Implementation and Initial Evaluation of Game in Which Password Enhancement Factor is Embedded

Masahiro Fujita, Mako Yamada, and Masakatsu Nishigaki(✉)

Shizuoka University, Hamamatsu, Japan
nisigaki@inf.shizuoka.ac.jp

Abstract. Embedding a security factor into entertainment instance is an effective approach to improving users' security awareness and/or skills. As an attempt of this approach, we have proposed a password enhancement scheme which enables users to memorize stronger passwords while playing games [5]. In this succeeding paper, we implemented a game instance and evaluated the practical effectiveness of our scheme. We asked five subjects to play our game in their free time for three days. All subjects memorized at least thirteen random characters in this game. After the experiment, we asked subjects to answer a questionnaire. The users' answers suggested that they had positive impression of our game. These results showed the effectiveness of our scheme, i.e. users are able to memorize stronger passwords naturally while playing our game.

Keywords: Entertainment · User authentication · Games · Password

1 Introduction

Consideration of human factors such as cognitive characteristic is indispensable in designing information systems. In this line, researchers have tried to improve security of information systems by combining security with entertainment (e.g. [1–4]). Enjoying entertainment is one of human factors.

There could be two approaches for combining security with entertainment; (i) an entertainment factor is embedded in security technology and (ii) a security factor is embedded in entertainment instances. The aim with approach (i) is to improve security and/or usability of information systems when users use these systems. On the other hand, that of approach (ii) is to improve security skills and/or awareness of users in their daily life. These approaches have different advantages; thus, it is important to investigate both. However, to the best of our knowledge, most previous studies were focused on approach (i). This motivated us to examine approach (ii). In our previous study, as an attempt, we focused on to embed a password enhancement factor (a security factor) into games (entertainment instance) [5]. We designed a password enhancement scheme that enables users to naturally memorize strong passwords while playing games. In this succeeding paper, we implemented a game instance and evaluated the practical effectiveness of our scheme through an experiment.

C. Stephanidis (Ed.): HCII 2016 Posters, Part I, CCIS 617, pp. 476–481, 2016.
DOI: 10.1007/978-3-319-40548-3_79

2 Password Enhancement Through Games

As an attempt of approach (ii), we have proposed a password enhancement scheme that enables users to memorize stronger passwords while playing games [5].

Game players often memorize commands naturally while repeatedly inputting them into games. So, let us consider commands as passwords. This deduces that users can naturally memorize complex commands (passwords) while playing a game if a game system prepares complex commands and requires users to input them at the appropriate time. For example, a roll-playing game prepares a command for using a special attack. While playing the game, users repeatedly input the command for using the special attack. Users will naturally memorize the command in the end, and the command can be later on used as a login password on a Web site or a master password for a password manager.

The idea of "reward" also can encourage users to memorize much stronger passwords. Reward is a factor that helps users play a game or give incentive to users. So, let us consider that the more complex the users' input commands are, the more they can receive a reward. In the example we described above, we can apply the idea of reward to decision regarding the power of a special attack. This works as follows. A user (game player) registers a command (password) on the game system and can use a special attack by inputting it. The user can re-register the command at any time. If he/she registers a short and easy command, the power of his/her special attack is twice the strength of that of an ordinary attack. If he/she registers a long and complex command, the power of his/her special attack is ten times stronger than that of an ordinary attack. Since users want to use a strong attack as possible, they will try to memorize much stronger commands (passwords).

3 Implementation of Game Instance

We implemented a game instance. The game is a simple dungeon exploration game. The dungeon has some floors (1F, 2F, …) and has a maze in each floor. Users need to explore the dungeon as deeply as possible. This game has three views: Dungeon view (Fig. 1), Battle view (Fig. 2) and Command check view. A Player have status: Current floor, Current level, Needed experience (experience points needed to reach the next level), Accumulated experience points, and The number of gained items. An attack command whose length is N characters is registered on the system.

In Dungeon view, the player moves by pressing up, down, right and left key. Each floor has three items. If the player collects them and moves to stairs, he/she is able to move to next floor. While exploring the dungeon, the player encounters an enemy with a constant possibility (=1.0 %). Whenever encountering an enemy, players automatically move to Battle view.

