ENHANCEMENT OF USER AUTHENTICATION SYSTEM BY REDUCING THE STORAGE OF HONEYWORD SCHEME

Rupali Gholap[1], N. L. Bhale[2]

[1] ME Student, Department of Computer Engineering, MCOERC, Maharashtra, India
[2] HOD, Department of Information Technology, MCOERC, Maharashtra, India

ABSTRACT

User Authentication enhancement system implement the secret sharing based Hybrid Technique using Honeypot and Honey index to analyzed the security of the begin authentication system by reducing storage of honey word scheme. If attacker use the login or attempts to login then system generate the alarm,but not password disclouser detect. system is used when an adversary tries to enter into the system with a honeyword, an alarm is triggered to notify the administrator about a password leakage or tell user someone can access your account. Proposed system focus is on the use of fake passwords and accounts. The administrator creates fake accounts and detects a password disclosure, if any one of the honeypot account get used it will detect by admin. .We used password file Secret sharing. Storing H(P) instead of storing the password hash we creating the secret and storing secret servey. Main focus is used share values then no need to change the password when attacker to attack .

Keyword : *Authentication, honeypot, honeywords, login, passwords, password cracking, secret sharing Hash inversion process*.

1. INTRODUCTION

Every year new approach against cyber security threats are introduced. But simultaneously the adversary also create new techniques those overcome these efforts. So considering for security and data protection as a priority new techniques are needed. So, there is one of the important security issues is with disclosure of password file. There are two issues that should be considered to overcome these security problems:

1) Passwords must be protected by taking appropriate precautions and storing with their hash values computed through salting or some other complex mechanisms. Hence, for an adversary it must be hard to invert hashes to acquire plaintext passwords.
2) Second point is that a secure system should detect the entry of unauthorised user in the system. Point is that a secure system should detect whether a password file disclosure incident happened or not to take appropriate actions.

In authentication process it becomes difficult to handle security of passwords thats why password became the most important asset to authenticate. But users choose the passwords that are easy to remember that can be predicted by the attacker using diffierent attacks like brute force,dictionary, rainbow table attacks,DoS attack etc. So Honeywords plays an important role to defence against stole password files. Speciffically, fake passwords placed in the password file of an authentication server. In this focus on fake passwords or accounts as a simple and cost effective solution to detect passwords disclosure. Honeypot is one of the methods to identify occurrence of a password database breach. In this approach, the administrator purposely creates fake user accounts to attacker and detects a password disclosure, if any one of the honeypot passwords get used.

1.1 Need of the System

For each user account actual password is stored with honeywords. Adversary who steals afile of hashed passwords cannot be sure if it is the actual password or a honeyword for any account. If adversary enters the honeyword for login then it will trigger an alarm notifying the administrator that password file breach. In our system, if the number of attempts exceeds the count of three or entered password other than honeywords then the access will be issued but the files available will be decoy files. Also we suggest an alternative approach to provide realistic honeywords a perfectly at honeywords generation approach and also to reduce storage cost of the honeywords scheme The proposed system is used when an adversary tries to enter into the system with a honeyword, an alarm is triggered to notify the administrator about a password leakage or tell user someone can access your account. System focus is on the use of fake passwords and accounts. The administrator creates fake accounts and detects a password disclosure, if any one of the honeypot account get used it will detect by admin.

1.2 Certain criteria should be considered in these search scenarios:

1. Fake user account created(Honeypots) .
2. If adversary enters the honeyword for login then it will trigger an alarm notifying the administrator that password file detect.
3. And tell user someone can access your account.

1.3 Existing System:

In this system, analyze the honeyword approach and give some remarks about the security of the system. If attacker use the login or attempts to login then system generate the alarm,but not password disclouser detect. Furthermore, point out that the key item for this method is the generation algorithm of the honeywords such that they shall be in distinguish able from the correct passwords. Therefore, propose a new approach that uses passwords of other users in the system for honeyword sets, i.e. realistic honeywords are provided. Moreover, this technique also reduces the storage cost compared with the honeyword method in.

