Reconciling Usability and Security: Interaction Design Guidance and Practices for On-Line User Authentication

Michael Lang

Business Information Systems, J.E. Cairnes School of Business & Economics, NUI Galway

Michael.Lang@nuigalway.ie

Abstract. Usability and security are often portrayed as though they are competing priorities in information systems development. Given that both are essential to the design of an effective system, it is important that these two prerogatives should be reconciled. In recent years, there is growing concern with the rising incidence of on-line impersonation, theft and other types of fraud. It is therefore important that an information system must have a secure and rigorous way of authenticating a user's identity. This paper reviews the sources of literature on interactive design guidance for on-line user authentication, and then compares the actual practices of a purposefully selected sample of twelve Websites against the recommendations from the literature. Alarmingly, the findings of this study are that many Websites have user authentication processes which contain basic design flaws that are potentially open to exploitation by Internet criminals.

1. Introduction

Government statistics reveal that the US Department of Justice received over 330,000 complaints of Internet crime in 2009, up 22% on the previous year, causing estimated losses of US$560m as opposed to US$265m in 2008. Of the complaints which were subsequently referred to law enforcement agencies, most related to e-commerce fraud, with identity theft (14%) and credit card misuse (10%) being notably to the fore [10]. Comparable figures released by the Data Protection Commissioner in Ireland indicate a similar trend (see Figure 1), leading him to comment that "*respect for privacy is part of the network of trust that our society relies upon ... a collapse of public trust in data-dependent services organisations would be hugely damaging*" [4]. Data security has long been recognised as an important issue in systems analysis and design [2], but given the modern situation where e-commerce and e-government systems which store and process private personal data are plugged into an expansive global communications network, security has now become a critical factor in Web-based systems development. In order to combat fraud, it is necessary to be able to authenticate the identity of a user with a high degree of confidence. The essential challenge is how to design a system that is both usable and secure while also respecting users' privacy [17].

The joint prerogatives of information systems security and usability can sometimes appear to be in conflict with one another [3,26,29], with Zurko & Simon [31] going as far as to say that "secure systems have a particularly rich tradition of indifference to the user". The more security controls that a legitimate user is required to pass through in order to authenticate himself, the more likely that user is to attempt to circumvent those barriers in order to minimise the effort required to access the system, perhaps even subverting the integrity, security and effectiveness of the system [15,28,29]. For example, free Web sites such as www.fakenamegenerator.com and www.bugmenot.com are specifically intended to bypass the registration process of e-commerce systems by providing false details, the reason being that sites which require compulsory registration are perceived by many users as a nuisance and an invasion of privacy. There is thus a need to strike a balance the vital considerations of security, usability, privacy, and data integrity, and with this in mind Norman [17] has called for "*a set of standardized scripts, templates, and system tools that allows [the community] to implement best practices in ways that are both effective and efficient, standardizing interactions across systems in order to simplify the lives of users*". The motivation for this paper is to go some way towards answering that call. Information systems security is multi-layered (e.g. the OSI security architecture), but because the focus of this paper is on user interaction, we shall concern ourselves only with the design of the front-end of the system, concentrating on the basic tasks of registration and logging in, both of which are central to user authentication and fraud prevention.

The structure of this paper is as follows: section 2 reviews the literature on interaction design guidance for on-line user authentication; section 3 outlines the research approach; and section 4 discusses the findings of an analysis of the actual practices used by a selection of Web sites, comparing those practices against the recommendations in the literature. Only one previous study has conducted a similar analysis [7], with most of the published work in the area of user interaction design for on-line authentication being of a theoretical or prescriptive nature. This paper therefore contributes towards building a better understanding of the fit between textbook theory and industry practice.

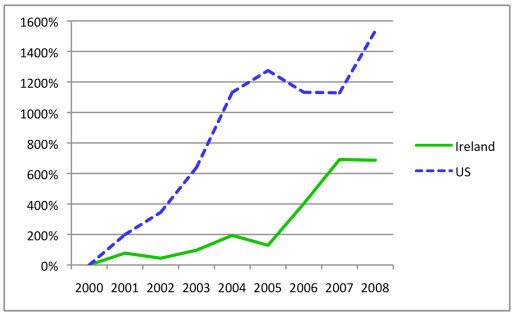

Figure 1. Growth in incidence of recorded Internet crime complaints since 2000. Sources: [4,10]

2. Literature Review and Commentary

Preece et al [18] define "interaction design" as the process of "*designing interactive products to support people in their everyday and working lives*". It is closely associated with the notions of "user experience design" and "information architecture", particularly within the context of Web-based systems development. Underpinning all of these is a central concern on usability, the essential aspects of which are: effectiveness, efficiency, utility, learnability, memorability, and safety. The practical discipline of interaction design is informed by a body of design guidance based on experience and theoretical knowledge accumulated within its own community and also inherited from cognate fields (e.g. industrial design). This body of design guidance encompasses knowledge in various formats, including design principles, design patterns, and use case scenarios.

Design principles are "generalizable abstractions intended to orient designers towards thinking about different aspects of their design" [18]. These general principles, such as "visibility" and "affordance", lead to what may be called design "heuristics" in practice, meaning the way a particular problem can be resolved (e.g. [29]). The notion of a "design pattern", first originated by Alexander [1] in the field of architectural design, refers to a way of formulating the description of a commonly recurring design problem as well as the core of the recommended solution to that problem. A design pattern typically comprises the following sections, though it need not include all of these: name, aliases, classification, motivation, problem statement, forces/rationale/intent, solution/sequence of events, participating patterns/classes, examples/known uses, consequences, and related patterns.

