

Class: IT & Strategic Opportunity
Section: 4002-873-90
Project: Microsoft .NET Passport
For: Professor Garrison
By: Robert M. Pufky
Due: 5/15/2005

Introduction

Microsoft released the .NET passport user authentication model in hopes that it would become the standard “1 user, 1 password” account for websites across the Internet. Their initial belief was that by simplifying access to secure sites on the Internet, their product would gain dominance as well as make accessing knowledge and resources on the Internet much easier. Years later, this software still has not taken root as the global standard for user authentication – what was the underlying causes as to why this software failed? Were the causes in concept only or were there other causes that lead to this non-acceptance as well? Are users really ready to move to a simpler form of authentication if they say they are? Understanding the successful concepts being used on the Internet today, and what wrong with the Microsoft solution we can determine what needs to be done for future systems implemented corporate or Internet wide. Single sign-on accounts for the Internet are a great knowledge management idea, but need to be strategically implemented for there success.

What is .NET Passport?

The concept that Microsoft .NET passport tried to implement was an internet application commonly called single sign-on software. This software allows a user to know a single username/password and use it to authenticate them as who they are. Conceptually, a single sign-on Internet application (such as .NET passport) would be a enormous advantage to Internet users everywhere – instead of knowing ten, twenty, and possible more username/password combinations for the Internet sites they commonly visit, it would allow them a single username/password for all sites. This removes the barrier of trying to remember these combinations to access information, effectively making access to knowledge simpler for everyone involved – businesses and customers alike.

Theories For Successful Implementation

To understand what happened with the Microsoft passport concept, a general understanding of where the Internet is today must be first understood. The Internet “E-conomy-already rivals the size of century-old sectors such as energy, transportation, and telecommunications” [2]. The size of the Internet requires “Business leaders worldwide [to] recognize the strategic role that the Internet plays in their company’s ability to survive and compete in the future” [2]. This enormous size of the Internet demands that any successful implementation of any product or software to be used across the Internet in its entirety, must consider the currently successful strategies that companies are using. Hartman and Kador state that there are six key points that should be considered for success on the Internet: *Customers* (ability to rapidly respond to demands and deliver value), *Globalization* (equal playing field for big and small companies), *Partnerships* (complementary business alliances that create a unique set of interwoven dependencies and relationships), *Employees* (make decisions in the best interest of the customers), *Culture* (dedicated to customer satisfaction), and *Access* (balance the need for both security and open access to information). These six factors play a vital role in the success of a product on the Internet. Businesses need to consider the entire world for their product development, not just the local culture they are in. They need to partner with complimentary businesses to provide superior customer satisfaction, as well as having dedicated employees to make it happen. Security needs need to be considered, as well as a good corporate culture. These six factors must be considered when designing future applications for the internet.

Regulation was also discovered to be critical in future Internet success. To “Truly hold the promise of borderless commerce, laws and taxes must be harmonized at an international level” [5]. The Internet knows no bounds, and the development of laws to enforce e-commerce cannot keep pace [1]. Recently, it was described that regulation essential for business success and required governments to do its part [5]. “Firms perceived a tradeoff between the benefits of e-commerce and the uncertainty and risk due to a lack of developed formal and informal institutions guaranteeing the fundamentals for a market to work: property rights and enforcement of contracts” [5]. Unmistakably, regulation must be enforced at some level to increase the usage of Internet applications for both the business and the consumer.

Another component for the success of internet implementations is security. Applications must be developed to combat the “challenges to the growth of electronic commerce: one is the security system on the Internet” [1]. Security, both at the encryption level and the protection of both the customer and the business must be maintained for Internet applications to be successful. “With the global, but insecure, Internet being the primary carrier of electronic commerce transactions, web sites can be counterfeited, identities can be forged, and the nature of transactions can be altered” [1]. This effectively creates a barrier to the success of e-commerce [5] through the non-enforcement of both the access and customer points of the Hartman and Kador model. Therefore, a global Internet application, such as single sign-on, must be written to use encryption to protect everyone involved in the use of that application.

