Strategies for using *Passfaces™ for Windows*

Passfaces™ for Windows is an "out-of-the-box" user authentication solution that strengthens access security on Windows networks.

Any security strategy that ignores people's behavior and abilities will fall short of its mark. Passfaces for Windows capitalizes on an innate human ability and introduces a second factor to the authentication process that facilitates user compliance with security policies and alleviates the risk of social engineering[1].

This document discusses how Passfaces for Windows can be used in an enterprise to provide an optimal level of security and how security policies may be adjusted accordingly to achieve improved usability and reliability of access - and administrative cost-savings. It is assumed that the reader has a working knowledge of IT security principles and is familiar with the Passface system.[2]

The Dilemma at Hand

In his book on Authentication, Richard E. Smith states: "Attempts to sacrifice usability in favor of security can sacrifice both." As administrators, responsible for the secure access to enterprise information, you face these competing demands on a daily basis. But as Smith suggests, both elements must be carefully considered in any password policy that is going to reliably deliver authentication.

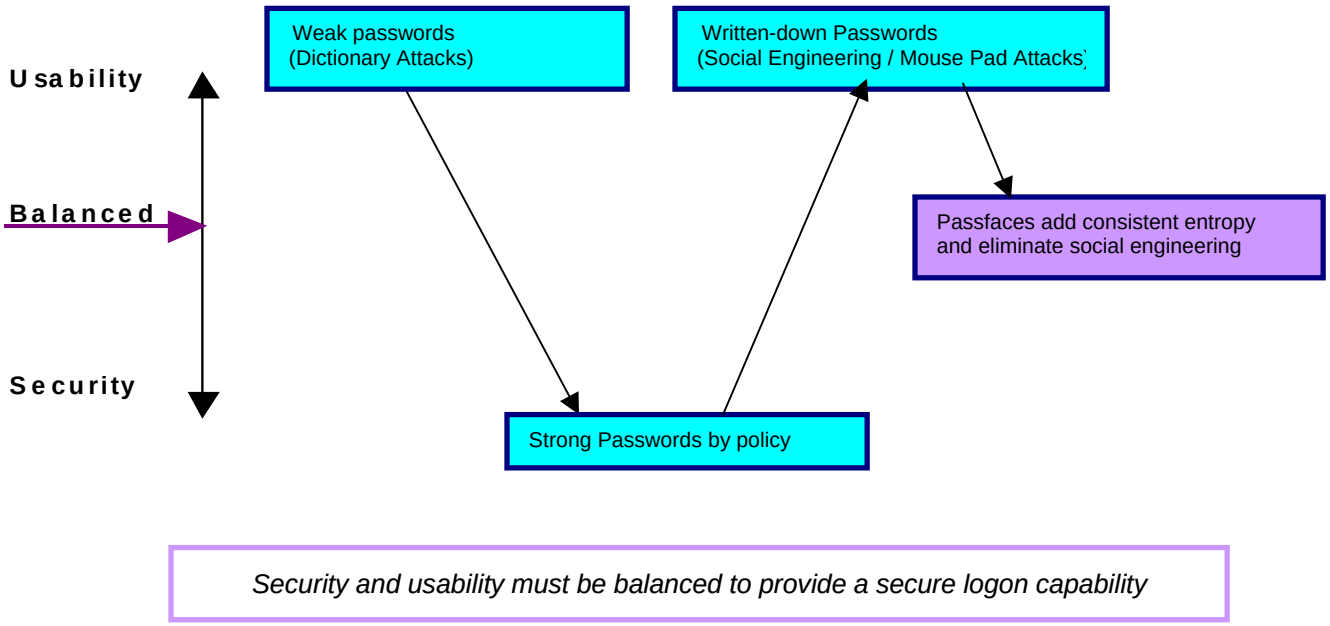
The principle behind authentication by password is very straightforward: if an individual has a "personal secret" associated with her identity that only she knows and she can demonstrate knowledge of that secret, then she must be the person who "owns" that identity. Unfortunately, the attacks that plague simple, easy-to-memorize passwords create a reactive strategy, pushing passwords sharply toward meeting theoretical security objectives. At the same time, the users continue to retake control by defeating those strategies in order to keep the process more usable. Requirements to change passwords frequently result in users recycling a short list of passwords or changing the password minimally. Unless there are crosschecks across systems, reuse of a single password is

[1] Social engineering refers to any activity, deliberate or casual, that allows one person to gain knowledge of another person's authentication secret for the purpose of impersonation. Many convicted hackers attribute much of their success to this activity.

