Authentication of User And Performing Deduplication of Data.

PROF.KALAM NARREN
Department of Computer Engineering Savitribai Phule Pune University, Pune, India,
sayali2019@gmail.com

Dr.HENRY
Department of Computer Engineering Savitribai Phule Pune University, Pune, India,
gulnarbanushaikh1@gmail.com

Dr.JULIE
Department of Computer Engineering Savitribai Phule Pune University, Pune, India,
kishantongrao123@gmail.com

Dr.SAI
Department of Computer Engineering Savitribai Phule Pune University, Pune, India,
sandhyapatil966@gmail.com

Dr.ECCLESTON
Department of Computer Engineering Savitribai Phule Pune University, Pune, India,
kirtivdeshpande@gmail.com

ABSTRACT
Authentication is identifying the user, on the basis of its valid username and password.Honeyword mechanism provides more enhanced way of authetication.This mechanism generates honeywords which are stored in honeypot.These honeywords are used to confuse the attacker ,making him believe that the honeyword found is the actual password.After Authentication when user logins successfully ,while storing the data perform dedupication .Deduplication wil check for depulicate files or duplicate data .The duplicated data will then be not stored in cloud.This deduplication will help in reducing the redundancy.

Keywords:
Authentcation,Honeywords,Sugarwords,deduplication,Splitting

1. INTRODUCTION
We know how important authentication is in each and every field now-a-days. Here in this paper we are putting ver forward a honeyword mechanism for authentication along with a deduplication mechanism performed when we will be actually storing the data file in cloud.

The honeyword mechanism is used to watch a hacker who endeavours to login with cracked passwords.In Honeyword mechanism we have three main terms i.sweet-word (sweet word is the actual main password of username login) , ii.sugarword (sugarwords are three extra words entered by user other than the sweet word) , iii.honeywords (our mechanism would now create n number of honeywords by mixing the existing sugarwords using various techniques).These honeywords are nothing but the fakewords to confuse the attacker and keep him away from the actual password.Also all these i.e sweet word , sugarwords and honeywords are stored using a key which adds the XOR bit to it and then stores it into database.Which then makes the attacker more difficult to make out the actual honeyword too. So here for every username there is only one sweet word which will give successful login and others are decoy passwords(honeywords).Hence, whenever any attacker tries to enter into the system usin a honeyword , an alarm is triggered via gmail or other specified media by user to notify the administrator relating the password leakage.

In addition to this , after succesful login of authentic user we perform a duplication check to the file which we need to upload on cloud. The deduplication mechanism would first generate a digestcode (using MD5 algorithm) of the file and using a java code we will search if the digest code matches upto minimum 60% with already existing digestcodes of other files on cloud. If the matching percentage is 60 or greater than 60 then that portion(matching portion) wont get stored on cloud, only the link of the actual file would be provided of the which is matched .Simply an entry is made in a table storing file name , location of matched file(if=>60%) (just a link of previous file) , and location of remaining unmatched portion of file is stored on cloud.Now the non duplicated portion of the file found will be splitted and then stored on the cloud.

2. PROPOSED SYSTEM

The propsed system will have two important mechanisms i.e the Honey word mechanism and the deduplication mechanism. The Honeyword mechanism starts working from registration itself. While user registers to the system , user is asked to insert the correct username and password(sweet word) for login purpose. After that user is asked to enter three different passwords which will act as sugar words. Then n number of honeywords are generated using these sugar words with the help of honey word generator algorithms. These honeywords are stored in the honeypot. These are then mapped with any random key and accordingly xor bits are added. Now whenever an attacker attacks the system he will be confused with honeywords instead of retrieving the actual password.

After successful login of authentic user , the deduplication mechanism starts working. If user wants to store a file to system , the file is first checked if duplicated.If found duplicated than it is not stored, instead the location of already existing part of file is provided. If not found duplicated then it is splitted and then stored.