To further prevent fraud and provoke success on the Internet, third party verification needs to be used. Without third party verification of parties, the “risk reflects the uncertainty a buyer or seller may have about dealing with an anonymous firm or customer through online processes” [5]. Without third party interaction and verification, establishing relationships would be nearly impossible [5]. Firms simply have to “reveal their policies on information disclosure, informed consent, and handling disputes” [5]. By using a third unbiased party, a neutral ground where these policies are shown will provide the customer with more background information on companies. With these third parties, reputation tracking of customers may also be kept (akin to the feedback system on e-bay). “Reputation tracking at least provides more information than previously existed” [5] – allowing companies to have knowledge about past customer history, as in non-payments. “Certification authorities serve as one form of trusted third party by authenticating the identity of each trading party in a transaction by issuing digital certificates ... [and] to protect security and privacy based on encryption technologies” [1]. Third party tracking as well as reputation tracking also derives from the Hartman and Kador model through partnerships and Culture. Separating the tracking of user and company history to a third party, as well as verification, bias's can be removed to allow a more friendly Internet experience.

All of these factors mix together to help create what can be called the basis for success of any Internet application: trust. “In the faceless electronic market, it is easy for people to commit fraud and misrepresent their products or services” [1]. These factors, including security, regulation, third parties, and the six key points to Internet success all help to enforce trust between the business and the consumer. By implementing these ideas, consumers and businesses that “inevitably face many difficulties in selecting reliable, suitable

vendors ... [and the] short-run temptations to cheat” [1] can avoid this altogether. By creating a system “governed by informal social norms” [1], social pressure and reputation will aid in both businesses and consumers being honest [1]. Implementing a third-party system, with high encryption, and no affiliations (an independent body); which has the power to enforce decisions made onto consumers and businesses alike all over the world, will boost trust in the Internet single sign-on application. In Microsoft’s case, they state that “web-services will allow code from one system to talk to code from another system. Code on one web server[,] can invoke code on another web server. This represents a loosely coupled environment. ... In this environment, the code cannot trust the code that it is invoking through a web-service” [3]. What Microsoft is trying to state here is that in the current situation, applications calling applications from other sites are inherently non-trust worthy. For future solutions to work, this breakdown in trust needs to be fixed. Simply put, correctly implemented, it will put a face and a history to a username and password – promoting trust on both a social and governmental level. Without fail, trust needs to be gained for any single sign-on program to be successful.

How Do These Apply To Microsoft’s .NET Passport

Applying these concepts to the Microsoft implementation of single sign-on; it can be seen where the breakdown could have happened. In Microsoft's case, there was no third party verification or reputation tracking – in fact, there is virtually no verification at all – anyone at anytime can have multiple .NET passport accounts. Additionally, regulation is virtually non-existent at both the governmental and Internet aspects of the .NET passport. Microsoft can ban the account, but a user simply has to re-create an account under other credentials with another e-mail address. There are currently no laws making this illegal. Security is another major issue of .NET passport. Whether a site users SSL to authenticate your .NET passport or not is entirely up to that site – allowing username and password combinations to be sent in plain text over the Internet. It has even been suggested that “Inadequate feedback and lack of control ... may even lead to users rejecting the technology” [4] meaning that if the interface that Microsoft had developed for .NET passport not been uniform and easy to use, this might have also contributed to the failing of this application. Each one of these things individually might not have affected the acceptance of .NET; but the combination of all of them unquestionably lowered the amount of trust in the system. Microsoft .NET passport could have failed in its acceptance of a single sign-on application from the lack of trust – lost by the aforementioned points.

Another Way?

With these general concepts, how does should an application for single sign-on be developed for the internet? One way is to consider what banks in Sweden are doing to promote online banking – giving out security boxes with their accounts. A security box is connected to a computer, and after the user enters a certain pin combination on the security box, a random password is generated from the box, and sent to authenticate on that user’s behalf. The authentication password in this case is many times stronger then a normal “user-entered” password. These banks do this because “trust relates strongly to users perceived control over personal information ... only a physical artifact is truly under he control of its user” [4]. This physical object gives the users perceived control over their information even though in reality, it is doing nothing more then entering a password for them. Increasing the trust

through control of information is a good starting point for future development of a SSO application, but should not be solely relied upon.

