Security Usability Principles for Vulnerability Analysis and Risk Assessment *

Audun Jøsang
Faculty of Information Technology
QUT, Australia
a.josang@qut.edu.au

Bander AlFayyadh
Information Security Institute
QUT, Australia
alfayyadhb@isi.qut.edu.au

Tyrone Grandison
IBM Almaden Research
San Jose, USA
tyroneg@us.ibm.com

Mohammed AlZomai
Information Security Institute
QUT, Australia
alzomaim@isi.qut.edu.au

Judith McNamara
Faculty of Law
QUT, Australia
j2.mcnamara@qut.edu.au

Abstract

Usability is the weakest link in the security chain of many prominent applications. A set of security usability principles should therefore be considered when designing and engineering IT security solutions. When improving the usability of existing security applications, it is necessary to examine the underlying security technologies used to build them, and consider whether they need to be replaced by totally new security technologies that provide a better basis for good usability. This paper examines a set of security usability principles, proposes how they can be incorporated into the risk management process, and discusses the benefits of applying these principles and process to existing and future security solutions.

1. Introduction

Poor usability of security systems and the consequences thereof have been pointed out by several authors. Whitten and Tygar's study [19, 20] on the usability of PGP is considered to be pioneering in this field. The importance of the usability aspect of security was discussed by earlier authors like Zurko and Simon [21], and even more than 100 years earlier by the Belgian cryptographer Auguste Kerckhoffs [11], who is most known for establishing the principle that security should not be based on obscurity. In fact, this is only one of several security principles that Kerckhoffs established, some of which relate specifically to usability.

Below is the list of Kerckhoffs' security principles[1]

1. The system must be substantially, if not mathematically, undecipherable;
2. The system must not require secrecy and can be stolen by the enemy without causing trouble;
3. It must be easy to communicate and remember the keys without requiring written notes, it must also be easy to change or modify the keys with different participants;
4. The system ought to be compatible with telegraph communication;
5. The system must be portable, and its use must not require more than one person;
6. Finally, regarding the circumstances in which such a system is applied, it must be easy to use and must neither require stress of mind nor the knowledge of a long series of rules.

Security principles 3 and 6 are in fact usability principles that are particularly relevant today, but that unfortunately have been mostly overlooked in the last 120 years [13].

The experience and skills gained in the contemporary discipline of CHI can now be applied to the domain of information security in order to better understand and improve the usability of security. This has given us a relatively good understanding of the role that usability plays in information security systems [15].

The main purpose of information security systems is to defend against adverse impacts. Generally, the strength of a

*Appears in the Proceedings of the Annual Computer Security Applications Conference (ACSAC'07) Miami Beach, December 2007.

[1] Translated by Fabien Petitcolas. The original articles with translations are available at http://www.petitcolas.net/fabien/kerckhoffs/

security system is determined by the weakest link. In many cases it is the human operator who represents the weakest link [16]. Social engineering attacks precisely target the human link, and represent a very effective attack vector. For example, reformed computer hacker Kevin Mitnick found that he never had to crack passwords by technical means, because he could always get them from people [12].

Security systems must be viewed as socio-technical systems that depend on the social context in which they are embedded to function correctly [14]. Security systems will only be able to provide the intended protection when people actually understand and are able to use them correctly. There is a very real difference between the degree by which systems can be considered theoretically secure (assuming they are correctly operated) and actually secure (acknowledging that often they will be operated incorrectly). In many cases, there is a trade-off between usability and theoretical security. It can be meaningful to reduce the level of theoretical security to improve the overall level of actual security. For example, the strongest passwords, from a theoretical perspective, are randomly generated. However, since it is very difficult to remember such passwords, people will write them down, and thereby undermine the system's security. Thus, it may be meaningful to allow people to choose passwords that are easier to remember. Although this reduces the theoretical strength of the passwords, it increases the security of the system as a whole.

The trade-off between usability and theoretical security is not generally accepted as a fundamental principle in security design. Some authors maintain that theoretical security does not have to be compromised if usability aspects are considered from the beginning of the system development life cycle. These authors describe specific approaches and development processes, which when followed, can improve the end result with regard to usability [15, 4]. This represents the *sustaining approach* to creating user friendly security because it does not question the underlying security building blocks, only how they are implemented.

However, even with a development process that focuses on good usability design, certain security building blocks are inherently unsuitable for designing user friendly security solutions. Some authors have pointed out that when security building blocks have limited potential for being implemented as user friendly, it can be necessary to invent radically new security building blocks in order to create security systems that are user friendly [17]. We will denote this as the *disruptive approach* to creating user friendly security because it questions the applicability of existing security primitives, and seeks to replace them with other primitives that better support user friendly security.

In this paper we describe a set of general security usability principles and show how they can be applied in the context of vulnerability analysis and risk assessment .

2. Principles of Security Usability

The usability of security is crucial for the overall security of the system, but is still a relatively poorly understood element of IT security. In [9] a set of general security usability principles were proposed in relation to identity management. We propose to use these principles as a basis for defining security usability vulnerabilities and for conducting risk assessments. The principles are described below.

