Assessing the Usability of Personal Internet Security Tools

T.Ibrahim[1], S.M.Furnell[1,2], M.Papadaki[1] and N.L.Clarke[1,2]
[1]Centre for Information Security & Network Research, University of Plymouth, Plymouth, UK
[2]School of Computer and Information Science, Edith Cowan University, Perth, Western Australia
cisnr@plymouth.ac.uk

Abstract: The popularity of the Internet and all the services it provides has driven the demand for computers in the home. Unfortunately, these home users typically represent a group of users who are generally poorly educated about the dangers and threats that exist when connected to the Internet. To this end, security vendors have provided a variety of integrated security solutions that provide Anti-Virus, Firewalls and Intrusion Detection Systems to enable home users to become better protected. However, the need to rely upon users to make decisions about potential threats they have little or no information about is concerning at best. An analysis of user interfaces that relate to security have shown they frequently lack in providing usable interfaces that users are able to make informed decisions from. The aim of the paper is to support these home users by proposing a set of novel design criteria to enable the development of usable security alerts which are triggered by home security mechanisms. Drawing from literature, the criteria that are proposed take into account the unique usability issues that exist when dealing with information security: explicit and useful information, the ability to make a timely response and a consistent presentation of information. A walkthrough using a potentially problematic dialog from Norton 360 is used as a case study to highlight the current issues with the interfaces and to evaluate the proposed criteria. The findings of the evaluation reveal that the novel criteria are promising and the assessment of other security tools are required to make consistent and valuable recommendations.

Keywords

Security, Usability, Human Computer Interaction, Intrusion Detection Systems, Home Users, Norton 360

1. Introduction

It is widely recognised that end-users encounter usability problems while performing their normal computer tasks. Frequently, these problems are not in performing the primary intended tasks, but relate to alerts and warning messages triggered by other software, such as security tools. Arguably some novice users will get annoyed, particularly in the case when the system is bombarding alerts at the them; which causes them to subsequently decide to uninstall the security software after a short time (i.e. hours or days) leaving them insecure. A significant inconvenience to the user is the inability to make an informed decision, with factors such as, lack of security knowledge and poor interface design hindering the decision making process. This can result in them often guessing as to whether to allow or deny a particular alert or action. This problem is exasperated because security notifications rarely form part of the primary activity the user is engaging with on the system and are therefore merely considered an inconvenience.

The ability to understand the alert notifications that many modern security applications use is no simple task. Prior research looking into what issues exist for commercial Intrusion Detection Systems identified skilled staff as a key element to an effective system (Ibrahim et al. 2008). Obviously, however, the idea of skilled staff within a home user context is simply not feasible. Therefore, it is imperative that security tools for home users must interface with the home user in such a manner to provide sufficient information for the user to make an informed decision in a timely manner but at the same time provide an interface that is friendly and usable. The purpose of this paper is to enhance the home user experience and provide the ability to deal with the security alerts effectively by proposing novel usability design criteria.

The paper is organized as follows: Section 2 presents the established research into usability and Human Computer Interaction (HCI), specially focused upon security aspects, before presenting and explaining the proposed HCI-Security (HCI-S) criteria. Section 4 applies the aforementioned criteria to a real security alert triggered by well-known security software, Norton 360 and analyses its effectiveness. Finally, Section 5 presents conclusions about the findings and future directions of the work.

