An experimental microworld for evaluating the tradeoffs between usability and security

[Workshop Paper]

Noam Ben-Asher
Department of Industrial Engineering
Ben Gurion University
Beer Sheva, Israel
noambena@bgu.ac.il

Joachim Meyer
Department of Industrial Engineering
Ben Gurion University
Beer Sheva, Israel
joachim@bgu.ac.il

Yisrael Parmet
Department of Industrial Engineering
Ben Gurion University
Beer Sheva, Israel
iparmet@bgu.ac.il

Sebastian Moeller
Quality and Usability Lab
Deutsche Telekom Labs
TU Berlin, Germany
sebastian.moeller@telekom.de

Roman Englert
Deutsche Telekom Lab
Ben Gurion University
Beer Sheva, Israel
roman.englert@telekom.de

ABSTRACT

Security mechanisms may require users to deal with the tradeoff between risky and efficient or safer yet less efficient use of a production system. We present an experimental system (microworld), based on the Tetris game, that can serve as a research tool for studying behavior regarding the usability and security tradeoff. This paper describes the system's main components and the variables the experimenter can manipulate. Also detailed here are the data collection process and the analysis methods. The collected data from the microworld allow researchers to explore and model usability and security tradeoffs in the context of user interaction with security systems and psychological acceptability.

Categories and Subject Descriptors

H.1.2 [**Models and principles**]: User/Machine Systems—*Human factors, Software psychology*; I.6 [**Simulation and modeling**]: Simulation Support Systems—*Environments*

General Terms

Experimentation, Security, Usability, Human Factors

Keywords

Human factors, uability, security, experimental system, microworld, alerts

1. INTRODUCTION

Information security is intended to decrease the likelihood that security related threats will realize and the severity of the consequences if they do. There are many security mechanisms, algorithms and dedicated tools, ranging from firewalls, anti-virus and spam filtering to data encryption tools, network monitoring and access control systems. However, these means are only effective when they are configured and used correctly [10]. Eventually, there are cases in which security tasks cannot be completely automated and inevitably the involvement of a human user is required [2]. A generalization of such a scenario is a situation in which a security related communication (e.g. an alert or warning message) is sent to the user. The user then faces two different tasks - progressing towards the primary goal, i.e. executing a production task, while dealing with the security issue, which may interrupt the workflow [8].

The conflict is often the decision between accepting and ignoring the alert from a security system, for example, not accessing a web page with a questionable certificate or starting a virus scan that will dramatically slow the computer. Thus, if the alert is followed, the user may not be able to do everything he or she intended to do, or may have to do it less efficiently. Alternatively, ignoring the alert can possibly cause severe damage. At such a point the user is confronted with the security and usability tradeoff [9].

A major obstacle to the studying and modeling of the security and usability tradeoff is the unavailability of real up-to-date data on security systems settings and users' interaction with them (e.g. responses to alerts and security indicators). If made freely available, such logs from work and personal environments, security policies and other "behavioral references" can perhaps be exploited by attackers to identify vulnerabilities [5].

To overcome this problem we created a novel experimental system which provides a controlled research environment for studying users' actions regarding the usability and security tradeoff during a prolonged interaction. This microworld is a flexible experimental platform, designed and

Symposium on Usable Privacy and Security (SOUPS) 2010, July 14–16, 2010, Redmond, WA USA
.

built for running experiments in various settings. It is a research tool for studying user interaction, and it provides data for both statistical analysis and modeling. Such controlled research environments (i.e. microworlds) can be particularly valuable when studying real time interaction and decision-making processes in dynamically changing environments. Microworlds can be suitable for laboratory experiments even in circumstance in which traditional research methods cannot be used [4].

2. THE EXPERIMENTAL SYSTEM

This experimental system was developed to study the factors which influence the interaction with security systems and lead to different security related behaviors. The need to deal with security-related tasks occurs while the user is engaged in an ongoing production task, and it can be triggered by security related communication. The experimental system employs a special version of the Tetris game. In this version the player can be attacked by a "virus" that randomly deletes part of the squares on the screen. Users receive indications about possible attacks from a security system (see Figures 1 and 2). We selected Tetris because it is a simple and popular game which requires no previous knowledge in computers or security and can be played by a wide variety of users. It imitates normal and prolonged computer usage with elements of fun. The Tetris game itself was already used in research on cognitive engineering (e.g. [7]). Performance in the game, e.g. the number of completed rows, can be easily translated into a monetary value, generating incentives for trying to minimize losses.