The user's interaction points with a security application are the *security action* and the *security conclusion* stages. We formally define these terms as:

- A *security action* is when users are required to produce information and security tokens, or to trigger some security relevant mechanism. For example, typing and submitting a password is a security action.
- A *security conclusion* is when users observe and assess security relevant evidence in order to derive the security state of systems. For example, observing a closed padlock on a browser, and concluding that the communication is protected by TLS is a security conclusion.

Usability principles related to security actions and security conclusions are described below.

Security Action Usability Principles:

A1. Users must understand which security actions are required of them.

A2. Users must have sufficient knowledge and the ability to take the correct security action.

A3. The mental and physical load of a security action must be tolerable.

A4. The mental and physical load of making repeated security actions for any practical number of instances must be tolerable.

Security Conclusion Usability Principles:

C1. Users must understand the security conclusion that is required for making an informed decision.

C2. The system must provide the user with sufficient information for deriving the security conclusion.

C3. The mental load of deriving the security conclusion must be tolerable.

C4. The mental load of deriving security conclusions for any practical number of instances must be tolerable.

These usability principles emerge from Kerckhoffs' security principles 3 and 6. In the next section we describe how these principles can be incorporated into the risk assessment process.

3. Integrating Usability in Risk Assessment

Information security management seeks to establish, implement, operate, monitor, review, maintain and improve the level of information security in an organisation [8]. Risk assessment forms an integral part of security management, because it is only when there is a real risk that it makes sense, from a business perspective, to implement security controls.

As already pointed out, it is recognised that usability often is the weakest link in the security chain of IT system. This means that poor security usability actually represents a serious vulnerability in those systems. It is therefore paradoxical that modern security and risk management standards do not seem to take security usability into account at all. The term *"usability"* is not even mentioned in *"ISO/IEC 27001:2006 Requirements for Information security management systems"* [8] or in *"NIST Special Publication 800-30 – Risk Management Guide for Information Technology Systems"* [18], which are prominent security and risk management references. It is also very telling that we could not find any reference to usability on the National Vulnerability Database website[2]. Although poor security usability clearly represents a significant vulnerability, a literature search on risk analysis and usability revealed no publication where this is explicitly mentioned. The references we found mentioned usability of the risk analysis process itself, but not usability as a factor in the risk analysis. Thus, it seems that poor security usability still does not appear on standard vulnerability checklists used by security analysts and experts. We think there is an urgent need to rectify this situation, and fortunately this is relatively simple to do. Below we briefly outline how security usability can be included in standard risk assessment.

Risk assessment forms part of the System Development Life Cycle (SDLC) [18] and is used to determine the potential threats and risks associated with an IT system. The output of the risk assessment helps to identify appropriate security controls for reducing risk to an acceptable level.

NIST [18] specifies the risk assessment process as shown in Table 1. We propose that the appropriate point in the process for considering the various forms of poor security usability is the shaded step 3: *"Vulnerability identification"*.

A *threat source* can be an agent with malicious intent, an agent susceptible to non-intentional error, or a natural phenomenon. A *vulnerability* is a weakness that could be exercised or exploited to cause adverse events. A *threat* is then defined as a potential adverse event or action caused by a threat source that successfully exercises a particular vulnerability. The *likelihood* of the threat to occur increases with the strength or motivation of the threat source, as well as with the degree of vulnerability. Associated with each threat is an *impact magnitude* which expresses the direct or indirect loss resulting from the threat occurrence. The *risk* of a threat is derived as the combination of the threat's likelihood and impact magnitude, as illustrated in Fig.1.

[2] http://nvd.nist.gov/ncp.cfm

Step 1.	System characterization
Step 2.	Threat identification
Step 3.	Vulnerability identification
Step 4.	Analysis of existing security controls
Step 5.	Likelihood determination
Step 6.	Impact analysis
Step 7.	Risk determination
Step 8.	Recommendation of new controls
Step 9.	Results documentation

Table 1. Risk assessment process [18]

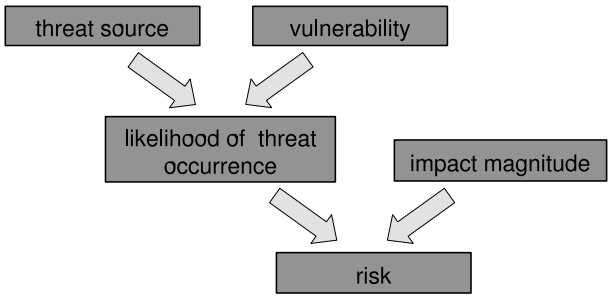

Figure 1. Principle for determining risk

A team conducting a risk assessment will try to identify all relevant vulnerability-threat combinations during steps 2 and 3 in the process of Table 1. The likelihood and impact of each threat are estimated during steps 5 and 6. The risk of each threat is determined during step 7 by a look-up matrix as illustrated in Table 2. The risk of a threat is given by the cell corresponding to the likelihood and impact of the threat. In this example, the possible risk levels are N (Negligible), L (Low), M (Medium), H (High) and E (Extreme).

