Loading ...
A Short Study into Security Delay Frustration

Alapan Arnab and Dave Nunez

CS06-03-00
September 29, 2006

Data Network Architecture Group and Collaborative Visual Computing Group
Department of Computer Science
University of Cape Town
Private Bag, RONDEBOSCH
7701 South Africa
e-mail: {aarnab, dnunez}@cs.uct.ac.za

Abstract

This report discusses a small experiment to investigate the amount of time users are willing to wait for applications to load, before they become frustrated. In particular, the experiment investigates whether, users are willing to wait longer, if they are aware that a security operation needs to be performed before loading the application. Our results show that users are willing to wait, on average 5.5 seconds longer for applications to load, if they are aware of a security operation.

1 INTRODUCTION

It is well known fact that security measures in computer programs (or in fact, for any other aspect of life) carry an overhead cost in performance. Most research exploring the performance degradation due to security, in both new and existing security programs and systems, focus on the effect on clock cycles.

However, with the increase in high speed desktop and server processors, there has to be sizable clock cycle degradation to have any noticeable effect for the end user. Furthermore, because security measures are often applied when loading documents or starting an application, the performance degradation is often infrequent and discrete in nature.

Thus, for many security applications and systems, it is more crucial to know the *wall time* performance degradation. Wall time performance measures the performance of a system or application in terms of the total time the application or system took to execute a certain task [1]. It is a concept used frequently in high performance computing, where performance improvements or degradation has only meaning if it has noticeable impact on the total time taken (often in the realm of minutes or even hours).

Wall time performance degradation measures the performance degradation from the user's perspective, but raises a difficult question: "What is an acceptable wall time performance degradation due to security?". This paper describes an experiment that aimed to investigate:

1. What is the average time, a user is willing to wait while loading an application?
2. Are users willing to wait longer if they know that there is a security operation that needs to be performed?
3. If users are willing to wait, on average, how much longer?

2 EXPERIMENT DESCRIPTION

We designed a simple experiment, taking a maximum of two minutes of the participant's time. Because of the short time required, we decided to use PDAs to run the experiment, and recruit participants on the street or from large gatherings. We used HP iPAQ Pocket PC PDAs running Windows Mobile 2003. The application itself was written in Java[1], and we used the Mysaifu JVM on the Pocket PC.

The aim for the participants was to navigate the interface, answering demographic questions (gender, age group and computer experience), and following instructions to load a photograph. The photograph was pre-loaded in the memory, and the participant was told before beginning that the photograph "*should load instantaneously*". Our application, created four experiment scenarios, which we relayed to the participant:

1. The photograph will load instantaneously. (Users not warned of delay, users not warned of encryption)
2. There could be a short delay before loading the photograph. (Users warned of delay, users not warned of encryption)
3. The photograph is encrypted (Users not warned of delay, users warned of encryption)
4. The photograph is encrypted, and there could be a short delay before loading the photograph. (Users warned of delay, users warned of encryption)

The scenario was chosen at random, and we did not have any control on which scenario was selected. Once the participant was informed of the scenario, (s)he was taken to a loading screen, which informed her/him "*Loading*

[1]We initially envisaged the experiment as a Java applet.

image. Press refresh if it takes too long". A large refresh button was placed at the bottom of the screen, and the aim of the experiment was to measure how long users waited before pressing the button. Once they pressed the refresh button, the photograph was displayed. Screenshots of the experiment application is shown in Figure 1.