Design patterns have been around for some time within the field of software engineering (e.g. the PLoP conference series) and are now also popular within the field of human-computer interaction (see for example UI-patterns.com). For Web-based applications development, most of the patterns in the literature are primarily concerned with hypermedia navigation, various aspects of interface design, software engineering, or programming (e.g. [6,9,13,15]), but there also exists a number of valuable contributions which describe interaction design patterns for on-line user authentication [16,22,24,30].

Similar in a number of regards to a design pattern is the notion of a "use case". This is a requirements specification technique commonly used in user-centred design approaches and is a normal feature of UML-driven development approaches. A use case scenario is a description of a particular task that an actor is required to perform, generally written up in a standard format which includes: use case name, participating actors, triggers, pre-conditions (if any), flow of events (including exception handling), post-conditions (if any), and relationships with other use cases (e.g. "includes", "extends"). For standard functionality that is similar across most systems, use case scenarios can provide a reusable source of design guidance. Of particular interest is the idea of "misuse" or "abuse" cases, where the malicious behaviour of a fraudster is anticipated when designing the system [14,20].

The sections which follow summarise the various sources of interaction design guidance found in the literature on the two basic tasks associated with on-line user authentication: user registration, and logging in.

2.1 User Registration Process

In order to permit access to the secure areas of an information system, it is necessary to able to authenticate a user's identity so the user must be registered. Out of respect for users' trust and privacy, Egger [5] recommends that the imperative to register should be delayed as long as possible e.g. by firstly letting a user browse the site and add items to a shopping cart as a guest user. This is referred to as the *Lazy Registration* pattern (see http://ui-patterns.com./pattern/LazyRegistration). When the user is eventually asked to create a profile, a full overview and justification for all required data attributes should be clearly provided and the registration process should collect the minimum amount of information needed [22,24]. To prevent registration on a Web site by a fraudulent automated agent as opposed to a human, the use of a CAPTCHA is recommended. A further precaution is not to activate a user's account until it has been verified, which is normally done by sending a confirmation request to the email address provided during registration [24].

As regards user authentication, the two essential attributes of the user's profile created upon registration are the password and the UserID, so we shall concentrate on design considerations impacting these two attributes.

In a previous study of Website password practices, Furnell [7] asserts that it is reasonable to expect users to be provided with guidance on how to choose a secure password, but most of the sites that he assessed were remiss in this regard, thus "leaving users to select passwords in a potentially *ad hoc* manner". There is general consensus within the literature on the basic principles of what constitutes a "strong" password, as well as on simple practices to protect passwords from being discovered easily:

- It is recommended that passwords should be at least 8 characters long [7,25,28], but numerous articles mention the trade-off between password complexity/length and memorability [11,27,28]. Users are more likely to write down passwords that they cannot remember, thus potentially compromising security [32]. As yet, no research has been conducted to investigate if the Password Manager feature of Web browsers has had any effect on the tendency to choose short passwords or to write longer passwords down.
- The password should ideally use a mixture of upper and lower case letters, numbers, and symbols [25,27,28]. In theory, extending the password length has a substantially greater impact on strength than just increasing the number of available characters [11], but this only necessarily holds if the password is a random sequence of the available characters.
- The use of passphrases (e.g. "EatSleepDrinkFootball-24-7!") as opposed to passwords has been found to increase password strength while still being memorable [11]. Alternatively, a passphrase can be used as a mnemonic to recall an alphanumeric password e.g. "I have to catch a plane to London at ten o'clock" = "Ih2cap2L@10oc" [7,25,27].
- Avoid simple patterns, such as a capitalised letter at the beginning or placing digits at the end [25].
- Reject passwords that can be found in precompiled dictionaries [7,11]. For 4-digit PINs, avoid obvious combinations such as "1234", and for 6-digit PINs avoid the temptation to base the PIN on an obvious DDMMYY date (e.g. the user's birthday).
- A particularly bad choice for a password is "password", but remarkably Furnell [7] has found that quite a number of major e-commerce Websites permit this.
- The risk of a "user information attack" can be reduced by avoiding passwords based on discoverable facts such as a person's date of birth, child's name, football team etc. [7,27,28].
- Users should be forced to change their passwords regularly, at least once every 3 months [7,27,28]. This also helps with password memorability [32]. The corollary is that organisations should deactive dormant accounts.
- Do not permit the reuse of previous passwords because these may have been compromised [7].
- As a password reminder "hint" for the purposes of reset/recovery, allow a user to define his own secret question [7], as opposed to using questions based on discoverable facts (e.g. place of birth) or facts that might be easily guessed (e.g. most people's favourite film is near the top of the Internet Movie Database chart at www.imdb.com/chart/top). This approach is more secure than sending

forgotten passwords by email, which is certainly not recommended for accounts containing sensitive information [22].

In contrast to the wealth of guidance on what constitutes a strong password, there is comparatively little advice on how to choose an appropriate UserID (assuming it is not assigned by the system). Vora [24] suggests that when registering for Websites, a user's email address is often a good choice for a UserID on the basis of uniqueness and memorability. However, there is a considerable security risk attached to the common practice of using an email address as a UserID. Users habitually use the same password for multiple accounts, and rarely change their passwords [12,17,32]. This problem is therefore compounded if users also have the same UserID for multiple Websites [7,11]. For example, a spoof Website offering the prospect of last minute bargains on unsold concert tickets could deceive a user into registering, but could then try to use the same UserID and password to log in to other sites, including the user's email account. A better resolution to the problem of a user having multiple Websites with the same UserID and password is to permit users to register using a "unified" service such as Windows CardSpace, SAML or OpenID [24]. Of course, if a user can choose his own UserID, there is no barrier in theory to the use of the same principles as previously outlined so as to create a "strong" UserID.