2. Usability Criteria for End-User Security Tools

This section focuses upon the related research including security criteria for designing a usable graphical user interface (GUI). Many studies have been completed in the field of (HCI). Jacob Nielsen developed ten usability criteria which many subsequent studies have used as a basis of their work (Nielson, 1994; Nielsen, 2005). Shneiderman and Plaisant (2005) presented a refined version of eight usability criteria, based upon the authors' experience over more than two decades. For our purposes, the limitation of both these studies is that they are general usability criteria and the authors did not consider the impact of security in their design. Chiasson et al. (2006), Chiasson et al. (2007), Garfinkel (2005), Johnston et al, (2003), Whitten and Tygar (1999), Yee (2002) and Zhou et al. (2004), have all presented alternative guidelines that consider security. Chiasson et al. (2007) in particular propose a set of design guidelines for designing security management interfaces. Whilst the study looks to design them with respect to administrators they can be usefully applied to home-users. Another example is the HCI-S criteria proposed by Johnston et al. (2003) in which the authors kept the *Visibility of the System Status* criterion from (Nielsen, 2005) and appended a new criterion entitled *Convey Features* (which shows users the availability of security features in the system, whereas the 'visibility' of features refers to their current status). Herzog and Shahmehri (2007) proposed more sophisticated guidelines for applications that set a security policy. The authors are interested in the limitation of some current security policies and the difficulty that novice users encounter when using it; especially for the first time.

Based upon the prior literature, the following 16 guidelines were developed:

1- Interfaces Design Matches User's Mental Model
The designer of alert interfaces should attempt to think as home-users to develop alert interfaces matches the users mental model. Initially, the user who receives a security alert will need to know the name of the security tool which triggered that alert. The user also needs to know how to respond correctly to that alert as fast as possible. Finally, the user who failed to respond or/and could not understand the response options, will need more help. In summary, the main interface of the alert should consist of four sectors: the alert detector sector, the alert description sector, the alert response sector and the alert support sector.

2- Aesthetic and Minimalist Design
Irrelevant or rarely needed information should not be displayed in the security alert. The alert interface design should determine the cause of the alert and impose the available response options to support the user to respond effectively. Bombarding the user with a lot of information might distract the user and force him to react randomly, just to return back to the indented primary task. Some alert interfaces manage to have a minimalist design but they do not have an aesthetic design (i.e. as will appear in the section 3).

3- Visibility of the Alert Detector Name
The appearance of the security tool name, which triggers the alert, is useful, specially, with the existence of more than one installed security tool on the home-use machine. This feature might guide the user to adjust the security settings of this particular tool. The reader should notice that the current criterion is not the same as the *Visibility of System Status* (Nielsen, 2005) criterion but perhaps a subset of it.

4- Establish Standard Colours to Attract User Attention
Users are most often attracted by the use of colours in the interfaces. Therefore, it is very important to focus on the use of colours as a major usability criterion. In general, the use of red and yellow colours in security alert interfaces are fairly standard, for example, the red colour informs the user that the alert severity is high; while the (orange or yellow) colour informs the user that severity of the alert is low. Moreover, we can consider this criterion as a subset of the *Visibility of the System Status* (Nielsen, 2005) criterion.

5- Use Icons as Visual Indicators
Users are most often affected by the use of pictures and icons in the interfaces. Therefore, it is very important to utilise this human feature to enhance our criteria. Muñoz-Arteaga et al. (2008) usefully

utilised the image of the traffic light to declare the security situation. This also supports the previous criterion, *Establish Standard Colours to Attract User Attention*. Finally, we can describe the icon and the previous colour criteria together as an implementation of the *recognition* feature from *Recognition Rather than Recall* guidelines (Nielsen, 2005).

6- Explicit Words to Classify the Security Risk level

The use of informative colours and icons, in the security alerts, to inform the user of the security risk level, as demonstrated in the previous two criteria, is excellent but not arguably enough. The user requires written confirmation of the security risk level and that information must be obvious in the main alert interface, not hidden in a secondary interface.