In Battle view, the player and the enemy have battle status: Hit point, Attack point, and Defense point. The status of the player is set based on the value of the player's

Fig. 1. Dungeon view

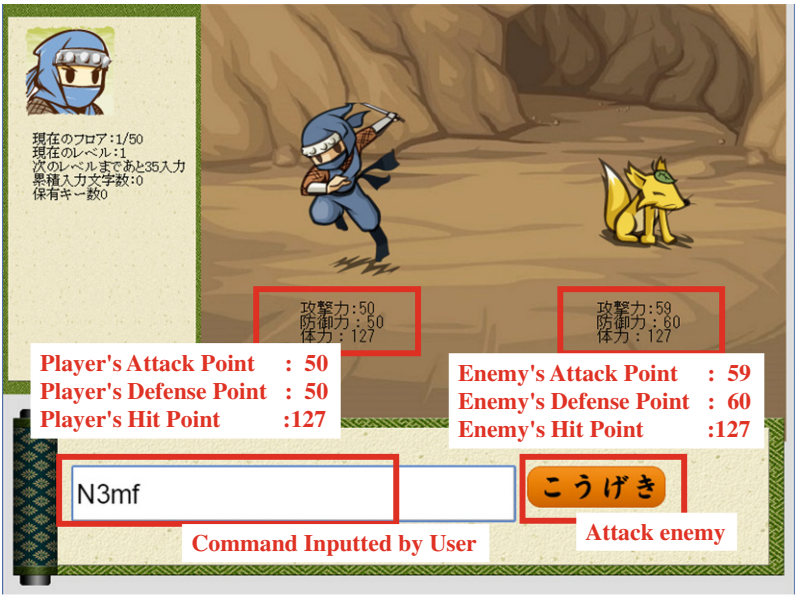

Fig. 2. Battle view

current level. That of the enemy is set based on the value of the current floor. The flow of the battle is as follows:

1. The player attacks the enemy by inputting the *k* characters from the beginning of the attack command.
2. The attack point of the user will increase depending on the value of *k*. In this game, we use the following setting: if *k* is less than four, the point increases by 1.01 times; if *k* is between five and seven, the point increases by 1.10 times; if *k* is between eight and ten, the point increases by 1.15 times, and so on. It should be noted that, if the player mistypes the command, the attack point will not increase.
3. The hit point of the enemy will decrease by the player's attack. The damage for the enemy depends on the attack point of the player and the defense point of the enemy.
4. If the hit point of the enemy becomes less than zero, the battle result is win. The game system goes back to Dungeon view.
5. The player is attacked by the enemy.
6. The hit point of the player will decrease. The damage for the player depends on the defense point of the player and the attack point of the enemy.
7. If the hit point of the player becomes less than zero, the battle result is lose. When going back to Dungeon view, the game system sets the number of gained items to zero and moves the player to the initial position.
8. Return to 1.

Players gain experience points when attacking enemies. Specifically, if a user is succeed to input the *k* characters from the beginning of the attack command, the game system adds *k* to the user's accumulated experience points. The system, while at the same time, subtracts *k* from the needed experience (experience points needed to reach the next level). If the needed experience becomes less than zero, the system adds one to the current level of the player and reset the value of the needed experience (the higher the current level of the player is, the bigger value the system sets).

In Command Check view, the player can check the attack command registered on the system. If players click the button at the lower right of Dungeon view, they are able to move to Command Check view (They are unable to move from Battle view to this view).

We have conducted a preliminary experiment beforehand. The parameters and default values used in the game, such as the parameter for damage calculation and the value for the needed experience to reach the next level, are decided empirically by the experiment. The details of them is omitted due to the limitation of space.

4 Experiment

4.1 Method

We asked five university students to play our game in their free time for three days. In this experiment, the command registered on the system was set by the experimenter (i.e., authors) as a random string with thirty characters. The command was "N3mf8-%x

RQkQeV)NMbnn*[sT(WW/". After the experiment, we asked the subjects to answer a questionnaire. The questionnaire includes the following two questions[1].

1. Did you memorize the command positively in our game? (yes/no) And also, please write why you think so.
2. Did you memorize the command naturally in our game? (yes/no) And also, please write why you think so.