2.2 System Login Process

In their seminal paper, Yoder & Barcalow [30] describe a design pattern named *Single Access Point* (alias *Login Window*), the basic premise of which is that there must only be one way into a system; otherwise put, everybody must enter via the front door as opposed to side and back doors. The standard solution is to create a single login screen through which all users must pass in order to gain authorised access to the restricted areas of the system. If a user attempts to access the system through a different access point, such as attempting to go directly to the URL of a restricted area, the user should be redirected through the normal login process – this simplifies control flow and encourages modular design by creating a reusable login "use case". A closely related pattern is named *Check Point* (alias *Authentication and Authorization*), which verifies the credentials of a user seeking to pass through the login process. The *Check Point* pattern recognises that users will occasionally make mistakes, such as entering an incorrect password, and that different actions need to be taken depending on the severity and number of mistakes. The recommended solution is to "create an object that encapsulates the algorithm for the company's security policy" [30] by considering every branch in the authentication logic, such as how to handle password failure or password expiration. Once a user's login is authenticated, the *Session* and *State* patterns then come into effect.

These general principles are by now well established, and in Web application design a user's login state is typically maintained through the use of cookies and HTTP sessions. A factor that must be considered is whether to permit a user to remain permanently logged in, or whether to log a user out after a certain period of inactivity [24]. Although it may be convenient for a user to remain continuously logged in to a system, this could raise a potential security problem if somebody else were to gain access to that user's workstation. Yee [29] emphasises the interaction design principle of "path of least resistance", meaning that a system should be designed so that a user cannot inadvertently compromise the default secure state of a system (e.g. by not logging out).

Although the implementation of the *Check Point* pattern will vary from one organisation to the next, many of the issues that need to be considered are common. For example, how many times should a user be permitted to enter a password incorrectly before the user's account is deactivated? Password cracker algorithms often use "brute force" attacks based on dictionary words or other known common combinations e.g. sequences of adjacent keyboard characters, or substitutions of digits for letters such as "4" for "A" or "0" for "O". Keith et al [11], working from the previous findings of Zviran & Haga [32], estimate that an unrestricted brute force attack could crack a typical password in just 2 seconds. Therefore, to prevent such attacks it is important to stipulate what could be regarded as a reasonable number of incorrect login attempts [19]. Vu [25] recommends a "six strikes and you're out policy" for failed logins as opposed to the common practice of permitting three strikes, his rationale being that users often have many passwords to remember and therefore can mix them up.

Of course, one cause of repeated failed login attempts may be that a user has forgotten the login details, or there may be a problem with case sensitivity. In most cases, the latter issue can be handled

simply by reminding the user that the password is case sensitive and by having a feature which warns if CAPS LOCK is on. As regards forgotten login details, it is important that the system has a process for handling lost passwords [22] as an extension of the normal login use case. Most systems have a "Forgot password" link which activates a reset or recovery mechanism, perhaps simply emailing the password (an insecure approach), or by emailing a once-off URL reset link, or else by asking the user a series of secret questions. Depending on an organisation's security policy, some systems may prevent users from changing their password unless personal contact is made with a human representative. On the basis of the author's personal experience, it is potentially risky not to have a mechanism to permit a user reset his password or at least temporarily disable his account without the need for the intervention of a system administrator: for example, if an organisation uses single sign-on to all its systems, an out-of-normal-hours compromise of any of those systems could place them all at risk if the password were not quickly changed.

Thus far, authentication has been discussed from the perspective of an organisation seeking to verify the identity of a user trying to access that organisation's resources. However, authentication is a "two-way street" and the user is also entitled and should be encouraged to verify that the party with whom he is interacting is not an impostor. Yee [29] refers to this as the "principle of the trusted path". For example, a user could receive an email or SMS message asking him to go to a bogus Web site or to call a telephone number, whereupon the user is asked some "standard security questions"; this is referred to as "phishing" and is one of a number of so-called "social engineering" techniques that confidence tricksters engage to lure users into disclosing personal details. Most password breaches arise this way, rather than through crude brute force attacks [17].

Unfortunately, the dilemma with Web security is that many of the issues which cause the most problems are not directly controllable. User interaction designers are limited in what they can do to prevent on-line fraud, firstly because the user's Web browser has inherent vulnerabilities (e.g. cookies, stored passwords, AutoFill options) that can potentially be exploited by malware, and secondly phishing/spoofing hoaxes can be so elaborate as to almost exactly emulate the look-and-feel of an actual Website (e.g. by registering a similar domain name, substituting visually-close characters such as capital "I" or the digit "1" for the letter "l" [26]). Thus, as Internet criminals engage in ever more devious means of entrapping users, the challenge for user interaction designers is how to maintain public trust by outwitting the fraudsters [17,19].

Egger [5] makes a number of recommendations as to how security and privacy issues can be handled in order to build and sustain a sense of trust. Amongst these are: prominent links to applicable policies (e.g. data protection, privacy/security, consumer rights), describe the technologies being used to assure security (e.g. SSL), and use trusted third parties (e.g. payment options, security certificates). Vora [24] and van Duyne et al [22] also emphasise the importance of having a clearly visible link to the organisation's privacy policy. Additionally, it is useful to provide advice to users about how to prevent and detect common types of fraudulent Internet "scams".

"Spyware" applications, "sniffers" and "Trojan horses" can potentially negate the use of Secure HTTP / SSL if they log keystrokes or grab passwords within the Web browser client *before* they are encrypted [19]. To oversome this, some financial institutions use hardware tokens to authenticate each transaction, but this places a burden on the user because simple tasks can become cumbersome [26]. Another mechanism which is being increasingly used to confound spyware is a CAPTCHA, an image-file of a randomly-generated character string which is readable by humans but not by computers [8].

