. The Mobile Token App is available for all leading mobile devices.

THEORETICAL FRAMEWORK

Advantages of Connected or Hard Tokens

The following are some advantages of connected tokens;

1. Confidence: The hard token gives the user confidence and assurance that contact and possession exchange has been made with the verifying mechanism.
2. Less authentication time: Time is not wasted in keying in generated one-time Pins. It just involves the user slotting the card or token into the stipulated space.
3. If stolen the connected token could be deactivated by the issuer and another token given to the user unlike the soft token generator device that would still be generating passwords.

Disadvantages of Connected Tokens

The following are some inherent drawbacks in the use of connected tokens;

1. They are not scalable: To secure access to important digital assets (for example VPN, large enterprises, such as a banks or government departments and agencies need to deploy multi-factor authentication to hundreds of thousands of employees and/or customers (Nancy,2012). With hard tokens, that means buying, supplying and managing a physical token for each individual.

2. They are easy to lose: Not only are hard tokens uncomfortable in your pocket and bulky on your key chain, they are also easy to forget at home, misplaced or drop down an elevator shaft. The point is that hard tokens are foreign objects that have been introduced into the users' lives for the sole purpose of seemingly far-off virtual threats, which makes user adoption a constant challenge.

3. They are expensive: The prevalence of lost tokens has forced most administrators to keep extra token inventory in stock to serve as replacement tokens. However, this problem goes beyond being an administrative headache because hard tokens are extremely pricey.Then there's the software license and eventual expiry to consider, at which point you would need to renew.

4. They are not cloud based: Hardware-based access management appliances like most hardware are subject to breaking down, especially over time. That means smart companies must keep two hardware appliances on site in case one breaks down. What happens if both break down? No one has access to any corporate resources, valuable time gets wasted and a special, immediate, and potentially inconvenient, trip to the office is required (Somini, 2012). Many companies are switching to cloud-based IT services across the board.

5. They are limiting: Not much can be done with a hard token besides generating an OTP-type code on the screen. This means that as multifactor technology progresses toward widespread incorporation of biometrics like fingerprint and iris scanning, wearables and contextual information such as geolocation, connected tokens will fall short.

6. It less convenient as it requires extra devices like wallet filler to avoid been lost or forgotten.

7. It is less flexible as its compatibility varies among different platforms, making it difficult for mobile users.

8. It has Out of Sync issues leading to problematic synchronization.

9. It is not eco-friendly as it constitutes environmentally irresponsible waste.

10. It has a limited user interface with high learning curve required.

Advantages of Disconnected or Soft Tokens

The following are some advantages of disconnected token. They include;

1. It is simple to use: Once the software is installed, disconnected tokens can be used. As far as the customer is concerned, this method works in essentially the same way as SMS OTPs, except that instead of waiting to receive an automatically generated SMS, the user runs the One-Time-Password application on his/her phone, generates an OTP, and uses it instantly (Prosecco: Publications, 2014).

2. Incurs no extra costs aside from the software download: As the user does not use any airtime or carrier services to generate the OTP, this method does not cost users anything. Depending on the carrier and download method, customers may have to pay to download the app. And unlike SMS delivery, which only cares about the number on the SIM card and not which handset the SIM card is in, customers who get a new mobile will need to download and install the application again.

3. It is immune to coverage, latency, and delivery issues: Doing everything on the phone itself very nearly completely immunizes users from the headaches of mobile networks thereby making the disconnected token approach a better choice for mobile payments and authentication.

4. Disconnected token has a PIN-code, allowing protecting an OTP passwords generator from unauthorized access in the case your phone for any reason enters the wrong hands.

5. It allows for flexible configurations in the choice of the password length and algorithm of its generation.

6. Many tokens could be created on one device.

7. There are versions available both for Android and iOS. Moreover, you can use smart watches and Android Wear to get OTP passwords as well.

8. It supports the data signing function (CWYS), which allows protecting transactions from such threats as data modification, replacement, and banking Trojans with automated transfer system (Prosecco: Publications, 2014).

Disconnected tokens offer more powerful, more flexible, more dynamic security infrastructure at a fraction of the cost. Ideally suited for the global, mobile user to stay connected in today's world. The key advantage of the disconnected token is that there are no new devices or wallet-fillers for customers – just an add-on to the device they already carry everywhere (Prosecco: Publications, 2014). Since customers already own the hardware which sometimes is the mobile phone.

Disadvantages of Disconnected Tokens

The following are the disadvantages of disconnected tokens. They include;

1. Web attacks: Any system which allows users to authenticate via an untrusted network such as the Internet is vulnerable to man-in-the-middle attacks. In this type of attack, a fraudster acts as the "go-

between" the user and the legitimate system, soliciting the token output from the legitimate user and then supplying it to the authentication system themselves. Since the token value is mathematically correct, the authentication succeeds and the fraudster is granted access (Somini, 2012). Citibank of USA made headline news in 2006 when its hardware-token-equipped business users became the victims of a large Ukrainian-based man-in-the-middle phishing attack.

