White Paper

Options for Two Factor Authentication

Authors: Andrew Kemshall
Phil Underwood

Date: July 2007

Table of Contents

1. Problems with passwords

Password Strength

In general the longer a password is the stronger it is, it relates directly to mathematics.

Passwords can be associated with a cryptographic value which is dependant upon a number of variables. Using additional variables such as upper case, lower case and numbers can generate even stronger passwords.

Long passwords are good, but consider the following

User 1 password = redcheese6
User 2 password = zglihalq

User 1 password is made up of 2 words and one number, assuming 20,000 easy to remember common words in the English language, strength is 20K * 20K * 10 = 4 billion or in terms of cryptographic strength, a 32 bit key.

User 2 password is 8 characters randomly generated therefore strength is 26 to the power of 8 = 208 billion combinations or in terms or cryptography strength, a 38 bit key.

The User 2 password is stronger and surpasses the strength of user 1. However, to gain this sort of strength usually requires the user to have a photographic memory or to write this password down.

Password attacks

Now that we understand password strength, how do we protect passwords from attack? Unfortunately there isn’t one silver bullet to fix this issue. Strong passwords are hard to remember and it is not uncommon to see a post-it note stuck to the computer screen with the password written on it.

Nearly half of all password attacks are physical, using social engineering skills to obtain them.

Simply reading a post-it note upon a colleagues computer is a simple attack, so is the “shoulder surfing” technique where you watch a person log on and remember the keystrokes that are used.

More sophisticated attacks use software to capture keystrokes at logon and which then sends them to an untrusted person for future use. Keylogging software can be installed on a computer from a virus infection, a Trojan program or a spyware program that was automatically downloaded from a web site (these can all happen without the user being aware). These attacks are especially bad, as you don’t know your password has been compromised and have no way of stopping it until it’s too late.

Network snooping is another prevalent attack, where programs like Cain and Able and Dsniff, capture passwords as they traverse the network. These programs capture Web, FTP and telnet logons (telnet is used with network communication equipment or Unix systems). They do this very effectively and with little user set-up or intervention.

Passwords traverse the network in one of two ways.

In the first method, a password is sent in plaintext, therefore anyone using a protocol decoder will be able to see any plaintext password!

The second way affords some protection by hashing the password. A hashing algorithm is a one way function where the plaintext password is transformed into a hash of fixed length. Common hash programs are MD5 and SHA-1 which have a fixed output of 128 and 160 bit hashes.

However it is easy for hackers to defeat a hash by using software to generate a dictionary file of different passwords and running them through the same hash algorithm. If the output from this algorithm is the same as the hash, the password is known. This technique is known as a brute force attack. Commercial programs are available today such as L0phtCrack.

Modern computing power shows that even very strong passwords can be defeated quickly. A modern Pentium computer running L0phtcrack can sustain password cracks of around 3 million cracks per second. If this program was used with our example users, all passwords would be cracked in the following times.

User 1	>	less than 23 minutes!
User 2	>	less than 20 hours

Finally passwords are hard to manage as the user typically decides their own passwords.

Shown below is an extract of an audit that was conducted for a major client.

342 user account passwords were audited.

29 users had the password **“password”**
1 user had the password **“password1”**
4 users only used numbers, of which two of these looked like a date of birth.
3 users only used 5 character passwords.

The key to defeating all these discussed password attacks is achieved with a one-time dynamic password that can only be used the first time it is sent. Any attempt to record and reply a password renders it useless as the initial password has already been locked. This method is also referred to as strong two factor authentication.

2. Issues with Certificates (without Smartcards)

The use of digital certificates within a user's PC would negate the need to use tokens or smart cards and solves all the discussed issues with password attacks, it does however have it's own management, support and security issues.

Roaming users would be required to carry their certificates with them in order to access services from; for example hotels, Internet café, smart phone or there home pc. Users that import their certificates into a shared computer open themselves up to a serious security issue if they do not remember to delete their certificate after use.

Digital certificates are held within the computer they are using. This leads to a series of security issues around multiple users that have access to the same computer. This approach is particularly problematic where computers are based in public environments, such as schools, libraries or hotels.

Strong authentication principles require that the physical device is linked to the person that is authenticating. This principle is flawed when the physical PC device is shared amongst more than one user. Most home PC's are shared between the family!

Users typically will not backup their private key within their browser and it will therefore be lost if the computer is reformatted or has a disk failure. With the advent of aggressive Adware programs causing negative performance issues to computers, it is prudent to expect unprotected computers to require a complete system rebuild at least once a year.

3. Principles of strong two factor authentication

In order to establish the identity of a person; there are three areas that can be used for authentication:

The first area is to have the user remember a secret, password or pin, something you know.

The second area of authentication is to use a physical device like keys, credit cards or a mobile phone, something you own.

The last area of authentication is a biometric for example a finger print, something you are.