Passwords can also be stolen by "shoulder surfing", where a fraudster observes a user entering his password either by loitering in close proximity or else through the use of an inconspicuous recording device. To mitigate this risk, passwords are usually masked on entry [24]. However, because the user cannot see what he has entered, two problems can arise with masks: firstly, the user may enter an incorrect password (especially if it is case sensitive); secondly, password-grabbing spoof Web sites can exploit the mask by luring a user into blindly disclosing all the digits of a PIN (e.g. "Please enter 1st, 2nd, and 5th digits" ➙ "Sorry, PIN failed. Please try again" ➙ "Please enter 3rd, 4th, and 6th digits" ➙ "System temporarily unavailable, please try again" ➙ re-direct to actual login screen). One potential improvement here would be if a user were given the option each time of whether or not to mask the characters entered in a password input box. Conversely, if there is a risk of a login process being observed, it would make sense to give the user the option of also concealing the UserID field with a mask.

3. Research Approach

Having reviewed the literature on interaction design guidance for on-line user authentication, the empirical phase of this study involved a comparison of actual practices used by Websites against the practices recommended by the literature. A purposefully selected sample of 12 Websites was chosen so as to cover a variety of Website types commonly targeted by Internet criminals, and also so as to include organisations from a number of different countries (Ireland, UK and USA/International). Because a proper analysis of Website security practices requires a longitudinal approach (e.g. to assess if periodic password changes are required), these sites were also chosen on the basis that the author is a habitual user with experience of having used all of them over an extended period of time:

- **Social networking / Personal communications (4):** Twitter, Gmail, Facebook, and LinkedIn.
- **On-line banking / Insurance services (3):** Bank of Ireland 365online (www.365online.com), Quinn Healthcare (www.quinnhealthcare.com), and Rabobank Online (www.rabodirect.ie).
- **Electronic commerce (3):** eBay, PayPal, and Amazon UK (www.amazon.co.uk).
- **Electronic government (2):** PAYE anytime (www.ros.ie), where Irish citizens can manage their income tax returns, and Motor Tax Online (www.motortax.ie).

The sites were assessed by creating a new "dummy" user registration (using an identity generated by www.fakenamegenerator.com), and where this was not possible an existing account to which the author had prior access was used. After initial registration, the sites were evaluated by following the available processes for logging in, retrieving lost/forgotten login details, and changing the user profile. Basic deliberate errors were made to test the security features, such as attempting to log in repeatedly with an incorrect combination until access is blocked, going directly to the URL of a restricted access area which requires prior login authentication, choosing a weak password, and leaving a user logged in but inactive for a period of time. The sites were also assessed for compliance with the norms of good practice as outlined in the previous section, such as clear links to appropriate privacy/security policies and periodic forced password changes.

4. Discussion of Findings

Assessment of Design of Password Mechanisms

The legend used in Table 1 for the types of characters allowable in passwords is: A = upper case letter, a = lower case letter; 9 = digit; % = symbol; + = required character type; n = password must include at least *n* different instances of the permissable character types. For example, if a password is of type [A,a,9+,%](*2*) this means it must contain at least one digit and also either an upper or lower case letter or symbol. If no restrictions are placed on the characters to be included in a password, this is indicated by [*]. Most of the Websites evaluated required that passwords be a minimum of 6 characters long, comprising a mixture of letters, digits and symbols. Three sites, - Bank of Ireland, PAYE anytime, and Motor Tax Online, - are accessed using a combination of facts and PIN, while the interaction with Rabobank Online is mediated using a physical hardware token. To test for the strengths of passwords permitted by each of the Websites, some typical choices were tried, including a very weak password ("abcd1234"), a 16-character dictionary word ("UnpredictabilitY"), a passphrase from the children's tale of Ali Baba ("OpenSesame"), a keyboard sequence in reverse ("Poiuytrewq"), a repeating 16-digit numeric sequence ("0123456789012345"), a variation of the initials and date of birth of the dummy user ("CEB300457"), the name of a football club and squad number of its star player ("Liverpool9"), and an abbreviation of an address ("1600PennAve"). The results are displayed in Table 2, and it is interesting to compare these findings against the password selection guidance provided on the sites. Although most of the sites warn against choosing a password based on a dictionary word, only one of them actually forbids this (QuinnHealthcare), this being because that is the only Website to insist that the password must contain a non-alphabetical character. Only QuinnHealthcare and eBay have adequate checks to ensure that a mixture of character types must be used, and also that neither a UserID nor a person's name can be used as his password. All of the Websites were case sensitive.