4.2 Results

Table 1 shows the result of game play. All subjects were successful in inputting at least thirteen random characters. Surprisingly, four out of five subjects was successful in inputting all thirty random characters. The result of the questionnaire suggested that the subjects had positive impression of our game. In question 1, all subjects answered "yes". They stated the reasons: it gives an advantage in the game; I wanted to increase the current level as soon as possible. In question 2, four subjects answered "yes". They stated the reasons: I was able to memorize the command gradually and naturally; when I encountered an enemy which I didn't beat, I tried to memorize longer command.

Table 1. Results of game play

User A

	Play time [h:m:s]	Number of characters that subject memorized	Number of command inputs
Day 1	0:06:20	11	6
Day 2	0:05:12	23	4
Day 3	0:10:16	30	6
Total	0:21:48		16

User B

	Play time [h:m:s]	Number of characters that subject memorized	Number of command inputs
Day 1	0:53:31	11	58
Day 2	0:29:43	23	26
Day 3	0:14:58	30	14
Total	1:38:12		98

User C

	Play time [h:m:s]	Number of characters that subject memorized	Number of command inputs
Day 1	0:07:06	9	9
Day 2	0:18:19	30	19
Day 3	0:17:44	30	19
	0:43:09		47

User D

	Play time [h:m:s]	Number of characters that subject memorized	Number of command inputs
Day 1	0:14:18	17	13
Day 2	0:21:11	26	19
Day 3	0:14:45	30	46
	0:50:14		78

User E

	Play time [h:m:s]	Number of characters that subject memorized	Number of command inputs
Day 1	0:05:00	8	4
Day 2	0:06:00	12	6
Day 3	0:05:00	13	5
	0:16:00		15

[1] We prepared five questions. But, due to the limitation of space, we in this paper report two questions and their results.

4.3 Discussions

Did subjects memorize stronger password through our game? Bonneau et al. reported that a 56-bit random password is a reasonable strength for most practical scenarios [6]. All subjects memorized at least thirteen random characters, which has an entropy of about 80-bit. In particular, four subjects memorized thirty random characters, which has much bigger entropy than the entropy of 56-bit. These result suggests that our game enables users to memorize an enough strong password.

Did subjects memorize password naturally through our game? All subjects answered "yes" in question 1. According to the reason why the subjects think so, subjects inputted the command (password) for playing the game smoothly or getting an advantage. The results suggest that the subjects recognized the command input not as "memorizing a password" but as "enjoying the game". Once a user can memorize a command while playing a game, later on they will be able to use it as a password. As a conclusion, our game enables users to memorize a stronger password *naturally*. In fact, in question 2, most subjects answered "yes".

5 Conclusion

In this paper, we developed a game instance for the password enhancing game proposed in literature [5] and conducted a user experiment. Its results showed the effectiveness of our scheme, i.e. users are able to memorize stronger passwords naturally while playing our game. This experiment is still initial phase, so we are going to conduct a more comprehensive experiment.

References

1. Yamamoto, T., Suzuki, T., Nishigaki, M.: A proposal of four-panel cartoon CAPTCHA. In: 25th International Conference on Advanced Information Networking and Applications, pp. 159–166. Biopolis (2011)
2. Mohamed, M., Gao, S., Saxena, N., Zhang, C.: Dynamic cognitive game CAPTCHA usability and detection of streaming-based farming. In: Workshop on Usable Security, San Diego (2014)
3. Kojima, Y., Yamamoto, T., Nishigaki, M.: Proposal of an image-based one-time authentication scheme using "Spot the deference". In: IPSJ SIG Technical report, 2007-CSEC-36, pp. 375–380 (2007). (in Japanese)
4. Ur, B., Kalley, P.G., Komanduri, S., Lee, J., et al.: How does your password measure up? The effect of strength meters on password creation. In: 21st USENIX Conference on Security Symposium, Bellevue (2012)
5. Fujita, M., Yamada, M., Arimura, S., Ikeya, Y., Nishigaki, M.: An attempt to memorize strong passwords while playing games. In: The 18th International Conference on Network-Based Information Systems, pp. 264–268 (2015)
6. Bonneau, J., Schecheter, S.: Towards reliable storage of 56-bit secrets in human memory. In: USENIX Security 2014, pp. 607–623 (2014)