IOSR IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. II (Mar-Apr. 2014), PP 87-93
www.iosrjournals.org

A Method for Bypassing Keystroke Recognition Based Security System Using Social Engineering

Devbhuti Shounak[1,] Ganguly Debojyoti[2], Majumder Debdeep[3], Payra Ritwik[4]
[1,2,3,4] *ESL, Chandernagore Lab CA-5 Salt Lake City, Kolkata – 64, India*

Abstract: *In this paper we aim to use the power of Social Engineering to bypass Keystroke Recognition based authentication system. We have designed a virtual chat system (basically a chat bot), coded in Python, which performs the much needed social engineering on the victim. A wise victim may think that the bot is trying to extract confidential credentials from him and so provides false credentials to it but in reality the bot has nothing to do with the credentials. Bot's only job is to record the typing speed of the individual which is the basic need of a security system based on keystroke recognition. Our system includes following prime specialties (assuming two machines, one is the victim's and another the attacker's).*

- *At the victim's side our virtual chat bot measures the victim's typing time and creates a database of it.*
- *An attacking program (designed in Python), integrated to the chat bot, uses the above created database and enters the credentials to the security system virtually at the same time as that of the victim. As a result access is granted to the victim's system.*

Hence we can effectively breach the security system based on keystroke recognition, which primarily uses a person's typing speed and relies on the fact that no two different person's typing speed are exactly same.

Keywords: *Keystroke Recognition, Python, Social-engineering, Virtual Chat System (Chat Bot).*

I. Introduction

Our project is mainly divided into three parts: (i) A dummy of the presently implemented security system based on Keystroke Recognition. (ii) A virtual Chat Bot whose main job is to perform social engineering on the victim by chatting with him and manipulating him to type different types of words. It then records the typing speed and creates a database for various categories of words. (iii) An attacking program which uses the database created by the chat bot and feeds the security system with the credentials at approximately same time as that of the victim. Here we are using an approach which is somewhat similar to social engineering but differs a lot in methodology. An intelligent victim may think that he is socially manipulated by the bot and the bot is trying to extract sensitive information such as usernames, date of birth, bank account numbers or any other kind of personal details which can cause risk to the security of his system and thus may provide wrong information to the bot. But this doesn't affect to our project at all as the bot is being designed to record the keystroke timing (basically the typing speed) and not for extracting personal details of any other kind. In this project we have used various technologies and one programming language:

- Keystroke Recognition
- Python
- Inbuilt modules for Python programming language:
  a) 'pysqlite'
  b) 'Tkinter'
  c) 'time'.
  d) 'os'
- Social Engineering

1.1 Keystroke Recognition: This is the detailed timing information that describes exactly when each key was pressed and when it was released.

1.2 Python: A widely used general purpose high-level language. Its design philosophy emphasizes code readability and its syntax allow programmers to express concepts in fewer lines of code than would be possible in languages such as C. Python supports multiple programming paradigms, including OOP and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large comprehensive standard library.

1.3 'pysqlite': This is a module under python programming language which renders database support from standard Structured Query Language (SQL) based database management system MYSQL.

1.4 'Tkinter': This is a module under python programming language using which standard Graphical User interfaces (GUI) are designed.

1.4 'time': This is a module under python programming language which gives access to the current system time. We used this module here mainly for calculating the typing time of the victim.
1.5 'os': This module imports routines for Mac, NT, or Posix depending on what system we are on. Programs that import and use 'os' stand a better chance of being portable between different platforms.
1.6 Social Engineering: Social Engineering uses influence and persuasion to deceive people by convincing them that the social engineer is someone he is not, or by manipulation. As a result, the social engineer is able to take advantage of people to obtain information with or without the use of technology.

