

A Process for Supporting Risk-Aware Web Authentication Mechanism Choice

Karen Renaud

Department of Computing Science, University of Glasgow

Abstract

Web authentication is often treated as a one-size-fits-all problem with ubiquitous use of the password. Indeed, authentication is seldom tailored to the needs of either the site or the target users. This paper does an in-depth analysis of all the vulnerabilities of authentication mechanisms, and proposes a structured and simple process which, if followed, will enable developers to choose a web authentication mechanism so that it matches the needs of their particular site.

1 Introduction

The Web offers unprecedented potential for reaching and enriching lives. Many e-commerce-based applications could, if they became popular, have a knock-on beneficial effect on gridlock, stress, pollution and global warming by reducing the need for consumers to be on the roads. It is far better for consumers to have their goods delivered than to drive to the supermarket to purchase them¹. E-commerce, although only 10 years old, has become an integral part of the national and international commerce infrastructure since more and more products and services are being marketed and traded via this route [1]. There are a number of complex reasons why many consumers have not yet started using e-commerce sites (the exception being book purchases[2]). One the one hand much attention has been paid to improving the usability of web sites and this attention has paid dividends, with many online resources offering valuable advice to developers². On the other hand, security problems are increasing [3,4]. E-commerce, since it involves an exchange of money, undoubtedly has an element of risk. It is very difficult for the average person to quantify risk in unfamiliar situations and the man in the street is more likely to be confused than informed by media hype. The end result is that he is probably

¹ http://adbusters.org/metasteco/truecosteconomics/true_cost.html

² <http://www.webcredible.co.uk/user-friendly-resources/web-usability/ecommerce-usability.shtml>

ill-informed about security issues [5]. The closest many people will come to any form of Web-based security is during authentication, and few users are sufficiently aware of the risks to behave securely during multiple authentication procedures.

The burden imposed by the authentication requirements of multiple systems is arguably one of the main remaining obstacles to wide-spread e-commerce use [6]. The most popular authentication mechanism is the password. These have been used for hundreds of years and are therefore well understood and widely accepted. The problem is not the password in itself, but rather the *abundance* of passwords. With any significant web usage and web-based purchasing, the user will have a significant number of different passwords to remember, each protecting a site that potentially holds sensitive details. With sublime naïvety many sites consider themselves islands, and ignore the fact that users will also purchase elsewhere, and have to shoulder the burden of multiple passwords, with the consequent unpleasantness of dealing with fraud if they fail to maintain good security by using different strong passwords for *all* web accounts. Schneier [7] claims that one in four victims of identity theft cannot regain their previous good credit standing, a statistic that is very worrying.

In order to address this state of affairs we need to widen our horizons somewhat so that we can consider using a variety of authentication mechanisms and not always reach for the faithful old password. Since this is relatively uncharted territory it might well be difficult for web developers to make an informed choice.

To support this process, therefore, we need first to understand the domain. This paper will perform an in-depth analysis of the weaknesses of the available web authentication mechanisms. This done, we should then have a structured way of incorporating the results of this analysis into the software development process. Van Wyk and McGraw [8] propose a tailored waterfall model, which includes security best practices, and this is clearly the best way to address security by incorporating it into all software development activities. This paper is focusing on the web authentication mechanism choice process. This is a fairly circumscribed problem — it doesn't impact on the rest of the website content or functionality. This paper will therefore propose that the authentication mechanism be chosen early in the software development life cycle, that the choice process considers the risks, target users and budgetary constraints, and that the process will produce some documents outlining the implementation and testing process as part of the normal implementation and testing stages during software development.

Section 2 explores the idea of risk, and how it impacts web sites. Section 3 introduces authentication — and discusses its features and weaknesses. Section 4 contrasts and compares various kinds of authentication mechanisms which can be used on the web. Section 5 explains risk in the context of the web and Section 6 proposes a way of quantifying the weaknesses of these authentication mechanisms to support ranking and informed choice. Section 7 proposes a risk-aware authentication

mechanism choice process. Section 8 concludes.

2 Risk

Risk is a complex topic and it would be folly to attempt to come up with a single definition when so many treatises on the subject exist. What *is* helpful is to gain an understanding of the most important facets of risk. One of the first issues to address is that of the *quantifiability* thereof. If risk *is* quantifiable then it is a simple matter to ascertain its magnitude and rank different risks. If not, we need to find other ways of drawing conclusions about risk severity.

Thompson and Dean [9] explain that there are different conceptions of risk, with the probabilistic at the one end and the contextual approach at the other. The probabilist considers risk to be essentially probabilistic and hence quantifiable. Contextualists consider probability to be merely one of risk's facets, but not an essential one, as the probabilists do. Starr and Whipple [10] strongly argue for a probabilistic approach. This approach is also followed by the Royal Society in its 1983 report, with their definition of risk [11]:

the probability that a particular adverse event occurs during a stated period of time, or results from a particular challenge.

Adams, on the other hand, [12] argues that whilst risk embodies the concepts of probability and magnitude with respect to danger, hazard, exposure to mischance or peril, it is, as a whole, *not* quantifiable. Plough and Krimsky [13] argue that risk cannot be considered in a purely probabilistic way but that it should be considered within context, including for example, cultural and experiential factors. They consider the quantification approach to be reductionist since it attempts to reduce a complex issue to a number. Finkel [14] also warns against viewing threats using only quantification techniques.

It therefore seems sensible to conclude that risk is a complex and multi-faceted concept that no mere quantity can adequately delineate, especially in the web context, which is so huge and with such unknowable attackers. There are two main concepts of risk that researchers of all persuasions *do* agree on, however:

Vulnerabilities — A weakness in the system that could be exploited by an attacker [15]. The word *hazard* is used in connection with risk by Beck [16] and the Royal Society [11]. Others refer to *threats* and *vulnerabilities* [17–20]. The American Chemical Society [21] refers to *source assessment*. Endorf [22] refers to *vulnerability exposure*. They all agree that these can be broadly thought of as situations that can lead to harm. For the purposes of this paper vulnerabilities can be considered to be weaknesses in authentication mechanisms.

Harm — The effects of the above-mentioned vulnerabilities are referred to by the Royal Society as *detriment*. Smith [23] and Kraemer *et al.* [24] refer to *outcomes*, Hertz and Thomas [25] talk about *outcomes* or *consequences* while Peltier [19] refers to *impact*. The American Chemical Society [21] refers to *effects*. Pfleeger [26] refers to *negative consequences*. For the purposes of this paper harm is the outcome of an intrusion into the website — the effect of a successful attack.

Thompson and Dean [9] warn against trying to come up with new risk definitions in order to accommodate both probabilistic and contextual approaches. Andrews and Whittaker [27] argue that security is essentially a trio made up of *assets*, *attackers* and *vulnerabilities*. The asset is to be protected from potential attackers, who will exploit vulnerabilities to attack. A definition that encompasses this trio is penned by Tiller [28] (p1064) as “*a measure of the loss of what you consider valuable, the impact of losing it, the threats to those assets, and how often the threats could be successful*”.

Our ultimate purpose is to *manage* the potential risks, which accords with Beck’s approach to risk, as “*a systematic way of dealing with hazards*” [16] (p21). Hence we need to come up with control mechanisms which can mitigate the vulnerabilities to reduce the opportunities open to an attacker. When we consider protecting any asset, in this case a website, we have to tailor the protection according to the vulnerabilities and the value of the asset.