Creating A Successful SSO Application

The first aspect of a SSO application to consider is whether it should be implemented as a separate multi-platform application, or as a server side implementation. A server side implementation would move the main authentication portion of the SSO application to the companies hosting the website, leaving only the end user to enter in a username and password – with very little feedback as to what is happening. This would open up opportunities for stealing user information and phishing. An individual application separates the end user, the company they are trying to connect to, and the verification process. This brings more control to the end user, allowing them to specify what information they want to give to the company; in turn that company can decide whether this information is enough to allow the user to use their site. This also incorporates the idea currently being used for online banking – separate devices to authenticate. For SSO applications to be successful it should be developed with the concept of maximizing the trust created between the three parties, splitting power between them; therefore, the SSO application should be written as a separate application.

Applying the concepts presented earlier, a major factor in the development of a successful SSO application would be the incorporation of security. As mentioned before, this is a critical need for users to have trust in the system, and as such, needs to have extensible security measures for continued use in the future when more secure encryptions are able to be produced. For this, the SSO application should have a public-key encryption module, as well as the main program, which lends itself to being upgraded as time progresses. Because of the need for this security module to be upgraded, the SSO application should check for updates from an update server after a successful login; and automatically download and install the encryption updates in the background. To ensure non-tampered package identities for server as well as a md5-hash will be verified before installation of the package. The application will be hard coded to report any package verification errors, as well as pertinent information (such as IP, reasons why it failed, certificates used, etc) to a reporting server. For the end users, the current encryption being used will always be displayed, as well as options on the application denying credential sending if the encryption isn't high enough (which by default should be set to the highest level). The more personal the information shared, the higher the required encryption level. Transactions with credit card information will always require the highest level of encryption.

In addition to the security, there should be separate levels of access granted by how a potential host wants to use this authentication program. At the most basic and free level, authentication will be provided, but with no access to personal information, even if the option in the SSO application is set to allow them access – the SSO application will simply be a way to “login” to the site to provide non-anonymous access. At the next level, which shall require personal contact information, and a legal contract to be signed with a small nominal fee (to provide the service); will provide the company with more access to personal information, such as mailing information, name, and other contact information if specified by the SSO application. The third and final level of access will require a company to setup contact information, sign legal contracts, pay a nominal fee, and require a verified physical

presence; will allow a company to access more personal information, including credit card information – again, only if the options are set to allow it on the SSO application for the consumer.

Reputation tracking is another vital aspect of the SSO application. By informing people of past interactions and experiences with the company, greater trust can be created allowing superior acceptance of this SSO application. In the application, the current access level that the company has will be displayed (as levels 1,2,3) along with a link showing the difference between the access levels; and another link allowing the end user to change what information they can access at any one time. In addition to this, current feedback from other customers using the site will be able to be viewed with a single click; showing their comments, whether they actually used the company; and the company's response. Any negative feedback, or a site that was previously on as another site, will use a tray-flashing heuristic to grab the user attention that something may be wrong. Included in this information, the physical company contact information, including phone numbers and addresses will be shown; as well as how long they have been using this service. Notes on the company, including vital history (pertaining to the customer) will also be able to be accessed through this application. Additionally, a status indicator light for quick reference will be used on the tray icon – allowing the user to always know at a glance what state the company is in. A green light will indicate that the company currently has had no major issues or problems; a yellow light will indicate that the company has not received enough feedback yet, has just started using the service, or has a small portion of negative feedback. A flashing red light heuristic will be used to alert the users that there has been a major problem with the company, and that extra caution should be used when dealing with the company. Clicking on this light at any time will display the information that is causing this warning to appear. See **figures 1, 2** for example diagrams of this type of program.