[2] For more information on Passfaces, see "The Science Behind Passfaces".

ubiquitous. This dilemma continues today: advocating strong passwords[3] inevitably results in the majority of users writing down their passwords. Meanwhile, due to our ever-increasing reliance on IT systems for all aspects of modern society, the need for assurance that the person present during the authentication process has the identity claimed grows more critical with every passing logon.

By adding Passfaces to your authentication strategy you are taking a step that will balance usability and security in order to improve both. Passfaces add strength to the authentication process without requiring the user to *recall* any additional information. Passfaces add further proof of the user's presence by taking advantage of people's natural ability to *recognize* familiar faces amongst a group of other (unfamiliar) faces. Because Passfaces are self-prompting, in order to prove their presence, the user need only show knowledge by demonstrating the ability to *recognize* their Passfaces rather than by recalling a string of symbols. This distinction is a subtle but critical one. The user is more in control and the system is more reliable because Passfaces play to innate strengths: the number of symbols that a person can hold in short term memory for *recall* is said to be between 5 and 9 because recollection is a poor skill; the number of faces that can be easily *recognized*, once they have become sufficiently familiar, is boundless.



Another aspect of usability that Passfaces address is in the secret selection process. Users rarely 'think up' their password until the moment they are required to change it, regardless of what else they are doing at the time. The training process on the new secret therefore is poor and the retention potential denigrated. Passfaces are randomly assigned and through an ergonomically designed familiarization process that takes less than one minute per face, the retention is almost perfect. All of the password caveats given to users about social behavior and responsibility for the secure handling of the secret are unnecessary here. Adherence to those policies is the default (i.e. the user feels no *need* to write anything down, and the sharing of details inadvertently is difficult at best).

Entropy and Security

Before a *Passfaces for Windows* logon strategy can be discussed, a few words about password and Passface entropy are appropriate. For the purposes of this discussion, entropy is a way to quantify the amount of work that

[3] According to Peter Tippett, in his February 2002 article in Information Security Magazine, *Stronger Passwords Aren't:* "In the real world, an eight character mixed alphanumeric password is no more secure than a simple four-character password."

needs to be done to gain access to the system. The more alternative choices required for an attacker to try, the more entropy (randomness) there is in the system. The amount of entropy needed in an authentication process is very dependent on other precautions that have been taken to prevent various attacks on the system. This will also play into developing a strategy.

It is a simple calculation to determine the theoretical number of alternatives that would need to be guessed to attack a password of a given length, or of a password in general if even the length is unknown. These calculations can provide some very large numbers. For example, lets consider passwords that are 10 characters long. Even if we restrict ourselves to passwords with combinations of lower case letters, there are 140 trillion combinations possible. In discussions about entropy, where the numbers can become very large and tedious to describe, the accepted fashion to express these numbers is in terms of the number of computer bits required to store the number of alternatives. That 'number of bits' is the log, base 2, of the number. In our example here, 140 trillion is equivalent to 47 bits of entropy or randomness.

However, the password space for user-selected passwords as exploited by attack programs and Internet worms has been shown to be biased to a subspace of about 15 bits (~32,000 choices). Our initial theoretical calculation assumed complete randomness in the selection of the characters in the password. It ignores several important factors that the attackers can count on. For one, language has structure. There are biases to certain letters and combination of letters. Each piece of information we consider reduces the amount of work that needs to be done to recover a password when we are attacking the group of passwords that provide access to the system. Even if the attacker knows that only 25% of the users select words for passwords, the work is seriously reduced. Some attacks use customized dictionaries that give preference to sets of words known to be common for passwords. These biases are introduced because the users want to make life easy on themselves.

To correct this weakness, a strong password policy with strict construction rules can raise that bias to 23 bits (8 million alternatives) and a FIPS 181 10-character random password generator produces a biased space of 40 bits (1.6 Trillion choices).[4] As these steps are taken however, the actual security decreases because users become more inclined to write their passwords down somewhere, like under their mouse pad, or on a piece of paper. It is less obvious how to quantify the entropy of a mouse pad search (1-4 bits) or yellow sticky search but these are real and undermine the security by exploiting the user's need to maintain usability. More importantly, as Peter Tippett notes[5]: "When it comes to strong passwords, anything less than 100% compliance is weak."