What is therefore needed is a risk management approach to authentication, which can be incorporated into any software development life-cycle. In traditional software development this will probably be the requirements stage but in the agile community this will occur as part of the risk analysis process.

Experts propose different risk analysis methodologies [21,15,29]. While the methodologies differ, all encompass the following basic concepts :

- (1) *Identify assets and possible harm* — Section 5.1 outlines the possible impacts of intrusions and provides a framework for ranking impact on assets in a qualitative fashion.
- (2) *Pinpoint vulnerabilities* — Greene [30] points out that hazards are perceived differently by people based on previous experience, cultural values and previous training. Slovic [31] argues that risk is socially constructed so that risk assessment is subjective rather than objective.

Web development is undertaken by a wide variety of people with different levels of training, experience and cultural backgrounds. Hence in choosing and tuning these mechanisms they will do this according to their own perceptions. Unfortunately these perceptions may not accurately reflect the true vulnerabilities so that potential harm is not fully comprehended. Section 5.2 will thus present a framework for considering the web-specific hazards to be managed by an authentication mechanism.

- (3) *Consider the possible control options* — Endorf [22] proposes a scheme based on *return on security investment (ROSI)* that works out a dollar value based on the cost of a single loss, the annual rate of occurrence, and the cost of the security controls. This scheme weighs up the value of the asset against the security controls put into place to protect it.

This paper addresses web-based risks so the first problem we encounter is that of comparing these risks. Finkel [14] says that whilst comparing risks may not be impossible, it *is* difficult. There is a significant chance that one will end up comparing very different types of risks, eg: those that come from different sources, errors of commission; errors of omission; the same risk at different places in the system; risks with immediate feedback as opposed to risks with no feedback; but which are equally “risky”; and comparisons of risks with benefits. Endorf [22] bases his ROSI scheme on the assumption that we will be able to assign a dollar value to loss that derives from a vulnerability being exploited, and that we will be able to correctly predict the yearly occurrence of such an occurrence. In the face of increasingly powerful technology which can be used to break into systems and potential access by any number of hackers (worldwide) to your system, it is going to be difficult to come up with these figures.

Section 6 presents a mechanism which can be used to support decisions about how to limit and control authentication mechanism vulnerabilities. It is important to note that budgetary restrictions will play a large role in determining which control mechanisms will be implemented.

- (4) *Decide on the actions to be taken.* This will be discussed in Section 7 and an example illustrates the process in Section 7.1.

This paper presents an approach to web authentication that attempts to address the current situation so that sites are adequately protected, in the face of differing developer perceptions and understandings of risk. The following section discusses authentication principles in general and Section 4 considers web authentication in particular.

3 Authentication

Web applications provide access to information (the primary asset of websites) usually stored in a database, and supports specific functionality, which usually acts on that information. There are three core properties to be preserved [26]:

- *Confidentiality* — access must be restricted to legitimate users.
- *Integrity* — modification of information must be carried out only by authorised individuals.
- *Availability* — information should be available to legitimate users at all times.

We have standard mechanisms for enforcing these properties and foiling potential attackers: 1) identification followed by authentication, and 2) authorisation, which dictates what information and functionality the user is permitted to access. We will be discussing only the former mechanism in this paper.

Granting access to a digital space, then, involves a two-step process: identification followed by authentication. Identification, in the context of the web, will usually occur when the user offers an identity as a character string which is either her email address or a user name. In some cases a smart card may be used, but on the web this is rare. Authentication is the step which *verifies* the identification. Authentication has four distinct phases, with the user entering at the first, iterating within the second and possibly entering the third occasionally before returning to the second phase again. The fourth phase is seldom implemented or offered to users but in a world of increasing privacy concerns it is likely to become a requirement:

- (1) *Registration* — matching the user with a secret (the authentication key). The key can be issued by the system or provided by the user, with the latter being more common. In this phase the secret is paired with the person.
- (2) *Authentication* — the user is challenged by the system to provide the pre-agreed secret, which is compared to the stored secret. If they match, the user is granted access.
- (3) *Replacement* — this occurs if the user forgets the secret and needs to provide a new one.
- (4) *De-Registration* — this phase allows a user to request complete removal of his or her record from the web site owner's database.

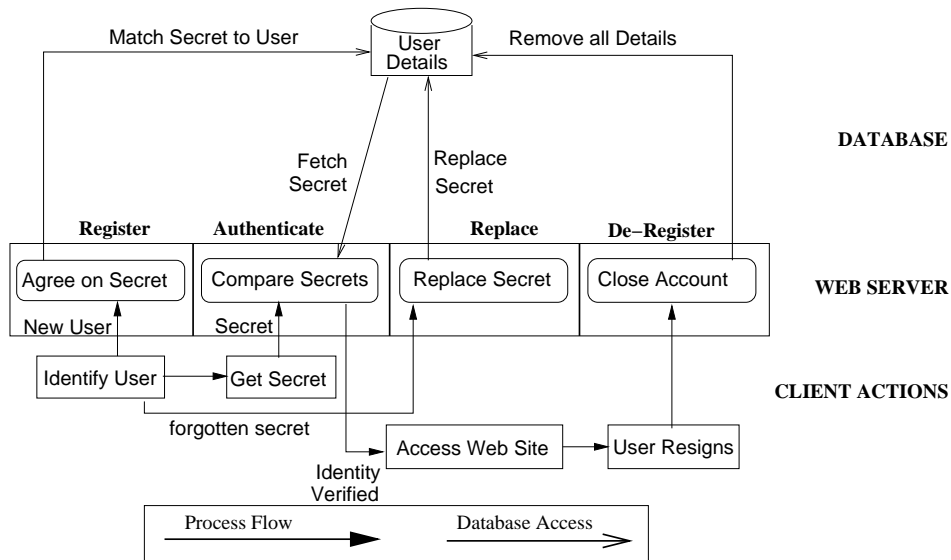


Fig. 1. *Authorising Users*

Authentication mechanisms traditionally have three types of weaknesses [32]:

- (1) *Guessability* — this is the traditional measure of strength of an authentication key: the size of the dictionary space. This is the probability that a person

guessing the key would succeed. So, for example, the guessability of a four digit PIN is 1 in 10 000 since there are 10 000 four digit numbers to choose from.

- (2) *Recordability* — this is the extent to which it is possible for a person to record his or her authentication key. The easier it is, the easier it is for someone else to obtain the key.
- (3) *Observability* — this is a measure of how easy it is to observe key entry and then to use that information to gain access to the system illegally.

There are two additional features to be considered specifically in terms of web-based authentication:

- (1) *Analysability* — this applies particularly to the software manning the mechanism. Sometimes the software has particular bugs which can be exploited and other times the authentication is a multi-stage process which offers valuable clues to the wily intruder. Any error messages produced by the system can also prove to be valuable clues to the potential intruder in helping to ascertain the operating system being used, web server software serving the web pages, database server and database login and password being used.
- (2) *Resistibility* — this is a measure of how resistible the mechanism is to penetration attempts, quite apart from the size of the dictionary. It refers to auxiliary attempts to secure the system. An example of this is the three-strikes-out policy that makes the guessability less important because the intruder only gets three attempts before the system becomes aware that a possible intrusion attempt is in progress. Only much later may the legitimate user become aware of the activity when he/she notices inconsistencies in the web account. Another way the system can resist attempts is by timing the authentication attempt and locking the person out if the authentication stage takes too long, which could indicate some kind of spurious activity, or if the inputs are given too quickly, which could indicate that the responses are being generated by a computer.

