A SUPPORT ARCHITECTURE FOR MULTI-CHANNEL, MULTI-FACTOR AUTHENTICATION

Karen Renaud, Richard Cooper, Mohamed Al Fairuz
Department of Computing Science, University of Glasgow
18 Lilybank Gardens, Glasgow, G12 8RZ
{karen,rich,mafairuz}@dcs.gla.ac.uk

ABSTRACT

As more and more critically confidential information is managed electronically by distributed information systems, efforts to gain unauthorised access to that information become more prevalent. Traditional authentication mechanisms, such as passwords and PINs, are fairly weak mechanisms for controlling access to critical resources and excluding unauthorised users. This is because mechanisms which utilise only one “factor”, such as a password or PIN, are increasingly easy to subvert. It has become essential for us to consider making use of multiple mechanisms and/or channels to strengthen security. For instance, an authentication attempt that requires a password to be entered may require verification by means of entry of a one-time password, on another channel, which is delivered to the user’s registered mobile phone. In this paper we propose an architecture to support multi-channel authentication. The architecture allows a range of authentication channels to be deployed and consults the user about his or her personal preferences within risk-based constraints. The user is given the flexibility to choose from an available selection of channels and mechanisms which will be combined to achieve successful, secure and flexible authentication. Such a mechanism can be associated with a secured resource and thus improve the security of the access mechanism. Furthermore, the personalised choices can be changed by the users, making it easier to foil potential intruders by introducing unpredictability into the system.

KEYWORDS

Authentication, Channels, Security, Architecture, Scenarios.

1. INTRODUCTION

Access to distributed applications has been revolutionised with the advent of ubiquitous, uninterrupted access from a variety of devices such as traditional personal computers, PDAs and mobile and touch-tone phones. Some distributed sites need to reliably identify their users in order to allow them to purchase items, or deliver personalised services. Identification cannot be accepted on face value and therefore is verified by means of an authentication step.

Authentication in the physical world is performed by means of people recognising each other. In a distributed virtual world, with no one-to-one physical access, there is a need for a viable alternative. The computing world thus adopted an old military tactic – the secret password. When computers first came into popular use the users were given a single password, and had no difficulty in using it. The mechanism worked, and worked effectively. As long as people kept the passwords secret we could be relatively sure that knowledge of the secret authenticated the user. However, computers are now ubiquitous and being used in everyday corporate life in ways that the early users would never had dreamt of. Everyone now has a computer on his or her desk and they have to remember various passwords and PINs – one for the network, another for the email account, yet another to get into the library website. At home, they have a number of bank cards, each with its own associated PIN, and any e-commerce they engage in requires its own password. It is no wonder that people resort to reusing their passwords (Ives, 2004), sharing passwords or writing them down (Gaw *et al*., 2006).

Even if the user behaves very securely an intruder can still get hold of the password, either by guessing it or by breaking it with easily obtainable brute-force software. There is a consensus that the era of the password is drawing to a close. The only problem is that a viable alternative has yet to emerge. Many alternatives have been proposed, but most simply do not fit the bill as well as the everyday password. Some require the use of specialised hardware, others are not accessible to all users in the same way as a password is. In the face of this lack of a viable alternative, we have to find ways of strengthening the lowly password, to extend its useful lifetime until such time as a viable alternative is available.

Authentication can traditionally be done in three ways – by knowledge, by biometric or by ownership of a token. The latter requires accompanying knowledge as well in order to prove that ownership is legal. Biometric authentication mostly requires the use of a biometric reader, which is not always available to end users and which is easily compromised in an uncontrolled environment such as the Web. Hence most distributed systems will utilise knowledge-based authentication, and this relies on the secrecy of the knowledge being maintained by the user. Unfortunately humans are not very good at keeping multiple secrets so they either reuse the same secret on different systems, or they choose weak and predictable authenticators, opening the way for attackers to gain access to their online accounts (Gaw *et al*., 2006).