The principle of two factor authentication is to use two of these three areas to give a much stronger level of authentication.

A good example of two factor authentication used in our daily lives is an ATM cash machine. In order to withdraw cash from an ATM machine you must first insert your credit card (something you own) and then enter your pin (something you know). If you loose your credit card you rely on the second factor (the pin) to protect your credit card until you can notify the bank that it is missing.

SecurEnvoy's two factor authentication works on the same principle of ATM systems except that the device you own is your mobile phone instead of a credit card.

Finally it is important that the device you own is authenticated in a manner that prevents all the attacks described in section 1. This is established by the use of a one-time Passcode which after authentication is locked to prevent any reply attacks from hackers trying to shoulder surf, key stroke log, sniff communication protocols or mount a brute force attack.

4. Something you own (Authentication types)

4.1 Smartcards

The key issues for the successful adoption of smartcards are around their dependence on smartcard readers. An authentication strategy based on this method may be feasible in an environment with managed readers such as ATM machines or Smartcard enabled kiosks. However to apply this approach to the on-line Internet community has serious limitations as follows:

Access from a home PC
The adoption of built-in smartcard readers within the home PC market has been almost non-existent. End Users would be required to buy the necessary hardware reader and install the required software. It should be noted for earlier PC's running OS types Windows 9* or Windows 2000 also adds an additional installation step as Microsoft's Smartcard Base software would also require installing. It is likely that a high number of technical support calls would be placed on your company's support helpdesk.

Access from Another Company's PC
Company issued PC's are not typically issued with Smartcard readers and are usually locked down to prevent unauthorised software being installed. Even if the end user obtained a smartcard reader it is unlikely they could install the required software. Use of other company issued Smartcards is NOT viable.

Access from Hotels, Cybercafés or Internet kiosks
Again the adoption of Smartcard readers is almost non-existent and as these environments are locked down, even if the end user had a reader, software would still need to be installed and again it is unlikely they would be allowed to install it. Use of Smartcards in this environment is currently NOT viable.

Access from Smart phones, Blackberry's, PocketPC etc.
Over the past 5 years there has been an exponential growth in connected smart phone devices that include Internet browsers. Due to the compact nature of these devices it isn't possible to include a smartcard reader. Use of Smartcards in this environment is also NOT viable.

4.2 Tokens

Token devices have clear advantages to their users in that they can be used in any environment and they do not require additional hardware or software loading at the users PC or Smart phone. However the following areas should be considered:-

Deployment
Each token must be assigned and deployed to a user that requires authenticating. This typically means a costly exercise of posting the relevant token to the correct user and then ensuring that this user has been received, before enabling it.

Lost or failed tokens
In a poll of 5 companies that had deployed tokens, an average of 10% of the tokens deployed had failed or been lost and therefore required replacing!
It is therefore prudent to allow an additional 10% cost for these additional tokens. In addition, emergency access should be considered for these users that are awaiting their replacement tokens and still require access.

Short term contractors or business partners

These types of users lead to additional deployment costs and don't always return their token after their contract is complete. This leads to additional expense with supplying, administrating and deploying new tokens.

Business To Business eCommerce
When two factor authentication is required as part of a business to business transaction, using tokens is a flawed approach. Consider a purchaser that wishes to conduct business with 20 companies that all require tokens. The purchaser would need to manage 20 tokens that may look identical and find a way of linking the correct token to the relevant company.
Not only is this approach flawed, it would be expensive both it terms of tokens and deployment on the companies that chose this approach.

Redeployment every 3 to 4 years
Some tokens for example those from RSA Security have a fixed life span of 3 to 4 years and will need replacing and re-deploying to all your existing user base. Other token types may seem to last longer however as with any hardware device that is frequently carried by users, wear and tear will eventually force their replacement.

Security
A token is only required when the owner is about to authenticate with it, it is unlikely that a stolen token will be missed or reported stolen until the user next logon on. By then it may be too late!

In comparison a users' mobile phone is their life line to the outside world and is much more likely to be missed especially when they next need to make a call. Therefore it will be reported missing at a much earlier stage.

End User convenience
Given a choice between carrying a mobile phone and a token or just a mobile phone, clearly the later is more convenient. It is frustrating when an end user finds that the only thing preventing them logging in, is the simple fact that they didn't bring their token!

4.3 Phone Based Authentication

Current estimates show that over 55 million mobile phones exist in the UK (a population of 60 million people) with some 2.4bn SMS messages being sent and received each year (Source Mobile Data Association). It is also estimated that for every PC sold world wide, 3.5 mobile phones are sold. This massive scale of mobile phone deployment now makes it viable to utilise this communication structure as a method of strong authentication.