Obviously one wishes to increase resistibility and reduce the weaknesses inherent in recordability, observability, guessability and analysability.

A discussion of authentication cannot be concluded without considering the most important person in the process, the user. We need to understand what to expect our users to be comfortable with and able to handle in terms of authentication. It should be borne in mind that the web user is essentially operating in a buyers market — he or she has the power to switch to a competitor with very little effort. Thus a great deal of effort is required to ensure that your website is usable and that security procedures are acceptable to users. Hence we should determine the range of ages and abilities of our users. The information gathered typically falls into the following categories:

Expertise — how will users manage any expectation that they will install extra

hardware or software on their systems? How well will they deal with exceptions caused by their inability to correctly interact with the authentication mechanism?

Disabilities — What range of cognitive, mobility or sensory disabilities do we have to cater for?

Age — What is the range of ages we need to accommodate? Many older users have difficulties remembering nonsensical “secrets” and this could reduce their use of the system, or scare them off completely.

Attitude — What are the confidence levels of the users and their state of mind?

The discussion in this section has focused on generic authentication. The following section will discuss web authentication, a far more constrained area, characterised by users we cannot hope to train, accessing the website from across the globe, using unknown software executing on a variety of different kinds of hardware.

4 Web Authentication Mechanisms

The password, the web authentication mechanism of choice, relies on uncued exact recall, so it becomes increasingly difficult to keep track especially since we have so many of them. Some schemes have been proposed which can accommodate less than perfect recall [33] but these require special software, and are not widely used on the web. Left with this increasingly onerous demand on their memories, users either use the same password everywhere, a practice which is bound to cause a problem sooner or later with the increasing probability of hacker activity on user machines [34]; or by abandoning sites that require authentication.

Quite apart from unrealistic expectations the other problem with the password is its limited propensity for tailoring to match intrusion impact. Passwords are far too one-dimensional — the only way they can be tailored to the security needs of the site is by enforcing particular stringency requirements. This usually entails including special characters or digits. Paradoxically, this also makes the password weaker because the harder it is to remember the more likely it is that the user will write it down and in that case it can no longer be considered a secret.

It seems that the real problem is that one software model is used in all contexts, something the carpenter or engineer would scorn, and a practice that needs to be addressed in the web environment. There *are* alternatives to passwords. A less problematical authentication mechanism, in terms of memory load, is the physiological or behavioural biometric. Unfortunately the former has limitations related to secure biometric capture at enrolment in an uncontrolled environment such as the web, and privacy issues [35,36] and also due to the impossibility of replacement should it be leaked. Behavioural biometrics, such as mouse movement patterns or keyboard monitoring, have some potential but are often difficult to measure in a web environment without specialised software being installed on the user’s machine, something

that many Web administrators are reluctant to require of their users. Furthermore, it is almost impossible to rule out the possibility of interception or alteration of a recorded biometric being submitted over the web.

The biggest problem with these alternatives, as we have already seen from the problems with biometrics, is that the web environment is severely constrained. In the first place we can assume only minimal hardware and software in client machines and in the second place we cannot really trust anything recorded by special hardware or software located at the client. Thus we need to revert to the use of a secret.

Whereas the use of a “secret” in web authentication is the closest to viable web authentication mechanism we have at present, it need not necessarily be based only on text or numerals — it is quite possible to use some other memorable secret. In this section I will be considering the use of *images*, which people tend to remember better than words [37].

A number of image-based authentication mechanisms have emerged:

Cognometrics The users are issued with a set of images at enrolment and at authentication time they identify “their” image out of a group of distractor images. Example recognition-based systems are Passfaces [38], Déjà Vu [39] and the Visual Identification Protocol or VIP [40]. Whereas the previously mentioned systems were developed and evaluated by researchers there are also some commercially available cognometric systems. Pointsec is a commercially available system developed for use on a PDA (illustrated in Figure 2)³. The user chooses a set of images from the challenge set to authenticate. Lockscreen⁴ was also developed for PDA use, and has the same paradigm. Both use straightforward representational images.



Fig. 2. Cognometric Authentication

³ <http://www.pointsec.com/core/default.asp>

⁴ <http://www.learningtogo.com/ls/forppc.php>

Locimetrics The user has to identify particular locations within the image to be authenticated, illustrated in Figure 3. In 1996 Blonder patented a graphical password which required the user to touch predetermined areas of an image in a fixed sequence for authentication [41]. One evaluation has identified severe flaws in the location-based paradigm, at least for the particular implementation of the loci-mechanism called Jiminy, which was evaluated in the study [42]. The main problems with the mechanism appear to relate to the predictability of choices which in turn is caused by the severely limited number of possible positions the user can choose in authenticating him or herself.



Fig. 3. *Locimetric Authentication*

Drawmetrics Drawmetric systems require the user to draw a previously-drawn image at authentication time, as illustrated in Figure 4. One example of this is the draw-a-secret system [43]. Unfortunately participants in an evaluation of this scheme were not able to reproduce the picture accurately enough. Furthermore, a recent study by Thorpe and van Oorschot [44] has demonstrated that the potentially unlimited dictionary for this kind of mechanism is reduced by users' tendency to draw symmetrical images. This kind of system fails completely for users with tremors or other deformities of their hands that make writing difficult.

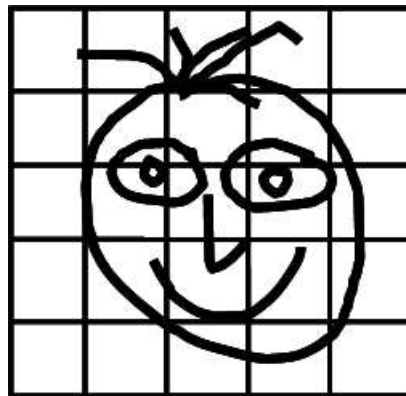


Fig. 4. *Drawmetric Authentication*

Visuo-biometrics The Handwing system [45], illustrated in Figure 5, provides a fairly robust mechanism in terms of memorability but it does sacrifice security and can only really be used for low to medium-risk systems. Users fill in a form at registration which records their written numerals, written postal code and a

line sketch called a doodle. At authentication they have to identify each part of their authentication key from a number of distractors, using the recognition of their own handwriting as a cue. Long-term trials with an older user group have delivered encouraging results with users gaining access to the system with very few exceptions (1%) over a 2 year period.