II. Motivation

All of us set password in our system for securing confidential and sensitive data. But none of the security systems is unbreakable. While enforcing security standards we generally overlook a basic vulnerability which is the human factor. It's natural to yearn for a feeling of absolute safety, leading many people to settle for a false sense of security. This makes the human factor security's weakest link. Hence our main challenge today is to secure any type of security system from the human factor as far as possible. In this project we put our conscious effort to highlight how the human factor can be exploited for bypassing even one of the toughest security system. This will help upgrading this security system in future and will gather more concentration on enhancement of other such type of security systems from the human factor beside renowned security breaching attacks like brute force, dictionary attack, rainbow table attack etc. The methodology section describes the algorithms in detail.

III. Methodology

In the following section we are going to describe the methodology of the project we have done.
3.1 Design of the dummy security system based on Keystroke Recognition: The working of this system can be divided into two parts: a) The training session, b) The final login phase.

- **The training session:** If the security system is being operated for the first time in a computer, then the user has to go through a training phase. During this phase the user is being given six random words, meaningful or meaningless, of different categories (the types of categories of words are being discussed later on). Through this the security system creates a database of the typing time against each category of words. Hence, the system gets well informed about the keystroke timing of the user. The training session occurs only for a single time, i.e., for the first time only.
- **The final login phase:** In this phase the user is given only one single randomly generated word which can belong to any category. If the user types the correct word as displayed and his keystroke timing matches with the database of the security system, then access is granted to the user, else the user is forbidden to use the system.

3.2 Category of words: For classifying various words into different categories we have followed a uniform rule of categorization. We have classified the keys of a standard QWERTY keyboard into three different categories based on the position of the keys. They are as follows:

- Category A: Keys at the leftmost side i.e., 'A','Q','W','S','D','Z','X','C','E'.
- Category B: Keys at the middle position i.e., 'R','T','Y','F','G','H','V','B','N'.
- Category C: Keys at the right most side i.e., 'U','I','O','P','J','K','L','M'.

Thus covering all the 26 alphabets of English language. We have excluded the numbers and special symbols. So any word of a Standard English lexicon can be formed by the combination of these three categories. Again we have also categorized the words formed from the combination of letters of the above categories. For example the word 'Bitter' is formed by 4 letters from second category, 1 letter from third category and 1 letter from first category. So we get combination as (1A, 4B, 1C). In this way we can categorize all words of English language as well as proper nouns. As a result we will get a large number of categories. But we have used only 5 such type of categories for our project and fixed the word length to six letters for testing purpose. This can be understood better from the Venn diagrams below (figure 1 and figure 2):

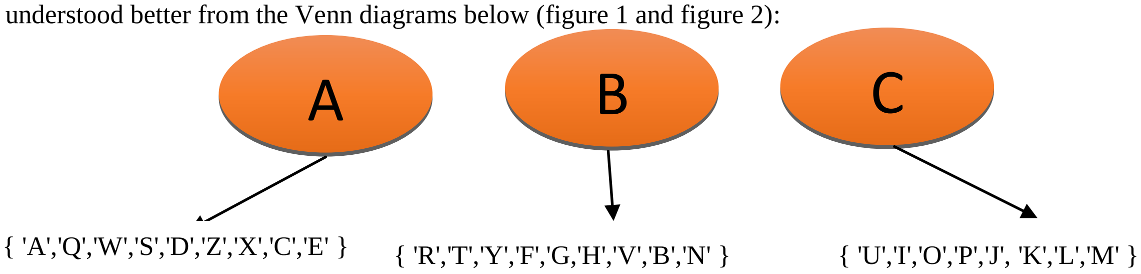

A
B
C
{ 'A','Q','W','S','D','Z','X','C','E' }
{ 'R','T','Y','F','G,'H','V','B','N' }
{ 'U','I','O','P','J', 'K','L','M' }

Fig. 1: Categorization of letters.

Again we have 5 sub categories of words formed by random selection of words from the above categories of letters. They are divided on the basis of combinations of letters selected randomly. As shown in figure 2.