Table 1. Summary of analysis of design practices used by Web sites in practice

✓ *Guideline implemented* × *Guideline implemented* *"N/A" = Not Applicable*	**Electronic government**		**Social networking / Personal communications**				**On-line banking / Insurance services**			**Electronic commerce**		
	PAYE anytime	Motor Tax Online	Google Gmail	Twitter	Facebook	LinkedIn	Bank of Ireland	Rabobank Online	Quinn Healthcare	Amazon (UK)	PayPal	eBay
"Lazy Registration" facilitated?	N/A (demo)	N/A	N/A	N/A	N/A	N/A	N/A (demo)	N/A (demo)	N/A	✓	N/A	✓
Verification of registration? (email or off-line)	✓ (off-line)	N/A	×	✓	✓	✓	✓ (off-line)	✓ (off-line)	×	×	×	✓
Use of CAPTCHA to register?	×	N/A	✓	✓	✓	×	N/A	N/A	×	×	×	✓
Use of CAPTCHA to change password?	×	N/A	×	×	×	×	×	N/A	×	✓	×	×
Password type	PPSN + fact + PIN	Vehicle Num. + PIN	[*]	[*]	[*]	[*]	UserID + fact + PIN	"DigiPass" device	[A,a,9+](2)	[*]	[*]	[A,a,9,%](2)
Password length (min,max)	6 digit PIN	6 digit PIN	8,100	6,-	6,-	6,16	6 digit PIN	N/A	6,-	8,20	8,40	6,20
Password/PIN strength check?	×	N/A	✓	✓	✓	×	×	N/A	×	×	×	✓
Password/PIN strength meter?	×	N/A	✓	✓	✓	×	×	N/A	×	×	×	×
Guidance on password selection?	×	N/A	✓	✓	✓	✓	×	N/A	×	×	✓	✓
Forbids password based on word?	N/A	N/A	×	×	×	×	N/A	N/A	✓	×	×	×
Forbids password same as UserID or person's name?	N/A	N/A	×	×	×	×	N/A	N/A	✓	×	×	✓
Password case sensitive?	N/A	N/A	✓	✓	✓	✓	N/A	N/A	✓	✓	✓	✓
Input masking on password?	PIN only	✓	✓	✓	✓	✓	PIN only	N/A	✓	✓	✓	✓
Forbids re-use of previous password?	×	N/A	×	×	×	×	×	N/A	×	×	×	✓
Forbids use of email address as UserID?	N/A	N/A	×	×	×	×	N/A	N/A	N/A	×	×	✓
"Single Access Point"/ . "Check Point"?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
User automatically logged out after a few minutes of inactivity?	✓	Not ascertained	×	×	×	×	✓	✓	×	×	✓	✓
No. of incorrect logins permitted	Not tested	Numerous (25+)	10	3	3	10	3	Not tested	Not tested	Numerous (25+)	10	3
Forces periodic password change?	×	N/A	×	×	×	×	×	N/A	×	×	×	×
Facility to recover lost password?	✓ (off-line)	N/A	✓	✓	✓	✓	× (off-line)	× (off-line)	✓	✓	✓	✓
Avoids recovery hints based on easily discoverable facts?	N/A	N/A	✓	N/A	N/A	N/A	N/A	N/A	×	N/A	×	×
Permits user-defined recovery hint?	N/A	N/A	✓	×	×	×	×	N/A	×	N/A	×	×
Links to privacy/security policies?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Description of secure technologies?	✓	✓	×	×	×	×	✓	✓	✓	×	✓	✓
Obvious link to fraud advice?	✓	×	×	×	×	×	✓	✓	✓	✓	✓	×

Table 2. Results of tests of the admissability of various password types.

× *Disallowed password* ✓ *Permitted password*	**Google Gmail**	**Twitter**	**Facebook**	**LinkedIn**	**Quinn Healthcare**	**Amazon (UK)**	**PayPal**	**eBay**
abcd1234	×	✓	×	✓	✓	✓	✓	×
0123456789012345	✓	✓	✓	✓	×	✓	✓	×
1600PennAve	✓	✓	✓	✓	✓	✓	✓	✓
Liverpool9	✓	✓	✓	✓	✓	✓	✓	✓
CEB300457	✓	✓	✓	✓	✓	✓	✓	✓
Poiuytrewq	✓	✓	×	✓	×	✓	✓	✓
OpenSesame	✓	✓	×	✓	×	✓	✓	✓
UnpredictabilitY	✓	✓	✓	✓	×	✓	✓	✓

Gmail did not permit the UserID to be used as a password, but did allow the selection of a password based on the user's name with initial capitals; its password strength meter even indicated that this was 'Strong'. The passwords listed in Table 2 flagrantly ignore all of the password selection tips in Gmail's checklist of "Things to Avoid", but nevertheless Gmail only rejected "abcd1234", with "OpenSesame" getting a 'Good' rating and all of the others rated as 'Strong'. In contrast, the Facebook meter disallowed not just the obviously weak "abc1234", but also rejected "OpenSesame" and "Poiuytrewq". Both "UnpredictabilitY"and "0123456789012345" were given a meter reading of 'Medium' but were still permitted, and the others were all deemed 'Strong'. LinkedIn, Amazon and PayPal do not have password strength meters and they accepted all passwords, as did Twitter, flying in the face of their own password selection advice:

- PayPal: "*We recommend that your password is not a word you can find in the dictionary, includes both capital and lower case letters, and contains at least one special character (1-9, !, *, _, etc.)*." In actual fact, PayPal accepts any combination of 8 or more characters of any type.
- LinkedIn: "*A good password should contain a mix of capital and lower-case letters, numbers and symbols*".
- Twitter: "*Be tricky! Your password should be at least 6 characters and not a dictionary word or common name. Change your password on occasion*."
- Amazon: "*Use at least 8 characters, a combination of numbers and letters is best; Do not use the same password you have used with us previously; Do not use dictionary words, your name, e-mail address, or other personal information that can be easily obtained; Do not use the same password for multiple online accounts*." Peculiarly, Amazon only provides advice when changing a password, not when creating an account. Only the first of these five recommendations is enforced by Amazon.
- eBay: "*Passwords must have a mix of letters, numbers and symbols… do not use a dictionary word*." However, eBay is willing to accept a dictionary word, as long as it is of mixed case.

Bank of Ireland 365online uses a "multiple input" approach in an effort to strengthen its login combination. A user is required to enter three pieces of information to access his account: a unique six-digit user ID, a personal fact (usually date of birth or the last four digits of a telephone number), and three random digits from a user-defined six-digit PIN. Of these three pieces, the second and third are vulnerable. A date of birth and telephone number can be easily compromised e.g. through a Facebook profile, phishing attacks of CVs by spoof recruitment agencies, or simply through the loss of a handbag. Six-digit PINs are vulnerable because people tend to base them on memorable DDMMYY dates. For example, if a hacker guessed that the six-digit PIN was based on a date between 1980 and 2009, and was asked to enter the 1st / 3rd / 5th digits, then the respective possibilities are [0,1,2,3] / [0,1] / [8,9,0], giving a maximum of 24 permutations. Given that the likelihood of the first digit being 3 is low (4.9% i.e. only 18 possibilities out of 365 days), then the number of permutations is further reduced.

