Passwords with No Trace!

Gagan Deep Bakhshi
Student
CMR Technical Campus, JNTU Hyderabad

***Abstract* - We have been introduced to the passwords since long back. It has been one of the best and widely adopted scale of security in various applications. There are many aspects of giving and receiving passwords, which include Text, audio, picture, problem solving etc, but one of the widely used is the text mode authentication. But here we are facing a serious drawback in the Password text mode .It has the vulnerability to be traced .This is what we need to improve and where my research is intended to.**

I. INTRODUCTION

In this paper, we have attacked to the vulnerability of the password visibility at the time of entering.

Vulnerability

Generally, when we enter the passwords, we try to encrypt them using some special characters like asterisk (*) or dots(.),and we feel it secure that nobody is seeing it, that's where we are wrong.

In most of the cases, when one's password is compromised, there is some small carelessness that does the trick. If a person sitting at your back sees what you are typing and gets to know some part of it, and the length of your password(as the encrypted text is visible),he may try and guess it.

Solution

We need to ensure that the password is not visible in any form, so that the small loophole which we all neglect can be overcome in all aspects (in all applications).

Details

Let us consider an example for the illustration of this technique. Suppose a password is set as "Hello World", If you note, it includes 11 characters, now suppose the password is typed by the user, it will be shown as **********,now the one thing which we know is that the password is of 11 characters length, which is what we disclosed knowingly. Now most of us like to keep our passwords simple, so we do not take much advantage and risk of using too much complexity in passwords as special characters, and Caps and small in an abnormal way. If a third party user, gets to know something like Hel….ld, one can easily crack the password using the length of the password guessing or a simple jumble.

This type of technique is adopted in Linux command mode authentication, but needs to be widely adopted. Especially in terms of the credentials, social networking, emails and others.

Explanation of Refinement

As per our technique, when we enter the password, the password BLINKER won't move forward with each character typed, instead it will blink at the same position even though the password is being typed, so that even if someone tries to catch up with the password, he won't be able to estimate the length of the password, which will increase the set of possibilities of password set to an infinite level, and will reduce the risk to much extent.

We used a C code, to implement the logic and we succeeded in doing so, and this will for sure bring an improvement in the password typing, and authentication.

- For refined approach, it should be used in a wide area of applications.
- The person needs to be perfect in terms of his password, and should not depend on the encrypted text being typed to guess the characters typed.

II. RESEARCH QUESTIONS

1. **What way is this technique helpful?**
   One of the drawbacks that we found in the current password mode authentication is that although the text becomes encrypted on typing still we are unable to hide the length of the password. As already explained your search may be refined just if you are aware of the length of the password. And as most of the people do not effectively use block letters and special symbols in all aspects, and then, they fall in the vulnerable category. Therefore, why not cover this aspect as well, so that the little loophole is fixed.
2. **From where the research is inspired?**
   This technique originated, when I successfully wrote a program to simulate this technique. Then the real world comparison came to my mind and as a result I thought to work on it.
3. **Is it a limited research or work has been completed?**

This research is on the various aspects of the application of this technique and its advantages. More research is possible, and the scope is clearly available on this plot.

III. ADVANTAGES

1. This technique is not based on any platform.
2. It has various fields of applications such as Web based, Transaction based, system apps etc.
3. Improves the authentication by not much expense.
4. Can be easily adopted by everyone.

IV. TEST CASES

If you consider a game of finding the word, and you are given with the information that the word starts with H and is of length 5 characters, ending with O, then you can easily guess it out with HELLO.

On the other hand, if the same word is given but you are not aware of the length of the word as 5 characters, then your guess is not a comparatively stronger one as it can be HERO,HELLO etc.

Note:

Many of the nonprofessional users, use the encrypted text as a hint for how much characters have they typed, and then count and proceed. For them, this technique may require some fluency in the credentials info and typing.

V. WHY WE NEED THIS TECHNIQUE TO WIDESPREAD?

Existing System

The basic requirement and challenge in front of us at present is that we need security, but we also need to make sure that we do not use something more tricky or complex that would be difficult for the user to recollect.

So, if the password can be typed, the only change we are making is the visibility of the password is restricted, so the existing system is not being altered much, and we are also covering the scope of intrusion. What we need to focus is if while changing the password in windows, or Linux, the visibility is restricted for security reasons, then why not it be spread in the other areas, because either it is mail or social networking account, the security aspect remains unaltered.

New System

As an update to it, if we follow the same technique everywhere then we are covering the loophole.

When my research is implemented, there will be no trace of the password in the authentication mode. The length can't be judged, and the existing system will be improved significantly.