2. Breach of Codes: In 2012, the Prosecco research team at INRIA Paris-Rocquencourt developed an efficient method of extracting the secret key from several PKCS #11 cryptographic devices, including the SecurID 800. These findings were documented in INRIA Technical Report RR-7944, ID hal-00691958 and published at CRYPTO 2012 ("Specification for Integrated Circuit(s)", 2010).

3. It is important to note that mobile OTP generators, if poorly implemented, are susceptible to fraudster attacks. Ensuring OTPs are generated securely only for intended users requires advanced technologies to mitigate key threats (Strong Authentication, 2007).

Mitigation of the Shortfalls in Disconnected or Soft Tokens

It is quite obvious from the comparative analysis above that the use of disconnected or soft tokens are better and more preferred for multi-factor authentication. This being the case, there is then need to check most the drawbacks inherent in this new adopted paradigm. Mitigation of the shortfalls in disconnected tokens can be done in the following ways;

1. **Phishing:** Ensure that each software token is bound to the device of the user on which the application is installed.

2. **Keystroke logging:** Preclude attackers from capturing OTP's using key-logging. Even with a captured PIN or activation code, the attacker will be unable to generate an identical (clone) mobile software token.

3. **Static code dump/Patch runtime debugging:** Even if the unique device IDs are spoofed, the mobile software token must have sophisticated levels of code obfuscation and symbol stripping, as well as an additional security layer in the form of a PIN, built-in (Voltage Secure Stateless Tokenization, 2012). These measures ensure that even through reverse engineering by an attacker, an OTP will not be generated.

4. **System resource manipulation:** In order to conduct this type of attack, a jail-broken or rooted device is required. The mobile software token does not operate on such a device thereby circumventing such an attack.

5. **Brute force:** The mobile software token must be PIN protected and designed to self-destruct after 5 incorrect entries entered consecutively. The mobile software token can also be protected with a layer of PIN camouflaging. In this case, an incorrect PIN will be accepted and an invalid OTP will be displayed.

6. The attacker should have no way of knowing if an input PIN is correct or incorrect.

7. **Dynamic memory access:** In order to conduct this type of an attack, the device would need to be in a vulnerable state such as jail-broken or rooted. The mobile software token should implement sophisticated layers of verification to determine if the device is compromised and ceases to operate.

8. **Chosen plain text brute force:** The attacker will not be able to mount this attack as it is computationally not feasible to obtain the token secret key in brute force.
9. **Screen capturing:** It should be possible to deploy the mobile software token with the configuration to generate OATH compliant time-based OTP and Challenge/Response with a short time validity for making it ineffective to capture and relay.

10. Additionally, all strong authentication solutions should be implemented as part of a larger, multi-layered, context-based security strategy that also includes device profiling, malware forensics, transaction verification, and mutual authentication between the user and the application (Voltage Secure Stateless Tokenization, 2012). This requires an integrated versatile authentication platform with real-time threat detection capabilities. The advanced fraud prevention seamlessly integrates with all major platforms and the threat detection piece is transparent, so that there is no software for the user to install.

SUMMARY

The objective of this paper to compare the two main types of tokens and determine which is best and most appropriate for defined situations was achieved. The data-flow modeling technique of the structured systems analysis and design methodology was used for this research. It involves the process of identifying, modeling and documenting how data moves around an information system by examining processes, data stores, and external entities and data flows of security tokens. The result of this paper is an educative insight and gross understanding of the best type of security token required during possession factor presentation in enhanced multi-factor authentication.

CONCLUSION

The disconnected token which also known as soft token that belongs to the impersonation type of token is the best choice of token to be used for possession factor presentation in multi-factor authentication. These reasons to buttress this statement can be seen in the body of this paper.

REFERENCES

Access Tokens. (2007). Retrieved from www.MSDN.net

American Express Introduces New Online and Mobile Payment

Security Services. (2014). www.americanexpress.com

Biba, E. (2005). Does Your Car Key Pose a Security Risk? PC World Press, New York, USA.

Biometric token. (2013). Retrieved from www.contactlessbletoken.com

DeBorde, D. (2007). Two-factor authentication. Siemens Insight Consulting Press, Michigan, USA.

How Access Tokens Work. (2014). Retrieved from www.MSDN.net

Nancy, O. (2012). Team Prosecco dismantles security tokens. (2014). Retrieved from www.phys.org

Prosecco: Publications. (2014). Retrieved from www.phys.org

Somini, S. (2012). Computer Scientists Break Security Token Key in Record Time. New York Times Publishers, New York, USA.

Specification for Integrated Circuit(s) Cards Interface Devices. (2010). Retrieved from www.usb.org

Strong Authentication. (2007). Retrieved from www.Securitypronews.com

Voltage Secure Stateless Tokenization Advances Data Security for Enterprises, Merchants andPayment Processors. (2012). Retrieved from www.reuters.com.