Protection of personal information will also be managed from all three entities using this system. As previously mentioned, access to what types of information a company can have through this application is determined by the contract level the site uses when applying for access. The user can set what personal information the companies are allowed to 'see'. This information will be accessible only if the access level of the company is high enough to allow them access to read that information. For instance, even if the user allows all companies to view their mailing information and name, only companies with level 2 or higher access will actually be able to read their information. This will also work in the reverse – if the user does not want the company to access some information, they can deny access to it; denying companies even with level 3 access the ability to 'read' that information. This also works the same way for companies. Companies will have the ability to set up to require information from the end user at what their access levels allow – users denying them access will have the option to deny access to the user, prompting as to why they were denied. Credit card information and personal contact information (such as phones numbers) will never be required to "login" to a site, only for actually purchasing an item.

The final and most important point to the success of the SSO application is the third party. Although this is the least technical of all the concepts needed to create a successful implementation, it is also one of the most important. A third party verification and maintainer of the SSO application will ensure an unbiased approach to companies and users. This party should be created from any number of groups wanting to be involved – allowing

each group equal access to the project. The application will be developed with a modified open source model – anyone belonging to this party will have access to it as well as the ability to suggest improvements on it. Additionally, this party will need to have the legal power to settle disputes created the ability to accurately verify both companies and end users. The social and political aspects of this third party are too complex to be covered in this paper.

Conclusion

Although there are many speculations as to why Microsoft's .NET passport implementation has failed to take root as the global standard of authentication for the Internet, it was shown that a major factor in the acceptance of these types of applications is trust. It can therefore be said that a major reason as to why .NET passport failed was trust – or lack thereof – in the system, in the medium or in Microsoft itself. This trust, built from a combination of security, access, third parties, regulation and the six points was clearly lacking in the majority of the .NET passport implementation. Learning from the mistakes made by Microsoft, researching theories and applying a social context, a better and more acceptable SSO application can be developed. This application, built on the concepts of developing, maintaining and expanding trust between users and Internet sites will effectively weed out untrustworthy sites, promoting and proliferating this application even farther. It is commonly known that Microsoft's .NET passport solution failed to gain global acceptance; by learning from this, researching new theories and applying at that has been learned to a SSO application, its success can be ensured.