Likelihood	Impact magnitude				
	Insignificant	Minor	Moderate	Major	Catastrophic
Certain	M	H	H	E	E
Likely	L	M	H	H	E
Possible	L	L	M	H	H
Unlikely	N	L	L	M	H
Rare	N	N	L	L	M

Table 2. Look-up risk matrix

Several hundred threats can be identified during a major risk assessment exercise. The team will normally use predefined checklists of threats and vulnerabilities for this task.

The identified threats are ranked according to risk level during step 7 of the risk assessment process. New security controls to be considered during step 8 will be aimed at mitigating the highest risks first.

We believe that vulnerability checklists used during step 3 traditionally have not included the various forms of poor security usability that are common in security systems today. As a result, many relevant vulnerability-threat combinations, and thereby significant risks, are routinely being overlooked.

In order for realistic threats, resulting from poor usability, to be captured by a risk assessment process it is necessary to explicitly consider poor security usability as a vulnerability. Relevant checklists must then be updated to include such vulnerabilities.

We propose to define security usability vulnerabilities as violations of the security usability principles of Sec.2. By adopting the abbreviation SUV to denote a Security Usability Vulnerability, each vulnerability can be referenced in a compact form. Table 3 shows the list of security usability vulnerabilities derived from the principles of Sec.2.

Security usability vulnerabilities on action	
SUV-A1	Users are unable to understand which security actions are required of them.
SUV-A2	Users do not have sufficient knowledge or are unable to take the correct security action.
SUV-A3	The mental and physical load of a security action is not tolerable.
SUV-A4	The mental and physical load of making repeated security actions for any practical number of instances is not tolerable.
Security usability vulnerabilities on conclusion	
SUV-C1	Users do not understand the security conclusion that is required for making an informed decision.
SUV-C2	The system does not provide the user with sufficient information for deriving the security conclusion.
SUV-C3	The mental load of deriving the security conclusion is not tolerable.
SUV-C4	The mental load of deriving security conclusions for any practical number of instances is not tolerable.

Table 3. Security usability vulnerabilities

Note that the assessment and determination of tolerability of the mental and physical load on individuals is an open research problem, which is beyond the scope of this paper.

The next two sections provide a brief usability vulnerability analysis and risk assessment of current web security solutions. This will demonstrate that serious risks can easily be discovered in this way. The selection of suitable controls corresponding to step 8 in the risk assessment process will be discussed in Sec.6.

4. Web Security Usability

Current web security technology is based on the Transport Layer Security (TLS) protocol. It is normally assumed that TLS provides the security services *message confidentiality* and *server authentication*. It will become clear that the server authentication provided by TLS is mostly theoretical, and meaningless in practice due to poor usability.

Despite being based on strong cryptography, there are a number of security exploits that TLS cannot prevent. For example, phishing attacks are a combination of social engineering and man-in-the-middle attacks aimed at obtaining sensitive information, like login identities and passwords from unsuspecting users. A phishing attack normally starts by sending email asking people to log on to a fake web site masquerading as a genuine web sites that requires login and authentication. There are always people who will fall victim to such emails, and they will not notice the fake web site despite using TLS. Technically speaking, the fake web site has been correctly authenticated if it uses TLS. Semantically speaking, this is a case of a *false positive*, i.e. the client has wrongfully authenticated the server. The problem is not due to weak cryptographic authentication mechanisms, but to poor usability of the overall authentication solution, of which TLS is only a small part. This type of attack, described in [10], is illustrated in Fig.2.

By comparing the security solution with the security usability principles described above, it can easily be seen why the security fails in this case.

The standard implementation of TLS in web browsers provides various types of information about server authentication. Unfortunately none of this information is sufficient to make an informed conclusion about the identity of the web server, as will be explained below.

The closed padlock in the corner of your browser represents one form of security information indicating that the web session is protected with TLS. It is simple to interpret and causes negligible mental load. However, the fact that is does not say anything about the identity of the server indicates the presence of vulnerability SUV-C2.

Additional security information is contained in the server certificate that can be inspected by double-clicking on the padlock. The mental load of analysing the content of a server certificate is at least intolerable, which indicates the presence of vulnerability SUV-C3 and C4. The vulnerability SUV-C2 could have been eliminated had the certificate provided sufficient information to derive the correct security conclusion, but it seems that not even that requirement is satisfied. The following analysis will make this evident.