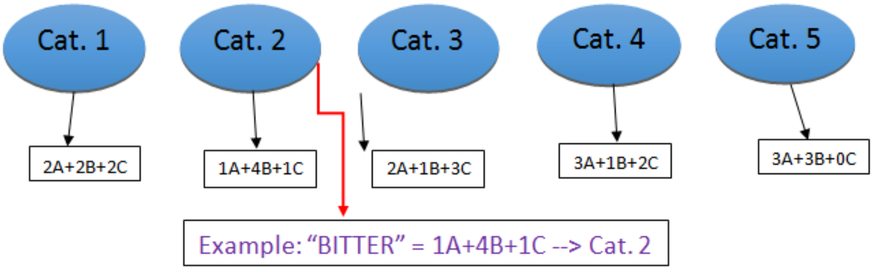


Fig. 2: Categorization of words.

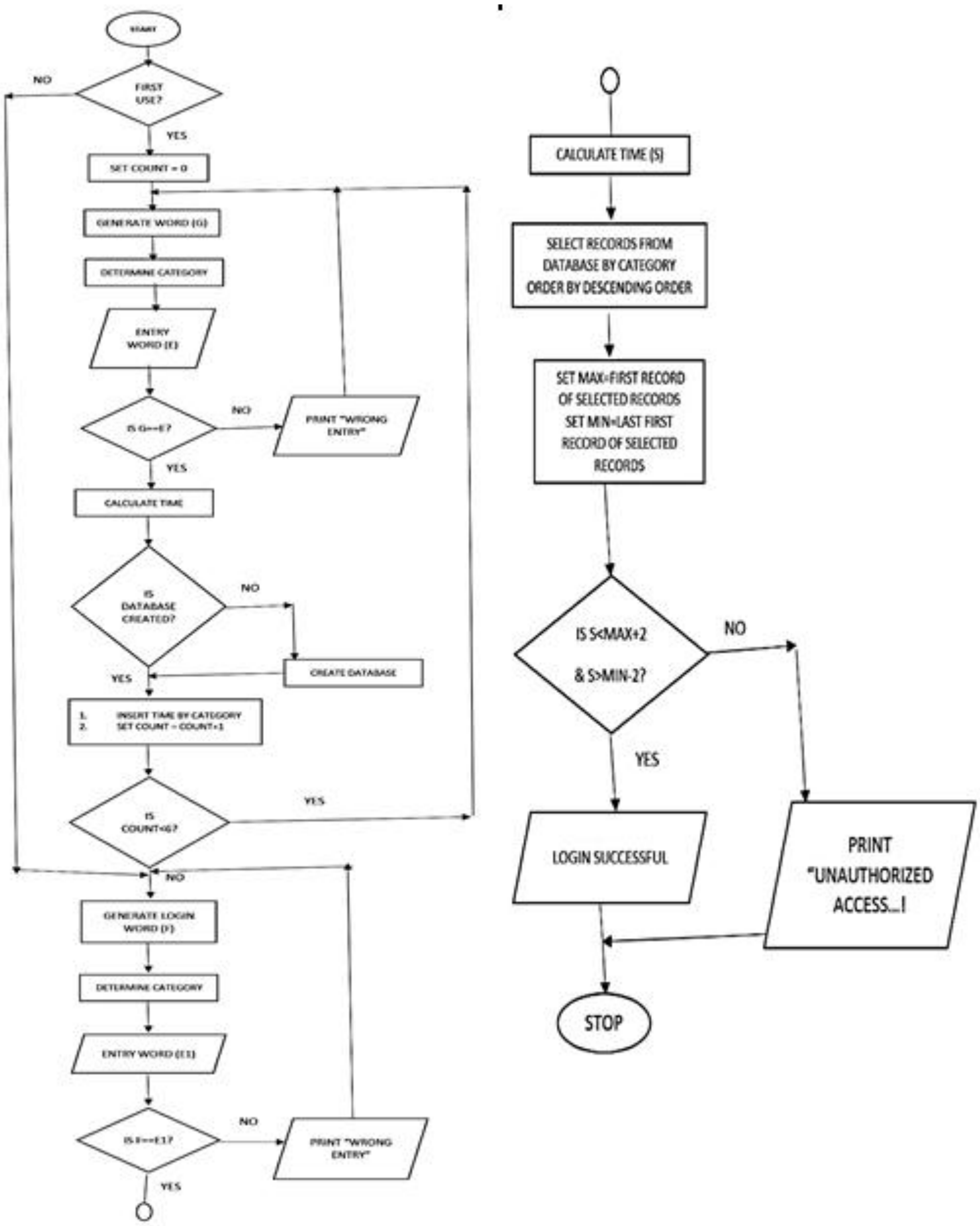


Fig. 3: Flow diagram of dummy security system based on Keystroke Recognition.

Figure 3 describes the flow diagram of the dummy security system designed by us. A new user will go through the training session. Whereas an old user will be directly taken to the login screen.

3.3 Design of the Virtual Chat System (Chat Bot):

