SafeNet
THE DATA PROTECTION COMPANY

A Security Survey of Strong Authentication Technologies

WHITEPAPER

Contents

Introduction

All authentication methods are based on providing the legitimate user with a method for proving his or her identity.

Such "proof" can involve different form factors, such as something only the user knows (like a password) or something only the user has (like an external piece of hardware or the user's biometric information). It could also be something that the user is, such as unique physical attributes, for example, a fingerprint or retinal scan.

Unfortunately, however, proof of authentication is rarely foolproof. For example, a user's password may be easily guessable, or "secret'' information about a user's history could be easily discovered. Likewise, an external piece of hardware can be temporarily accessed by others. Thus, strong authentication uses multiple methods, aiming to make impersonation difficult while not overly inconveniencing the user.

This paper gives a detailed overview of several different types of authentication methods and their underlying security mechanisms. It also discusses how the various strong authentication methods are effective in mitigating different types of attacks.

Authentication Methods

Password Authentication

The classic method of authentication is a user password. The limitations and drawbacks of passwords are well known—they are easily stolen and are hard to remember, and as such are typically not chosen well. Attempts to make this method more secure by requiring longer, more complex passwords further undermines their effectiveness, as often this leads to passwords being written down, making them vulnerable to exposure. Nevertheless, despite these drawbacks, it is assumed that passwords are not stored anywhere and reside only in the user's memory where they cannot be stolen. Thus, most other authentication methods are combined with user passwords, which provides multi-factor authentication, making it difficult for an attacker to impersonate the user since it needs to obtain all factors (for example, physical access to hardware as well as knowing the user's password).

Additional vulnerabilities related to password authentication are the result of weak security in storing passwords. For example, to prevent passwords from being leaked or stolen, they must be hashed (with random salt) at the server and not stored in any other form. While the weaknesses of passwords are well documented, recommendations for password-based authentication are beyond the scope of this white paper.

8	4	5	9	1
9	5	4	0	2
0	2	8	3	7
3	3	7	9	6
7	6	8	1	7

Authentication and Human Memory

Passwords are not the only authentication method that relies on human memory. Research has shown that humans can remember patterns very well, and often more easily than random passwords. One example of such a scheme is SafeNet's GrIDsure. In this solution, the user's "secret" is a subset of the cells within a 5 by 5 cell grid. To authenticate, users are presented with a grid of random numbers or characters and they must enter the numbers that appear in the subset of cells that are their secret. Such a solution can be used together with a password to provide a higher level of security. Later in this document, we will discuss the security benefits of this solution.

Hardware-based Authentication Methods

The basic motivation for using additional hardware in authentication is exactly the same as that of a physical key. The additional hardware is something that the user physically possesses. Without this physical object, authentication is not possible. It is possible to divide hardware-based authentication into two main classes or types:

- **Secure storage:** In this case, the hardware consists of secure storage that holds a secret or secrets which are used to carry out authentication. This has many advantages; in particular, it may be random and very long since the secret need not be memorized, thus making it essentially impossible to guess. The disadvantage of this method is that, in order to authenticate, the secret is downloaded to the computer from which the user is logging in. Thus, if this computer is compromised (for example, contains some type of malware), the secret can be stolen and used later by an attacker.
- **Cryptographic processor:** Here, the hardware consists not only of secure storage, but also has the capability of carrying out onboard cryptographic computations. This addresses the disadvantage of secure storage methods because the secret used for authentication never leaves the hardware device. Thus, even if the computer being used is corrupted, it is not possible to steal the secret for later use[1]. Two common examples of authentication devices of this type are smart cards and one-time password (OTP) devices.

1 Note that any present malware can carry out malicious operations after the user authenticates because at this point it is assumed that any operations originating from the computer are those of the legitimate user. However, the important point is that the damage is limited to this session; once the session is closed, the attacker cannot re-authenticate, in contrast to secure storage methods. In any case, it is important to keep in mind that even the best authentication methods should not be used on untrusted systems

A special category of hardware cryptographic devices is one that adds a user-presence test to prevent malicious code from using the hardware device. One example of such a solution is a smart card reader with a PIN pad that requires the user to provide the smart card PIN on the hardware card reader itself. As such, malware present on the host device cannot use the card on behalf of the user unless the user is present and types in the smart card PIN. Another example of such a device would be a hardware token that requires a button to be pushed in order for the cryptographic operation to be completed.