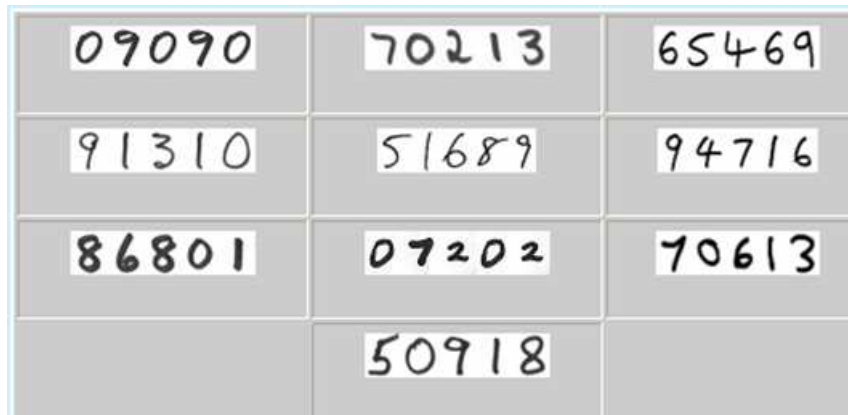


Fig. 5. Visuo-biometric Authentication

Manipuometrics Another kind of mechanism, which is gaining popularity due to its potential to foil shoulder surfing and key-logging software, relies on the use of arrow keys or the mouse only to move images around to line them up. Most of the manipuometric mechanisms have some redundancy so that the observer is not able to deduce the key from casual observation but has to either observe the user entering the key many times or carry out an error-prone deduction of the key based on a few observations. The v.Crypt system from Bharosa⁵, illustrated in Figure 6, requires the user to use arrow keys to line up a shape on the bottom row with an alphanumeric key on the top row. This is done for as many letters and numbers there are in the key. The redundancy introduced by the multiple images makes shoulder surfing far less likely to be productive.



Fig. 6. Maniupuometric Authentication

The following section discusses risk in the context of web authentication.

5 Risk and Web Authentication

Risk, in the context of web authentication, based on the previous discussion, can be considered to be:

⁵ <http://www.bharosa.com>

the possibility of an intrusion, and consequent harm, to a website, by means of exploitation of authentication mechanism vulnerabilities.

Web-based risk can now be discussed in terms of the concepts of *harm* and *vulnerability* introduced in Section 2.

5.1 Harm

O’Gorman [46] argues that it is better to estimate the impact of an attack and then implement security so that the risk of such an attack is reduced to an acceptable level of probability. We should therefore assess the impact of an intrusion and the higher the impact, the smaller the window of opportunity must be in order to offset the potentially harmful effects of the intrusion.

The first step in this analysis is a consideration of the web site content. Websites can hold the following kinds of data [47]:

- (1) *Sensitive* — the integrity of this information is paramount, such as financial transactions.
- (2) *Confidential* — personal information such as hospital patient records.
- (3) *Private* — information intended for use within a certain setting, such as school or church records.
- (4) *Public* — any information that cannot be classified as any of the above. For this kind of information no authentication mechanism would be required.

Any website that offers access to confidential or sensitive information should be considered to be a very high risk web site because of the potential impact of this information being leaked, which might well lead to litigation. Other impact factors to be considered are [48]:

- (1) the cost of recreating the information should an intruder modify or delete it.
- (2) the value of the information to the web site owners. The liability to the organisation be if it leaked out.
- (3) the impact if the information was unobtainable to legitimate users.
- (4) the exclusivity of the information: consider how much someone might be prepared to pay for it.
- (5) the impact on the organisation if the website were unavailable. This could range from a lack of confidence and possible abandonment by site users to minor inconvenience at the lowest end of the scale.
- (6) the cost both to the company and to the legitimate user should an intruder perpetrate a scam. This could include a loss of trust in the particular customer and also any other customers who hear about the problem. The company may need to reimburse the customer if they have been remiss in their security policies or procedures, to prevent such negative effects.

An attempt to quantify impact accurately is fraught with difficulty since many of the measures are subjective. The final decision will have to be based on the individual organisation's level of risk tolerance and the value *they* place on the asset. Hence, for the purposes of risk analysis and to avoid the subjectivity that could be encompassed in a quantification scheme, a categorisation scheme will be followed and impact will be classified by the system developer, in consultation with the customer, as low, medium, high or very high.

5.2 Vulnerabilities

Consider that the information controlled by the web site is to be secured by holding it within a *virtual* safe. Blaze [49] presents an interesting perspective to digital security by explaining how traditional metal safes are constructed and how they can be breached. This analogy serves well in considering the vulnerabilities of web sites and web authentication vulnerabilities will be considered in this context throughout this section.

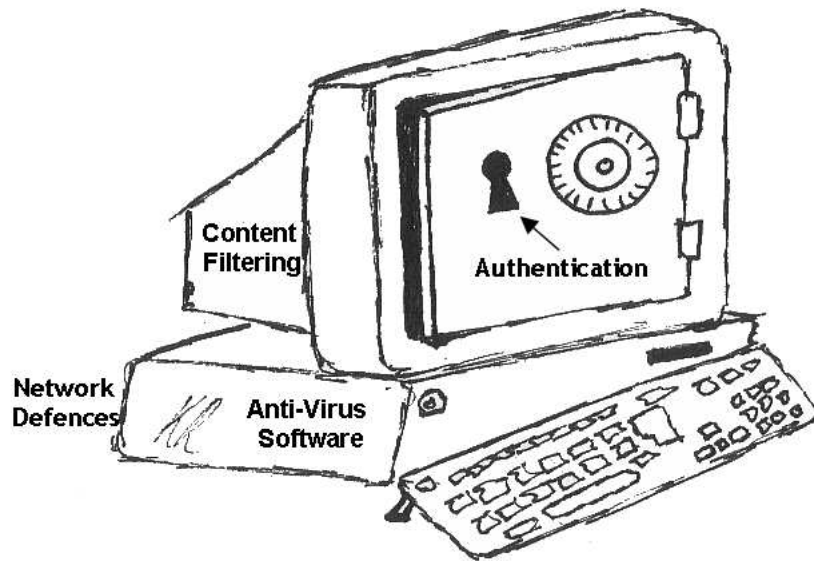


Fig. 7. *Virtual Safe*

Boundaries — The walls of the virtual safe are constructed using firewalls and network defences installed to prevent people gaining access to the system in any way other than the legitimate way — via the “lock”: the web server authentication mechanism.

Lock/Authentication Vulnerabilities — A *lock* needs to be constructed in such a way as to withstand penetration attempts and only to permit the legitimate user, who knows the key, to gain entry to the web site. There are basically two ways for an intruder to get in: the first is to obtain the key, and the second is to get past the “lock” in some other way.

The user is usually required to remember a secret sequence of numbers or letters without any cues. Since this is often difficult, and the consequences of forgotten keys unpleasant, people tend to write down these valuable keys. Table 1 enumerates the weaknesses which permit intruders to obtain the key.

Web Site	Weakness
Writing down authentication key	Recordability
Telling someone the authentication key	Recordability
Observation of Key entry (Shoulder Surfing)	Observability

Table 1

User-Related Vulnerabilities

Social engineering is a distinct threat in the digital world. Most organisations have centralised system support departments with workers who are not personally known to the employees and who appear to have the right to request authentication keys. It is hard for an employee to refuse such a request. Furthermore, as the computer user's boundaries stretch to encompass the entire Web so do the numbers of people who can request authentication keys, apparently legitimately. It is extremely difficult for a computer user to distinguish between a genuine request and a "phishing"⁶ attempt.

The next way an intruder can get in is simply by trying many combinations of the key until one succeeds. There is a difference between the potential strength of an authentication mechanism and the strength in reality. For example, passwords may be drawn from an infinite number of possible alphanumeric character strings, but in reality they will be drawn from a much smaller subset made up of meaningful words or numbers. Table 2 contrasts the weaknesses which permit such activities in both worlds:

Web Site	Weakness
Dictionary Size	Theoretical Guessability
Effective dictionary size	Realistic Guessability

Table 2

Combination Space Vulnerabilities (Exhaustive Search)

The intruder, having failed to gain entry in these ways, may now resort to other means of gaining entry. Some examples of vulnerabilities are:

⁶ The practice of sending a person an email containing a link pretending to link to the legitimate website which in reality links to another website which simply harvests user names and passwords to be used to access the genuine site.