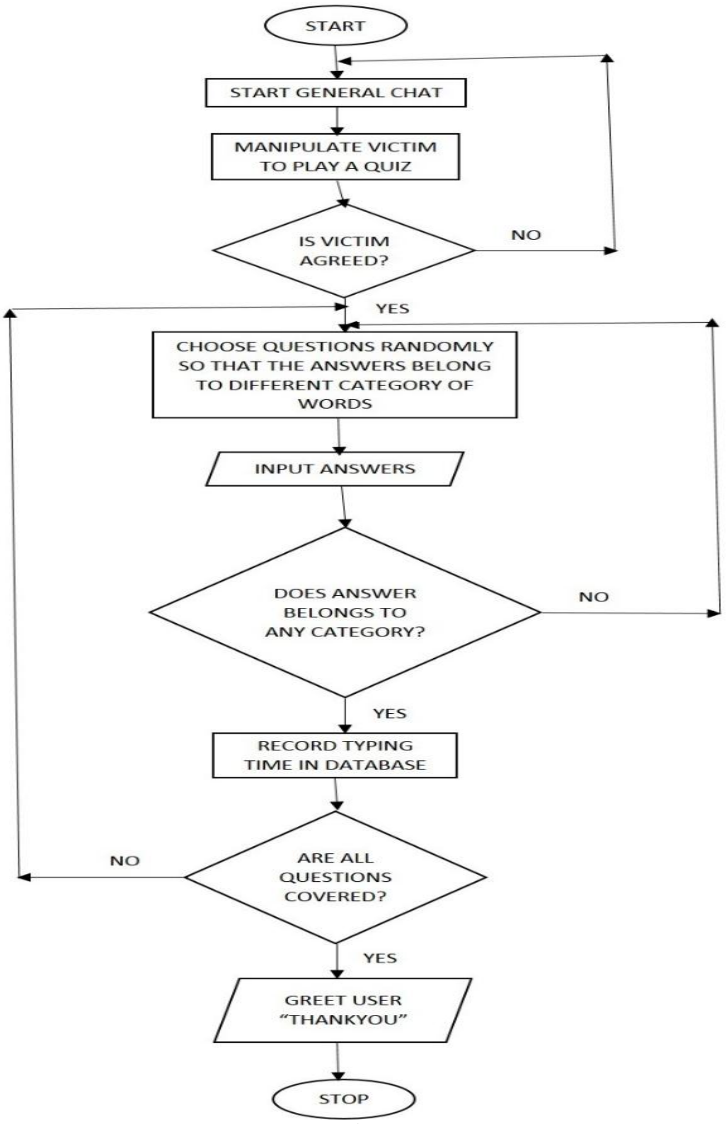

START
START GENERAL CHAT
MANIPULATE VICTIM TO PLAY A QUIZ
IS VICTIM AGREED?
NO
YES
CHOOSE QUESTIONS RANDOMLY SO THAT THE ANSWERS BELONG TO DIFFERENT CATEGORY OF WORDS
INPUT ANSWERS
DOES ANSWER BELONGS TO ANY CATEGORY?
NO
YES
RECORD TYPING TIME IN DATABASE
ARE ALL QUESTIONS COVERED?
NO
YES
GREET USER "THANKYOU"
STOP

Fig. 4: Flow diagram of Virtual Chat (Chat Bot).