The changes in the game aimed to make the gains of the player susceptible to security-related events. Hence, unlike in the original Tetris game, completed rows are not removed automatically but stay visible and susceptible to threats until the player initiates a protective action which saves the gains in a safe place. This protective action is clicking the 'Clear Rows' button (see Figures 1 and 2). Security threats are viruses that delete Tetris bricks from the display. An attack can turn completed row to incomplete ones and thereby lessen the player's gains. Similar to real life situations, performing a protective security action and saving the gains is associated with usability costs which decrease productivity in the primary task. When saving the gains, the game is paused for a given period, interrupting the workflow and shortening the time available for playing.

2.1 Main Components

The experimental system consists of three main components, which each represent a different aspect of the microworld. The experimenter sets the attributes of the components and they operate independently during the experiment.

The first component, and the primary task for the user, is playing Tetris. As described above, the original game was slightly modified, allowing the user to collect multiple completed rows before saving them. To save gains from possible attacks the user has to click a button labeled 'Clear Rows'. This action pauses the game for a predetermined period of time. The duration of the pause is essentially the usability cost for using the security system with, for example, 5 seconds or 15 seconds pauses representing more usable versus less usable security systems. The experimenter also controls the gains based of the following function: $Gains = R * cP_R + kP_R^{\alpha R}$. Where R is the number of saved rows, P_R is the amount paid for one row and c, k, α are coefficients that allow linear and non-linear gain functions. Using the parameters in the gain function, it is possible, for example, to encourage the player to accumulate rows before clearing them by increasing the gain exponentially as the number of cleared rows increase. In this case the gains from clearing 2 completed rows at once can be larger than the sum of gains from twice clearing a single completed row. The actual values of the protected and unprotected gains are visible to the player throughout the game.

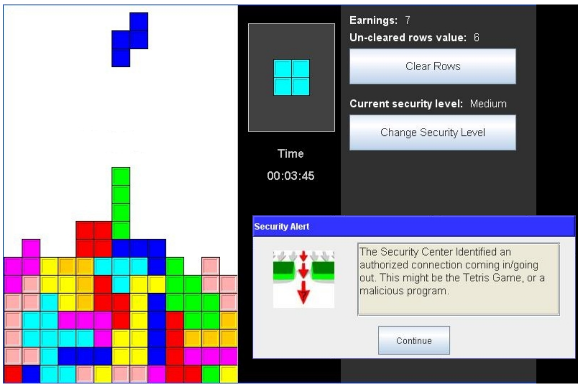
Earnings: 7
Un-cleared rows value: 6
Current security level: Medium
Time
00:03:45
Security Alert
The Security Center Identified an authorized connection coming in/going out. This might be the Tetris Game, or a malicious program.

Figure 1: Screen capture of the experimental system when an alert appears.

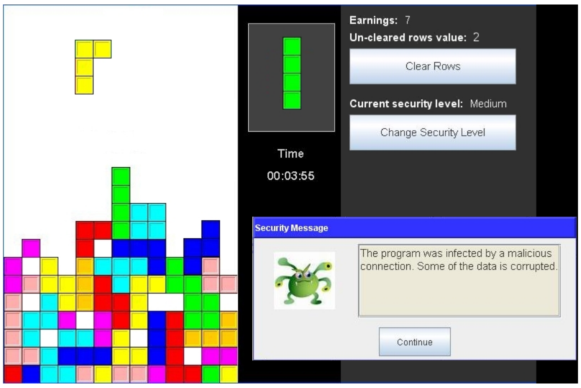
Earnings: 7
Un-cleared rows value: 2
Current security level: Medium
Time
00:03:55
Security Message
The program was infected by a malicious connection. Some of the data is corrupted.

Figure 2: Screen capture of the experimental system following an attack.

The experimenter controls the duration of the game and its level of difficulty. Increasing the velocity with which Tetris bricks descend from the top of the screen to the bottom makes the game more difficult and can increase the mental load and concentration required from the player. Such a manipulation makes it possible to control the workload in the primary task. Thus, we can evaluate how changes in workload and stress might affect tolerance to usability costs of a security action, responses to security related communication and risk taking.