7- Consistent Meaningful Vocabulary and Terminology

The alert sentence(s) should be simple, short and informative and the words used in these sentence(s) should be familiar to the user. It is recommended that security terms that some users might be not aware of, such as the term *phishing attack,* should be avoided. Moreover, if possible, it would be better that each alert sector consist only of one sentence. However, the current criterion includes the main features of the Neilson criteria *Match Between System and the Real World*, *Consistency and Standards* and *Aesthetic and Minimalist Design.*

8- Consistent Controls and Placement

Users need to be able to find the security features they need in an appropriate location and in a reasonable time. Buttons are one of the most common user controls that are provided in interfaces. Unfortunately, in some security tools the appearance of these buttons reflects the existence of a poor design, at least from a usability perspective. For example, *Allow* and *Block* buttons exists in some security alerts without providing the user with any clue about the impact of this selection (i.e. the allowance or the blocking might be permanent or temporary). Therefore, this sort of information should be designed explicitly in the screen to give the user more control and freedom.

9- Learnability, Flexibility and Efficiency of Use

The security alert should be flexible and efficient to use, and enhance the user ability to learn the required security basics. The current criterion stresses on the use of explanatory tooltips for concepts or/and security terms which appears in the alert window to enhance the system flexibility, while providing links to access a built-in library or/and an Internet web page, in some other cases to increase the system efficiency.

10- Take Advantage of Previous Security Decisions

This criterion consists of two parts as follows:

- The home user alert history: only the user's previous experience with the alert: The user deserves to obtain information about the triggered alert. This information reports whether this type of alert has occurred before or not, and how the user previously reacted to it. The use of simple statistics which summarize this information will also be very helpful for the user in the decision making process. Moreover, these statistics should also be available to the user to give them the chance to investigate later, to evaluate the effect of his decision.

- Social feedback: other home-users previous experience with the alert: Develop a process by where users are able to benefit from other users' experiences. For instance, a security software database could receive reports of the user responses for every alert generated in the home user's machines. All users should have access to that database as soon as one of these alerts is triggered in the user machine. The existence of the criterion increases the home-user *learnability,* one of Johnston et al. (2003) HCI-S criteria. Moreover, the criterion is an enhancement of (Nielsen, 2005) *Help Users Recognize, Diagnose, and Recover from Errors* criterion.

11- Online Security Policy Configuration

The security tool designers should develop an efficient default configuration for the security policy. The aim of the criterion is in guiding the user to adjust the security settings to avoid, if possible, any conflict between the intended primary tasks and the security configuration (i.e. for instance, to avoid the triggering of frequently low level security alerts). It is anticipated that the current criterion would enhance (Johnston et al. 2003) HCI-S criterion *Convey Features.*

12- Confirm / Recover the Impact of User Decision
The security alert interfaces should be designed carefully to prevent home user errors. Sometimes, user errors are inevitable and vary from simple mistakes to dangerous errors, as follows:

- The user might press a button or click a link unintentionally by mistake.
- The user might respond randomly to the security alert and feels later that he made a mistake.
- The user decision might have an unanticipated impact on the configuration.
- The user decision might have a vital impact that seriously affects the security of the machine.

Therefore, the user should receive a confirmation message after performing any response which will affect the security of the system. The confirmation message should contain information about the possible impact of the decision. This facility gives the user the chance to recover the error, modify the response, extract a rough evaluation of the reaction and make a more informed decision.

13- Awareness of System Status all the Time
The user deserves to obtain a simple report declaring the state of the system as a result of the home user response to the alert. This report could be raised immediately after the user responds to the security alert or/and could be saved, where the user can access it after performing his intended task.

14- Help Provision and Remote Technical Support
The security alert should be designed to let the users be self-sufficient; however, some will still require further support. Tools should therefore provide built-in help and remote technical support. In this paper, term “help” means providing the user with extra information at the time of the alert and advice on an appropriate response. In practice, information in the accompanying help is not always sufficient to enable the user to respond correctly. Therefore, they can use the “remote technical support” facility as a final attempt to solve the security problem via support from the security vendor.

15- Offer Responses that Match User Expectations
Home-users usually make security decisions based upon factors such as the security alert feedback, the response options available, and their own hypothesis of the impact that the response would have. However, the *actual* impact of the available alert responses options does not always match the user’s expectation. Therefore, good alert design is not only what is required to obtain a secure system but also to ensure the user’s correct comprehension and understanding.