The previous discussion focused on the use of a single authenticator: so called single-factor authentication. There is overwhelming evidence of the weaknesses of single-factor authentication and the owners of various financial websites are moving towards two-factor authentication in an attempt to strengthen the identification mechanism (Adams & Sasse, 1999; Besnard & Arief, 2004, Ives *et al*, 2004). The second factor enhances security by introducing some randomness into the authentication process. So, for example, the user may receive a special PIN via his mobile phone, or be issued with a card that produces one-time passwords. The rationale behind this is that the authenticator, if observed, will not be useful for authentication at a later time because it is only valid for one authentication attempt. This makes the mechanism much stronger because it becomes resistant to shoulder surfing (covert observation by third parties) or password recording by means of key logging software which is active on a person's computer without their knowledge. It does not protect users against what we will call "real time" attacks – attacks that occur while user is busy accessing his account, after he has been authenticated. There are at least two kinds of attacks that fit into this category (Schneier, 2005):

1. *Phishing*: An attacker sends an email to a user, with a link in it which directs the user to a website that appears to be the genuine website, but which is, in fact, a fake, proxy-type website. The user then logs into the website giving his password, and the random part as provided by SMS, for example. The fake website quickly relays this information to the genuine website and gains access to the user's account.
2. *Trojan attack*: the attacker installs a piece of software on the person's computer. This software watches the user logging into the website and then submits fraudulent transactions while the user is logged in using valid session variables to slip under the security radar.

Schneier argues that we should not be authenticating users, but that we should rather authenticate *transactions*. Hence any action that does not have side effects can adequately be protected by a password, accompanied by the judicious use of a three-strikes lockout policy to prevent brute force attacks. As soon as the user wishes to carry out an action with side effects, such as a transfer from a bank account, or access to sensitive information which could be used against the user, an extra authentication is carried out. Simply using another password or PIN here does not achieve the level of security required. Based on Schneier's argument, this is where the random authenticator should be introduced.

ASB Bank (New Zealand) already makes use of this kind of mechanism. They attempt to authenticate Internet users' transactions by providing a random PIN via an SMS message. The user logs into his Internet banking account using a username and password as usual. However, to transfer money out of his account, he will need to provide a PIN, which is delivered to his mobile phone. Unfortunately, this does not mitigate the threats posed by Phishing and Trojan attacks. Consider a user who is logged into a fake website, which is relaying all his entries and receiving and relaying responses. The fake website may submit a fraudulent transaction in place of the transaction the user wishes to carry out. Perhaps the user wishes to pay his utility bill, and authorises a transfer of £50 to the utilities company. The fake website sends a request for a transfer of £1000 to the attacker's account. If the bank requires the user to enter a special PIN to validate the transaction, the fake website will simply relay that PIN and the bank will assume that the transaction has been approved. Soon after two-factor authentication was vaunted as the "solution to the password problem", stories of successful phishing attacks on two-factor authentication systems were being published (Sanders, 2006).

We will propose a mechanism in the following section that deals with these real-time threats by utilising separate channels which cannot be predicted or observed by Trojans or fake website attackers, thus protecting the user from these kinds of attacks. Mizuno *et al.,* (2005) propose using multiple channels during authentication, but their main focus is to reduce the risks of people using publicly accessible client machines.

They make use of an auxiliary trusted channel, such as a mobile phone, to confirm the user's continued presence in front of an untrusted client machine. They use a 2D bar code, which is capture by their mobile phone and submitted via a separate channel to the server. Our scheme is both simple, more flexible and more secure. We make use of multiple auxiliary trusted channels, standard mobile features such as SMS texting and we authenticate *transactions*, rather than users, as recommended by Schneier.

2. MULTI-CHANNEL, MULTI-FACTOR AUTHENTICATION

Authentication mechanisms were originally developed in the context of a single means of accessing a particular system, with all its associated applications. Most of these "users" were, in fact, dedicated computer professionals who understood security requirements and had no difficulty with the access control mechanisms. This was followed by a trend for increasing numbers of stand-alone applications requiring passwords. Two things happened: firstly users now had multiple passwords to remember, not just one, and this became more challenging. Secondly, the access control mechanism was now in the public arena, being used by people who did not have a computing background and only a rudimentary understanding of the need for access control. The situation really got out of hand with the advent of distributed applications such as web-sites and ATMs which opened the way for a whole new type of attack and multitudes of unknown and unknowable attackers were able to exploit weak password choices..