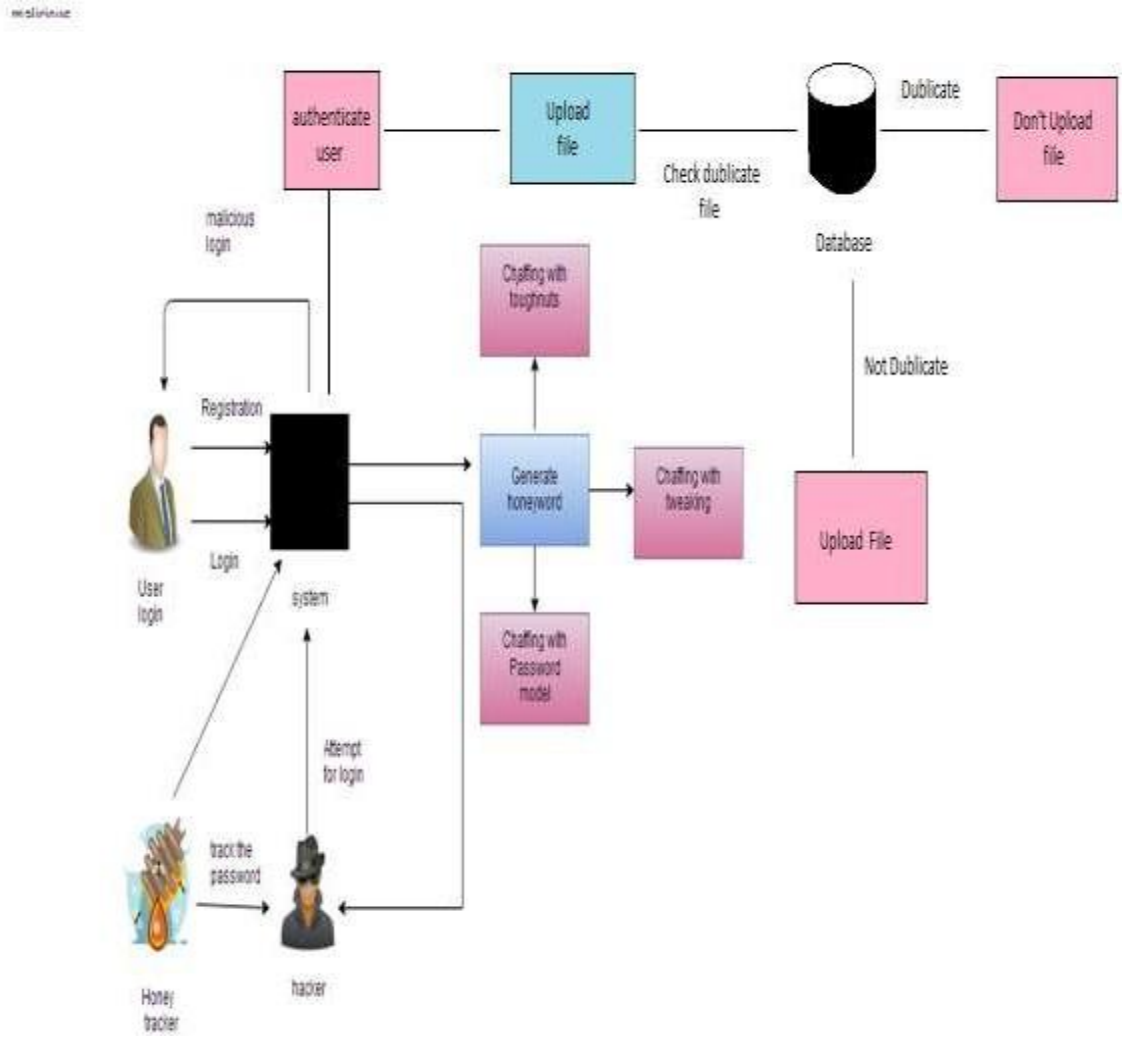

Fig 1: System Architecture

3. TECHNOLOGY USED

For generating honeywords we use four methods.

i.Chaffing by tweaking digits.

ii.Chaffing with a password model.

iii.Chaffing with toughnuts.

iv. Chaffing with tail.

1. Chaffing with tweaking , this method would generate honeywords using the three sugarwords.This method will tweak the selected location of the sugarwords to generate n number of passwords.The choosen characters of selected locations are then replaced by randomly any character.

Gen(m;d)

d=number of characters to be replaced

Example:

Input: pqr456 (sugarword)

d=2

Output:

pqr143,rru456,pwt456,…

2. Chaffing with a password model, this method the algorithm will takes the sugarwords and depending on the probabilistic model of sweet word, produces the honey words.The passwords is splitted into character sets.