We stress that not all hardware authenticators are created equal. Some are designed using standard cryptographic algorithms, are rigorously tested for logical and other errors, and deploy sophisticated physical mechanisms for preventing access to the protected secrets. However, there are also hardware authenticators that are not well designed and are easily broken by anyone with reasonable expertise in hardware hacking. Thus, it is not enough to choose an authentication method; one must also check authentication device certifications, vendor credibility, and, where necessary, use independent lab testing to check the security of the device.

Out-of-Band Authentication Methods

Out-of-band authentication is a multi-factor alternative to hardware-based authentication. In hardware-based authentication, the user has a special-purpose authentication device as the second factor of authentication. In contrast, in out-of-band authentication, a device that is already in the user's possession – and that can be used to receive information securely – is used as the additional factor. Examples of out-of-band authentication methods include email and SMS. These methods work in the following way. When a user wishes to login, an authentication code is sent to the user via the out-of-band channel (email, SMS, etc.). Then, the user types in the authentication code together with their password. In order for an attacker to impersonate the user, he or she must know the user's password and either take possession of their out-of-band device (e.g., mobile phone, which might also be locked with a password or fingerprint), or somehow intercept the authentication code en route. This is typically difficult to do. (We remark that for email, it may depend on the security of the email being used. For example, the security level is higher when using enterprise email than when using public, unencrypted email services.)

One significant advantage of out-of-band authentication is that no secret is stored on the user's device. Thus, if an attacker steals the user's phone, no secret can be stolen for later use. Likewise, the authenticating server can generate an independently random authentication code every time. Thus, even if the server is somehow breached, there is no long-term secret that can be stolen for later use.

As with hardware, not all software-based authentication methods are equal. The important issue is what protections are deployed against malware and physical access.

Software-based Authentication Methods

Authentication methods of this type deploy a software application on the user's computer, smartphone, or mobile device. Typically, software-based methods are clones of hardware authentication devices that work in software. Thus, the software may be a password-storage application or an application that carries out cryptographic computations (like smart card operations or in order to generate one-time passwords).

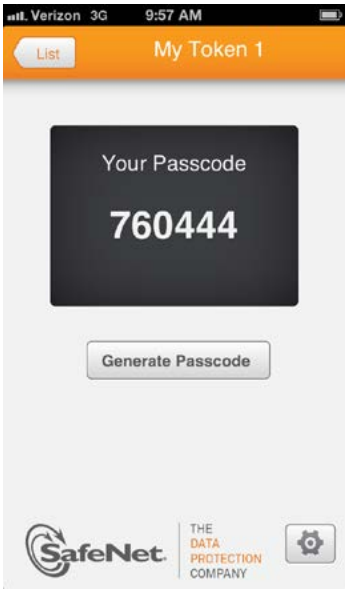
The main disadvantage of software-based methods is their vulnerability when an attacker obtains physical access to the user's computer or mobile device, or when malware is present. (In the case of malware, all secrets can be obtained, even those used in cryptographic computations).

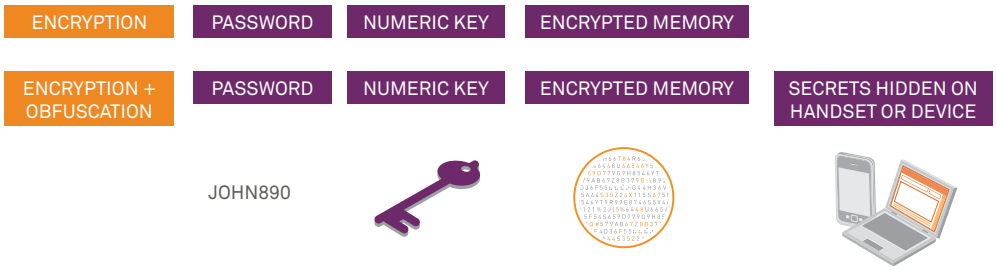
As with hardware, not all software-based authentication methods are equal. The important issue is what protections are deployed against malware and physical access. It is important to understand that in contrast to hardware-based solutions, it is almost impossible to thwart an attack on software-based solutions by a professional attacker who has physical access and is making a concerted effort. This is because all software-based protections are inherently obfuscated, which, given enough time and expertise, can be broken. However, it is possible to make the attack difficult and time-consuming. In order to illustrate this point, we compare two possible methods of protection:

- **Password encryption:** A popular way of protecting software authentication methods is to encrypt the application (or, more exactly, the memory where the secrets are stored) using a password that the user chooses (a cryptographic key should be derived from the password using standard key derivation techniques). Then, in order to authenticate, the user enters his/her password, which is then used (after key derivation) to decrypt the necessary secrets for carrying out authentication; for example, decrypting the private key or OTP seed.