It has been obvious for some time that the use of passwords to control access is, at best, a sub-optimal mechanism (Morris & Thomson, 1979). Passwords can be observed, recorded (and then stolen), guessed or broken by brute force. The undeniable fact is that reliance on a single factor or channel for authentication simply does not deliver the level of security that is required for any but the most innocuous access to distributed applications. The relatively recent advent of ubiquitous, uninterrupted access to distributed applications from a variety of devices, such as wireless computers, PDAs and mobile and touchtone phones, has opened up a new avenue for bolstering the ever weakening password artefact. Walton (2005) has advocated a similar scheme by recommending the use of multiple biometrics in order to authenticate people. Unfortunately biometrics are problematical because biometric reading devices are not yet ubiquitous and they are notoriously easy to spoof in a distributed environment (Furnell & Clarke, 2005, Roberts, 2007).

The most popular use of multiple channels is to offer various alternatives for access to a website. For example, a banking customer can access her account on the web, through an ATM or by using her mobile phone. Another option that opens itself up with the availability of auxiliary channels is to make use of them to verify identification or to authorise transactions that have side-effects. The availability of multiple devices which can be used to augment the primary access channel means that the ubiquitous password can be shored up and strengthened with minimal outlay from the resource owner. It also allows us to differentiate between website actions that are fairly innocuous (requiring weak authentication) and those that have potentially damaging side-effects (requiring strong authentication) and only asking the user to put the effort in that is commensurate with the risk that attends their actions. In this paper we describe an architecture that utilises multiple channels in concert.

Having multiple devices available means we can now strengthen the access control mechanism by making use of multiple channels through which identity can be verified. For example, a banking customer can access an account on the web, through an ATM or a mobile phone. The user thus chooses his primary access channel simply by using it as the channel of first contact with the website. He will then usually have various auxiliary access channels available to him, which we can utilise in order to verify identity. If an intruder wishes to gain access to an account it is not enough merely to guess a person's password – he also has to gain control of the person's auxiliary channels, which means that he needs physical access to his victim.

We therefore propose that we utilise multiple independent channels in order to strengthen the traditional and almost ubiquitous one-factor, one-dimensional authentication in popular use today. However, it is necessary first to identify the weaknesses inherent in some of the two-factor authentication mechanisms currently being touted as the solution to the "password problem". The current forays into two-factor authentication have two weaknesses:

1. They utilise a single entry channel. A random PIN is delivered to the user by a different channel, but is entered by means of the same potentially compromised channel used for the initial

authenticator. Levene (Levene, 2007) reports on a rollout of new chip and pin mechanisms by UK banks, which will deliver a one time password to users. This mechanism does indeed confirm that the user holds the valid token for the account, but does not prevent the real-time attacks mentioned in the previous section. Since the channel may be compromised, the attacker could grab the entered one time PIN and use it to authenticate a fraudulent transaction.

2. Current mechanisms are inflexible because they require the use of two specific and hard-wired channels. Hence, if the system requires the use of a mobile phone and the user does not have a mobile phone, or the user’s mobile has been stolen or broken, he cannot carry out side-effect transactions on his online account. The use of rigidly defined channels also makes it easier for attackers to predict and perhaps observe one time passwords delivered to these channels. Our proposal, on the other hand, has the following core features:
   - The user should be given the flexibility to choose the channels which suit him best, at any time. The minimum number of channels to be used will be determined by the resource owner after assessing the risk associated with the resource or kind of access being requested. The user himself is then free to increase the number of channels to satisfy his own desire for increased security. The user may also specify a number of channels through which authentication might be attempted, but require only a subset at any one time.
   - We will utilise independent entry and delivery channels so that even if one particular entry mechanism is compromised, the attacker will have to also compromise auxiliary channels to gain knowledge of all required authenticators to authorise a fraudulent transaction
   - Users will not simply be asked to blindly enter a PIN delivered to their mobile phone. They will be asked to actively authenticate a specific transaction. Exact details of the transaction will be delivered to a chosen auxiliary mechanism, together with a one-time authenticator to be used for that specific transaction. The user then delivers the PIN via an auxiliary pre-registered channel to authenticate that particular transaction. Another code will also be included, which allows the user to lock the account. This would be used if the user received such a conformation message, while they are not actively using their account, which means that someone is breaking into their account.