In contrast to the Bank of Ireland approach, Rabobank uses a physical "Digipass" device which generates a token to authenticate the various steps of a transaction. However, simple processes such as transferring funds between accounts are substantially more cumbersome in the Rabobank system than with Bank of Ireland. It is well acknowledged within the substantial body of literature on users' attitudes towards technology adoption and usage that perceived effort and perceived utility tend to be closely correlated, so users may be deterred by the "Digipass" device if they feel that the marginal gain in security that it offers is offset by the additional effort required to operate it.

The login combination for the *PAYE anytime* system is very similar to Bank of Ireland, except that in place of a bank customer UserID, it uses an individual's Personal Public Service Number (PPSN). This is a unique identity number given to each citizen in Ireland, which can be found on documents such as payslips, pharmacy prescriptions and wallet cards. Thus *PAYE anytime* is more vulnerable to an attack than the Bank of Ireland system because the three pieces of required information are either relatively easy to discover (i.e. PPSN, telephone number) or guessable (i.e. a weak PIN based on a significant date). This is all the more serious because *PAYE anytime* grants a user access to the income tax returns of himself and his partner for the previous six years, and stores details of names, dates of birth, address, telephone numbers, email, employment details, and bank account details.

Of all the sites evaluated, the one which contained the most sensitive information was QuinnHealthcare, an insurance provider. Once logged in, a person has access to extensive private data about a policy holder and his dependents, including names, genders, dates of birth, address, telephone number, email, PPSNs, and bank account details. Alarmingly, this was one of the most insecure Websites visited. The interaction design of this system is very unusual in so far as the very same details are required to register as to retrieve a forgotten password. In both cases, what is required is the user's membership number and date of birth, details that can easily be accessed by a criminal by stealing a user's wallet or robbing a medical centre. Another poor aspect of this site's design is that there appears to be no upper limit on the number of characters that can be entered as a password: a repeating sequence of 500 characters generated a "Password saved" message, but the subsequent login failed, most likely because the password was truncated when stored in the underlying database field.

Review of Security Checks for Login and Logout

The policies regarding login timeouts varied across the twelve Websites. Bank of Ireland, Rabobank, PayPal, eBay and *PAYE anytime* all force a user to log back in if he has been inactive for a few minutes. This is understandable given that these five Websites are all in the financial services or e-commerce sector. On the other hand, Gmail, Facebook and a number of other sites provide the user with an option to remain permanently logged in. The rationale for allowing habitual users of a system to remain continually logged in is to make the user experience hassle-free, but the cost-benefit of requiring a user to re-authenticate himself at least every few hours must be evaluated against the threat of the user's account being accessed illegitimately, as could for example happen if it were left temporarily unattended in an open workspace. eBay has a happy middle-ground where a user can elect to remain logged in for one day, but the logged in state does not persist beyond that period.

Another important user interaction design consideration is how to handle multiple consecutive failures to enter the correct login combination. Bank of Ireland and eBay both handle this by blocking access to an account after three failed login attempts. Facebook also permits a user to have three clear strikes, but then the user must decipher a CAPTCHA to request that a password reset code be sent to his email address. Twitter follows much the same idea as Facebook.

A number of other Websites allow a user have ten incorrect login attempts. Thereafter, PayPal blocks access and requests a user to make a telephone call to Customer Service. With both Gmail and LinkedIn, every subsequent attempt beyond the tenth consecutive failure requires a user to decipher a CAPTCHA in addition to entering the UserID and password, which is clearly intended to defeat brute force attacks. However, there seems to be no upper limit on the number of subsequent incorrect login attempts that are made, so a patient hacker with information about a user might be able to keep continually trying. Motor Tax Online and Amazon have no apparent policy for handling unsuccessful login attempts because 25 incorrect combinations were entered on both of them with no consequences.

Bank of Ireland stood out as an exception in that, when users were logging in, they were clearly notified of the time and date of the last successful login on that account. With Gmail, there is a feature to check the last activity but this is not prominent, and none of the other systems featured similar notices. The relevance of clearly letting a user know when an account was last accessed is that the user is made aware of any illegitimate activity that may have taken place.

Analysis of Procedures for Changing and Recovering Passwords

It is noteworthy that of the twelve Websites evaluated, not one compels a user to *ever* change his password. The author ashamedly admits that he retained a weak password, now altered, for more than 10 years on one of these sites. If a user is not regularly forced to change his password, he is likely to mislay it [32]. All of the systems surveyed have a mechanism for recovering lost passwords/PINs, though in the cases of *Bank of Ireland 365online* and *PAYE anytime* this necessitates off-line postal correspondance. For those systems that permit a user to change or recover a password on-line, the processes vary. If a user knows his password and is logged in, all that is normally required is to go to the account settings and specify the old and new passwords and perhaps one or two other authenticating facts; this is to prevent somebody else changing your password. In cases where a user does not know his password and cannot log in to this account, he is generally required to authenticate himself by providing certain facts or answers to questions that were configured during the account registration process. Once these facts and questions are correctly answered, what generally happens is that a confirmation reset code is sent to the designated email address of the user. Additionally, a number of sites use CAPTCHAs at various points in the interaction workflow so as to prevent automated agents trying to intercept password recovery.

However, there are a number of potential security risks in the way that user authentication is designed in this type of password recovery process. Firstly, many of the so-called "secret" questions actually pertain to facts which are likely to already be known to others or are readily discoverable, such as a telephone number, date of birth, postcode, or mother's name. Other questions may not be known but can be guessed with a reasonable likelihood of success, such as PayPal's question about one's favourite cartoon character (apparently Bugs Bunny is the usual suspect, as evidenced by numerous Internet polls). Some of the sites provide obscure question sets (e.g. Gmail gives the options of your frequent flyer number, first telephone number, first teacher's name, or library card number), which leaves the user with no option but to nominate his own secret question. The dummy user that was created in this study chose to exercise this latter option and suggested that his secret question should be "What is your favourite film?" to which he responded, being a regular sort of fellow, "The Shawshank Redemption". The point is that when users are given the option of nominating their own questions, there is a strong chance that they will choose a predictable question with a predictable answer.