1.4 Proposed System:

We used password file Secret sharing.Storing H(P) instead of storing the password hash we creating the secret and storing secret servey.Main focus is used share values then no need to change the password when attacker to attack.

1.5 Terminology

a) Honeywords : Honeywords means decoy passwords or fake passwords. The idea is the insertion of fake passwords called as honeywords associated with each users account. When an attacker gets the password file, he recovers many passwords for each account and he cannot be sure about which word is genuine. Hence, the cracked password files can be detected by the system administrator if a login attempt is done with a honeyword by the adversary.For generating these Honeywords , use some methods these are Chaffing-by- tweaking, Chaffing-with-a-password-model, Chaffing with-Tough Nuts and Hybrid Method.These methods are useful and decrease the chances of guessing correct password.

b) Sweetwords : Honeywords and true passwords are placed into a list of Sweetwords.

c) Honeypots : A Honeypot is a security capability whose value is being probed, attacked or comprised. A honeypot is a secure source.A Honeypot could just as simply be one of your old PCs, a script or even a digital entity like some fabricated patient records. Whose value is being probed, attacked or comprised.

d) Honeyindex : Instead of honeywords we use honeyindexes, for every account we created a new and unique honeyindex. The correct honeyindex is store with the hash of the correct password in a list. In another list we have integer list with the username, the integer list is a honeyindexes of other accounts as well as their own account honeyindex this list is called as a honeyindex set.

e) Honeychecker : In our approach, the auxiliary service honeychecker is employed to store correct indexes for each account and we assume that it communicates with the main server through a secure channel in an authenticated manner. Honeychecker can be anything background service at remote secure location.

2. LITERATURE SURVEY

The most important concept is information security requirement in this which is secured using some authentication method. Various authentication method are existing such as Patterns, Passwords, PIN's etc.. Now-a-days most generally used technic for authentication is passwords. Security of password is an important part in security. A password is a secret word, which a user must input during a login, this word is match only after that it is possible to get access.Generally disclosure of password files is a several security problem that has affected millions of users and many companies and software industries store their data in database, Like facebook, Yahoo, RockYou, Gmail and Adobe. Generally user name and passwords are stored in a database. Since stolen passwords make the users target of many possible attacks. These recent events have proved that the weak password storage methods are currently used by many people on websites. For example, the LinkedIn passwords were using the SHA-1 algorithm without a salt and similarly the passwords in the eHarmony system were also stored using unsalted MD5 hashes.Once a password file is leakage, attacker by using the password cracking technique it is easy to capture most of the plaintext passwords. In this respect, there are two issues that should be considered to avoid these security problems: First, passwords must be protected by taking proper caution and storing with their hash values computed through some other correct complex mechanisms. Hence, for an advance it must be hard to include hashes value in plaintext passwords. The second point is that a secure system should detect whether a password file leakage incident happened or not to take appropriate actions. Honeypot is one of the methods to identify occurrence of a password database breach. In this approach, the administrator purposely creates deceit user accounts to lure adversaries and detects a password disclosure, if any one of the honeypot passwords get used. In the proposed system focus on the honeyindex and deal with fake passwords or accounts as a simple and cost effective solution to detect compromise of passwords. Many researchers have already worked for password security approach. Earlier, to protect online banking accounts from brute-force attacks, Herley and Florencio proposed a new approach to detect the malicious behaviour on every incorrect or unauthorized login. For every single user false login attempts with few passwords will generate honeypot accounts (fake accounts) so that malign behaviour is caught. Recently, Juels and Rivest have presented the honeyword mechanism to detect an adversary who attempts to login with cracked passwords. Imran Ergulers Achieving Flatness by Selecting the Honeywords from Existing User Passwords,this suggest an alternative approach that selects the honeywords from existing user passwords in the system in order to provide realistic honeywords a perfectly at honeyword generation method and also to reduce storage cost of the honeyword scheme. The propose system in concept is that for each username they build a set of honeyindexes in which one is real index and the others are false. When detecting the honeyword than alarm is trigger which noti_es the administrator about the password file breach.