16- Trust and Satisfaction
Home-users typically trust the security tool on their computers until the occurrence of a performance failure. Unfortunately, the lack of understanding or/and the inability of some home-users to react correctly to some alerts can have a strong influence on the trust or/and satisfaction factors. In some cases, such events might lead them to improve their security knowledge (i.e. they still trust the security tool), but others might prefer to uninstall the software and thereby avoid further inconvenience.

Table 1 presents a comparison between the proposed criteria and some established usability guidelines (note: the guidelines are referenced via the names of lead authors listed in the References section, with a year added in cases where multiple papers from an author have been listed). The main purpose of this comparison is to demonstrate the real-world requirement to develop usability criteria specifically for security alerts. The findings suggest that our criteria have a role to play, in the sense that no individual example from the established guidelines covers the full range of issues.

Table 1: Comparing the proposed criteria against existing usability guidelines

	Proposed Criteria	Chiasson (2006)	Chiasson (2007)	Garfinkel	Herzog	Johnston	Nielson (2005)	Shneiderman	Whitten	Yee	Zhou
1	Design Interfaces Match User Mental Model	✓	✓	-	✓	✓	✓	✓	✓	✓	
2	Aesthetic and minimalist design	-	✓	-	✓	✓	✓	✓	-	✓	
3	Visibility of the Alert Detector Name	-	-	-	✓	-	✓	✓	-	-	
4	Establish standard colors to attract user attention	-	-	-	-	-	✓	✓	-	-	
5	Use icons as visual indicators	-	-	-	✓	✓	✓	✓	-	-	
6	Explicit Words to Classify the Security Risk level	-	-	-	✓	-	-	-	-	✓	
7	Consistent Meaningful Vocabulary and terminology	-	-	✓	-	✓	✓	✓	-	✓	✓
8	Consistent Controls and Placement	-	-	✓	-	-	-	✓	-	✓	
9	Learnability, Flexibility and Efficiency of Use	-	-	-	✓	✓	✓	✓	-	-	✓
10	Take Advantage of Previous Security Decisions	-	✓	-	-	-	-	-	-	-	
11	Online Security Policy Configuration	-	✓	✓	✓	-	-	-	-	✓	
12	Confirm / Recover the impact of User Decision	✓	✓	-	✓	-	✓	✓	✓	✓	
13	Awareness of System Status all the Time	✓	✓	-	-	✓	✓	✓	-	✓	✓
14	Help Provision and Remote Technical Support	-	-	-	-	✓	✓	-	-	-	✓
15	Offer Responses Match User Expectations	✓	✓	✓	-	-	-	✓	-	✓	
16	Trust and Satisfaction	✓	-	-	-	✓	-	✓	✓	-	

3. Assessing Alerts in Practice

This section presents a detailed assessment of a typical security alert, and a walkthrough of the process that a user might take in order to understand it. The example is taken from Norton 360; a package that is widely recognized and popular among end-users. The choice is not intended to imply that Norton 360's usability is worse than others in its class, and indeed it has actually scored highly on 'ease of use' in comparative evaluations (Which, 2009). Therefore, it is expected some of the limitations mentioned here might also exist in some other well-known products. Indeed, the Norton case represents one example from a wider study being undertaken by the authors, and is intended to be illustrative of the problems that can be encountered in practice rather than being presented as a significant finding in its own right.

The analysis presented here uses a simple alert that many users would have encountered. Having installed Mozilla Firefox and started the application for the fist time, an alert appeared, as illustrated in Figure 1. This is a trivial case compared to others that might occur, but is notable in that it may still confuse some users (particularly novices), and cause them to devote time to an event that actually would not cause any harm to their system.