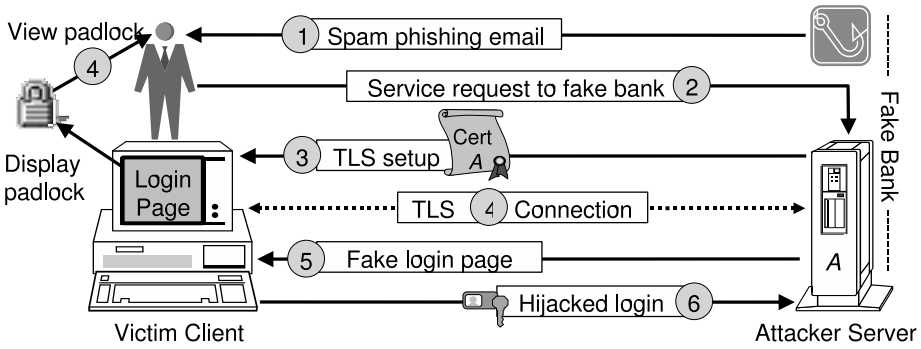

Figure 2. Typical phishing attack scenario

We will consider the real fraudulent phishing site with URL `http:\\www.hawaiiusafcuhb.com` that targeted the Hawaii Federal Credit Union in March 2007.

Assuming that potential victims want to inspect the server certificate for its authenticity, it is interesting to see that it actually provides very little useful information. Fig.3 shows general information about the attacker's certificate as it can be viewed through the MSIE[3] browser.

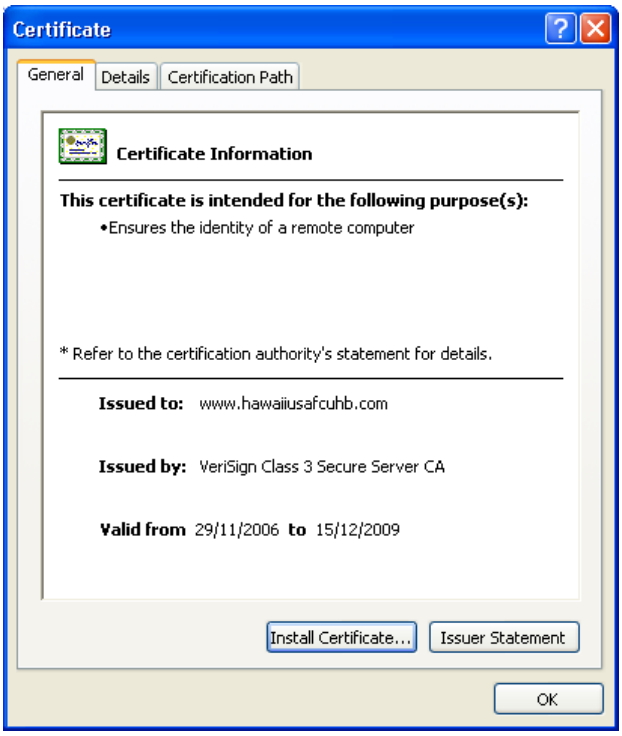

Figure 3. Fake certificate general info

More detailed information can be viewed by selecting the *"Details"* and *"Certification Path"* placeholders on the certificate window. This gives the fraudulent certificate's validity period and the certification path from the root to the fraudulent certificate. However, this additional information gives no indication that the certificate is fraudulent.

The unique identifier of the fraudulent certificate's owner is the domain name to which the fraudulent certificate is issued, specified as `www.hawaiiusafcuhb.com`, which is equal to the URL of the fake login page.

The question now arises whether this represents sufficient evidence for the user to detect that the certificate is fraudulent. In order to find out, it is necessary to compare the fraudulent certificate to the genuine certificate of the genuine Hawaii Federal Credit Union illustrated in Fig.4.

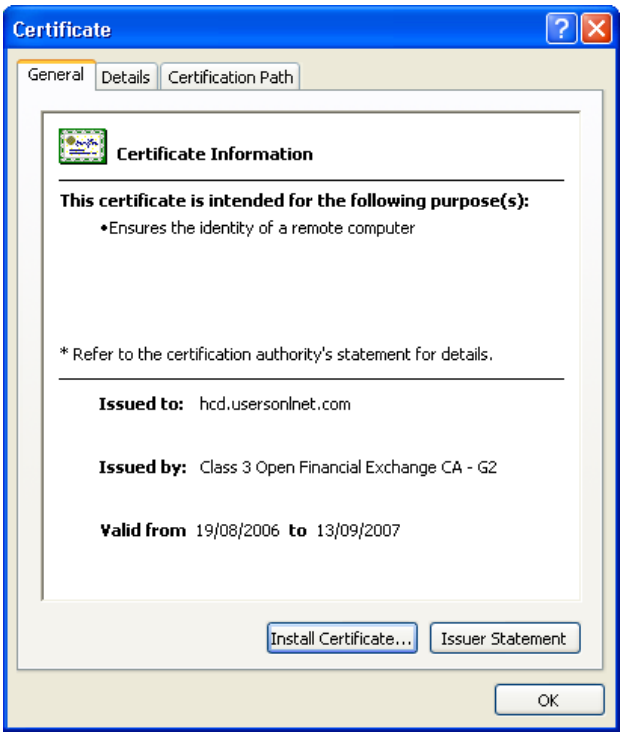

Figure 4. Genuine certificate general info

[3]MicroSoft Internet Explorer

The unique identifier of the genuine certificate's owner is the domain name to which the genuine certificate is issued, specified as `hcd.usersonlnet.com`. Interestingly this domain name **does not** correspond to the URL of the genuine Hawaii Federal Credit Union which is `www.hawaiifcu.com`. Intuitively this fact seems to indicate that the login page is not related to the genuine Hawaii Federal Credit Union. Based on this evidence, users who inspect the certificate could therefore falsely conclude that the genuine login page is fake.