The problem with this method is that it is inherently vulnerable to an offline dictionary attack. This means that an attacker can try all possible/likely passwords and see which one works. It is typically possible to try millions of passwords a second, and so, for most users (who use passwords that can be remembered), such an attack could be successful after a short amount of time. In addition, since the attacker can copy the application (or encryption portion) given physical access to the user's computer, it is possible to carry out this attack on the attacker's own computer, far away from the user's scrutiny.

- **Password encryption plus obfuscation:** In order to improve the security of password encryption protection, it is possible to also use obfuscation. This means that, in addition to encrypting the secrets under the user's password, they are somehow hidden on the computer or mobile device. There are many ways to obfuscate, and the level of protection achieved depends very much on how well the obfuscation is implemented. Some systems, such as Microsoft Windows operating systems, provide native obfuscation services that are maintained and improved over time (Windows Data Protection API (DPAPI) mechanism)[2]. Using the native tool for obfuscation causes security to automatically improve every time DPAPI is improved. Of course, it is always a good idea to implement additional obfuscation on top of this.





The growing use of software authentication solutions in conjunction with mobile devices warrants an additional discussion on the protection afforded by the various mobile operating systems. Additional details on this topic can be found in a SafeNet white paper that reviews the various security mechanisms designed to protect SafeNet's software-based authentication solutions[3].

2 As with everything else, DPAPI can also be used in more and less secure ways. My recommendation is to use DPAPI of the ``current user'' and to provide the user password for protecting the secrets as the DPAPI ``additional secret'', called pOptionalEntropy. This means that, in order for an attacker to obtain the secrets, it must either break DPAPI or carry out an offline dictionary attack on the user's machine while using the DPAPI mechanism. This is difficult because it means that the attacker must also be able to log in to the user's account on the machine.

3 See "SafeNet Authentication Service Security Considerations" Security White Paper.

Biometrics are more suitable for physical security than remote authentication or IT access. When biometrics are used for remote authentication, additional security measures are advised, such as using it in conjunction with another authentication factor.

Biometric Systems

Another completely different type of authentication mechanism is based on biometrics. The typical biometric used for authentication is the user's fingerprint. Biometrics have the strong advantage that users don't need to carry anything with them since their physical attribute is the additional factor of authentication. Having said this, fingerprints can be a problematic form of authentication. First, in order to reduce false positives, stringent parameters are often applied. This can result in many false negatives, requiring users to attempt authentication multiple times until their fingerprint is accepted. These false negatives may often be the result of users having a scratch or dirt on their finger. Second, fingerprints are not secret and cannot be replaced. Thus, if a user's fingerprint is found, it is possible to build a model of it and use it to impersonate a user. Fingerprint reader manufacturers often deny that this is possible; however, it has been demonstrated time and time again.

There are a number of ways that fingerprint authentication is used. The basic question is where the authentication takes place and what is being protected. Following is a discussion of these two elements.

- **Device protection and on-device authentication:** One use of fingerprints is to protect a device by having a fingerprint reader embedded into it or connected to it. (Some laptops and phones, such as the latest iPhone 5s, have embedded readers). As described above, usability issues often arise, which depend on the quality of the reader and how stringently it is configured (regarding the tradeoff between false negatives and positives). Beyond this, an image of the user's fingerprint can often be found on the device itself. This means that if the user loses his or her laptop or phone, an attacker can use the user's fingerprint to log on. (The attack is non-trivial and, depending on the reader, may involve printing the fingerprint or constructing a silicon imprint. In many cases, the fingerprint is used instead of a password, providing only one factor of authentication. Better security is obtained when the fingerprint is used in conjunction with a good password, rather than only using a password. The same threats arise when a separate reader is used instead of an embedded one.
- **Remote authentication:** Another way of using fingerprints is when they are used for remote authentication. In this case, the user's fingerprint (or in actuality a vector of measurements taken from the fingerprint) is sent to a remote source for authentication. This use of fingerprints is very problematic since a single theft of the fingerprint image renders the authentication method eternally compromised since the attacker can always replay the measurement and impersonate the user.