Figure 1: A real example of Norton 360 security alert

The events and thought processes from this point are documented from the perspective of the user. The first comment is that the main interface provides no information about the cause of the alert and there are no explanatory tooltips (the cause was relatively obvious in this case, because the user had intentionally launched Firefox immediately beforehand, but other cases may be less clearcut). Arguably therefore, the main interface of the alert did not achieve the *Learnability, Flexibility and Efficiency of Use* criterion. Moreover, it is clear that the user's *mental model* was not completely considered during designing of this alert.

Assuming that the user decides to read the rest of the content (rather than investigating the *Help* and *Support* links), the alert wording is direct and simple, which satisfies our seventh criterion. The user can assume that the exclamation mark icon and the yellow colour indicate only a warning case, which increases assurance that there is no high risk. This confirms the importance our fourth and fifth criteria *Establish Standard Colours to Attract User Attention* and *Use Icons as Visual Indicators*, respectively. Nonetheless, the summary view of the alert did not mention explicitly, by words, the risk level status, which represents a design limitation, from the usability perspective.

At this stage, the user has a general idea about the alert and is presented with an explicit question, *"Should Norton 360 allow this access?"* (consequently managing to mention '*Norton 360'* for a third time in the same dialog, while other relevant information is missing). The user may assume that the *Show Details* link will give more guidance about how to respond, but this actually reveals more details about the cause of the alert (see Figure 2). This consequently reveals a minor conflict with the *Consistent Controls and Placement* criterion, as the link has been placed at a point in the dialog where the user is making a response rather than understanding the alert.

Looking at the consequence of selecting *Show Details* (Figure 2), it can be noted that all of the terms are mentioned without any further links. The user can now see the *Name* of the executable program that raised the alert, and the related *Path*. Moreover, further down the list, the user is given an explicit indication of the *Risk Level*. However, of the eight items listed, these are likely to be the only ones that will be meaningful to a wider audience. The inability to get any further description (e.g. via tooltips) will mean that many users are confused rather than informed by items such as the *Remote Url*, *Protocol* and *Direction*. No links in the *Show Details* interface is a remarkable limitation. In fact, even items such as the *Name* could merit further assistance. While the user might well be expected to recognise it in this example, other cases may not be so readily obvious and having a lookup to reference the names of known applications could be beneficial.

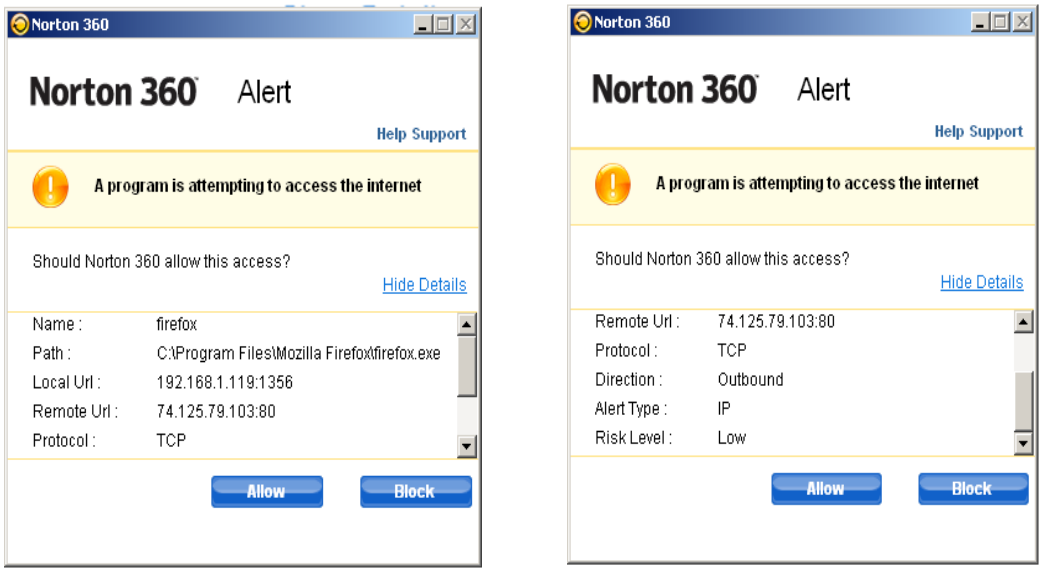

Figure 2: The expanded view of the alert, having selected the *Show Details* link