- *Bypassing the Authentication Mechanism* — there are a number of ways of doing this:
 - if an attacker is able to gain access to the underlying infrastructure such that he can create an account for himself on the system the authentication step will no longer be a barrier to the system.
 - sometimes the attacker can hijack another user's session and take over another user's account in this way. A technique that can be very successful in the digital arena is the Trojan horse trick. The attacker installs a special piece of software on the target's computer that watches activity and records user names and passwords as the user enters websites. These details are sent to the attacker's computer to be used to gain access to the system.
- *Exploiting Known Software Bugs* — examples are:
 - *Buffer overflows*: some web application components don't operate correctly over buffers and this, if exploited skillfully, allows a hacker to gain control over the authentication process.
 - *Injection Flaws*: web applications pass their parameters in a variety of ways and if it is not done securely an intruder can embed malicious commands in the URL.
- *Obtaining authentication source code or database* — Insecure storage of vital application code or databases can be a gift to a hacker. Cryptographic code is often hard to use correctly. If used incorrectly it can open doors to informed intruders.
- *Analyse user/browser interaction* — This refers to attempts made by an intruder to determine the operating system, web server software type and protocols used in order to exploit known weaknesses in the software. Examples of this kind of vulnerability are:
 - *Unvalidated input*: inputs with embedded apostrophes can, if not caught by the underlying system be used to fool the system into allowing an intruder into the system.
 - *Broken access control*: systems have various categories of users, with different levels of access. If this is enforced incorrectly users can gain access to information they should not access.
 - *Improper Error handling*: error messages containing information intended to support analysis of the problem rather than re-orientation of the client, can be very useful to an intruder.

Table 3 rates each of these in terms of weaknesses.

Exploiting authentication mechanism weaknesses makes it easier to gain access to the system illegally. The next section will consider the evidence that such an attempt leaves behind.

Evidence of Penetration Attempts A pertinent issue which is relevant to this discussion is the detectability of attacks which contributes to the resistibility of the mechanism. They may be *surreptitious* if they leave no evidence, *covert* if they

Web Site	Weakness
Bypass Authentication	Low Resistance
Exploiting known software bugs	Analysability
Obtaining authentication source code or database	Increased Guessability
Analyse user/browser interaction	Analysability

Table 3

Application-Related Vulnerabilities

leave hard-to-notice evidence, or *forced* if they leave obvious evidence. We need to ensure that web-based attacks always leave evidence that can be used to detect illegal activity. Indeed it is much easier to obtain such evidence in the virtual world since everything is logged.

Surreptitious attacks on web sites, especially if they do not succeed, may not leave any evidence unless the website displays a list of previous login attempts (both successful and unsuccessful) when the legitimate user logs in successfully. Covert attacks, especially if successful, may be evidenced by the same display but there is no guarantee that the user will notice the illegal attempts. However, if a covert attempt succeeds, the impostor may merely change some settings such as the user's address, or obtain some personal details which may be used elsewhere. A forced attack will become obvious to the user when fraudulent transactions become evident or the user is locked out of her own account. In all cases the system log may deliver evidence of attacks.

The biggest problem in uncovering penetration attempts is the sheer mass of information in log files. Kevin Mitnick [50] relates stories of successful penetration attempts and points out that *all* the hackers' activities were recorded in the logs but that they were not noticed due to the volume of recorded information and the fact that system administrators are usually overloaded and unable to spend the time trawling through the logs. This problem is not going to get resolved in a hurry and since many hackers manage to gain access to accounts via weak passwords attention paid in that area will pay large dividends. Auditing, whilst tiresome and often unrewarding, does make an essential contribution to the overall strength of the authentication mechanism by augmenting the resistibility of the mechanism.

The same four key weaknesses mentioned in Section 3 emerge: *guessability*, *observability*, *recordability* and *analysability*. The extent to which these weaknesses can be buttressed will determine how hard it will be for an intruder to penetrate the system. We have a valuable tool at our disposal — increasing resistibility.

The security officer has many tools at his/her disposal to secure the boundaries of the website. Andrews and Whittaker [27] enumerate three general areas that can be attacked in a system: *services*, *applications* and *user actions*. Service vulnerabilities are well understood and there is a vast armoury of tools at our disposal to deal

with these. Application vulnerabilities were described in Table 3. The user-related vulnerabilities are far harder to control than either of the others. The authentication of users is the one place where the administrator exercises limited control. The technical strength of the authentication mechanism comes to nought in the face of insecure human behaviour. Users can behave insecurely simply because the security requirements are unrealistic or because they have been fooled by someone carrying out a social engineering attack. Hence one has to consider the users' needs in improving the security of an authentication mechanism.

6 Controlling Vulnerabilities

As stated before, we have no idea of any website's actual exposure to any of the vulnerabilities mentioned in Section 5.2. While acknowledging this, what we *can* do is to narrow the window of opportunity that any potential attacker can use to breach the defences of the authentication mechanism. Based on the discussion in Sections 3 and 5, opportunity can be defined as:

$$Opportunity = \frac{f(Guessability, Observability, Recordability, Analysability)}{Resistibility}$$

Since opportunity needs to be compared and contrasted it is as well to use a quantification mechanism to come up with a single number depicting the opportunity level. In order to do this we will quantify the four identified weaknesses and one strength. This section proposes one possible opportunity quantification mechanism⁷ which could be used to support decision-making:

- (1) *Guessability* — this is the traditional measure of strength of an authentication key: the size of the dictionary space. The guessability of a four digit PIN is 1 in 10 000 since there are 10 000 four digit numbers to choose from hence any key that is as strong or stronger than this is assigned a 0. Weaker keys will be assigned a proportionally higher guessability figure.
- (2) *Recordability* — As regards recordability, the systems can be assigned values as follows:
 - (a) 1 if the code is easily recorded
 - (b) 0.5 if it was harder to record or describe, or if recording of the key does not provide an observer with the full key
 - (c) 0 if it is difficult or impossible to record or describe, such as, for example, a biometric

It should be noted that recordability is an extremely difficult weakness to counteract, mainly due to inbuilt operating system features such as the “Print Screen” button and browser print functionality, which allows the user to print

⁷ In [32] we used a similar mechanism, which quantified the *security* of graphical authentication mechanisms. Hence the numbers assigned there are inverted.

the authentication screen and mark off the required images to offset memory lapses.