Figure 1: Security settings for end user

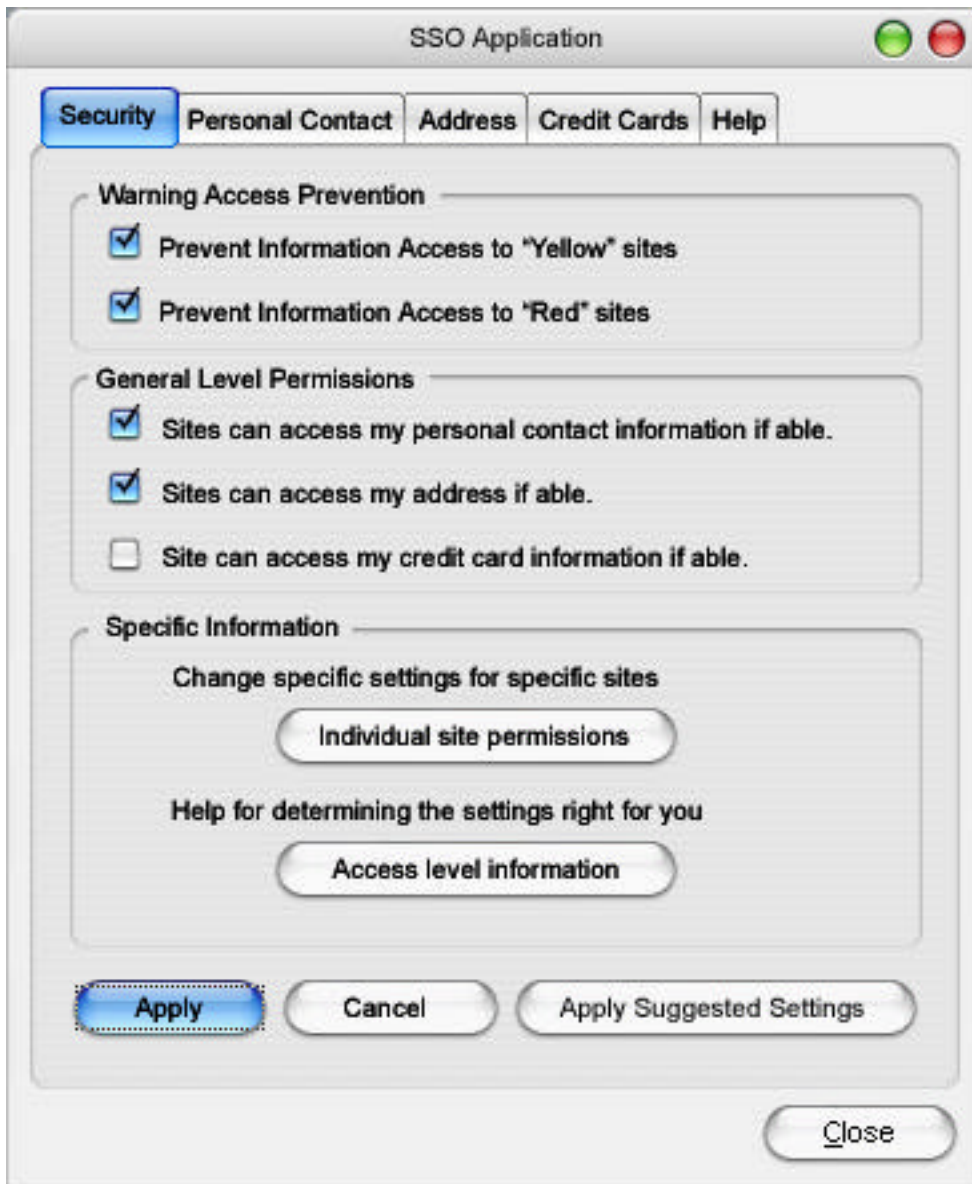
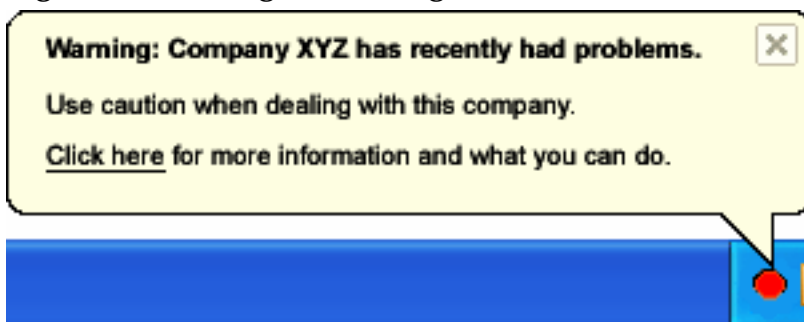


Figure 2: "Red Light" warning screen



References

- [1] Ba, S., Whinston, A. B., & Zhang, H. (1999). Building Trust in the Electronic Market Through An Economic Incentive Mechanism. *Proceeding of the 20th International Conference on Information Systems*, 208-213. Retrieved April 20, 2005, from ACM Digital Library database.
- [2] Hartman, A., & Kador, J. (2000). *Net Ready: Strategies for Success in the E-economy, 1st edition*. Retrieved April 13, 2005, from ACM Digital Library database.
- [3] Lenci, M., Brown, D., Holliday (Microsoft), N., & Beckleman (TSRI), S. (2002, May 16). *Microsoft / TSRI .NET Platform Strategy*. Retrieved April 13, 2005, from <http://www.mainframemigration.org/ShowPost.aspx?PostID=384&fm=wp>
- [4] Nilsson, M., Adams, A., & Herd, S. (2005). Building Security and Trust In Online Banking. *Conference on Human Factors in Computing Systems*, 1701-1704. Retrieved April 27, 2005, from ACM Digital Library database.
- [5] Schoder, D., & Yin, P.-L. (2000, December). Building Firm Trust Online. *Communications of the ACM*, 43(12), 73-79. Retrieved April 19, 2005, from ACM Digital Library database.