The second problem with this general type of password recovery process is that, like any system of control checks and balances, it is only as robust as its weakest link. The user's email account is a key actor in the process, being the channel through which changes to other accounts are notified and verified, so the implicit assumption is that the user's email account is secure. However, if a user's email account were to be broken into, an impostor could change settings in that user's other accounts and clear his tracks by deleting all the confirmation messages, meaning that the user may not find out until some time later that one or more of his accounts was misused. Given that many people now habitually use a suite of Websites (e.g. Gmail/Facebook/Twitter/YouTube, eBay/PayPal) there is a risk of a domino effect occurring if they all, as is not uncommon, share the same UserID and password.

Finally, in relation to resetting passwords, the only site amongst those evaluated which prevents the reuse of all former passwords is eBay. The reuse of passwords is a security hazard because many people like to recycle the same combinations again and again; this is akin to handing someone a bunch of keys and saying "I don't know which one it is, but if you try all of these you'll get in".

Policies and Information for Users

It is very notable that on many of the Websites evaluated (e.g. Gmail, Amazon, LinkedIn, eBay), the link to the Privacy Policy is inconspicuously hidden within the page footer in a place that most users would rarely notice or care to look. The ethics of this type of interface design are questionnable; it almost appears as though organisations are intentionally keeping their privacy policy out of clear sight of customers. The Bank of Ireland 365online Website takes a very different approach, being exemplary in the visibility and clarity of information that it provides in relation to privacy, security, and fraud prevention. Contrary to the approach taken by the aforementioned examples, Bank of Ireland has very prominent links on its entry level pages directing customers to useful and intelligible sources of

guidance. The other bank evaluated, Rabobank, provides substantially less information as regards advice for users, preferring instead to proclaim its "100% No-Fraud On-Line Banking Guarantee".

PayPal provide an obvious link to a Security Center where users can obtain advice against phishing, spoofing, and identity theft. It is interesting however that the PayPal site permits a user to enter a very weak password (see Table 2) which is in direct contravention of the recommendation in their own Identity Theft Guide to "always choose strong passwords to protect accounts; mix upper and lowercase letters; use symbol characters". Unlike PayPal, the link on eBay's Website to their Safety Center is buried in the small print. Confusingly, the "Search" button in eBay only searches auction listings; there is no obvious way of searching or browsing eBay pages, so users are forced to really drill-down or use an external Internet search engine in order to locate information about fraud prevention on eBay's site.

5. Conclusions

Many of the design principles outlined in the literature review of this paper appear to be self-evident, yet the findings of this research indicate that most of the Websites analysed do not enforce simple validation rules on password selection and take a rather lax approach to privacy assurance. At a time when Internet criminals are devising ever more sophisticated schemes to defraud on-line users, it would seem that a lot of Websites are "leaving the key under the mat" by not taking adequate precautions to safeguard UserIDs and passwords from being violated. Of particular importance is the need to lock down email accounts with strong passwords because if an email account is violated it can be abused as a launchpad to gain access to further accounts, thus raising the likelihood of identity theft and financial fraud. It is quite absurd to think that a common document like a Curriculum Vitae may be all that a wily "social engineer" requires to initiate an elaborate scam.

To the extent that security is considered in the literature on information systems development, the main focus has been on the software engineering of secure protocols and services, security modelling, and programming aspects. There is comparatively little work on the human-centred aspects of security and their implications for user interaction design, data validation rules, and process logic. One of the mantras of user interaction design is "don't make me think!", meaning that a system should be designed in such a way that it is obvious to users what to so. However, as the findings of previous studies [7,12,32] reveal, users don't seem to think about the probability or consequence of their account being violated, most likely because it is not obvious to the average user what the risks are or what to do to mitigate those risks. It is therefore important to take pre-emptive action during systems analysis and design by anticipating the mistakes that users are likely to make, which requires the requirements analyst to be aware of how the system is actually going to be used and potentially abused or misused. Norman [17] makes the point that "developers who lack an understanding of real human behavior tend to impose logical rules". Software engineers and programmers have traditionally taken an inside-out approach to systems development, concentrating primarily on the internal back-end components. On the other hand, interaction designers take an outside-in approach, concentrating on the user's front-end experience with a particular emphasis on usability and the (mis)conduct of users. The principle of the "path of least resistance" tells us that a user will take convenient shortcuts around the intended course of action if he is not prevented from doing so. This means that in order to develop a secure information system, the back-end and front-end designers must work together to devise solutions which are cognisant of behavioural tendencies that potentially compromise security. For example, most "password strength meters" currently work on the assumption of a brute force attack, but in actuality it is usually much easier to crack a password using a personal information attack. Therefore, additional intelligence should be designed into the logic of database validation rules so as to also disallow elementary passwords based on information stored in a user's profile.

Although security and usability can sometimes be regarded as getting in each other's way in the design of Web-based information systems, what they have in common is that historically they were largely ignored until the latter stages of the traditional systems development lifecycle but are both now recognised as being foremost design issues that must be considered from the outset [30]. Security design is now being incorporated into the newer generation of systems development methods and modelling techniques [19,21], and indeed methods from previous generations can also be used for security analysis and design [2]. Thus to conclude, what is now needed to progress the current state of knowledge is to bring together researchers and practitioners from the fields of user interaction design, security engineering, psychology, organisational behaviour, and Internet criminology so as to gain a

multi-faceted understanding of the individual, organisational, and societal aspects that need to be considered when designing secure user-centred information systems.