The second main component of the system is the attack generator. In this microworld attacks occur at random times and are unrelated to the player's actions. During the game, every predetermined period the attack generator decides whether or not to initiate an attack. The experimenter has direct control over the frequency the attack generator

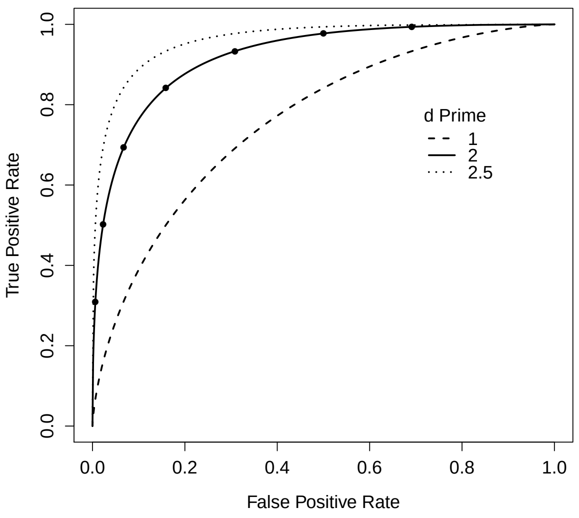

Figure 3: ROC curves for 3 security systems, differing in their sensitivity index (d').

is initiated, the likelihood attacks occur (P_{Attack}) and the delay between a True Positive (TP) alert and an attack (see Figure 4). This provides the ability to imitate the non-immediate correlation between a past security alert and a currently experienced security breach [9]. Additionally, the experimenter controls the damage caused when an attack is realized. The virus attack randomly deletes a given percent of the visible bricks on the screen, leading to the loss of unprotected gains.

The security system component provides the player with alerts on possible threats. This component operates based on signal detection theory (e.g. [6]), where attacks are designated as signals. The experimenter controls the quality index (d') of the security system in terms of correct and incorrect detections. On the other hand, the player interacts with the front-end of the security system at the beginning and during the game. The player adjusts the current security level of the system by selecting one out of seven possible levels, ranging from 'Very Low to 'Very High' security. Figure 3 depicts the characteristics of 3 possible security systems. Changing the d' value affects the ability of the system to distinguish between a signal and noise. Each point on a specific curve is a setting of the security system which can be selected by the player and change the rate of True Positive (TP) and False Positive (FP) alerts. A high security level will bring about many alerts, most of which will be false alarms, but there will be almost no missed detections. A low security level will lower the false-alarm rate, but will cause the system to miss some threats. At the beginning of the game, the player has to set the security level and can change it during the game, without costs, based on personal preferences and/or past experience with the system. The experimenter controls the available $P_{TruePositive}$ and $P_{TrueNegative}$ values of the security system which the player can select from. Figure 4 demonstrates the relations between the attack generator which operates autonomously and the security system which is adjusted by the player.

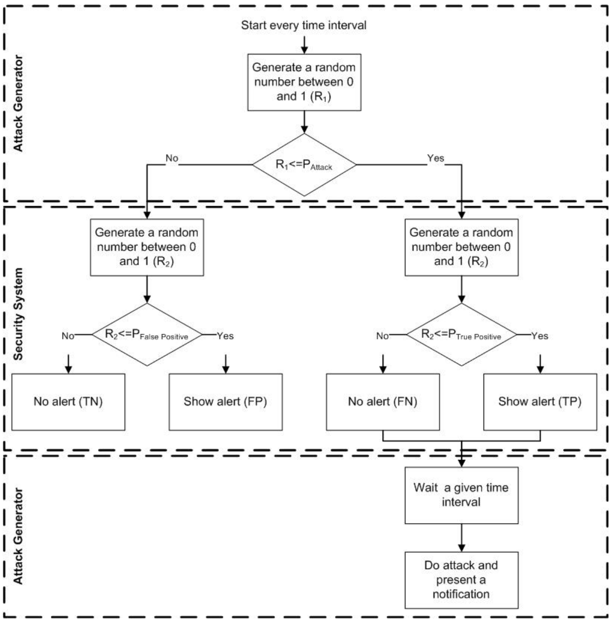
Attack Generator
Start every time interval
Generate a random number between 0 and 1 (R1)
No
R1<=PAttack
Yes
Security System
Generate a random number between 0 and 1 (R2)
Generate a random number between 0 and 1 (R2)
No
R2<=PFalse Positive
Yes
No
R2<=PTrue Positive
Yes
No alert (TN)
Show alert (FP)
No alert (FN)
Show alert (TP)
Attack Generator
Wait a given time interval
Do attack and present a notification

Figure 4: The flow of attacks and alerts generation.