In general two approaches are taken by manufacturers:

1. **Software installed on a phone that creates a one time passcode**

The main issue with software installed on a phone is how to start the application! The current diverse range of user interfaces on different mobile phone types leads to a significant challenge for support staff. They would need to be fully trained in all supported phone types to help guide the end users through the relevant menus and sub menus needed to navigate to the "Java" section on the phone and then start the relevant authentication program.
In addition some phones require a connected PC to install additional software and other types can use the phones browser to download software. Both methods require the end user to understand how to install the software!
It is generally accepted that any approach to support more that one operation system will lead to significant technical challenges. An approach that requires software to be installed on a mobile phone should only be considered for a deployment that supports one or a very limited number of mobile phone model types.
Finally is should be remembered that users that do not have a company issued mobile phone should be encouraged to use their own private phone. Adding software to a private phone is not only un-supportable but invasive to the user's property. In comparison, sending a one

time SMS message to them is no more invasive that any other person communicating with them especially if you can demonstrate that their phone number is kept secret and will only ever be used for SMS authentication messages.

2. **Authentication information sent via SMS in real time**

By utilising SMS (texting) all GSM mobiles can be supported without the need to add or support additional software on the phone. However sending authentication information to a users' phone in near real time is a flawed approach. Expecting this text to arrive after the user has entered their user name and pin is inconsistent as SMS text messages suffer from delivery delays at peek times. In addition, if the user authenticating is located in an area that can not receive a mobile signal especially buildings with large stone walls, the incoming SMS message can not be received!

4.4 Easy to use

In general it is accepted in the token authentication market that a 6 digit number should be used as this is easy to read and when combined with the pin allow for authentication codes that are 10 digits (4 digit pin) to 14 digits (8 digit pin).

Some approaches use complicated human based encryption, where SMS messages contain 20 digits. The user authenticating must then extract their authentication code by looking up their pin in the first 10 digits and translating it with the next 10 digits to form a new encrypted version of the pin. This approach is complex and time consuming for users and in general is likely to be rejected by users that it is forced on.

5.0 The SecurEnvoy Solution

One-time Passcode

A One-time dynamic passcode prevents all the discussed password attacks in section 1.

Easy to use

SecurEnvoys' approach is to use simple 6 digit codes in the SMS text message as this is easy to read and easy to use. When combined with a pin it represents a 10 digit (4 digit pin) or 14 digit (8 digit pin) authentication codes.

No additional software is required on the end users mobile phone

This ensures that all mobile phones are supported and eliminates any support issues associated with managing such software on a diverse range of mobile phones.

Does not require a "real time" SMS message when the user is logging on

When a user first enrols, their first one time passcode is sent to them. Pre-sending the first required passcode gives plenty of time for the user to receive the 6 digit passcode. If the end users mobile phone is temporarily out of range, switched off or the mobile phone provider is busy, the SMS message is stored and retried regularly, typically this would be up to 4 days until it is successfully sent to the users phone.

In the unlikely event that a user does not have access to a mobile phone, their 6 digit passcode can be sent to a regular land line in the form of a voice synthesised message via mobile operators such as BT, C&W, and Orange etc.

The end user then connects to their protected company resource and is prompted for their LoginID, Pin and Passcode.

LoginID:	This is the same as the one used in Microsoft© or other LDAP Directory.
Pin :	4 – 8 digit number or Microsoft Password
Passcode:	6 digit code previously sent to the users mobile phone via SMS

Note: The Pin can be configures to be the current users Microsoft© password or a 4 to 8 digit code.

If an incorrect pin or passcode is entered, a new passcode is sent to the end users mobile phone ensuring that even if the required passcode had been deleted, a passcode is still available.

If more that 10 bad pins or passcodes are entered, the end user is disabled and no new passcodes will be sent. This security measure prevents hackers running brute force attacks.

Passcodes are sent to end users phones in a manner that causes the phone to overwrite the old message. You don't need to delete old passcode messages. The end users mobile phone will only ever have one SMS passcode message from the security server that is dynamically updating.

The Security Server integrates directly into most common directories servers that support LDAP, preventing the unnecessary burden of recreating users or trying to keep them in synchronisation with other databases.

6.0 SMS COSTS

A typical remote access user would only expect to authenticate once per day and would rely on their VPN or a cookie in their browser to maintain the connection for the rest of the session (typically 8 hours). If we take the worst case situation, a user that is always working remotely and thus authenticates every working day. There a 270 working days a year which if you allow for holidays equates to around 250 authentications per year.

High street single user packages from a UK Telecom providers offer SMS messages at 3 pence per message. Corporate high volume accounts can reduce this to 1 to 2 pence per message. Therefore the worst case cost for SMS authentication, based on 2 pence per message is 250 * 2 pence = 5 UK pounds per user per year.

Some company mobile phone business packages allow inter-mobile phone calls or messages to be free if they are between the company's own user-base. These free SMS messages can be utilised via a Wavecom or Siemens modem that supports a GSM SIM chip.