Passfaces do not suffer these same biases. The theoretical entropy remains consistent in practice because the faces are system assigned. A user's ability to recognize faces is a well-documented, innate aptitude. *The usability is not at odds with the security because the user is not engaged in activities that are difficult for him.* There is no compunction to write anything down. In fact, it is difficult to share the information unless it is done quite deliberately.

The theoretical entropy of Passfaces, using the standard configuration of 5 faces, is 16 bits (3.2 bits per face). It has no personal or language biases and so the theoretical entropy is the entropy in practice. Where more entropy is needed, additional grids of faces can be added. Later we will see ways in which this can be used to bolster the overall security of an authentication process.

Reliability and Cost

In the real world, no discussion of a strategy is complete without discussing reliability and cost. In an authentication scheme, reliability means that I have confidence that I am authenticating the correct person. This means both that there is no impersonation (no false accept) and that the real person has ready access (no false reject). Passfaces can be used to add confidence in this regard. Because Passfaces depend on an ability to *recognize* self-prompting images, the user's presence for the process is more assured. Unintended sharing of the secret does not plague Passfaces. Because faces are 'unforgettable', the secret requires less resetting and is consequently more reliable.

[4] Authentication, Richard E. Smith, Addison-Wesley, 2002, p.99. See here as well for a more detailed discussion of how to calculate the entropy of a system.

[5] See footnote 3.

The cost of particular importance in an authentication process is the cost of maintaining the system. This includes the help desk cost of secret reset and the cost of user education and (re)training. The retention of Passfaces is excellent because there is nothing to *recall*; *recognition* is easy for the user when the faces are self-prompting. The need to have Passfaces reset and the costs associated with that activity are therefore significantly less than with passwords. The Real User enrollment process is optimized so that the users spend the minimum amount of time familiarizing themselves with their Passfaces (typically 3-5 minutes for five faces) that is consistent with the faces being "imprinted" on their memory. This process also requires minimal support from administrative personnel.

Configuration Strategies

The goal of your strategy is to balance entropy and ease of use in order to maximize security while keeping an eye on cost and reliability. This goal is the same one you have always faced but now you have at your disposal a tool that makes the balancing easier. As you know, the most well intentioned password construction policy can be undone at the drop of a yellow sticky. The plan here is to discuss how you can minimize the impact of user behavior on the security of the authentication process with the help of *Passfaces for Windows*.

Two Factors

➢ Passwords

Let's review what we know about passwords strengths and weaknesses.

- Passwords today are the most common form of authentication because they are built into everything and so the initial cost of implementation is low.
- Modern password systems are built on cryptographic hashes and challenge/response mechanisms, which means that the password is not stored in the clear anywhere (except under the mouse pad). This removes the confidentiality risks on the system side.
- 'Something you know' is an excellent authenticator, given that the secret remains a secret at the user end as well.
- Passwords are expensive because of the administrative costs to reset. The need for complex passwords that are hard to memorize only exacerbates these costs.
- For users, password management is a problem. Their need to write them down and their susceptibility to social engineering are hard to overcome.

➢ Passfaces

And what does *Passfaces for Windows* bring to the equation?

- *Passfaces for Windows* brings a second factor to the authentication process that is user friendly, cost effective, and easy to introduce.
- Like passwords, Passfaces are not stored in the clear anywhere. Only the 'alphabet' or set of decoy screens is stored on the client machine.
- *Passfaces for Windows* provides confidentiality for the secret *at the user's end*. Because Passfaces are system assigned and 'memorable', the burden of *recall* is taken away from the user. The user participates in a way that is natural and easy for him to do.
- *Passfaces for Windows* improves the security of the logon by increasing the entropy of the logon credential and by removing the security risks associated with social engineering and written down secrets.
- Because there is no need to write Passfaces down, the need to reset them frequently is less compelling.

- *Passfaces for Windows* is also low cost to implement. Training and installation are minimal as is the help desk costs for reset.

Combining the Two Factors

Passfaces for Windows is designed so that the two factors are combined for presentation on the server side. One thing this implies is that no attack can be initiated on either factor alone. This is an important consideration and maximizes the benefits of having two factors.