The system provides the player with two types of indicators, alerts on a possible attack and a notification that an attack has occurred. It is possible to customize visual attributes (e.g. an icon) and the contents of the security indicators. During the period of time between an alert and the consecutive attack, the player has to decide whether to respond to the alert with a protective action or to dismiss it. Attributes of the experimental system that may bias this decisions are the cost of protecting the unsecured gains by saving the completed rows (i.e. the period for which the game will pause), the selected security level coupled with previously experienced reliability of the security system and the possible damage from an attack based on the amount of unprotected gains.

2.2 System Architecture

The experimental system is a client-server application. It uses a database server for data collection and it is capable of running simultaneously multiple experiments with large numbers of participants from various locations. The database also stores the configurations of the microworlds and their values. When the player starts the experiment, the client retrieves the values of the parameters controlled by the experimenter and operates independently from other clients which play at the same time.

Additionally, the experimental system includes two questionnaires. The first is administered before interacting with the system and provides an opportunity to collect general demographic data, as well as data on computer usage and knowledge in information security. A second questionnaire is administered after the user completes the game. Here subjective impressions from the system are collected, e.g. the perceived effectiveness of the security system or trust in the security alerts.

3. DATA COLLECTION

The experimental platform collects data in an event-driven manner. It documents the operations of all independent

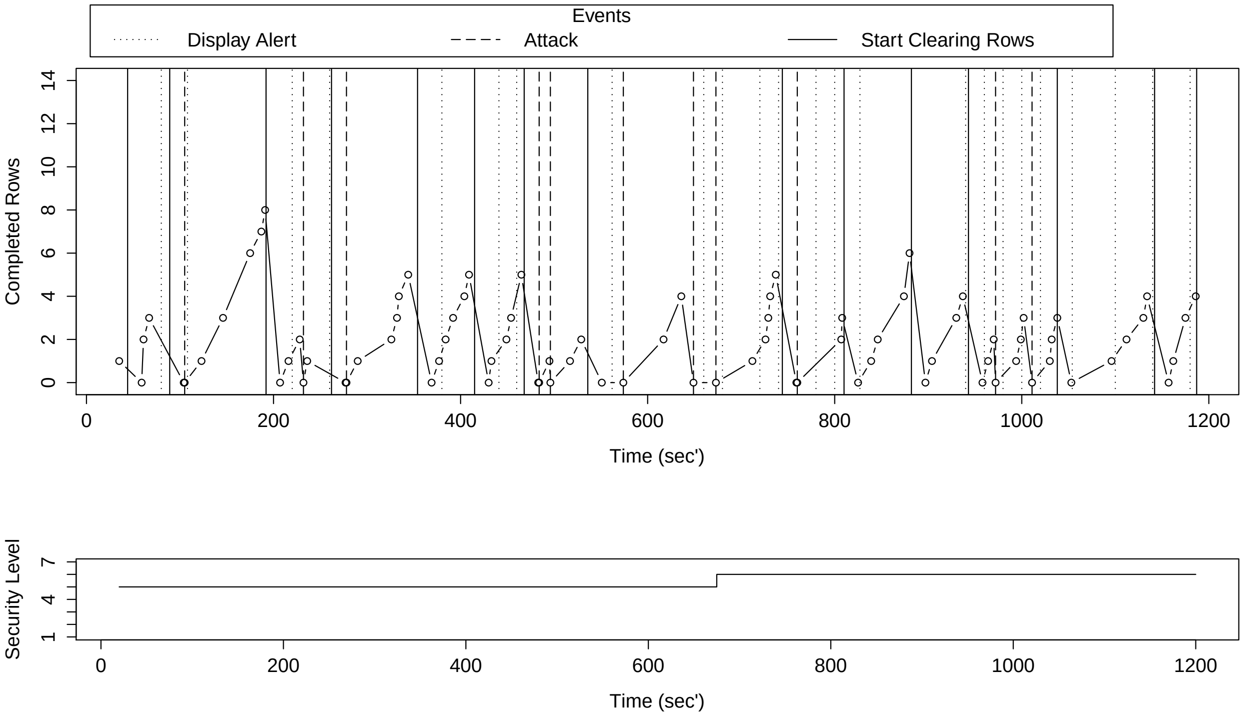

Figure 5: The actions of a player in a microworld with a high attack likelihood.