Biometrics are considered by many to be highly secure since they are used in military and other installations. However, in these cases, they are usually used where there is video or in-person monitoring, such as when a guard stands next to the fingerprint reader and verifies that the user's real finger is used. When such precautions are taken, it is much harder to carry out attacks. For this reason, biometrics are more suitable for physical security than remote authentication or IT access. When biometrics are used for remote authentication, additional security measures are advised, such as using them in conjunction with another authentication factor.

Other Authentication Methods

Other methods, beyond those mentioned above, include authentication using an additional online server (as in single sign-on scenarios), and are beyond the scope of this white paper.

Classes of Attacks on Authentication Mechanisms

In this section, we describe the main classes of attacks that will form the basis of our comparison of different authentication mechanisms. The main division is between external attacks and internal attacks, where an external attack is one that takes place by an attacker who does not have direct access to the user's authentication device or to the user's machine (be it physical access or malware-based access). Conversely, an internal attack is one that is perpetrated by an attacker who does have physical access to the user's machine or device (be it physical or malware-based). Examples of external attacks include eavesdropping attacks, active man-in-the-middle attacks, phishing attacks, and social engineering attacks[4]. Internal attacks, on the other hand, include malware attacks, physical access to a user's machine and hardware device theft. It is worth noting that attack types sometimes overlap. For example, man-in-the-middle attacks can be achieved in a number of ways, one of them being phishing. In addition, phishing attacks are also a type of social engineering attack.

In a shoulder surfing attack, the attacker looks over the shoulder of the user as they enter an access credential.

External Attacks:

- **Eavesdropping:** The attacker is able to passively eavesdrop on all messages that are sent between the user's machine and the authenticating server. Note that such attacks can easily be thwarted using SSL.
- **Shoulder surfing:** The attacker literally looks over the shoulder of the user as he or she enters an access credential, in an attempt to steal it.
- **Guessing:** The attacker simply tries to guess the user's credentials.
- **Phishing:** In a phishing attack, a user is duped into giving up sensitive information to an attacker. It is important to note that in such an attack, the user may provide information that would never be requested in a normal authentication process. This distinguishes phishing from a typical man-in-the-middle attack where the user expects to see the usual login page. In many cases, however, phishing is just used as a way of effectively carrying out a man-in-the-middle attack.
- **Social engineering:** This class of attack refers to cases where a user is persuaded to give up her credentials or otherwise actively help the attacker. It includes phishing attacks (where users are warned that if they don't enter their credentials, their account will be locked) but also extends far beyond it. An important example of a social engineering attack is one where an employee is called by someone from the IT department and asked to provide information or carry out a seemingly innocuous activity. These types of attacks are extraordinarily effective.
- **Active man-in-the-middle:** In this type of attack, the attacker can actively inject messages of his own into the traffic between the user's machine and the authenticating server.

Internal Attacks

- **Malware:** This is the classic internal attack where the user's machine is infected in some way by malicious software. Although there are many ways to carry out this attack, for the purposes of this paper, we make no distinction between them and simply assume that once a user's machine is infected, attackers can obtain any unencrypted information and carry out any operation they wish.

4 An excellent example of social engineering is the following. A new employee is called by someone from the IT department and introduced to the security policy of the company. In particular, they are warned to never give up their password, even to an IT manager. Following this, the employee is asked to run through a series of steps to check their account, including the "change password" process. Since the employee cannot tell the IT worker their password, they are asked to change their password to a default password, and then a few minutes later to change it back. This all sounds very innocent. However, at the moment that the employee changes the password to the default (and before it is changed back), the attacker logs in remotely to the employee's account using the default password