This analysis indicates that not even the information found in the certificates is sufficient to make the correct security conclusion. The presence of vulnerability SUV-C2 has therefore been firmly established.

It seems that the Certificate Authorities are aware of this problem, and are careful to have policies that avoid any liability of practical misuse of the certificates they issue. The certificate window of Fig.4 provides a click-able button called *"issuer statement"* that opens a new window with the certificate issuance policy, which is a 2,666 word document (approximately four full standard pages in MS Word). While it might provide sufficient information to judge the legal status of the certificate, the size of this document alone clearly indicates the presence of vulnerability SUV-C3, as well as SUV-C4. In order to better understand why TLS can lead to a false positive authentication conclusion, it is useful to look at the very meaning of authentication.

According to the standard definition, peer-entity authentication is *"the corroboration that a peer entity in an association is as claimed"* [7]. When looking at message 3 in Fig.2 and the certificate owner of the fraudulent certificate of Fig.3, the attacker claims its own identity in the formalism of TLS, and the TLS client simply verifies the correctness of that claim. However, the claimed identity expressed in the certificate of Fig.3 does not correspond to the identity that the user assumes. Thus, the problem is one of identity representation and identity mapping.

The identity of the genuine bank assumed by the user is different from the identity of the same genuine bank assumed by the TLS client. Thus, the bank is an entity with multiple identities. From the user's perspective, the ordinary name and logo of the bank constitute a large part of the identity. From the client browser's perspective, this identity cannot be used because normal names can be ambiguous and visual logos can not be interpreted.

Certificates, which must be unambiguous, require globally unique identifiers in order to allow efficient automated processing. Domain names mostly satisfy this requirement[4] and have therefore been chosen to represent the identity of the bank in server certificates. Having different identities for the same entity can obviously cause problems. A simple way of solving this problem could be by requiring that users learn to identify online service providers by their domain names. Unfortunately this will not work because online banks often use multiple domain names depending on the service being offered.

As the example of the certificate of the genuine Hawaii Federal Credit Union bank of Fig.4 shows, many companies' secure web sites have URLs with non-obvious domain names that do not correspond to the domain names of their main web sites. Another vulnerability is the fact that distinct domain names can appear very similar, for example differing only by a single letter so that a false domain name may pass undetected. How easy is it for example to distinguish between the following URLs?
`http://www.bellabs.com`,
`http://www.belllabs.com`
`http://www.bell-labs.com`?

The crux of the problem is that domain names are designed for Internet applications and provide poor usability for naming organisations in the real world. Ordinary names are suitable for dealing with organisations in the real world, but not for online authentication. The consequence of this is that the users do not know which service provider identity to expect when accessing online services. Without knowing which identity to expect, authentication becomes meaningless. In other words, the users do not know which security conclusion to make, which indicates the presence of vulnerability SUV-C1.

To summarise, the above analysis of web security has exposed the presence of vulnerabilities SUV-C1, C2, C3 and C4. We now know that these vulnerabilities have been exploited by criminals to mount a large number of successful phishing attacks. Given that the risk of phishing attacks is well known it would be superfluous to conduct a usability risk assessment in this case. We nevertheless considered it important to demonstrate the relative simplicity of identifying security usability problems. Had this vulnerability analysis been conducted by banks before they rolled out online banking applications with TLS it would have been possible to predict and possibly prevent these attacks.

Current approaches to solving the problem include anti-phishing toolbars. Most anti-phishing toolbars are based on one or a combination of the following elements: blacklists, whitelists, ratings, heuristics [3]. However, none of these elements attempt to solve the fundamental problem of mapping the unique domain name contained in the certificate to a user friendly identity that the user can recognise. Thus they do not improve the users' ability to authenticate the server, but is an attempt to flag malicious servers. Only the Mozilla TrustBar [6] seems to solve the problem by making the authentication meaningful. The Mozilla TrustBar solution consists of personalising every server certificate that the user wants to recognise. The personalisation can e.g. consist of linking the certificate to an image or a audible tune

[4] Can change over time. See also TAG URI: http://www.taguri.org/

that the user chooses.

In the next section, we examine another application domain that purports to address some of the concerns mentioned in this section.