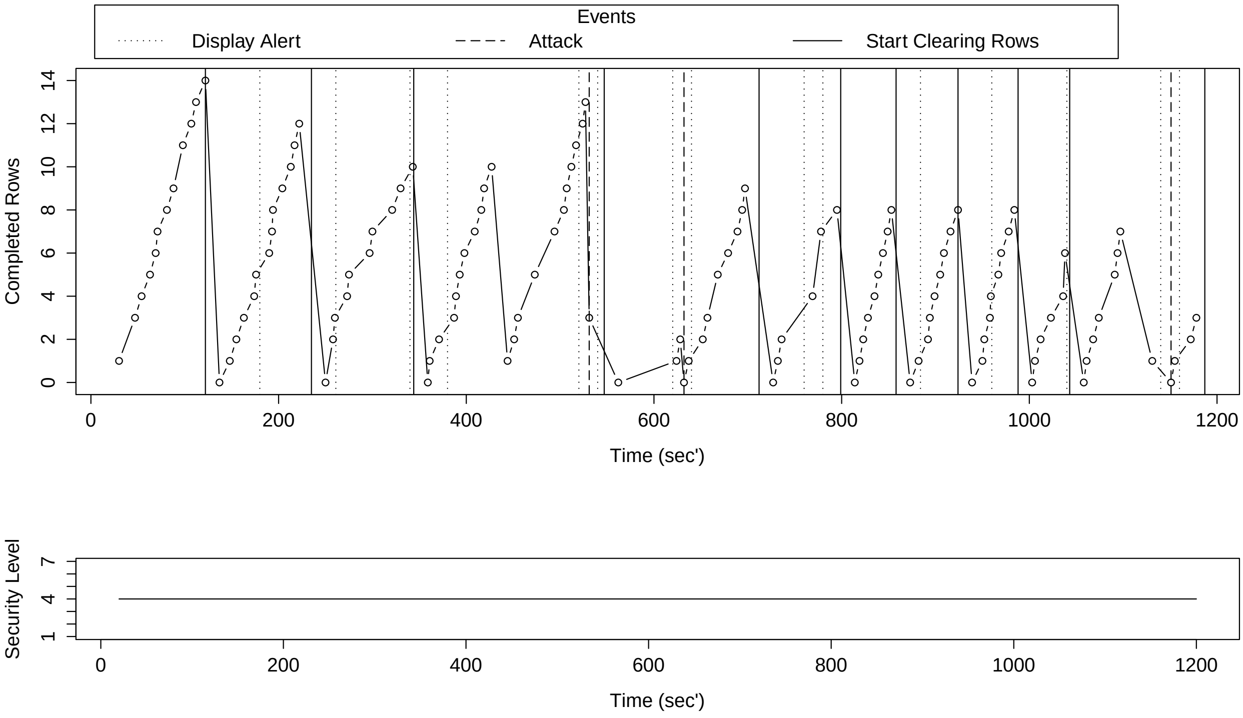

Figure 6: The actions of a player in a microworld with a low attack likelihood.

components (i.e. the attack generator and the security system), as well as the player's action and the status of the game. The main actions the system documents from the player's point of view are: (i) setting the initial security level of the system, (ii) progress in accumulating rows, (iii) saving gains from completed rows, and (iv) changes in the security settings during the game.