2.1 Summery

In existing system honey indexing and honey pots are not used so storage cost is more. So proposed system implement the Hybrid Technique using Honey pot and Honey index to analyzed the security of the begin authentication system by reducing storage of honey word. User Authentication Enhancement through Honeywords using Hybrid method comes under the Data Mining.In this user enter the username and password at the time of login then system find out enter password is honeyword or not .It improve performance and efficiency
of the System.

3. DESIGNING AND MODELING

Using different modelling diagrams one can understand the system.

3.1 Use case diagram

The actor can be a human or an external system. In the Figure 3.1, User is the actor who by performing different tasks like login and other functions, user gets access to the system and the work is done.

Actors

1. User : Responsible for entering Password and waiting to get relevant result .
2. Hacker : Hacking the system.
3. System : Responsible for security of system.

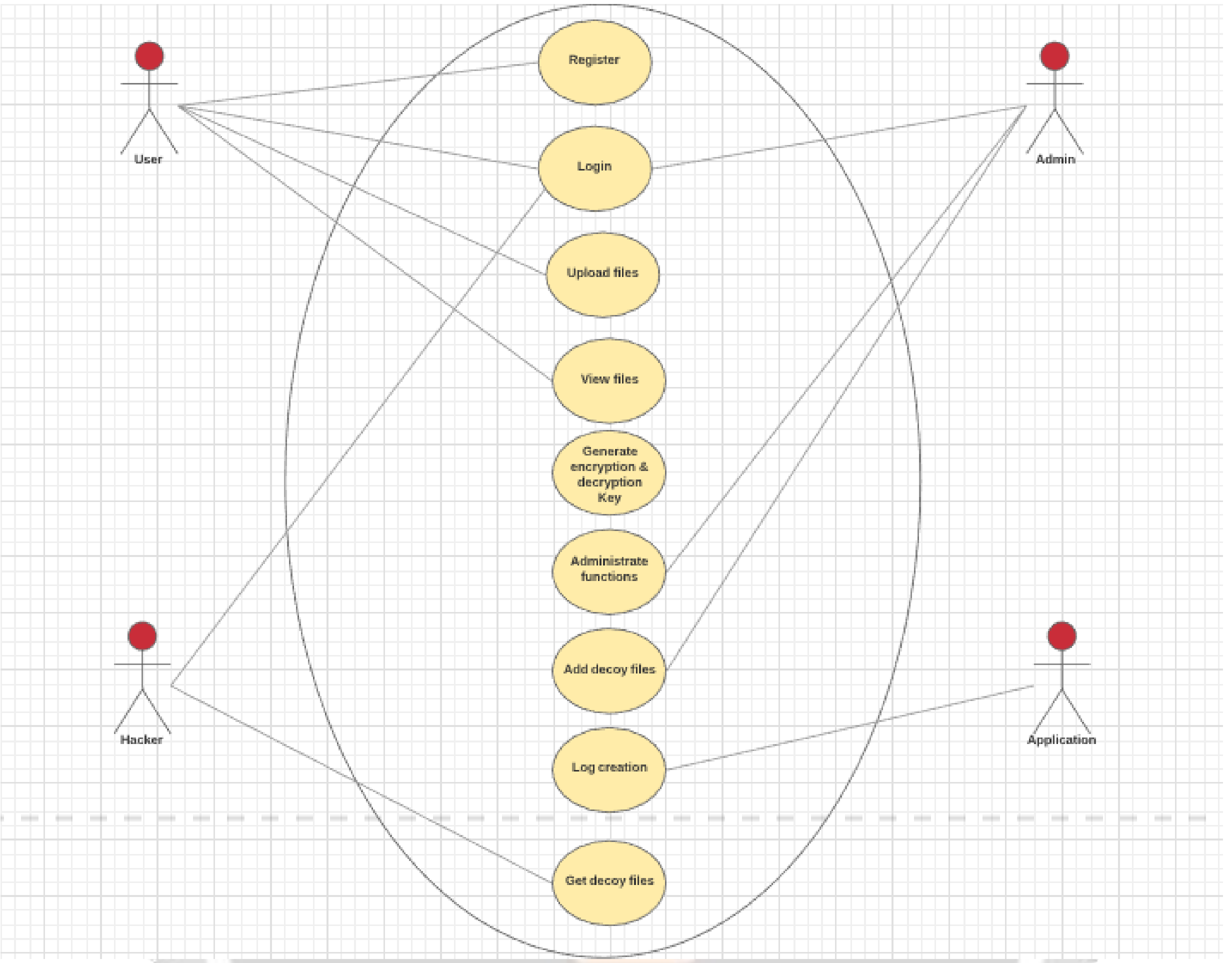


Fig -3.1 Use case diagram of system

3.2 Sequence Digram

This allows the speci_cation of simple runtime scenarios in a graphical manner.Figure 3.2 shows the sequence diagram for system.