Figure 1 illustrates the proposed architecture. The user will identify himself either by providing an alphanumeric identifier or by proffering a token. When the user registers with the system, he is given some options of potential extra channels and verifiers to be used in authenticating transactions with side-effects. Each channel can be personalised to choose the verifier the user wishes to make use of. There are basically two choices – either the user makes use of a previously known fact, such as an extra password or answer to a secret question, or the system delivers a random verifier to a particular channel, which the user then returns to the system in order to authenticate the transaction. Examples of this are one time passwords or PINs. The delivery and entry channels may well be different, but neither should be the primary interaction channel with the website since that might have been compromised.

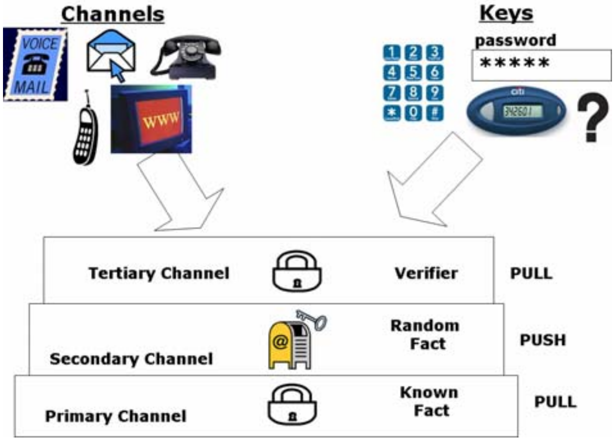

Figure 1: Multi-Channel Authentication

The organisation owning the resource will decide on which channels to offer to the users, both in terms of how easily they can integrate them into their system and how much they are willing to trust them. For example, few websites will trust biometrics delivered over the web because they are too easy to spoof. They will also decide on the minimum number of verifiers they require. Within these constraints, the user is given a number of options and will decide how many channels to make use of, and the kinds of transactions he wishes to be consulted about. For example, he may well decide that any transaction over £100 should be verified, or any transfer to a previously unused bank account, or any accounts that he has not personally registered with his bank.

An example will illustrate the mechanism:

- *Enrolling*: Janet goes to her bank and registers to use her website. Her bank makes use of a primary entry channel (the Web), secondary delivery channel (Email) and tertiary entry channel (SMS). She is issued with a user name and password, which she will use to check the balance of her current account at any time. She also registers her email address and her mobile phone number with the bank. She is told that she will receive an email to confirm any transfer together with a one time password, which she must text to the bank using her mobile phone in order to validate each transaction which transfers money out of her account.
- *Authentication:* Janet logs into her online bank website with her username and password. She views her account and then decides to pay her utility bill. She requests the payment and is told to follow emailed instructions. She receives an email from the bank with full details of the requested transaction. A telephone number and a PIN are provided for her to authorise the transaction. She is requested to send the PIN via SMS to the given telephone number. On receipt of the PIN, the bank authorises the transfer and the payment goes ahead.