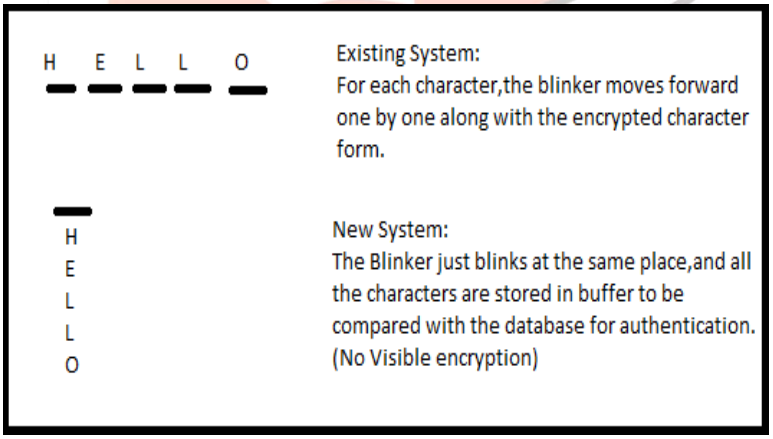

Fig 1

Program Logic in C

This algorithm was written as a program in c programming language and was tried and tested by storing a static password inside an array:

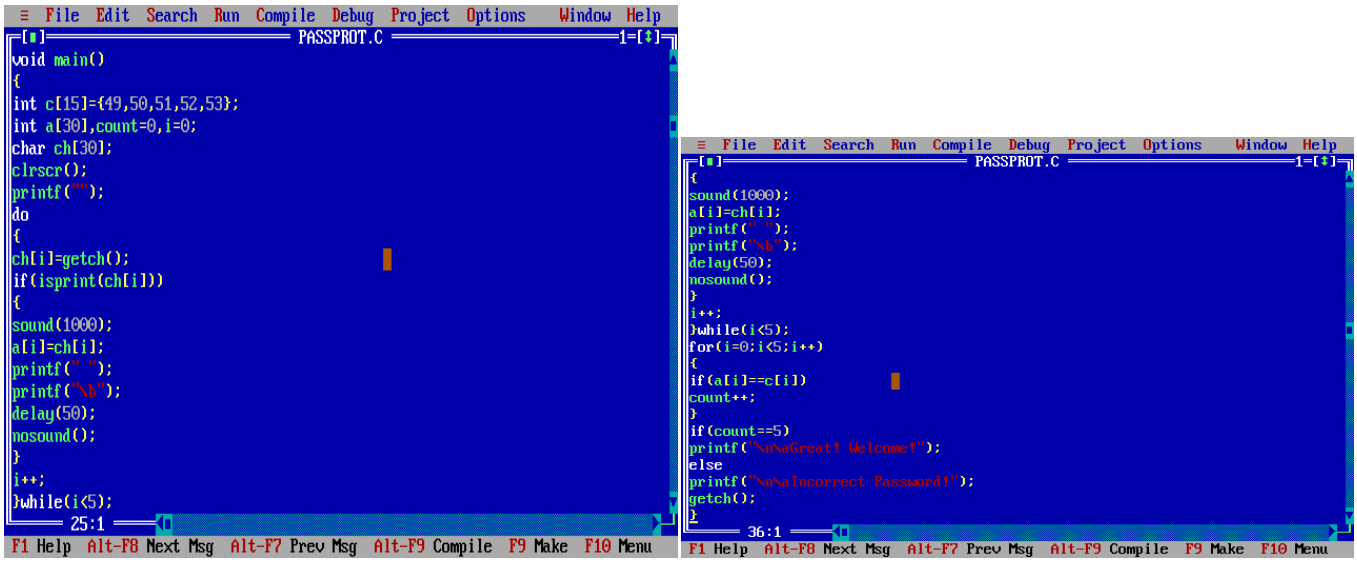

Fig 2

The ascii codes are stored for 12345 in the array as shown above.
This with no encryption and no blinker movement, under the new system will generate the following output:

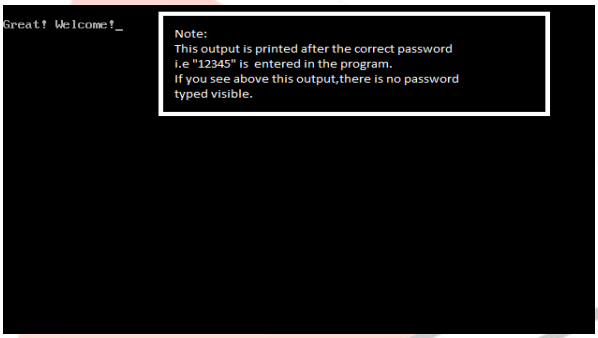

Fig 3

VI. Conclusion

This technique will for sure bring refinement to the authentication modes of text , some more research will further clarify the concept with added modules. If adopted in the correct sense, this technique will soon be widely accepted.

VII. Acknowledgments

Some of the members which contributed for this approach with some positive and negative feedback include: Mr. Sharma SaiRam Dept of CSE, CMRTC and Mr. K. Srujan Raju HOD CSE Dept. CMRTC

APPENDIX

A C program was written to first implement this technique, and after it was successful, WE made its set of possible applications. Then, we found one of its APPLICATIONS in the linux command mode authentication although it is also not the exactness what this paper covers.

hence, the research has shown some positive signs of applications in various fields as specified above in the paper and its implementation can be encouraged in multiple platforms and this theme can be decorated with multiple ideas, and more research can be encouraged on this theme.