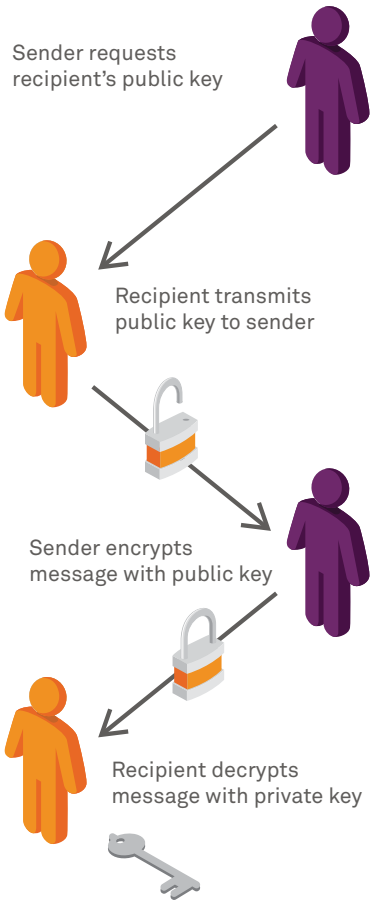
- **Physical access to the user's machine:** This attack can be carried out by a perpetrator posing as someone who works on the premises, such as maintenance professional or a similar function. Here, the attacker has physical access to the machine when the user is logged off. We assume that the attacker can read the hard drive (this can be achieved by booting from an alternative operating system or by physically extracting the drive and connecting it to the attacker's laptop).
- **Device theft:** This is essentially the same as the previous attack except that the perpetrator faces no time constraints to carry out the attack. Note that in this scenario, the user will probably detect the attack relatively quickly, as they will realize that their device is missing.

Security Analysis of Authentication Mechanisms

This analysis describes the properties required for thwarting the types of attacks detailed in this paper, leading to a greater understanding of the types of authentication mechanisms available and their effectiveness.

Eavesdropping

An eavesdropping attacker is completely thwarted whenever encryption is used. In fact, SSL suffices even if neither party (neither the client nor the server) has a certificate. Thus, all authentication mechanisms provide sufficient protection against eavesdropping.



Shoulder Surfing

Shoulder Surfing is most successful when only plain passwords are used for authentication. One-time passwords (OTPs) provide protection against shoulder surfing since the password that is stolen has already been used and is no longer relevant (each time the user logs in they are required to generate a new password, so the lifespan of the password is very short and cannot be used more than once). Likewise out-of-band authentication is not vulnerable to shoulder surfing since each code is randomly generated and used only once.

GrIDsure (pattern-based authentication) is essentially a type of password, but it too provides good protection against shoulder surfing, since it is very hard to record the values typed and to correlate them to the one-time random grid that was presented to the user without being detected. In addition, even if an attacker does succeed in recording the grid pattern, a single GrIDsure authentication session does not provide full information about the password since multiple patterns match to the series of values that are entered.

Guessing

Guessing attacks can only be successful when the credentials being used are short. Password-based authentication is typically vulnerable to guessing attacks due to the fact that most users do not choose good passwords. However, a guessing attack is usually only successful for the most common passwords (like password1, 123456, the user's name, and so on). If the user chooses a short random password, then it will be less vulnerable to a guessing attack since standard retry counters prevent an attacker from trying too many possibilities. Thus, one-time passwords and out-of-band authentication codes, which are also "short" (comprised of 6 digits), are effective against guessing attacks. Likewise, guessing attacks are very unlikely to succeed when pattern-based authentication, such as GrIDsure, is implemented, requiring users to remember a visual pattern rather than a word or phrase.

Man-in-the-middle and Phishing

Man-in-the-middle (MitM) attacks can be thwarted by using SSL with server authentication (for example, using a server certificate) and then authenticating the client over the SSL connection. While this is true for any authentication method, it is worth noting that server certificates are only effective if the server certificate is correct and valid, which is not always the case. For example, many users fail to correctly validate server certificates and will click through warnings displayed on their screens.

Thus, although SSL with server authentication only may afford a degree of protection against man-in-the-middle attacks, carrying out these attacks is still possible by using phishing or other methods. The only method of authentication that provides complete protection against man-in-the-middle attacks is when SSL is used with client authentication—when the user has a digital certificate. Note that this method is most effective when the user's secret key associated with the certificate is stored securely on a hardware authentication token, ensuring that it cannot be stolen by malware.

SSL with mutual (server and client) authentication effectively protects against man-in-the-middle attacks owing to the design of SSL, which provides protection even if the server certificate is not correctly validated by the user, as can occur in a phishing attack. Thus, the only method that provides complete protection against these attacks is x.509 certificate-based authentication. (Of course, if the server certificate is valid, as in the vast majority of cases, then SSL with only server authentication is safe.)

OTPs and out-of-band authentication provide much better protection against MitM and phishing attacks than that offered by plain passwords since the theft of a one-time password or authentication code enables only a single fraudulent login by the attacker, in contrast to long-term access, as in the plain password case.