5. Transaction Authorization Usability

In reaction to the stream of phishing attacks, many banks have rolled out additional security solutions. Some banks issue special hardware tokens that can generate one-time authorization codes, whereas other banks rely on out-of-band communication to the customer's communication device of choice, e.g. their mobile phone. In the latter approach transactions can be authorized using SMS messages sent to the user's mobile phone. Although the user has been authenticated and is already logged in, this allows authentication of the transaction request itself. Below we conduct a simple usability risk assessment of this method of transaction authorization.

5.1. Transaction Authorization with SMS

SMS messages sent from the bank to the user's mobile phone pass through the cellular network, which is assumed to be independent of the Internet. The user can manually transfer data from the mobile phone to the client terminal. By verifying the correct transfer of data from the mobile phone to the client terminal, the bank can conclude that the user received the data through the cellular network, read it and submitted it through the Internet. This is then interpreted as a genuine intent to submit the transaction. The security of this scheme is based on the assumption that it is difficult for an attacker to steal the user's personal mobile phone and to attack the cellular network. The scenario is illustrated in Fig.5.

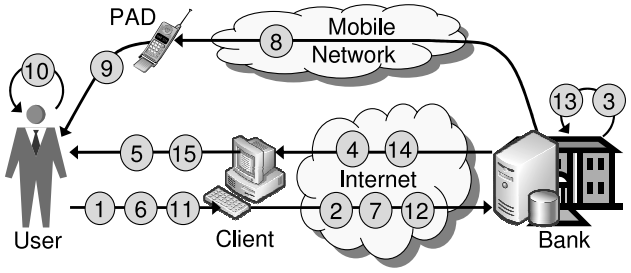

Figure 5. Transaction authorization with SMS

The SMS authorization code is computed as a function of the origin and destination accounts, as well as the amount of money to be transferred. It typically consists of 8 digits which is the same length as a normal telephone number, and can therefore be copied manually from the mobile phone to the client terminal without too much effort. A typical SMS message as used in the scheme of National Australia Bank is illustrated in Fig.6.

Msg ♯	Message description
1	Produce Login Id and authentication token
2	Transmit Login Id and authentication token
3	Verify Login Id and authentication token
4	Transmit service options
5	Present service options
6	Transaction request
7	Transmit transaction request
8	SMS message with authorization code
9	Read SMS message
10	Verify amount and bank account number
11	Copy authorization code
12	Transmit authorization code
13	Verify authorization code
14	Transmit transaction confirmation
15	Present transaction confirmation

Table 4. Messages of Fig.5

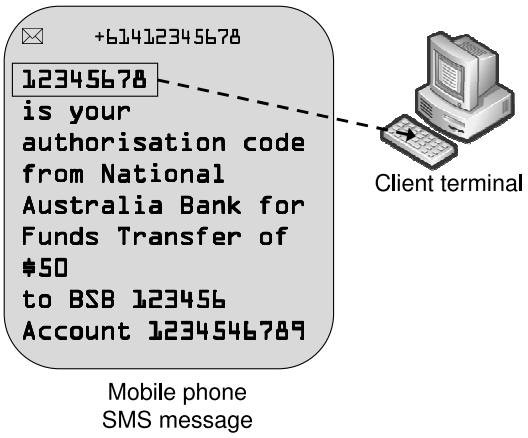

Figure 6. Example SMS message with authorization code

Assuming that the user is able verify the correctness of the amount and of the bank account number in SMS messages consistently and reliably, this scheme is secure against attacks on the client terminal, and is in fact independent of the security of the client terminal. This would represent a considerable security improvement.

This scheme assumes that the mobile terminal can be trusted, i.e. that no attacker is able to take over the control of mobile terminals, in contrast to standard desktop client terminals. If it were possible to take over the control of the mobile terminal, an attacker could change the SMS message, and present the expected amount and the bank account

number, so that the SMS message that the user reads is not the same as the SMS message that the bank sent.

The scheme also depends on the security of the mobile phone networks, and it assumes that no attacker is able to modify SMS messages sent to the user while in transit through the mobile network. Even if interception and cryptanalysis of the SMS messages sent over the air were possible, it requires that the the attacker is physically present in the same base station coverage area, and this excludes attacks from many places in the world.

5.2. Risk Assessment of SMS Transaction Authorization

Assuming that an attacker changes the amount and/or the destination account number, e.g. by a Trojan program on the client terminal, the modified amount and account number will appear in the SMS message. It is assumed that the correctness of the amount and of the destination account number is verified by the user when copying the authorization code from the SMS message. However, this can be quite tedious and could therefore represent a security usability vulnerability. If a user victim fails to notice that the destination account number specified in the SMS message is not the same as the intended destination account number, and submits the authorization code through the client terminal, the attack will succeed. Despite being the victim of an attack, the liability could be put on the user because he accepted the SMS message.

In a study of the usability of SMS authorization, it was found that 21% of participants failed to notice when the destination account number was modified under a simulated attack [1]. This indicates the presence of vulnerability SUV-C4. The study did not focus on whether users were able to correctly copy the authorization code from the SMS message to the browser window as a possible indication of vulnerabilities UV-A1 - 4. Below we will conduct a simple usability risk assessment of the out-of-band SMS authorization method.