The user enters his password. The user identifies his Passfaces amid the decoys. These are the two independent factors you have to work with. Because *Passfaces for Windows* takes advantage of the underlying Windows logon process, a single secret is sent to the server for authentication. The authenticating secret is not either factor alone but rather a *combination of the two*. The results of the Passface selections are concatenated to the password entered by the user to create the authentication secret that is interpreted as a 'password' by the server. Since the user cannot inadvertently share one factor (Passfaces) of this secret, the secret, from the system perspective, remains a secret at the user's end.

Leverage Points

So what combinations do you have at your disposal? Revisiting your security risk assessment and password policies at this time would give you a backdrop for the options you now have at your command. Regardless of how you decide to combine them, the fact that there are two factors is an immediate gain because the entropy added by the Passfaces is consistent, i.e. it remains the same in practice as in theory. The password guessing tools today have difficulty with concatenated words; this same difficulty works to the advantage of Passfaces concatenated to a password.

➢ Number of Passfaces

One of the parameters over which you have control is the number of Passfaces a person is requested to recognize. The standard implementation of 5 faces is equivalent to a user-chosen password but without the social drawbacks, so the standard implementation increases the combined entropy from 16 bits to 32 bits (This represents an increase of over 4 Billion alternatives). Even in the event that the user discloses the password, entropy equivalent to a user-chosen password remains, thanks to the Passfaces! Each additional Passface that is utilized adds another 3.2 bits. Keep this in mind as you consider your risk factors and are balancing usability with security.

➢ Reduced Password Requirements

Under the appropriate system conditions, alleviating the costs associated with password resets might be achievable in a couple of ways. Adding Passfaces could allow the password requirements to be relaxed. In so doing, the passwords chosen might be more easily recalled and therefore result in fewer administrative resets.

➢ Reduced # of Passfaces

The standard Passfaces presents 5 faces. For sophisticated user groups where the caliber of passwords is high and the security consciousness is high as well, using fewer faces may be enough to gain the benefits inherent in the combination.

➢ Passfaces Alone

Alternatively, given the right system conditions, using Passfaces alone without a password, could be a strategy that completely alleviates the user's password management problem and significantly reduces the administrative reset task. In trials conducted at the University of London, 70% of users were able to recognize their Passfaces 3 to 6 months following only the minimal enrollment process; for users that used their Passfaces to logon during the week following enrollment, close to 100% were able to recognize them 3 to 6 months later.

➢ Password/Passfaces Change Policy

As alluded to earlier, the full authentication secret, from the system perspective, is now less vulnerable to inadvertent disclosure. One of the drivers for password resets is to balance against this risk of disclosure – which is low for passfaces since they cannot be easily written down, nor inadvertently disclosed, nor used on other systems (such as the Web). Therefore the time period before resetting either the password or the Passfaces can be extended, improving the user experience and the cost profile for administration without introducing additional risk.

One exception to this is in the event of a password reset by you as the administrator. Because good system design precludes recording either factor of the credential (i.e. Passfaces and password), you are not able to keep the current Passfaces in place when assigning a temporary password and forcing the user to select a new permanent one. After logging on with the temporary password, when the user selects his new permanent password, new Passfaces will be presented for training as well.

This is not the case if the *user* decides to change his own password (using Ctl-Alt-Del). His successful authentication makes the current Passface information available to combine with the new password. The Passfaces will remain the same unless you as the administrator have decided otherwise.

Risk factors

Cognometric[6] authenticators are susceptible to attacks that leverage 1) lack of confidentiality and 2) unrestricted network access (dictionary attacks). The risk of disclosure, or lack of confidentiality for the secret on the user end is mitigated by using Passfaces. As discussed earlier, most systems today address the system confidentiality concerns for transport and remote storage of the authentication secret by using cryptographic hashes and challenge/response protocols.

On the other hand, if the authentication process you are using has no lock out or delay after repeated logon attempts, or if there is reason to consider off line dictionary attacks viable, then the amount of entropy you want in your authentication secret will go up. Keep in mind that dictionary attacks against the database of secrets are looking for the weakest secret in the group. With Passfaces as part of your strategy, this minimum secret has been significantly strengthened.