- (3) *Observability* — Observation of the code involves two equally important features:
 - (a) being able to actually see the key on the screen, and to use it — we assign 0.5 if the key can be used if observed only once to obtain the full key; 0.25 if key entry needs to be observed multiple times to obtain the key and 0 if the key cannot be observed.
 - (b) being able to judge the position of the key based on where the person is pointing at the screen or on the keyboard — we assign 0.5 if observation of the key location is meaningful and 0 if not.
- (4) *Analysability* —
 - (a) *User names* — users should be assigned user names rather than email addresses because email addresses are too easily obtainable and make it easier for hackers to gain access to the system. A 1 is assigned if the system uses email addresses as user names and a 0 if unique user names are used and not visible to other users.
 - (b) *Error messages* — error messages need to be provided on various levels. The developer obviously needs a different kind of error message that will enable him/her to analyse problems with the web site. Once the site is deployed, however, the messages should become targeted at the needs of the user, and no longer inform as to actual failure codes or database errors, but rather in terms of actions the user needs to take to recover. This limits the usefulness of error messages to the potential intruder. Hence a 0 is assigned if error messages have been tailored in this way and 1 otherwise.
 - (c) *Default Keys* — this particularly bad practice earns a rating of 1 because many users will not redefine their key or an intruder can take advantage of the default setting before the user logs in for the first time.
 - (d) *Forced changes* — the reasoning behind this is that a leaked authentication key will only be useful to an intruder for a limited period of time. However, routine forced renewals actually decrease guessability since users need to come up with new passwords every time it is renewed and they eventually start choosing easy-to-remember passwords. Hence an application with this policy earns a 1.

A much better way of dealing with the effects of others using a stolen password is by displaying, in very visible and attention-grabbing format, the last login time and duration every time the legitimate user logs into the system. Also allow only one session per user and inform the currently logged-in user of other attempts to log in. In this way the user participates in the attack detection and the attacker's activities are more likely to be noticed.

- (e) *Key retrieval* — Forgotten keys should never be emailed. This is simply too easy for an intruder to intercept. The current practice of asking the user to confirm the answer to a particular question reduces the authentication key space to a very small space indeed and one that can probably be

uncovered by a research based attack. A better mechanism is to reset the password and email the user a secure link, which requires the user to set a new password. For a secure site a more secure option may be required, such as, perhaps, sending an SMS message to the user and requiring her to confirm the request via SMS before the secure link is emailed. A policy that emails authentication keys or uses confirmation questions to confirm identity earns a 1 for analysability.

- (f) *Backward browsing* — intruders can often try to obtain information by using the back button, which is impossible to disable. Hence we will assign 0 only if the system ensures that authentication pages expire immediately they are processed. A weakness of 1 will be assigned otherwise.
 - (g) *Choice of distractor images* (if applicable) — some recognition-based authentication mechanisms rely on the user choosing one image from a group of distractor images. If distractor images are varied at each attempt it is a simple matter for the intruder to observe the interaction over an extended period of time to identify the target images, or to refresh the display repeatedly. It is more secure to fix distractors for a particular user and to use these repeatedly. This policy will be assigned a 0 and a policy of varying distractors is assigned a 1.
 - (h) *Choice of background image* (if applicable) — some mechanisms make use of a single large image, which needs to have particular characteristics: In this case the image needs to have many features which can be chosen by the user, but not too many, which could cause confusion. Hence we assign a 1 to an image with fewer than 10 identifiable features and 0 to an image with more than 1000 features. Numbers in between are assigned on a proportional basis.
- (5) *Resistibility* — We could quantify this as follows:
- (a) *Lockout policy* — 1 if there is a strikeout policy and 0 otherwise
 - (b) *Key Strength* — stronger keys are less prone to brute-force attacks so many systems enforce password policies that require passwords to have a specific length, a digit, upper and lower case letters and special characters. Unfortunately people cannot remember long and complicated strings and this increases the likelihood that it will be recorded. A much better way of strengthening a key is by using length rather than complication. For example, password users can be encouraged to write a whole sentence rather than a simple word. If a key complicating policy is applied assign a 0, if a key lengthening policy is applied assign 1.
 - (c) *Timeouts* — If an authentication takes too long it is likely that an intruder is trying to determine which the target images are by doing some kind of research. The legitimate user can be expected to home in on her images very quickly. Hence a time limit should be applied to the authentication step. We assign 1 if there is a policy and 0 otherwise
 - (d) *Auditing* — 0 if no regular auditing takes place, 1 if auditing takes place at weekly intervals and 2 if it occurs more often than that.

- (e) *Evidence* — 1 if previous login attempts are displayed to the user at login time; 2 if the user is apprised by email or SMS when someone logs into their account; 0 if no historical data is provided.
- (f) *Ease of change* — if you make it easy for users to change their authentication keys they are more likely to do so. Hence assign a 1 if this is easy to do, but only if they have to authenticate themselves before changing it.

A way of combining these assignments will now be proposed in order to control the vulnerabilities. Guessability, observability and recordability each have a maximum weakness of 1 whereas the weakness assigned to analysability can be as much as 7. This is because a highly analysable system will easily sabotage the best of authentication mechanisms. The maximum value of resistibility is 6, but only if the system has all the required resistance mechanisms. Thus we should try to drive the opportunity value down below 1, whereas it could easily be infinitely large which would indicate a very weak authentication system. The proposed formula for determining opportunity is:

$$Opportunity = \frac{Guessability+Observability+Recordability+Analysability}{Resistibility}$$

This formula does not attempt to give more or less weight to any of the first three weaknesses because no empirical evidence exists, at present, to justify such a weighting. It is entirely possible that time and experience will suggest a realistic weighting scheme.

The alternative mechanisms discussion in Section 4 can be rated in terms of observability, guessability and recordability, as shown in Table 6. An additional column considers disqualifying factors ie. disabilities or difficulties experienced by system users that would disqualify this mechanism from use.

	Observability	Guessability	Recordability	Disability Disqualification
Passwords	0.5	0	1	Memory/Dyslexia
Cognometrics	0.5	0.2	1	Vision
Locimetrics	1	1	1	Vision
Drawmetrics	1	0	1	Vision/Dispraxia /Tremors
Visuo-biometrics	0.5	0.16	1	Vision
Manipuometrics	0.25	0	1	Vision

Table 4
Alternative Authentication Mechanism Weaknesses

We need to counteract the inherent weaknesses of the chosen mechanism by augmenting the resistibility and reducing the analysability of the system *to the extent justified by the web site content*. Unfortunately, the stronger and more resistant the authentication mechanism, the higher the cost. One needs to balance the extent to which you will guard against various kinds of attacks depending on the kind of information you're protecting and the impact of an intrusion.

The following section will present one possible methodology to match risk to “lock” strength by proposing a risk-oriented process for web authentication mechanism development and incorporation into the software development life cycle.

7 Risk-Oriented Web Authentication Choice

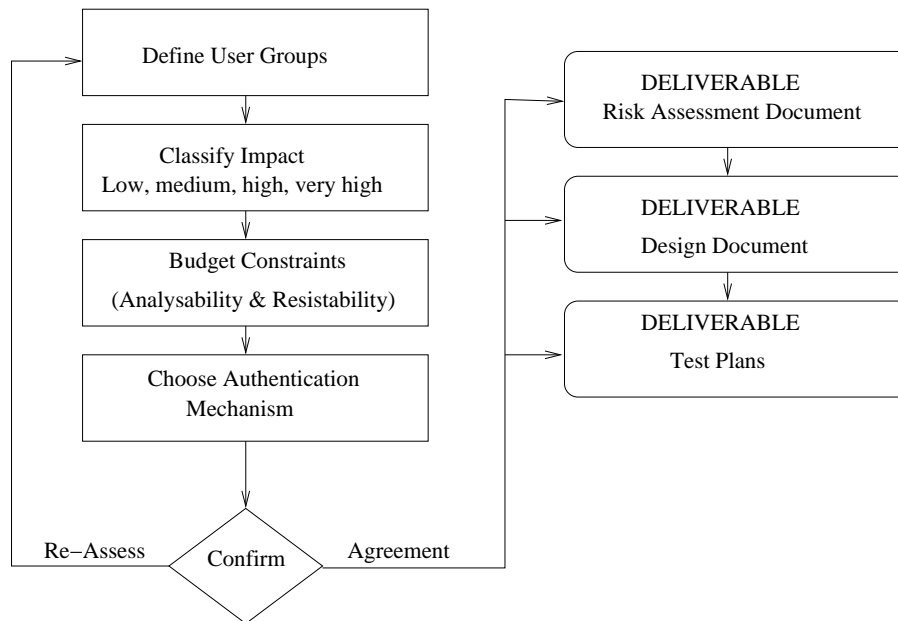


Fig. 8. Risk-Oriented Choice Process

To choose a *suitable* authentication mechanism, in terms of afore-mentioned vulnerability and impact, and for our users, we should progress through the following steps:

- (1) *Define User Groups* — Identify the characteristics of your target user group. This means determining accessibility factors such as possible disabilities which need to be considered. For example, if the target users are expected to be elderly one has to consider visual acuity and dexterity.
- (2) *Classify Impact* — Determine the possible impact of an intrusion.
- (3) *Decide on the Budget* — the amount assigned to preventing analysability and increasing resistibility. (Some of this cost is accrued during implementation, and the rest after deployment as part of maintenance activities.)

- (a) Specify the analysability features that need to be eliminated for the mechanism. For example, for cognometrics, make recommendations about the choice of distractor images.
- (b) Enumerate resistibility features you can afford to apply to offset mechanism weaknesses.
- (4) *Choose Authentication Mechanism* —
 - (a) Identify some candidate authentication mechanisms for your site which meet the target user needs. For each, do the following:
 - (i) Quantify the guessability, observability, and recordability.
 - (ii) Now work out the opportunity value for the mechanism.
 - (b) Rank the mechanisms from low to high opportunity.
 - (c) Choose an authentication mechanism with the opportunity level to match the impact.
- (5) *Confirm* — at this stage we will need to consult with all stake-holders and ensure that they are satisfied with the choice. Another iteration of the previous 4 steps may need to occur in order to gain acceptance and to proceed to the following step.
- (6) *Make Test Plans* — determine how the mechanism should be tested and feedback facilitated.

There are a number of deliverables from this process, which will feed into later stages of the software development process:

- *Risk Assessment Document* — list the qualitative assessment of the impact of possible intrusions. Also enumerate the possible hazards and the control mechanisms that should be implemented to protect against them.
- *Design Document* — specify the choice of mechanism, and include instructions about how this is to be implemented. This includes specification of authentication-related logging required in order to facilitate feedback about the efficacy and efficiency of the mechanism.
- *Test Plans* — these are usability and penetration tests that should be carried out before the site is deployed, in order to ensure that control mechanisms are working as anticipated.

7.1 Example

Assume we have a website which has been set up to provide a directory of special services available to a group of elderly users. The authentication choice procedure proceeds as follows:

- (1) *Define User Groups* — The target user group have short memory problems, impaired vision, possible tremors and arthritic hands.

- (2) *Classify Impact* — The site is classified as being *medium impact* since it does not hold any financial or other sensitive personal details.
- (3) *Decide on the Budget* — The budget for analysability and resistibility is determined. We can afford the resistance and analysability options shown in Table 5.

Analysability	
User names assigned	0
Error messages curtailed	0
No default keys and no forced changes	0
Key Retrieval by Email	1
Backward browsing restricted	0
Distractor images fixed at enrolment (Cognometrics)	0
Background image unrestricted (Locimetrics)	1
Analysability Total	1 (2 for locimetrics)
Resistibility	
3-tries lockout policy	1
Key strength not enforced	0
Timeouts will apply	1
No Auditing	0
No evidence of previous accesses will be provided	0
Change not facilitated	0
Resistibility Total	2

Table 5
Analysability and Resisitibility Options

- (4) *Choose Authentication Mechanism* — Opportunity is calculated by adding the analysability total given in Table 5 to the observability, guessability and recordability figures given in Table 6, and dividing by the resistibility total given in Table 5. The final opportunity quantification is given in Table 6.

Note that passwords and drawmetrics are disqualified by the needs of the user group and manipuometrics and locimetrics are disqualified by the needs of the website. so for our medium impact system we have a choice of two alternatives to passwords to match the impact rating ie. cognometrics or visuo-biometrics.

We now need to confirm this choice, including the decisions and classifications that led to the choice, with all stakeholders. Once the choice has been

made, we will need to specify details about the implementation of the mechanism for the design document.

- (5) *Make Test Plans* — Usability and Penetration tests will be specified to ensure that analysability and resistibility features of the mechanism are tested. Penetration tests should specifically test the analysability and resistibility features budgeted for.

	Opportunity	Mechanism Security Ranking	Suits Website Ranking	Suits Users
Manipuometrics	1.125	High	No	Yes
Passwords	1.25	Medium/High	Yes	No (Memory)
Visuo-biometrics	1.33	Medium	Yes	Yes
Cognometrics	1.35	Medium	Yes	Yes
Drawmetrics	1.5	Low	No	No (Tremors)
Locimetrics	2.5	Very Low	No	Yes

Table 6

Alternative Authentication Mechanism Opportunities

The developer will probably choose either cognometrics or visuo-biometrics for this website. This is an informed choice, based on the opportunities offered for intrusion, the classification of intrusion impact and the budgetary constraints. She will also document her decision process so that the implementation and testing of the mechanism are fully specified.

8 Conclusion

The password, despite its widespread use, is *not* the only way to authenticate web users. It is not even a particularly effective or efficient way. The ubiquitous use of the password on the web is unworthy of a discipline of growing maturity. Controlling access to websites is not the simple one-size-fits-all problem that this monotonous use of the password suggests it is.

This paper has proposed a risk-aware procedure for supporting choice and development of a web authentication mechanism which takes the asset value and impact of possible intrusions into account, and considers user needs. A quantification mechanism to support a choice between different authentication mechanism alternatives so that vulnerabilities are controlled was presented to support the process.

Acknowledgements

My thanks to Brad Glisson and Dora Galvez-Cruz for their extremely helpful comments on this paper.

References

- [1] S. Fox, J. Q. Anderson, L. Rainie, The future of the internet, Tech. rep., Pew Internet and American Life Project, http://www.pewinternet.org/pdfs/PIP_Future_of_Internet.pdf. Accessed May 2006 (2005).
- [2] E. Joyce, Amazon's profit jumps, but it eases outlook, *internetnews.com* (21 October 2004).
- [3] DTI, Information security factsheet, <http://www.dti.gov.uk/bestpractice/assets/security/intro-to-info.pdf> (2005).
- [4] Deloitte, Global security survey, www.ladlass.com/ice/archives/files/deloitte
- [5] S. Hills, Millions 'are wide open to online crime', *Metro* (28 October 2005).
- [6] W. Rash, Password chaos threatens e-commerce (15 February 2002).
- [7] B. Schneier, Cryptogram newsletter, <http://www.schneier.com> (September 2005).
- [8] K. R. van Wyk, G. McGraw, Bridging the gap between software development and information security, *IEEE Security & Privacy* 3 (5) (2005) 75–79.
- [9] P. B. Thompson, W. R. Dean, Competing conceptions of risk, *Risk: Health, Safety Environment* 7 (1996) 361–384.
- [10] C. Starr, C. Whipple, Risks of risk decisions, *Science* 208 (1980) 1114–1119.
- [11] Risk assessment: Report of a royal society study group, The Royal Society. London (1983).
- [12] J. Adams, Risk, University College London, 1995.
- [13] A. Plough, S. Krimsky, The emergence of risk communication studies: Social and political context, *Science, Technology & Human Values* 12 (3-4) (1987) 4–10.
- [14] A. M. Finkel, Comparing risks thoughtfully, *Risk* 7 (4) (1996) 325–59.
- [15] C. P. Pfüeger, Quality time: The fundamentals of information security, *IEEE Software* 14 (1) (1997) 15–16, 60.
- [16] U. Beck, Risk Society: Towards a New Modernity, Sage Publications, 1986.