Example:

Input: cat12late

It is decomposed as 3 letter+2 digit+4 letters.

So L4+D1+L5 will replaced with the same composition like "lay38gold".

3. Chaffing with toughnuts, this method is used to generate some honeywords using the sugarwords where intensionally the sugarwords's hah values are inverted and any random bit strings are set as new hash values.

4. Chaffing with tail, in this method various methods are combined e.g. chaffing-with-a-password-model and chaffing-by-tweaking-digits.

In this method the random password model will yield seeds for tweaking-digits to generate honey words.

Example:

Input: PQR567

It will add one Special Character at tail

Special Character(@,#,&)

Output: PQR#567.

4. CONCLUSION

Hence we have studied the Honeyword mechanism and various techniques like chaffing with toughnut, chaffing with tail, chaffing with password model to generate the honeywords.And then performed Authentication of user with the help of these honeywords generated.If any malpractice found or any kind of attack detected then system would notify the user regarding the same.If there is no attack or authentic user tries to login with correct username and correct password then , he would be able to successfully login to the system .After successful login we would now allow the user to access the cloud . When user wants to upload a file to cloud the file is checked if duplicate.We know if same files, same data is stored again and again on cloud it would ultimately increase the redundancy.To avoid redundancy , deduplication is performed. Whatever portion found duplicated is then not uploaded and remaining file is splitted and uploaded on the cloud. This splitting provides more security as the file contents are not present at one loctaion , but at various so if any non-authorised person tries to fetch a specific file he wont get the complete file, neither at one location.So overall idea is to provide security . Security while login, security while storing.

5. ACKNOWLEDGMENTS

We thank to our project guide Prof.Kirti Deshpande who has guided us throughout and contributed towards research of this work.

6. REFERENCES

[1] M. Weir, S. Aggarwal, B. de Medeiros, and B. Glodek,"Password cracking using probabilistic context-free grammars," in Proc. 30th IEEE Symp. Security Privacy, 2009, pp. 391–405.

[2] K. Brown, "The dangers of weak hashes," SANS Institute InfoSec Reading Room, Maryland US, pp. 1–22, Nov. 2013,[Online]. Available: http://www.sans.org/reading-room/ whitepapers/authentication/dangers-weak-hashes-34412.

[3] F. Cohen, "The use of deception techniques: Honeypots and decoys," Handbook Inform. Security, vol. 3, pp. 646–655, 2006.

[4] M. H. Almeshekah, E. H. Spafford, and M. J. Atallah,"Improving security using deception," Center for Education and Research Information Assurance and Security, Purdue Univ., West Lafayette, IN, USA: Tech. Rep. CERIAS Tech. Rep. 2013-13, 2013.

[5] C. Herley and D. Florencio, "Protecting financial institutions from brute-force attacks," in Proc. 23rd Int. Inform. Security Conf., 2008, pp. 681–685.

[6] D. Mirante and C. Justin, "Understanding password database compromises," Dept. of Comput. Sci. Eng. Polytechnic Inst. of NYU, New York, NY, USA: Tech. Rep. TR-CSE-2013-02, 2013.

[7] A. Vance, "If your password is 123456, just make it hackme," New York Times, Jan. 2010.

[8] D. Quick, B. Martini, and K. R. Choo, Cloud StorageForensics.Syngress Publishing/ Elsevier,2014[Online] .Available: http//www.elsevier.com/books/cloud-storageforensics/quick/978-0-12-419970-5.

[9] K. R. Choo, M. Herman, M. Iorga, and B. Martini,"cloud forensics: State-of-the-art and future directions," Digital Investigation, vol. 18, pp. 77–78, 2016..

[10] K. R. Choo, J. Domingo-Ferrer, and L. Zhang, "Cloud cryptography: Theory, practice and future research directions," Future Generation Comp. Syst., vol. 62, pp. 51–53, 2016.

[11] D. Quick and K. R. Choo, "Google drive: Forensic analysis of data remnants," J. Network and Computer Applications, vol. 40, pp. 179– 193, 2014.

[12] Y. Yang, H. Zhu, H. Lu, J.Weng, Y. Zhang, and K. R. Choo, "Cloud based data sharing with fine-grained proxy re-encryption," Pervasive and Mobile Computing, vol. 28, pp. 122–134, 2016.