Let us assume the user felt stuck at this point, and still wanted to obtain more information about exactly what was causing the alert. The use of Norton 360 *Help* is shown in Figure 3. The user wrote the terms *Firefox* and *firefox.exe* separately in the *Index* but failed to provide any result. Next, the user wrote the same terms in the *Search* but he did not find any useful information.

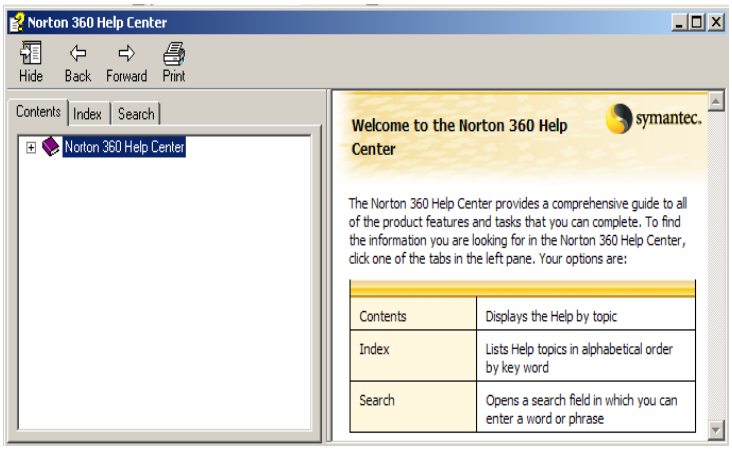

Figure 3: Norton 360 Help

Trying another route, the user may select the *Support* option from Figure 1. Selecting *Search Solution Library* yields the dialog shown on the right hand side of the Figure. Once again the user typed the term *Firefox*, the results focused upon the cause of the alert but only indicated Internet Explorer web browser and requested the user to check whether it is the default web browser or not. Hence, the user may assume that the cause of the alert was related to a default web browser issue, which is a computer setting rather than a security issue.

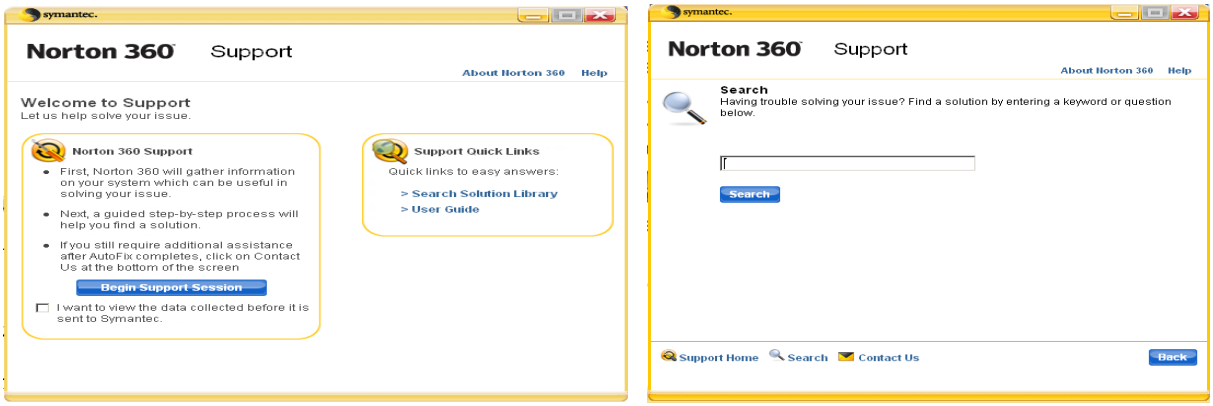

Figure 4: Norton 360 Support – main interface and search

From this point, the user only has one further line of investigation within the tool; namely to select the *Contact us* link shown at the bottom of Figure provide the user with three options to obtain Norton technical support; live chat, e-mail and phone calls, as shown in Figure 5. Although each of these are likely to yield a satisfactory result (especially in the case of this specific example), it seems a rather long way for the user to have to go in order to obtain a fairly baseline level of clarification.