Figures 5 and 6 present the collected data on players' actions and responses to events generated by the microworld along a timeline. These are taken from an experiment where the between-subject condition was attack likelihood. One group of subjects (n=10) played in a microworld where the likelihood of being attack was high and thus experienced a high number of attacks. The other group (n=10) played in a microworld where the likelihood of being attacked was low. Each subject played with the same settings for 20 minutes on three consecutive days. The data used to plot the figures are from the third session where it is possible to assume that the players have already formed a strategy. As seen in Figure 5, a player in the high likelihood condition started the session by setting the security system to the 'Medium High' level and following 2 miss events he changed the security level to 'High'. The player in the low likelihood condition (Figure 6) used the default security level 'Medium' for the entire session and on average accumulated a high number of rows before saving them. His pattern of saving rows was almost independent from the alerts, while the player in the high likelihood condition tends to comply with the alerts.

4. DATA ANALYSIS

From the collected data, several measurements were extracted. Some are ratio-scale measurements which can be calculated for a player in a session or over sessions, such as the average time spent in a security level, the number of changes in the security level and the number of completed rows saved by the player. The analysis of these measures is relatively straightforward and is done using Analyses of Variance (ANOVAs) or linear regressions. It is also possible to calculate correlations between the player's responses in the questionnaires and these measures.

One main point of interest is the player's response to a security alert. Players have a "window of opportunity" to save their unprotected gains following an alert and before experiencing an attack. Hence, complying with an alert is a binary response variable. The probability of responding to an alert was analyzed using generalized linear models (GLM). The dependent variable is clearing rows following an alert (1) or ignoring it (0) and the independent variables can include selected attributes of the microworld (e.g. attack likelihood, usability costs, sensitivity of the security system) and player status (e.g. number of completed rows and selected security level). The resulting models can be relatively complex. Therefore, model reduction and selection was done based on Akaike Information Criterion (AIC) [1].

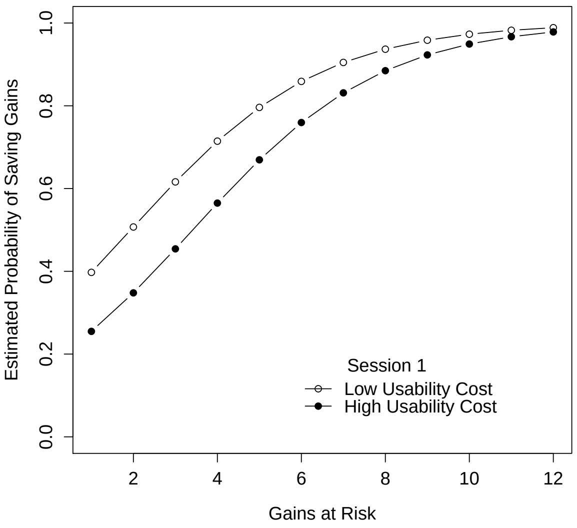

Figure 7: The probability of saving completed rows in the first session as a function of the number of rows and the duration for which the game pauses.

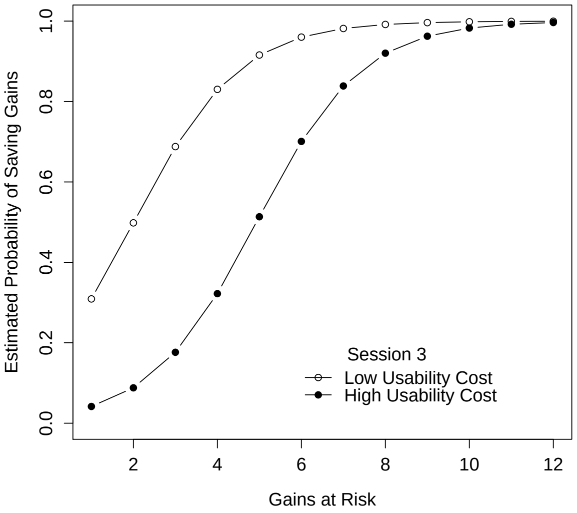

Figure 8: The probability of saving completed rows in the third session as a function of the number of rows and the duration for which the game pauses.

Two approaches enhance the validity of the analysis. The first is using generalized linear mixed models (GLMM), where the mixed random effect represents an individual player. Therefore, the GLMM analysis takes into account the individual differences between players. Such differences might originate from various characteristics of the player, ranging from risk aversion to the skillfulness in playing Tetris. Another interesting approach is to incorporate previous events into the model, i.e. the individual short history of the player within the microworld. The time since the last attack or alert and the damage experienced in the last attack were found to contribute significantly to the likelihood to perform a security action following an alert. They also improve the goodness of the statistical models based on AIC.