Environmental Threats

Below are some possible strategies that respond to three different sets of environmental conditions. Any strategy must be customized and different strategies may be needed for different aspects of a system's authentication profile. Threats do not necessarily translate into risks. Risk is the result of identifying a threat and assessing its likelihood and potential loss within the context of the system's security profile. An authentication mechanism is a part of that profile and should be adjusted as the risk profile dictates.

[6] A cognometric is a knowledge factor. It includes those that are based on *recall*, such as PINS and passwords, and those that are based on *recognition, i.e. Passfaces.*

1. **Environment:** Extranet, Internet, or Remote Logon to Intranet
Threat profile: Dictionary Attack is possible; no lock out policy
Strategy: Add Passfaces, a second independent factor, to the login process.
Benefits:

Security:
- Introduces a second factor that cannot be attacked separately
- Increased entropy: the level of the attack is increased by requiring 60,000 attempts to be tried for *each word* in the attack dictionary. The 'weakest' credential in the hashed password file is strengthened.
- Dictionary attack is more difficult to construct because of the concatenation of the Passfaces outcome to the password.
- Social Engineering is hindered – complete authentication credential cannot be obtained.

Usability:
- Passfaces do not demand that the user recall more arbitrary information (as with a PIN for a second factor)

Cost:
- Training on Passfaces is automatic upon enrollment and requires no administrative involvement.
- Passfaces presents reduced burden on help desks (as opposed to PIN used as second factor)

Reliability
- By design, Passfaces give more assurance that the owner of the credential is present during the authentication transaction

2. **Environment: Intranet or extranet with lockout in place;**
Threat Profile: Dictionary Attack not likely
Strategy: Reduce the requirements on the construction of the password or relax the reuse policy for passwords, and/or reduce the number of Passfaces.
Benefits:

Security:
- Two factor authentication in place
- Social engineering is hindered
- Consistent entropy is increased
- Dictionary attack is more difficult

Usability:
- Reduced burden on user to *recall* password since a simpler password is required

Cost
- Scheduled password changes can be less frequent
- Less password retraining required
- Reduced demand on Help Desk

Reliability
- By design, Passfaces gives more assurance that the owner of the credential is present during the authentication transaction

3. **Environment:** Internal enterprise access from known machines by known employees
Threat profile: Dictionary Attack (exhaustive logon attack) presents low to no risk
Strategy: Omit password and use Passfaces alone
Benefits:

Security
- 5 Passfaces provides equivalent entropy (to a user chosen password) but cannot be inadvertently compromised
- Because Passfaces are self-prompting, introducing a lockout or delay policy is possible without creating a burden for the help desk
- Social engineering is eliminated

Usability

- Completely user friendly – no random data to *recall*
- Minimal help desk support required
- Scheduled credential change can occur less frequently

Cost

- Minimal training and help desk costs

Reliability

- By design, Passfaces gives more assurance that the owner of the credential is present during the authentication transaction

Conclusion

Successful personal authentication depends upon a security profile that addresses the risks inherent in a given system. No single component of any security strategy can provide THE comprehensive solution. Modern authentication systems today have secure database management and transport protocol designs in place to mitigate the risks associated with exhaustive and/or dictionary attacks. Moore's law[7] reminds us daily that no user can recall a password that will not succumb to tomorrow's computerized attacks. For this reason, cognometric based authentication needs to employ lockout or delay policies that stop computerized attacks through the user interface. With these other system contributions to the authentication process in place, confidentiality of the authentication secret *on the user end* remains as the weak link in a password based system. By adding Passfaces to the authentication credential, we add entropy without taxing the user to recall additional random information. Passfaces are not susceptible to social engineering, cannot be easily written down nor exposed on other systems; and because they take advantage of skills that come naturally to the user they allow the user to comply with the security best practices without difficulty. How Passfaces are configured into your particular authentication process depends on the other components of your security risk profile. In many cases, it can be used to reduce the high costs associated with password resets. In all cases, however, security and reliability are improved without sacrificing usability.

[7] In 1965, Gordon E. Moore, now Chairman Emeritus of Intel Corporation, first observed the "*doubling of transistor density on a manufactured die every year*". The doubling of computational power of microprocessors every 18-24 months became known as Moore's Law. The phenomenon has held true since that time and is currently expected to hold true for at least the next 10 years.