- [17] C. J. Alberts, A. G. Behrens, R. D. Pethia, W. R. Wilson, Operationally critical threat, asset and vulnerability evaluation (octave) framework. version 1.0, Tech. Rep. CMU/SEI-99-TR-017, Carnegie Mellon University Software Engineering Institute, <http://www.sei.cmu.edu/pub/documents/99.reports/pdf/99tr017.pdf> (June 1999).
- [18] S. Vidalis, A. Jones, Threat agents: what infosec officers need to know, *The Mediterranean Journal of Computers and Networks* 1 (2) (2005) 97–110.
- [19] T. R. Peltier, *Information Security Risk Analysis*, Auerbach Publishers Inc., 2005, <http://www.peltierassociates.com/frap.htm>.
- [20] L. C. Briand, K. E. Emam, F. Bomarius, Cobra: A hybrid method for software COst estimation, Benchmarking, and Risk Assessment, in: *20th International Conference on Software Engineering (ICSE'98)*, 1998, p. 390, <http://www.riskworld.net/index.htm>.
- [21] A. C. Society, Understanding risk analysis. a short guide for health, safety, and environmental policy making, <http://www.rff.org/rff/Publications/loader.cfm?url=/commonspot/security%/getfile.cfm&PageID=14418> (1998).
- [22] C. F. Endorf, Measuring roi on security, in: H. F. Tipton, M. Krause (Eds.), *Information Security Management Handbook*, 5th Edition, Auerbach Publications, 2004, pp. 685–688.
- [23] N. J. Smith, *Managing Risk in Construction Projects*, Oxford, Blackwell Science, 1999.
- [24] H. C. Kraemer, A. Kazdin, D. Offord, R. C. Kessler, P. S. Jensen, D. J. Kupfer, Coming to terms with the terms of risk, *Archives of General Psychiatry* 54 (1997) 337–343.
- [25] D. B. Hertz, H. Thomas, *Practical Risk Analysis and Approach Through Case Histories*, John Wiley and Son, Chichester, UK, 1984.
- [26] C. P. Pfæger, S. L. Pfæger, *Security in computing*, 3rd Edition, Prentice Hall, Upple Saddle River NJ, 2003.
- [27] M. Andrews, J. A. Whittaker, Computer security, *IEEE Security and Privacy* 2 (5) (2004) 68–71.
- [28] J. S. Tiller, Outsourcing security, in: H. F. Tipton, M. Krause (Eds.), *Information Security Management Handbook*, 5th Edition, Auerbach Publications, 2004, pp. 1061–1072.
- [29] J. M. Carroll, *Computer Security*, Butterworth-Heinemann, 2004.
- [30] A. Greene, A process approach to project risk management, in: *Doctoral Research Workshop: Construction Process Research*, Loughborough University, 2000, pp. 14–25.
- [31] P. Slovic, Trust, emotion, sex, politics and science: Surveying the risk-assessment battlefield, *Risk Analysis* 19 (4) (1999) 689–701.

- [32] A. De_Angeli, L. Coventry, G. Johnson, K. Renaud, Is a picture really worth a thousand words? reflecting on the usability of graphical authentication systems, *International Journal of Human-Computer Studies: special issue: HCI research on Privacy and Security* 63 (1-2) (2005) 128–152.
- [33] C. Ellison, C. Hall, R. Milbert, B. Schneier, Protecting secret keys with personal entropy, *Future Generation Computer Systems* 16 (2000) 311–318.
- [34] B. Ives, K. R. Walsh, H. Schneider, The domino effect of password reuse, *Commun. ACM* 47 (4) (2004) 75–78.
- [35] C. Braghin, Biometric authentication, <http://citeseer.ist.psu.edu/436492.html>. Accessed 13 April 2005 (Nov. 02 2000).
- [36] H. Berghel, Identity theft, social security numbers, and the web, *CACM* 43 (2) (2000) 17–21.
- [37] S. Madigan, Picture memory, in: J. Yuille (Ed.), *Imagery, memory, and cognition: essays in honor of Allan Paivio*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1983, pp. 65–86.
- [38] S. Brostoff, A. Sasse, Are passfaces more usable than passwords? a field trial investigation, in: S. McDonald (Ed.), *People and Computers XIV - Usability or Else! Proceedings of HCI 2000*, Springer, 2000, pp. 405–424.
- [39] R. Dhamija, A. Perrig, Déjà vu: A user study using images for authentication, in: *Proceedings of USENIX Security Symposium*, Denver, Colorado, 2000, pp. 45–58.
- [40] A. De_Angeli, M. Coutts, L. Coventry, G. I. Johnson, VIP: a visual approach to user authentication, in: *Proceedings of the Working Conference on Advanced Visual Interfaces AVI. 2002*, ACM Press, 2002, pp. 316–323.
- [41] G. E. Blonder, Graphical password, united States Patent 5559961 (1996).
- [42] K. V. Renaud, A. De_Angeli, My password is here! Investigating authentication schemes based on visuo-spatial memory, *Interacting with Computers* 16 (6) (2004) 1017–1041.
- [43] I. Jermyn, A. Mayer, F. Monrose, M. K. Reoter, A. D. Rubin, The design and analysis of graphical passwords, in: *Proceedings of the 9th USENIX Security Symposium, 2000*, p. electronic proceedings, <http://www.usenix.org/publications/library/proceedings/sec2000/technical.html>.
- [44] J. Thorpe, P. van Oorschot, Graphical dictionaries and the memorable space of graphical passwords, in: *13th USENIX Security Symposium, 2004*, pp. 135–150.
- [45] K. Renaud, A visuo-biometric authentication mechanism for older users, in: *Proc British HCI 2005. Sept 5-9, Edinburgh, 2005*, pp. 167–182.
- [46] L. O’Gorman, Comparing passwords, tokens, and biometrics for user authentication, *Proceedings of the IEEE* 91 (12) (2003) 2019–2040.

- [47] IASEP, Data security protocol for education, Center for Information Assurance and Security and the Indiana Assessment System of Education Proficiencies. Purdue Research Foundation, http://iasep.soe.purdue.edu/Protocol/home_page.htm (September 2000).
- [48] J. C. Miller, Risk assessment for your web site, IRMI.com. International Risk Management Institute, <http://www.irmi.com/Expert/Articles/2000/Schoenfeld.aspx> (Sep 2000).
- [49] M. Blaze, Safecracking for the computer scientist, Tech. rep., U. Penn CIS Department (2004).
- [50] K. D. Mitnick, W. L. Simon, The Art of Intrusion, Hungry Minds Inc, Indianapolis, 2005.