Figues 7 and 8 demonstrate how the compliance with alerts changes from the first to the third session as a function of the usability of the security system. This experiment included 40 participants and the usability costs were a between-subjects condition. In the low-costs condition (n=20) the game paused for 7 seconds, while in the high-costs condition (n=20) the game paused for 22 seconds when rows were cleared. The number of completed rows had a significant effect on the probability of clearing rows. However, in the first session no significant difference was found in the probability of protecting the gains between the two usability conditions. In the third session, after gaining experience with the game and the security system, a significant difference emerged between the two groups. High usability costs of the security action decreased the probability of players carrying it out, leading to more risk taking behavior.

5. LIMITATIONS AND FUTURE RESEARCH

So far we used the experimental system in three experiments with different experimental designs. The experiments provided valuable results and insights on how users interact with security systems and how they cope with the usability and security tradeoff. However, while running the experiments and analyzing the collected data, several issues appeared and some improvements were made in the system and the experimental protocol. While the relatively simple security system did not require any prior knowledge in security, the performance in the production task was affected by the participant's Tetris expertise. Although all participants were familiarized with the game and had some experience with it, the different levels of expertise lead to different security behaviors. Thus, the revised experimental protocol includes a separate session with no alerts or attacks that is used to estimate the player's ability in the production task.

Moreover, the security system did not provide full protection. As the player could save only completed rows, there were cases when an alert appeared but the player could not protect the nearly completed rows. This issue had also impact when trying to model user behavior using economical models. The virus deletes bricks randomly, and therefore, when receiving an alert, the player could only estimate the number of completed rows that will be damaged in the coming attack. We are now developing a new version of the system to address these issues.

There is evidence, though not from information security research, that compliance with alerts or warnings decreases as the costs in terms of effort, discomfort or time increases (e.g.[3]). These findings are consistent with the results obtained using the microworld. Validating the experimental results obtained with the microworld in real interactions with security systems will be one focus of our future research. Several directions are explored, including the use of questionnaires and long term user tracking.

6. CONCLUSIONS

The microworld seems to be a promising research tool for studying usability and security tradeoffs. It provides quantitative and qualitative data on how users interact in a world that includes a production task and a supporting security task. Even in relatively short experimental sessions, coping strategies were formed and different security behaviors were observed. Both researchers and developers can use the collected data to minimize the tradeoff between usability and security and to understand how users select to balance it under various conditions. Valid models of security behaviors can be used to develop and test usable security systems more efficiently.

7. ACKNOWLEDGMENTS

This research is partly funded by Deutsche Telekom AG as part of the Telekom Laboratories @ BGU activities.

8. REFERENCES

[1] K. Burnham and D. Anderson. *Model selection and multimodel inference: a practical information-theoretic approach.* Springer Verlag, 2002.

[2] L. Cranor. A framework for reasoning about the human in the loop. In *Proceedings of the 1st Conference on Usability, Psychology, and Security*, pages 1–15. USENIX Association, 2008.

[3] T. Dingus, S. Wreggit, and J. Hathaway. Warning variables affecting personal protective equipment use. *Safety Science*, 16(5-6):655–673, 1993.

[4] C. Gonzalez, P. Vanyukov, and M. Martin. The use of microworlds to study dynamic decision making. *Computers in Human Behavior*, 21(2):273–286, 2005.

[5] J. Gonzalez and A. Sawicka. A framework for human factors in information security. In *WSEAS International Conference on Information Security, Rio de Janeiro, Brazil*, 2002.

[6] D. Green and J. Swets. Signal detection theory and psychophysics. 1966.

[7] P. Maglio and D. Kirsh. Epistemic action increases with skill. In *Proceedings of the eighteenth annual conference of the cognitive science society*, volume 16, pages 391–396. Citeseer, 1996.

[8] A. Sasse. Usability and trust in information systems. *Cyber Trust & Crime Prevention Project. University College London*, 2004.

[9] R. West. The psychology of security. *Communications of the ACM*, 51(4):34–40, 2008.

[10] A. Whitten and J. Tygar. Usability of security: A case study. *School of Computing Science, Carnegie Mellon University, Technical Report CMU-CS-98-155*, 1998.