The identified vulnerability SUV-C4 can be combined with relevant threats to form a set of vulnerability-threat combinations. The vulnerability and threat source that we will consider are given below.

- **Vulnerability SUV-C4:**
  Users failing to notice that the destination account has been changed. Then making the wrong conclusion that the transaction integrity is preserved.

- **Threat Source:**
  Hackers and computer criminals attempting to conduct fraudulent bank transactions

In combination with the mentioned vulnerability and threat source we will consider the following two threats:

T1. **Smart Trojan Threat:**
Sending out spam email inviting users to access a web site that will install a smart Trojan on users client computer. This Trojan will observe activities on the client computer and get into action when the user starts an online bank session. When the user specifies a funds transfer transaction, the Trojan will alter the amount and destination account without displaying the alteration on the screen. The online bank will thus receive a transaction request with the false amount and destination account. Even when the transaction requires authorization via an SMS message, a significant percentage of users will fail to notice that the transaction details have been altered.

T2. **Pharming and Man-in-the-Middle Threat**
Interception of the communication between the user and the bank server and impersonating both. This could happen by directing the user to the attacker's website through pharming attacks. This consists of luring the user to access a website where malicious code will be uploaded, which in turn will poison the DNS cache of the client computer so that the URL of the legitimate bank will be translated to the IP address of the attacker. When the user sends a transfer transaction request to the attacker website, the attacker can send a similar altered transaction request (i.e. by changing the destination account number) to the real online bank. Upon receiving the altered transaction request, the online bank will then send an SMS message containing the authorization code and the false transaction details to the customer.

Given that about 20% of users will fail to notice alteration of destination account numbers, 1 out of 5 attacks of the above described form will be successful. While the smart Trojan threat and the man-in-the-middle threats require advanced technical skills to be executed, we consider their likelihood to be *"Possible"* in terms of the risk matrix of Table 2.

Assuming that attackers are able to conduct fraudulent transactions either through the smart Trojan attack or the man-in-the-middle attack, considerable amounts of money can be diverted. We therefore consider the impact magnitude to be *"Major"* in terms of the risk matrix of Table 2.

The likelihood and impact together indicate that this poses a *"High Risk"*. We predict that it is only a question of time before this risk will materialise.

6. Security Usability Controls

After a vulnerability analysis and risk assessment the next step is to specify suitable security controls to mitigate against the risks. In this section we will briefly dis-

cuss general strategies for finding suitable security usability controls. The question is whether it is possible to reduce security vulnerabilities by improving the user interface, or whether the current user-unfriendly technology has to be replaced by totally new security technology that allows better usability. The first approach can be called a *sustaining approach*, and the second a *disruptive approach*. Without using the same terminology, other authors have also pointed out the need for radically rethinking security [2, 5] in order to improve its usability.

6.1. Sustaining Approach

The sustaining approach to improving security usability consists of keeping the security building blocks more or less unchanged while improving the interface and changing the way users interact with the system. Whitten & Tygar's study [19, 20] demonstrated that usability of security has different requirements than usability of IT in general. A sustaining approach must take this finding into account, meaning that simply applying best practice in usability design will often be inadequate for security. A vulnerability analysis of security usability can be conducted in order to determine whether this approach can give a satisfactory result. Alternatively, a disruptive approach should be considered.

6.2. Disruptive Approach

The disruptive approach to improving security usability consists of replacing existing security building blocks with totally new ones that have a better potential for being implemented in a user friendly way. The examples described in the previous sections indicate that improving the usability of security can be challenging and require significant innovation in the underlying security technology in order to be successful. Innovation in the usability of security will therefore often require disruptive technologies. Smetters & Grinter for example describe how identity based cryptography can be used instead of traditional certificates combined with PKIs in order to allow mutual authentication between parties in open computer networks [17]. Identity based cryptography means that an entity's public encryption/verification key can be derived directly from the entity's identity such as an email address. This is an example of a highly disruptive technology because it does not rely on PKIs, but still serves the same purpose as a traditional PKI.

6.3. Usability Metric

A potential tool to assist with improving security usability is to apply a metric, i.e. a quantitative method for assessing the degree to which a particular security element is usable. A metric would not only be practical for assessing the usability of existing security solutions and thereby provide improved understanding of the security effectiveness overall, but could also predict security usability before the security design is implemented. In developing this metric, relevant factors for security usability should be considered:

- *User Level of Expertise.* The difference between users in terms of their knowledge or expertise about security. That difference could be general, like education level, or specific to certain aspects, such as being an intrusion detection expert. It has been observed that novice and expert users interact differently with a system. This difference is due mainly to the user's cognitive characteristics.
- *Security Concepts.* Users can be overwhelmed by difficult security concepts. For example, users need to have a certain degree of knowledge about PKI in order to use PGP, and to many non-professional users, PKI is a concept that is hard to understand. Other security concepts, such as passwords, are relatively easier to understand and use. The level of difficulty of a concept, *the cognitive load*, decides or at least greatly influences the level of usability. Security concepts could be divided into Hard, Medium and Easy. Ranking these concepts in a security system could be used as a factor in the calculation of a security usability index.