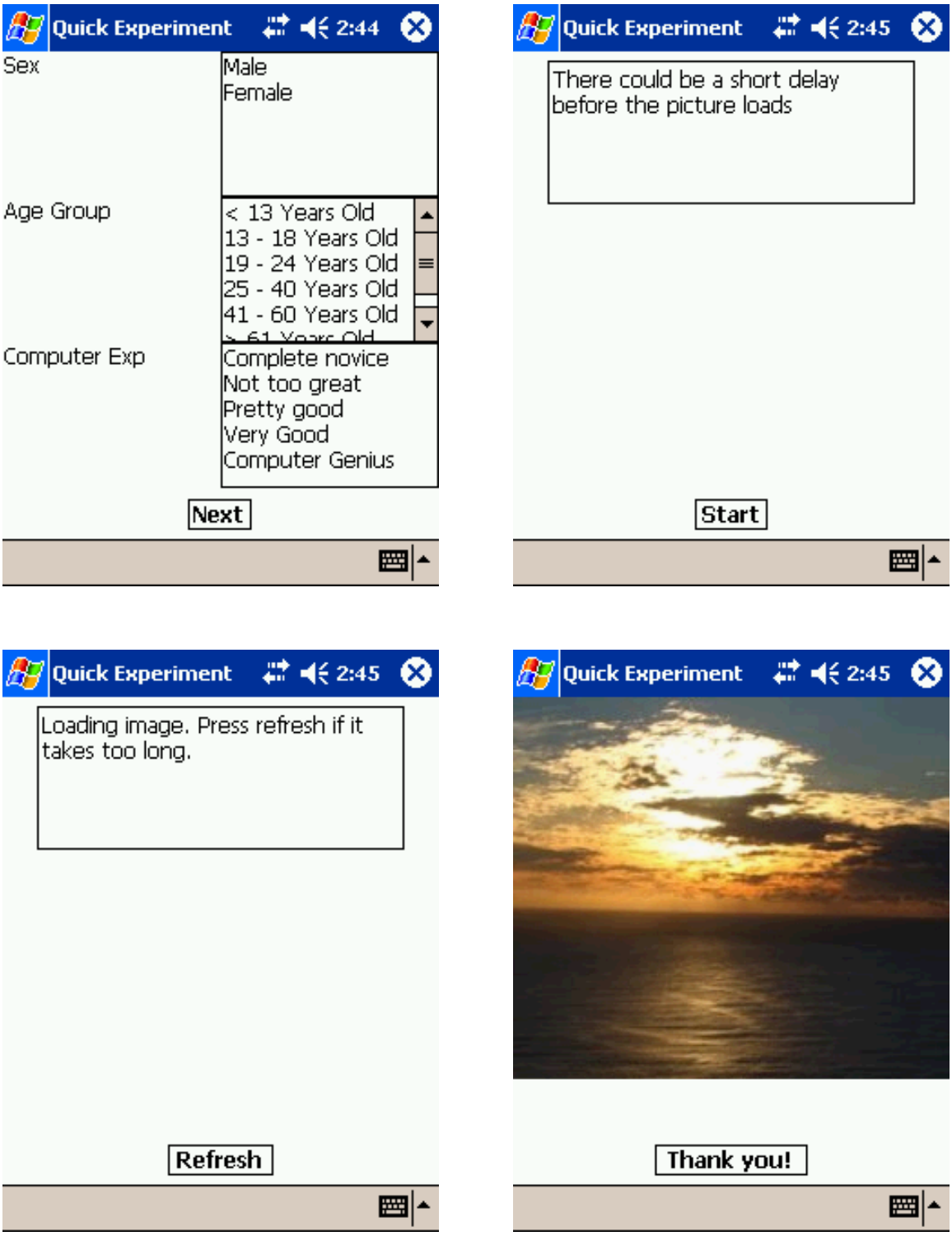

Figure 1: Screenshots of the Experiment (Screen 1: Top Left; Screen 2: Top Right; Screen 3: Bottom Left; Screen 4: Bottom Right)

3 RESULTS

We managed to recruit 119 particpants for the experiment, largely from the student population during lunch time and other student gatherings. We managed to get a 55%:43% split between our male and female respondents (with 2% of our respondents not answering the question). The summary of the results with respect to age are shown in Table 1. The Student t-test analysis shows that the gender has a significant effect on waiting time ($t(115) = 1.99$, $p < 0.048$), with women willing to wait longer (on average 4 seconds) than men.

Gender	No. Responses	Mean (ms)	Std Dev (ms)
Male	66	11075.76	8284.02
Female	51	15000.00	12921.30

Table 1: Summary of results with respect to gender

Because the majority of the respondents were students, most were in the 19 – 24 age bracket. The computer experience of the respondents varied; with most participants rating themselves as "Pretty Good" (42%), followed 25% of them rating themselves as "Not too great", 18% as "Very Good", 12% as "Computer Genius" and the remainder (3%) as "Complete Novice". Correlations were performed to test the effects of computer experience and age on waiting time. Neither age ($r = 0.04$, $p > 0.05$) nor computer experience ($r = -0.18$, $p > 0.05$) were significantly related to waiting time.

Experiment Scenario	No. Responses	Mean (ms)	Std Dev (ms)
No Delay, No Encryption	39	9794.87	7259.04
Delay, No Encryption	23	12652.17	8477.83
No Delay, Encryption	26	15884.62	15497.94
Delay, Encryption	31	15032.26	11680.42

Table 2: Summary of results with respect to experiment scenario

Table 2 summarieses the results according to the experiment scenario. We analysed the above data using a 2x2 analysis of variance (ANOVA), using encryption and delay as the factors. The analysis showed only a main effect for encrypt ($F(1, 115) = 4.31$, $p < 0.04$), but not for delay or the interaction of delay and encrypt. The effect on encrypt was in the expected direction; that is, subjects who were warned that the data need to be decrypted first waited longer (for an average difference of approximately 5.5 seconds).

4 CONCLUSION

Our experiment also shows that users are willing to wait, on average 9.8 seconds before becoming frustrated with the delay. Furthermore, our experiment clearly shows that users are willing to wait longer, one average by 5.5 seconds more) if they know that the delay is due to a security operation. The large variances in responses however shows that the average of 9.8 seconds as the baseline for delay may not be appropriate for all users, and could vary with applications in question.

5 ACKNOWLEDGEMENTS

We would like to thank Jakkaphan Tangkuampien for arranging the PDAs and Paolo Pileggi for helping to collect responses.

This work is partially supported through grants from the University of Cape Town (UCT) Council and the

National Research Foundation (NRF) of South Africa. Any opinions, findings, and conclusions or recommendations expressed in this paper/report are those of the author(s) and do not necessarily reflect the views of UCT, the NRF or the trustees of the UCT Council.

REFERENCES

[1] DICTIONARY, T. F. Wall clock time definition.
URL: http://computing-dictionary.thefreedictionary.com/wall+clock+time.