References

1. Alexander, C. (1964) *Notes on the Synthesis of Form*. Cambridge, MA: Harvard University Press.
2. Baskerville, R. (1993) Information systems security design methods: implications for information systems development. *ACM Computing Surveys*. 25(4), 375-414.
3. Cranor, L. F. & Garfinkel, S. (2005) *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly Media. ISBN 10: 0-596-00827-9.
4. Data Protection Commissioner (Ireland) (2008) *Twentieth Annual Report of the Data Protection Commissioner*. http://www.dataprotection.ie (accessed April 20, 2010).
5. Egger, F. N. (2001) Affective design of e-commerce user interfaces: how to maximise perceived trustworthiness. In Helander et al. (eds) *Proc. International Conf. on Affective Human Factors Design*. London: Asean Academic Press.
6. Fernandez-Buglioni, E., Hybertson, D. & Sommerlad, P. (2005) *Security Patterns: Integrating Security and Systems Engineering*. Wiley.
7. Furnell, S. (2007) An assessment of Website password practices. *Computers & Security*. 26(7-8), 445-451.
8. Gao, H., Liu, X., Wang, S. & Dai, R. (2009) A new graphical password scheme against spyware by using CAPTCHA. In *Proc. Symposium On Usable Privacy and Security (SOUPS)*, July 15-17, Mountain View, CA, USA.
9. Halkidis, S. T., Chatzigeorgiou, A. & Stephanides, G. (2006) A qualitative analysis of software security patterns. *Computers & Security* 25(5), 379-392.
10. Internet Crime Complaint Centre (2009) *Internet Crime Report*. http://www.ic3.gov/media/annualreports.aspx.
11. Keith, M., Shao, B. & Steinbart, P. J. (2007) The usability of passphrases for authentication: An empirical field study. *International Journal of Human-Computer Studies*. 65(1), 17-28.
12. Lang, M., Devitt, J., Kelly, S., Kinneen, A., O'Malley, J. & Prunty, D. (2009) Social Networking and Personal Data Security: A Study of Attitudes and Public Awareness in Ireland. In *Proc. International Conference on Management of e-Commerce and e-Government (ICMeCG)*, Nanchang, China, September 16-19. IEEE CompSoc, pp. 486-489.
13. Lyardet, F., Rossi, G. & Schwabe, D. (1999) Discovering and using design patterns in the WWW. *Multimedia Tools and Applications*, 8(3), 293-308.
14. McDermott, J. & Fox, C. (1999) Using Abuse Case Models for Security Requirements Analysis. In *Proceedings of 15th Annual Computer Security Applications Conference (ACSAC'99)*, Phoenix, Arizona, USA.
15. Microsoft (2005) *Web Service Security: Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements* (WSE) 3.0. http://msdn.microsoft.com/en-us/library/aa480569.aspx (accessed April 20, 2010).
16. Muñoz-Arteaga, J., González, R. M. & Vanderdonckt, J. (2008) A classification of security feedback design patterns for interactive Web applications. In *Proceedings of Third International Conference on Internet Monitoring and Protection*, pp. 166-171. IEEE Computer Society.
17. Norman, D. (2009) When security gets in the way. *ACM interactions* 16(6), 60-69.
18. Preece, J., Rogers, Y., & Sharp, H. (2002) Interaction Design: Beyond Human-Computer Interaction. Wiley.
19. Schneier, B. (2004) Customers, passwords, and Web sites. *IEEE Security & Privacy* 2(4), 88.
20. Sindre, G. & Opdahl, A. L. (2005) Eliciting security requirements with misuse cases. *Requirements Engineering* 10(1), 34-44.
21. Siponen, M. & Heikka, J. (2008) Do secure information system design methods provide adequate modeling support? *Information and Software Technology* 50(9-10), 1035-1053.
22. van Duyne, D. K., Landay, J. A. & Hong, J. I. (2003) The Design of Sites - Patterns, Principles and Processes for Creating a Customer-Centered Web Experience. Addison-Wesley. ISBN 020172149X.
23. Villarroel, R., Fernández-Medina, E., & Piattini, M. (2005) Secure information systems development: a survey and comparison. *Computers & Security* 24(4), 308-321.
24. Vora, P. (2009) *Web Application Design Patterns*. Morgan Kaufmann. ISBN 012374265X.
25. Vu, K.-P. L., Proctor, R. W., Bhargav-Spantzel, A., Tai, B.-L., Cook, J. & Schultz, E. E. (2007) Improving password security and memorability to protect personal and organizational information. *International Journal of Human-Computer Studies*. 65(8), 744-757.
26. Wadlow, T. & Gorelik, V. (2009) What can be done to make browsers secure while preserving their usability? *Communications of the ACM* 52(5), 40-45.
27. Yan, J., Blackwell, A., Anderson, R. & Grant, A. (2004) Password memorability and security: empirical results. *IEEE Security & Privacy*. 2(5), 25-31.
28. Yapp, P. (2001) Passwords: use and abuse. *Computer Fraud & Security* 9(1), 14-16.
29. Yee, K.-P. (2002) User interaction design for secure systems. In Deng, R. et al. (eds.) *Information and Communications Security*, Springer LNCS Vol. 2513, pp. 278–290.
30. Yoder, J. & Barcalow, J. (1997) Architectural patterns for enabling application security. In *Proceedings of 4th Conference on Pattern Languages of Programming (PLoP '97),* Monticello, Illinois, USA, September 3-5.
31. Zurko, M. E. & Simon, R. T. (1996) User-centred security. In *Proceedings of New Security Paradigms Workshop*, Lake Arrowhead, California, USA, pp. 27-33. ACM Press. ISBN 0-89791-944-0.
32. Zviran, M. & Haga, W. J. (1999) Password security: an empirical study. *Journal of Management Information Systems*. 15(4), 161-185.