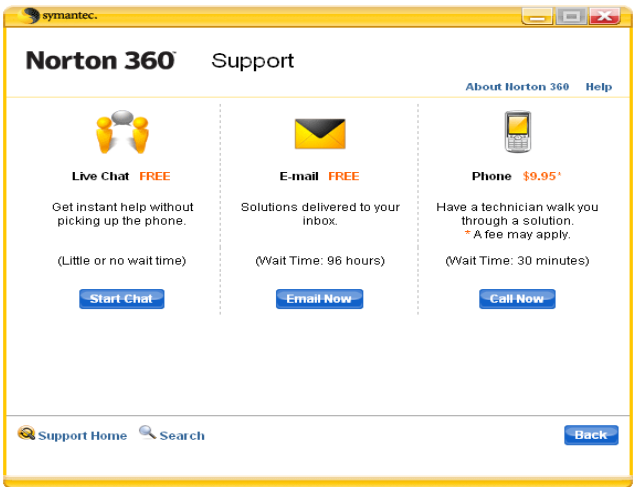

Figure 5: Norton 360 Contact us

The findings of this walkthrough suggest that some home-users who receive such alerts will require more help. The alert dialog provides three options which are *Help*, *Support* and *Show Details*. Unfortunately, they do not provide the user with the sort of information that might support a decision (for instance, there are no tooltips or links to more information). We applied our proposed criteria on this example and summarised the findings in Table 2.

Table 2: Evaluating a real Norton 360 security alert using the proposed criteria

No	Novel Criteria	Evaluation
1	Design Interfaces Match User Mental Model	No (the interface consists of the suggested four sectors but the contents does not match the user mental model)
2	Aesthetic and Minimalist Design	No (minimalist, but not aesthetic)
3	Visibility of the Alert Detector Name	Yes
4	Establish Standard Colours to Attract User Attention	Yes (e.g. Yellow = Low Risk Severity)
5	Use Icons as Visual Indicators	Yes (e.g. exclamation mark = Warning)
6	Explicit Words to Classify the Security Risk level	No
7	Consistent Meaningful Vocabulary and Terminology	Yes
8	Consistent Controls and Placement	No (no indication of whether the effects of selecting an option are permanent or temporary)
9	Learnability, Flexibility and Efficiency of Use	No (no tooltips or links to web sites)
10	Take Advantage of Previous Security Decisions	No
11	Online Security Policy Configuration	No
12	Confirm / Recover the Impact of User Decision	No
13	Awareness of System Status all the Time	No (Norton 360 provides only a general status for the whole system)
14	Help Provision and Remote Technical Support	No (“Help” is not useful & “Support” is time-consuming and sometimes costs money)
15	Offer Responses Match Expectations	No
16	Trust and Satisfaction	Low

As an example of the proposed criteria in use, Figure 6 represents the same alert with some simple modification. The design helps the user to follow the scenario of the alert from the top to the bottom without distracting him to look at every single location in the security interface all the time. The user will be able to scan the alert without the need to go backward and forward to be sure that he did not miss vital information. It is also worth mentioning that the alert was not overly serious in this example and the user was almost aware of what caused the alert. The user was not performing an important or an urgent task. He was therefore not panicked and had the opportunity to investigate and confirm what had caused the alert and how to respond to it. The reader can imagine how painful the case would be if the user receives an alert, has no basis to understand what triggered it and does not have the time to investigate it.