Figure 4 shows the flow diagram for the virtual Chat Bot. This bot will manipulate the victim to play a quiz and will pose the questions in such a way that the answers are of one word and the words belong to the pre-set categories as we discussed before. The bot will then prepare its own database of typing time of the victim. This database will be used by the attacker program for bypassing the security system as we will see later on. In our bot we have fixed a set of questions for the quiz keeping in mind the answers will belong to the pre-set categories. As soon as the quiz is over, the bot will greet the victim thanks and will shut down.

3.4 Design of the Attacker Program:

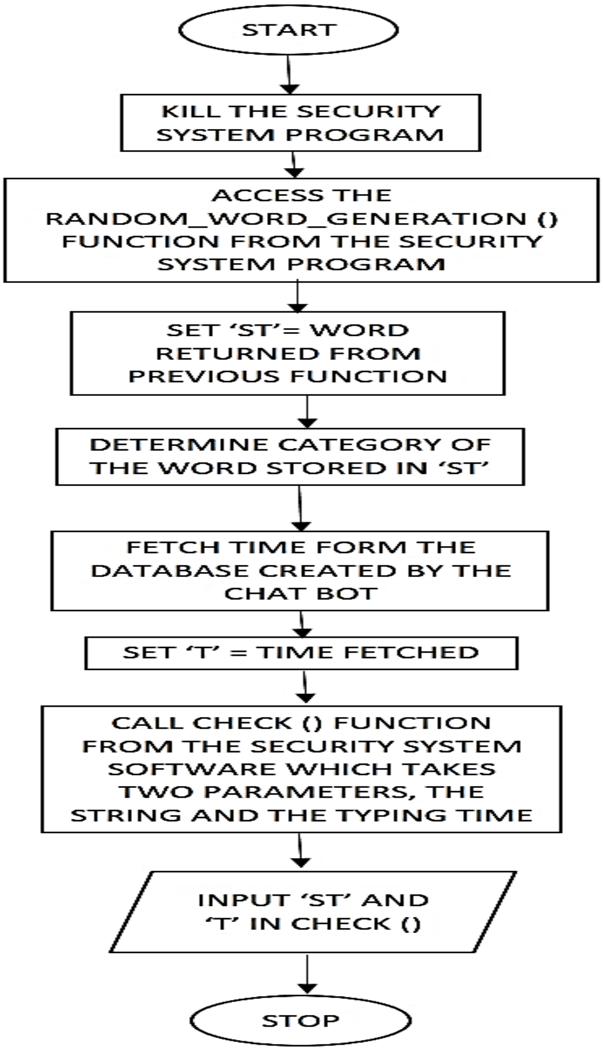

START
KILL THE SECURITY SYSTEM PROGRAM
ACCESS THE RANDOM_WORD_GENERATION () FUNCTION FROM THE SECURITY SYSTEM PROGRAM
SET 'ST'= WORD RETURNED FROM PREVIOUS FUNCTION
DETERMINE CATEGORY OF THE WORD STORED IN 'ST'
FETCH TIME FORM THE DATABASE CREATED BY THE CHAT BOT
SET 'T' = TIME FETCHED
CALL CHECK () FUNCTION FROM THE SECURITY SYSTEM SOFTWARE WHICH TAKES TWO PARAMETERS, THE STRING AND THE TYPING TIME
INPUT 'ST' AND 'T' IN CHECK ()
STOP

Fig. 5: Flow diagram of the Attacker program

Figure 5 shows the flow diagram of the attacker program. This program kills the process of the security software, (which has been described before) and calls the function which is generating the random word for the security system. The program then determines the category of the word generated by the security system and fetches the typing time from the database created by the chat bot. After doing this, the attacker program calls the function that does the checking of the typing time in the security system of the victim. This function takes two parameters, the generated string and the typing time. The program then supplies the sting which it already fetched from the security program and the time itself which it fetched from the database created by the chat bot. The algorithm for the attacker program is given below.