Likewise, a single man-in-the-middle attack on GrIDsure will only yield part of the credential (the user name and password) and so partial security is maintained.

...hardware-based secure storage and smart cards are non-transferrable and therefore not vulnerable to social engineering.

Social Engineering

Social engineering relies on the ability of the attacker to fool users into giving up their credentials. Since users are typically vulnerable to these attacks, any method that relies on a credential that can be transferred (like passwords or GRIDsure) is vulnerable to social engineering attacks.

The most successful cases of social engineering are carried out remotely, over the phone, or via email or the Internet. Physical transfer, which involves physical interaction, is less common, as people are less likely to hand someone they don't know their smart card or secure storage device.

While a user would most likely not turn over their OTP hardware authentication device to an attacker, phishing attacks that request users to generate and provide several one-time passwords have been known to be successful. Thus, one-time passwords and out-of-band authentication are vulnerable to these types of social engineering attacks. In contrast, hardware-based secure storage and smart cards are non-transferrable and therefore not vulnerable to social engineering.

The status of software-based secure storage and smart cards is very dependent on the implementation. On the one hand, many popular implementations enable a user to copy and paste the credential, making it transferrable and therefore vulnerable. On the other hand, it is possible to prevent the user from doing this, in which case the solution does provide protection (without expert hacking skills).

It is worth noting in this respect that biometrics provide full protection against social engineering, and this is one of their primary strengths.

Although, in principle, malware can undermine any authentication method, there is a fundamental difference between authentication solutions that are vulnerable to generic malware versus those that are only vulnerable to targeted malware.

Malware

In principle, as soon as a user's machine is infected with malware, the perpetrator can do anything that the user can. Thus, if a user logs in to her bank account with an infected computer, the perpetrator can perform any type of bank transfer that the user can perform.

Since vulnerability to malware is inherent to any authentication method, for the scope of this paper, we will define a method as providing protection against malware attacks if the attacks that can be perpetrated with the stolen credentials are limited to the period of time during which the malware is present on the user's machine. Thus, once the malware is detected and removed, the attacker must no longer be able to impersonate the user.

Non-impersonation of the user can only be achieved if the user's credentials cannot be obtained by the malware. Thus, any software-based solution is vulnerable, as too are hardware secure storage solutions (these solutions work by downloading the credentials to the user's infected machine).

Although, in principle, malware can undermine any authentication method, there is a fundamental difference between authentication solutions that are vulnerable to generic malware versus those that are only vulnerable to targeted malware.

For example, plain passwords are vulnerable to keyloggers, which are a common type of generic malware. In contrast, one-time passwords are not since the keylogger will only capture old passwords.

GrIDsure is not vulnerable to keyloggers since only a random series of values is captured. Thus, compromising GrIDsure passwords requires targeted malware. In addition, a single user authentication session is not sufficient to obtain the user's full credentials. If the user authenticated once from an infected machine, the credential is not fully exposed.

Smart-card-based authentication with X.509 certificates provides full protection against all types of malware since the secret key is never exposed. As we have mentioned, it is possible for malware that is currently present on a user's machine to utilize the smart card to fraudulently log in. However, no secret can be stolen that can be used later by the perpetrator from another system.

Out-of-band authentication methods provide complete protection against malware since each authentication code is randomly generated and completely independent of all previous codes.

Physical Access to a User's Machine

Any additional hardware authentication device provides protection in the event of a perpetrator's physical access to a user's machine, for the reason that the credentials do not reside on the machine. This stands in contrast to software-based solutions that are inherently vulnerable to this type of attack.

Good implementations of software-based solutions can provide reasonable protection in the event of physical access by utilizing obfuscation and other measures to make it difficult for the perpetrator to gain access to the authentication secret.

In the case of biometrics, and where authentication is used to authenticate the device itself, a secure element must be used to protect the biometric information.

Interestingly, plain passwords (including GRIDsure) fare well against physical access to a user's machine, as long as passwords are not cached. This is due to the fact that the password is not stored anywhere on the machine. For this reason, it is highly recommended to not cache important passwords (for example, for your bank account).

Out-of-band authentication methods are compromised if the attacker has access to the device to which the out-of-band code is sent. However, the damage is limited to the time that the attacker has access, since no information about future out-of-band codes can be obtained.