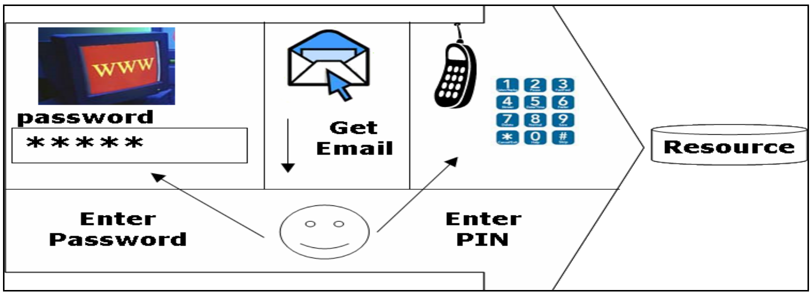

Figure 2: Authentication Process

We are aware of the fact that this is a fairly heavy-weight authentication mechanism, and that this kind of mechanism is only warranted for financial websites where the side-effects of fraudulent activity are expensive and potentially difficult to recover from. This mechanism also relies on the user being able to register in person at one of the bank's branches in order to register the mobile phone and obtain a user identifier and password. Banks are currently issuing pieces of hardware to produce one-time passwords at some cost to themselves (Levene, 2007), which suggests that they are prepared to spend money to deal with security issues. This scheme can cost less than an extra piece of hardware if email is used as a delivery mechanism, and it is more effective because of its use of independent channels.

The scheme does rely on the user being able to use his or her mobile phone to confirm each transaction, and this may well cause difficulties if the network is busy or unavailable. However, the user has the choice of using an alternative mechanism to confirm the transaction, perhaps by means of a telephone call, so that the problem can be bypassed.the text here.

2.1 A Support Environment for Multi-Device Authentication

The architecture to support multi-channel multi-factor authentication is proposed in Figure 3. Customer and session data is held in a central server, which supplies the Web server with information, and which probably executes on a separate machine in order to ensure that the database is not compromised by nefarious activity on the web server. There are two modes of interaction with the customer:

Pull mode: The customer initiates an interaction with the web server and accesses his or her account. The customer is authorised by means of a password or PIN via the same channel.

Push mode: When the customer wishes to carry out a transaction with side effects, an auxiliary authorisation is required, and this will be achieved by pushing a random authentication key to a secondary channel. This will be done in an asynchronous fashion, so that the web server is not delayed by slow service providers, for example. The user then submits this key via a tertiary channel in order to authorise each transaction.

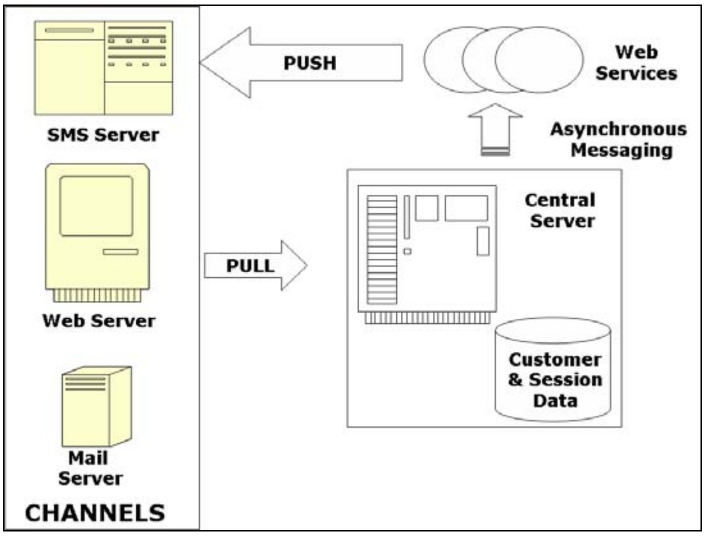

Figure 3: Support Architecture

Different servers handle different kinds of channel inputs both for the initial known fact and to receive confirmation of receipt of the random fact by means of the tertiary channel. The random keys are dispatched to secondary channels by means of web services which can act asynchronously so as not to delay the central server if there are delays with the transmission of the messages to the user's registered delivery channel.

Different kinds of channels have differing measures of compromisability. For example:

1. The Web browser is very easily compromised. We have referred to two commonly occurring attacks on the Web channel: phishing and Trojan software.
2. A landline phone is probably impervious to attacks, because it is fairly inaccessible and does not have embedded software.
3. Mobile phones are possible to attack because they can be stolen or cloned. It is possible, too, to write software to attack phones, but there is a much wider range of operating system software on mobiles and the attacker therefore has a limited chance of succeeding.
4. The Nintendo Wii is currently almost impossible to compromise as far as client software goes. It has no entry channels apart from a DVD reader, which only reads Nintendo-certified DVDs and USB ports are disabled. Furthermore, only Nintendo-certified software can be downloaded onto the device from the Nintendo website. However, the Wii could still be used in a Phishing attack.