IV. Algorithms

- **DUMMY SECURITY SYSTEM**

Step 1: Start.
Step 2: If first use of the program then go to Step 3. Else go to step 12.
Step 3: Set count=0.
Step 4: Generate random word (G).
Step 5: Determine the category of the word generated in Step 4.
Step 6: Input word from user (E).
Step 7: If (G==E) then go to step 8 else print “Wrong Entry” and go to step 4.
Step 8: Calculate typing time.
Step 9: If database is already created then go to step 10 else create database and then go to step 10.

Step 10: Insert calculated typing time in database by category of word.

Step 11: Set count=count+1.

Step 12: If (count<6) then go to step 4. Else go to step 13.

Step 13: Generate login word (F).

Step 14: Determine category of the word generated in Step 13.

Step 15: Input word from the user (E1).

Step 16: If (F==E1) then go to step 17. Else print "Wrong Entry" and go to Step 13.

Step 17: Calculate typing time (S).

Step 18: Select records from database by category order by descending order.

Step 19: Set MAX= first record of selected records. Set MIN=last record of selected records.

Step 20: If (S<MAX+2 and S>MIN-2) then go to Step 21. Else print "Unauthorized access." And go to Step 22.

Step 21: Login successful.

Step 22: Stop.

- **VIRTUAL CHAT SYSTEM**

Step 1: Start.

Step 2: Initiate general chat with the victim.

Step 3: Manipulate victim to play a quiz.

Step 4: If the victim agreed to play then go to Step 5. Else go to Step 2.

Step 5: Choose questions randomly so that the answers belong to different category of words as fixed before.

Step 6: Input answer from the victim.

Step 7: If answer belongs to any category then go to Step 8. Else go to Step 5.

Step 8: Record typing time of the victim in the database.

Step 9: If all the questions of the quiz is covered then go to step 10. Else go to Step 5.

Step 10: Greet the user "Thank you".

Step 11: Stop.

- **ATTACKER PROGRAM**

Step 1: Start.

Step 2: Kill the security system program.

Step 3: Access the Random_Word_Generation () function from the security system program.

Step 4: Set 'st'= word returned from previous function.

Step 5: Determine category of the word stored in 'st'.

Step 6: Fetch time form the database created by the chat bot.

Step 7: Set 't' = time fetched

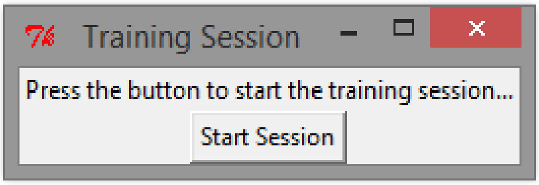
Step 8: Call check () function from the security system software which takes two parameters, the string and the typing time.

Step 9: Input 'st' and 't' in check ().

Step 10: Stop.

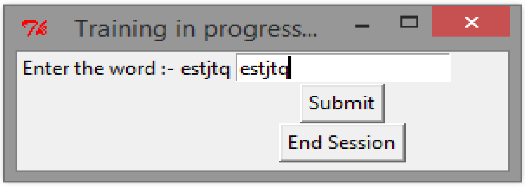
V. Results

Here goes some of the screen shots of our project which shows dummy security system and the virtual chat program respectively. Figure 6 is showing the training session start screen of the security system. Figure 7 is showing the training session in progress.



Press the button to start the training session...