Device Theft

Theft is only relevant to hardware-based solutions. If a user password is required for using the device, then this provides good protection against device theft. We stress, however, that this is only the case when highly secure hardware is used. A simple chip that provides memory protection on a logical level is not sufficient for achieving protection against device theft. Thus, if a password is used for boot protection and disk encryption without additional hardware, then this can be compromised. In contrast, if a separate smart card is used for decryption, then the secret key is not on the device and so cannot be stolen. It is also possible to use password protection in combination with a TPM, or Trusted Platform Module, which is a secure hardware module built into many laptops (as is possible, for example, with BitLocker). This provides security as long as the TPM device is indeed secure.

Summary

The tables below summarize the analysis presented above: ✓ is used to denote that a method provides protection against a given attack and **X** denotes the reverse. In cases where the answer is not a definitive "yes" or "no", we have used the notations ✓* and **X***.

Threats Countered by Hardware-based Authentication Methods

	Password Only	Hardware Secure Storage	Hardware OTP	Hardware Smart Card	Finger-print
Eavesdropping	✓	✓	✓	✓	✓
Guessing	X*	✓	✓*	✓*	✓
Man-in-the-middle	X	X	X*	✓	X
Phishing	X	X	X*	✓	✓
Social Engineering	X	✓	X*	✓	✓
Malware	X	X	✓	✓	X*
Physical access (machine)	✓	✓	✓	✓	✓
Device Theft	N/A	✓*	✓*	✓*	✓
Fingerpritnt Theft	N/A	N/A	N/A	N/A	X

Threats Countered by Software-based and Out-of-Band Authentication Methods

	Password Only	GridSure	Software Secure Storage	Software OTP	Software Smart Card	Out-of-Band Authentication
Eavesdropping	✓	✓	✓	✓	✓	✓
Guessing	X*	✓	✓	✓*	✓	✓*
Man-in-the-middle	X	X*	X	X*	✓	X*
Phishing	X	X	X	X*	✓	X*
Social Engineering	X	X	✓*	X	✓*	X*
Malware	X	X	X	X	X	✓
Physical access (machine)	✓	✓	X*	X*	X*	✓*
Device Theft	N/A	N/A	N/A	N/A	N/A	N/A
Code Interception	N/A	N/A	N/A	N/A	N/A	X

Observations

As can be deduced from the above tables, the main differences between analogous hardware and software solutions lie in their effectiveness against malware and physical access attacks. Also, as mentioned above, the protection these solutions afford in the presence of malware, whether hardware OTP solutions or certificate-based smart cards, is only with respect to future sessions. As previously noted, in the current session, the malware can assume the user's identity, and carry out any activity the perpetrator wishes. In light of these observations, it is worth noting the following:

- A hardware solution is always at least as secure as its analogous software solution. Thus, this does yield a direct security/cost trade-off, as hardware-based authentication devices are usually more costly than software-based solutions.
- Apart from hardware smart cards, the other solutions are all incomparable, as each of them is more effective in some scenarios and less effective in others; for example, there are threats that are better mitigated by software-based tokens rather than passwords (such as phishing and guessing), and, conversely, there are threats that are better mitigated by passwords, such as physical access to a user's system. Of course, when considering all other factors, passwords are rarely the best choice.
- Although fingerprints ostensibly provide security against many threats, the threat of fingerprint theft is very real in many settings. Combining this vulnerability with the fact that the credential does not change over time, fingerprints cannot be regarded as a secure option when used as the sole authentication method.
- This analysis does not include certain factors; in particular, usability and applicability to various use cases are not taken into account.

Conclusions

There is no doubt that the most secure method of authentication is a hardware-based smart card solution. However, if the cost of deploying such a solution is not justified, then the choice of alternative methods depends very much on the expected threat. For example, hardware secure storage only outperforms a software smart card in the case of physical access to the user's machine. Thus, if the user's machine is well-protected on a physical level, it may be preferable to choose the software solution. Due to the prevalence of malware, we do not believe it is possible to rely on software-based solutions for high-security scenarios, unless the user's computers are heavily locked down and the network is highly protected. However, it often makes sense to incorporate a mix of hardware, software and out-of-band solutions for different users, depending on the level of security needed for each user and the sensitivity of the information being accessed.

Choosing a solution that offers a choice of authentication methods has the advantage of enabling the deployment of different methods for different groups of users based on their role in the organization and the level of sensitivity of data and applications they need to access.

Contact Us: For all office locations and contact information, please visit **www.safenet-inc.com**
Follow Us: www.safenet-inc.com/connected