If a compromisable channel is used as the primary channel, the system needs to utilise the multi-channel functionality provided to address the weaknesses of the channel. If the channel is perceived to be reliable, such as is the case with a landline phone, authorisation authentication could well be carried out using the same channel.

2.2 Evaluation

Since the concept of "quality" with respect to architectures is a fluid concept, Kazman *et al*, (1994) argue that the quality of a particular architecture can only be assessed on how it measures up based on its behaviour and use in a particular operational context. The way they use to represent this context is by using scenarios, which are intended to ensure that the system supports the activities the system was designed for and the kinds of changes that can be anticipated over time (Kazman *et al*., 2002). To evaluate our architecture, we use the following scenarios:

- *Phishing* – Janet clicks on a bogus link within a SPAM email message, which sends her to a fake website. This website is relaying all messages to the bank's website and displaying responses so that Janet has no idea she is using a fake website. She requests a transfer to her utility company to pay her bill. The fake website sends this information to the bank, but changes the transaction details to transfer money to a fraudulent account. The bank receives the request and sends an email to Janet to request authorisation for the transfer. Janet will immediately become aware of the fact that her transactions are being altered and realise that she is using a fake website. She can then contact her bank and change her username and/or password details. The incident ends with no money being transferred out of her account since she has not authorised a transfer. This relies on Janet's vigilance but since she is the one affected by fraudulent activity, and she has the transaction information to hand, it is hoped that she will be able to spot changes to her transaction.

- *Trojan software* – Alice connects to her website and requests a payment to her utility company. The Trojan software sends a fraudulent transaction to the bank using her session details. The bank now sends two emails to Janet, one for the genuine transaction, and one for the fraudulent transaction. She will immediately spot the extra transaction and can simply not authorise it. Suppose, however, that the Trojan software is able to intercept her email, obtain the PIN, and text the PIN to the bank. This will not authorise the transaction since the SMS must come from the registered mobile number. Hence the person who installs the Trojan software will also have to gain access to Janet's mobile phone, which is much harder than installing secret software over the Web.
- *Extensibility*: The Nintendo Wii game console has become very popular and users wish to use this as one of their registered channels. The architecture can easily accommodate this by means of the inclusion of functionality to exchange messages with Nintendo Wii's either on a new server or on an existing server. A new web service can be written to send random confirmation messages to these devices.
- *Schedule Implications*: A user logs into her account, carries out a transaction, and waits for the bank to send the random PIN to her secondary channel, her mobile phone. The mobile phone network is down, and the user cannot receive her random number, and cannot authorise the transaction. This is an undeniable problem, not with our proposed architecture, but with the external systems upon which it relies in order to achieve its functions. Of course even Web usage is dependent on the user's Internet Service Provider providing an uninterrupted service but our architecture adds extra dependencies and this could fail. The bank has two options: one is to ask the user to register extra devices which can be used if a secondary or tertiary channel fails. The other is to email the user a tailored link, which will send the user to a Web address to confirm the transaction. This is not as secure as the previous option because Trojan software could easily change the link but is a work around that could be used if the user agrees to this at registration time.
- *Brute Force*: If someone manages to break into an account, the user may well receive an authorisation request via another channel. This request will include two PINs: one to authorise and another to lock the account. The user can then, by sending the lockout PIN, immediate deactivate the account, and foil the hacker's attempts.
- *Emergency Authentication*: Brainard *et al.* (2006) propose a new kind of authentication based on somebody you know. The idea is that the legitimate user designates a number of people who may enter his account on his behalf. If the user then needs to authenticate, but cannot because the password has been forgotten, for example, he will ask one of these designated people for assistance. The helper authenticates as usual and obtains a special vouch code, which she delivers to the user, who uses it to get into his account. A system using multi-channel authentication could implement such a scheme to cover the primary authentication step. To deal with the eventuality that the "user" asking the designated person is actually a hacker, we could deliver an authorisation message to an secondary channel, asking the user to confirm the request for the designated person to obtain a vouch code and a hacking attack can be averted.