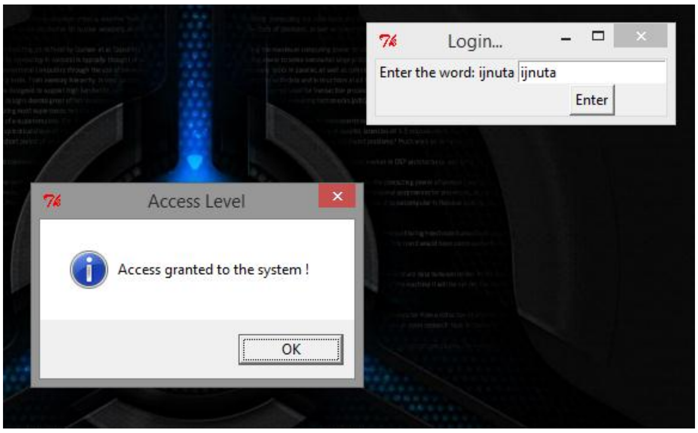
Fig. 6: Training session start screen.



Enter the word :- estjtq estjtq

Fig. 7: Training session in progress.

Figure 8 is showing the final login screen of the software which comes after the training session for new users and directly for old users.

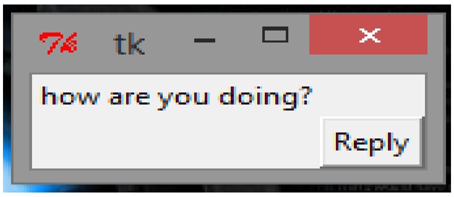


Enter the word: ijnuta ijnuta

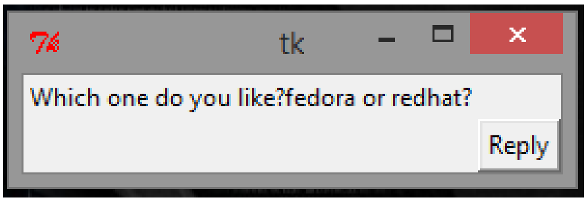
Access granted to the system !

Fig. 8: Final login screen showing access granted to the user.

Figure 9 shows the virtual chat system doing general chat with the victim before starting the quiz. This is important for gaining trust of the victim so that he can be easily persuaded to play the quiz which is our main target. Figure 10 shows the quiz in progress with the victim. The reply button is used to give reply to the bot.



how are you doing?



Which one do you like?fedora or redhat?

Fig. 9: Virtual Chat system doing general chat.

Fig. 10: Virtual Chat system playing quiz.

We also observed that the attacking program successfully bypassed the security software that we have designed previously.

VI. Future work

In near future we will work on the virtual chat and will make it more human by integrating Artificial Intelligence Mark-up Language (AIML). This can be done efficiently by using ‘pyaiml’ module of python. We will also work for securing the present Security system based on keystroke recognition from social engineering attacks and robots by using visual CAPTCHA for verification. We will also search for any other vulnerabilities in the above security system.

VII. Conclusion

The methodology section describes the implementation of the systems in detail and the result section shows that the systems are working as per our expectation. There are some assumptions incorporated in this system. They are:

i. Users will put correct spelling during the virtual chat

ii. Users won’t use any abbreviation and special symbols during the chat

We hope that the concept described in this report will aid the developers’ community in near future.

References

[1] Fabian Monrose, Aviel D. Rubin, “Keystroke dynamics as a biometric for authentication”, Future Generation Computer Systems, 2000 ELSEVIER, Volume 16, Issue 4, February 2000, Pages 351–359.

[2] Jiyou Jia, “CSIEC: A computer assisted English learning chatbot based on textual knowledge and reasoning”, Knowledge-Based Systems, ELSEVIER, Volume 22, Issue 4, May 2009, Pages 249–255.

[3] T Thornburgh, “Social engineering: the Dark Art". Paper presented at InfoSecCD '04 Proceedings of the 1st annual conference on Information Security Curriculum Development

[4] Rusch, J, “The "social engineering" of Internet fraud”. Paper presented at the 1999 Internet Society's INET'99 conference. Retrieved from http://www.isoc.org/isoc/conferences/inet/99/proceedings/3g/3g_2.htm

[5] Manske, K, “An introduction to social engineering”, Information Systems Security 9, Pages 53-59. Retrieved from GALILEO: Computer Source database.