Figure 6: A simple modification on Norton 360 security alert

5. Conclusions and Future work

Home users require an efficient security tool to protect them. Unfortunately, the analysis performed in this study has illustrated that the interfaces provided by such tools are not always sufficient to enable users to make intelligent and informed decisions. The criteria developed in this paper are an attempt to rectify the problem; utilising existing HCI based design criteria and applying them specifically to the problem of security software. The Norton 360 example illustrates the nature of the problems that can be encountered, even in the case of a baseline, low risk alert.

Additional research will be undertaken to validate the proposed criteria, through focussing upon a number of security interfaces across the most common security tools. Using this evaluation, the criteria will be re-evaluated and subsequently applied to software to ensure they are appropriate and robust criteria to be utilised more widely within the security industry for designing systems.

7. References

Chiasson, S., van Oorschot, P. C. and Biddle, R. 2006. A Usability Study and Critique of Two Password Managers. Proceedings of the 15th conference on USENIX Security Symposium, Vancouver, Canada, 31 July – 4 August 2006.

Chiasson, S., van Oorschot, P. C. and Biddle, R. 2007. Even experts deserve usable security: Design guidelines for security management systems. Proceedings of Symposium on Usable Privacy and Security (SOUPS 07), Pittsburgh, PA, 18-20 July 2007.

Dhamija, R., Tygar, J. D. and Hearst, M. 2006. Why Phishing Works. In Proceedings of SIGCHI Conference on Human Factors in Computing Systems (CHI '06), Montreal, Canada, 22-27 April 2006, pp581-590.

Garfinkel, S. L. 2005. Design Principles and Patterns for Computer Systems That Are Simultaneously Secure and Usable. PhD thesis, Massachusetts Institute of Technology, May 2005.

Herzog, A. and Shahmehri, N. 2007. Usable set-up of runtime security policies. Proceedings of the International Symposium on Human Aspects of Information Security and Assurance (HAISA 2007), Plymouth, United Kingdom, 10 July 2007, pp99-113.

Ibrahim, T., Furnell, S. M., Papadaki, M. and Clarke, N. L. 2008. Assessing the challenges of Intrusion Detection Systems. Proceedings of the 7th Annual Security Conference. Las Vegas, USA. 2-3 June 2008.

Johnston, J., Eloff, J. H. P. and Labuschagne, L. 2003. Security and human computer interfaces. Computers & Security, 22, (8) pp675-684.

Muñoz-Arteaga, J., González, R. M. and Vanderdonckt, J. 2008. A Classification of Security Feedback Design Patterns for Interactive Web Applications. The Third International Conference on Internet Monitoring and Protection, 29 June– 5 July 2008, pp166-171.

Nielsen, J. 1994. Enhancing the explanatory power of usability heuristics. Proceedings of ACM CHI'94 Conference. Boston, Massachusetts, USA. 24-28 April, pp152-158.

Nielsen, J. 2005. Ten usability heuristics. Available online at: http://www.useit.com/papers/heuristic/heuristic_list.html (Accessed: 14/12/2008)

Shneiderman, B. and Plaisant, C. 2005. Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th edition), Addison Wesley.

Which. 2009. Security software. *Which? Computing*, January 2009, pp44-49.

Whitten, A. and Tygar, J. D. 1999. Why Johnny can't encrypt: A usability evaluation of PGP 5.0. In Proceedings of the 8th USENIX Security Symposium, Washington, D.C., USA, 23–26 August 1999

Yee, K-P. 2002. User interaction design for secure systems. In Proceedings of the International Conference on Information and Communications Security (ICICS'02), Lecture Notes In Computer Science, Vol. 2513, Springer-Verlag Berlin Heidleberg. pp278–290.

Zhou, A. T., Blustein, J. and Zincir-Heywood, N. 2004. Improving intrusion detection systems through heuristic evaluation. 17th Annual Canadian Conference of Electrical and Computer Engineering (CCECE 2004), Niagara Falls, Canada, 2-5 May 2004, pp1641-1644.