This list of scenarios is not exhaustive, but does address the most likely scenarios that need to be catered for, and which our architecture deals with easily. The most obvious hurdle to be overcome is user acceptance of the mechanism, and its attendant demands on their time and attention. We will carry out acceptance tests once the system has been completely implemented.

3. CONCLUSION

In this paper we have proposed an architecture for strengthening authentication for secure financial systems. Our architecture is able to utilise extra channels in order to require confirmation of all transactions with side effects. It has the potential to render phishing attacks and Trojan software unfruitful. The cost to the banking institutions will be minimal as compared to current two-factor authentication proposals, and the benefits will be a reduction in identity theft. We are currently in the process of carrying out end-user evaluations of this architecture.

REFERENCES

Adams, A and Sasse, M A, 1999. Users are not the enemy: Why users compromise security mechanisms and how to take remedial measures. *Communications of the ACM*, 42(12), pp. 40-46, December.

Besnard, B and Arief, B, 2004. Computer security impaired by legitimate users. *Computers & Security*, 23 (3), pp. 253-264, May.

Brainard, J, Juels, Ari, Rivest, R L, Szydlo, M, and Yung, M. 2006. Privacy and authentication: Fourth-factor authentication: somebody you know. October 2006. *Proceedings of the 13th ACM conference on Computer and communications security CCS '06*

Furnell, S and Clarke, N. 2005. Biometrics: no silver bullets. *Computer Fraud and Security*. August. pp 9-14

Gaw, S and Felten, E W. 2006. Password Management Strategies For Online Accounts. In *Proceedings of the 2006 Symposium On Usable Privacy and Security*, 12-14 July, Pittsburgh, PA.

Ionita, M T, Hammer, D K and Obbink, H, 2002. Scenario-Based Software Architecture Evaluation Methods: An Overview, *Workshop on Methods and Techniques for Software Architecture Review and Assessment at the International Conference on Software Engineering*, Orlando, Florida, USA, May.

Ives, B, Walsh, K, and Schneider, H, 2004. The domino effect of password reuse. *Communications of the ACM*, 47(4), pp. 75-78, April.

Kazman, R, Abowd, G, Bass, L and Clements, P. 2002. Scenario-based analysis of software architecture *IEEE Software*. 13(6): 47-55. November

Kazman, R, Bass, L, Abowd, G and Webb, M. 1994. SAAM: A Method for Analyzing the Properties of Software Architectures. *Proc 16th Int Conf on Software Engineering*, Sorrento, Italy. May, p81-90.

Levene, T. 2007. No phish with home-made chips. *Guardian Newspaper*. 12 May. (http://money.guardian.co.uk/saving/banks/story/0,,2077620,00.html)

Mizuno, M, Yamada, K and Takahashi, K. 2005. Authentication using multiple communication channels. *Proceedings of the 2005 workshop on Digital identity management.* Fairfax, VA, USA, Pages: 54 – 62. 11 November.

Morris, R and Thompson, K, 1979. Password Security: A Case History. *Communications of the ACM*, 22(11), pp. 594-597, November.

Roberts, C. 2007. Biometric attack vectors and defences. *Computers & Security* vol 26. pp 14-25

Sanders, T. 2006. Phishers crack two-factor authentication. http://www.vnunet.com/vnunet/news/2160250/phishers-crack-two-factor. 13 July.

Schneier, B 2005. The Failure of Two-Factor Authentication. March 15, 2005. (*CACM April 2005*) http://www.schneier.com/blog/archives/2005/03/the_failure_of.html. Accessed May 2007-05-14,

Richard Walton. 2005. Combining biometric measurements for security applications. *Computer Fraud and Security*. April pp 7-13