This only represents a preliminary set of factors that should be considered when developing a usability metric. The purpose of a security usability metric is to have a framework for assessing the security usability of security systems without having to conduct user experiments. This can serve as a tool for selecting appropriate security usability controls.

7. Conclusion

Many significant risks are caused by poor security usability. It is necessary to consider security usability as part of vulnerability analysis and risk assessment in order to properly manage current and emerging risks.

We have described a set of security usability principles, and shown by examples how these can be used to define vulnerabilities for conducting risk assessments. This approach makes it possible to compare risks caused by security usability with other security risks. As a result it is possible to quantify the trade-off between theoretical security and practical security, which in turn can be used to determine a better balance between the two.

We have pointed out that suitable controls can be identified either through a sustaining approach or a disruptive approach. In case the sustaining approach fails, it will be required to invest in technological innovation specifically targeted at developing new security technology with better potential for usability.

References

[1] M. AlZomai, B. AlFayyadh, A. Jøsang, and A. McCullag. An Exprimental Investigation of the Usability of Transaction Authorization in Online Bank Security Systems. In *The Proceedings of the Australasian Informatin Security Conference (AISC2008) (to appear)*, Wollongong, Australia, January 2008.

[2] C. Brodie et al. Usable Security and Privacy: A Case Study of Developing Privacy Management Tools. In *Proceedings of the Symposium On Usable Privacy and Security (SOUPS2005)*, 2005.

[3] L. Cranor, S. Egelman, J. Hong, and Z. Y. Phinding Phish: An Evaluation of Anti-Phishing Toolbars. Technical Report CMU-CyLab-06-018, Carnegie Mellon University CyLab, 13 November 2006.

[4] P. Dourish and D. Redmiles. An Approach to Usable Security Based on Event Monitoring and Visualization. In *Proceedings of the New Security Paradigms Workshop*, pages 75–81. ACM Press, 2002.

[5] I. Flechais, C. Mascolo, and M. Sasse. Integrating Security and Usability into the Requirements and Design Process. In *Proceedings of the Second International Conference on Global E-Security*, 2006.

[6] A. Herzberg and A. Gbara. Protecting (even Naïve) Web Users from Spoofing and Phishing Attacks. Technical Report 2004/155, Cryptology ePrint Archive, 2004.

[7] ISO. *IS 7498-2. Basic Reference Model For Open Systems Interconnection - Part 2: Security Architecture*. International Organisation for Standardization, 1988.

[8] ISO. *ISO/IEC 27001:2006 - Information technology – Security Techniques – Information security management systems – Requirements*. ISO/IEC, 2006.

[9] A. Jøsang, M. AlZomai, and S. Suriadi. Usability and Privacy in Identity Management Architectures. In *The Proceedings of the Australasian Information Security Workshop (AISW), CRPIT Volume 68*, Ballarat, Australia, January 2007.

[10] A. Jøsang, P. Møllerud, and E. Cheung. Web Security: The Emperors New Armour. In *The Proceedings of the European Conference on Information Systems (ECIS2001)*, Bled, Slovenia, June 2001.

[11] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX(38):5–38 (January) and 161–191 (February), 1883. Available at F. Petitcola's Website: http://www.petitcolas.net/fabien/kerckhoffs/.

[12] K. Mitnick and W. Simon. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons, 2002.

[13] S. Ross. Security through Usability. *Securius Newsletter*, 4(1), 2003.

[14] M. Sasse. Computer Security: Anatomy of a Usability Disaster, and a Plan for Recovery. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI2003), (Workshop on Human-Computer Interaction and Security Systems)*, 2003.

[15] M. A. Sasse and I. Flechais. Usable security: What is it? How do we get it? In L. Cranor and S. Garfinkel, editors, *Security and Usability: Designing Secure Systems that People Can Use*. O'Reilly, 2005.

[16] B. Schneier. *Secrets and Lies*. John Wiley & Sons, 2000.

[17] D. Smetters and R. Grinter. Moving from the Design of Usable Security Technologies to the Design of Useful Secure Applications. In *Proceedings of the New Security Paradigms Workshop*, pages 82–89. ACM Press, 2002.

[18] G. Stoneburner, A. Goguen, and A. Feringa. Risk Management Guide for Information Technology Systems – NIST Special Publication 800-30. Technical report, National Institute of Standards and Technology, 2002.

[19] A. Whitten and J. Tygar. Usability of Security: A Case Study. Computer Science Technical Report CMU-CS-98-155, Carnegie Mellon University, 1998.

[20] A. Whitten and J. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th USENIX Security Symposium*, Washington, D.C., August 1999.

[21] M. Zurko and R. Simon. User-Centered Security. In C. Meadows, editor, *Proc. of the 1996 New Security Paradigms Workshop*. ACM, 1996.