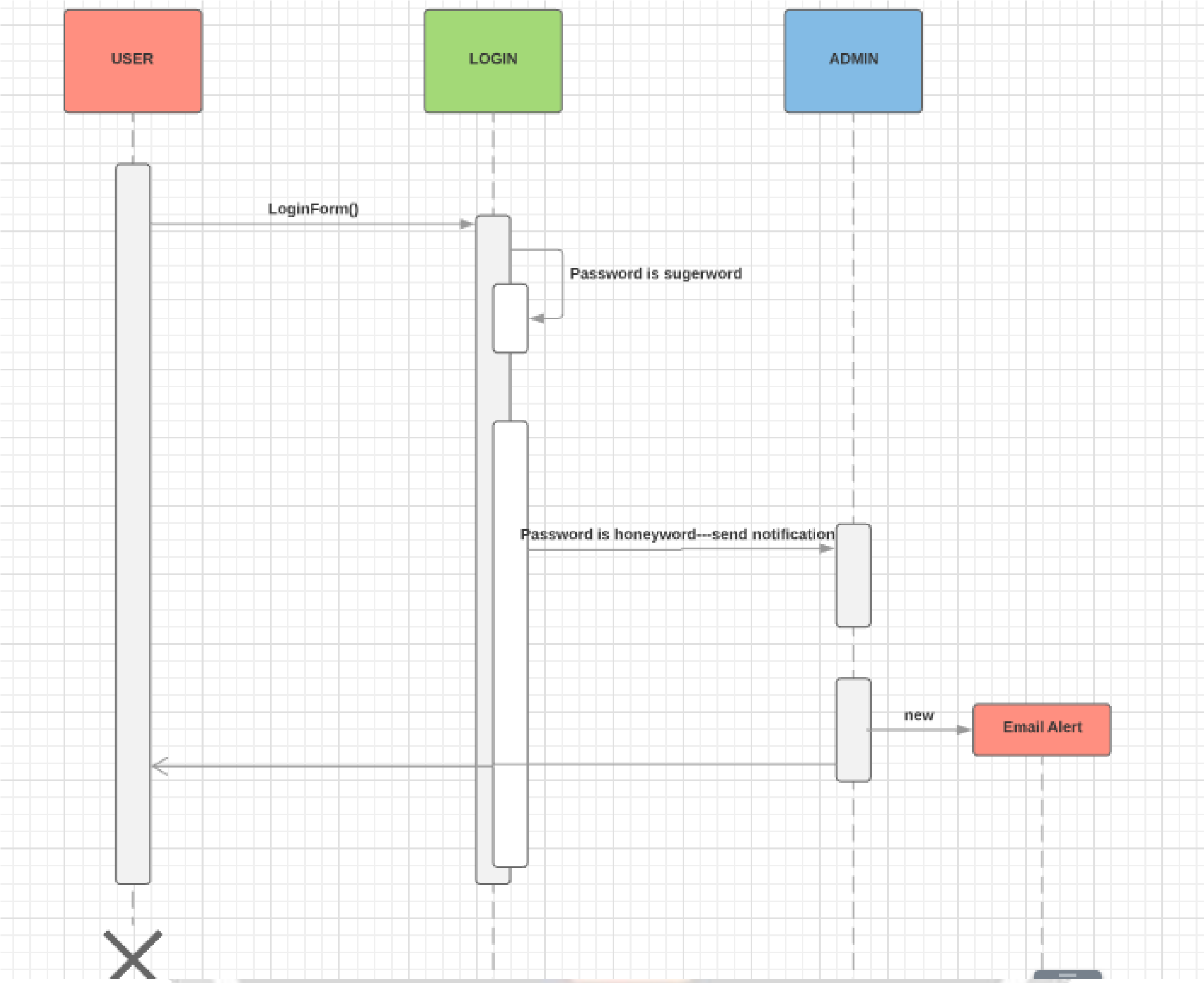


Fig-3.2. Sequence diagram

3.3 Design

Design phase is basically used to understand the data flow of the overall system. It includes Data flow diagram.

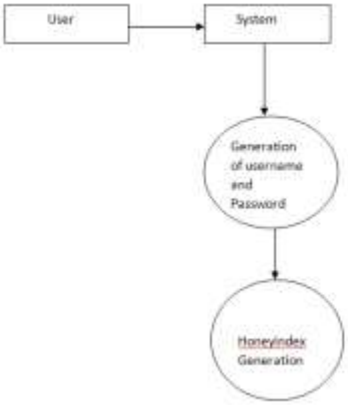


Fig-3.3 Data flow diagram level 0

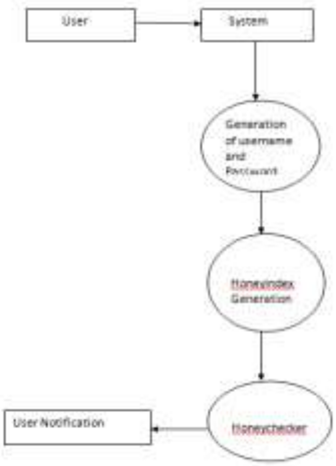


Fig-3.4. Data flow diagram level 1

4. SYSTEM ARCHITECTURE

In this there are 3 phases they are as follows:
1. Initialization
2. User Registration
3. User Authentication

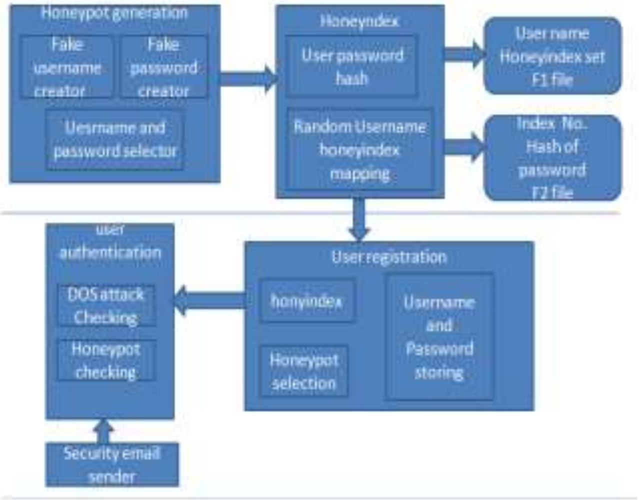


Fig-4.1. Architecture of system

A. Initialization:
In this include :
1) Honeypot generation: In this create a fake user name and fake password .Generate fake user name by using spam trap,spammer and online fake account generator. And generate fake password by using the honeypot generation methods.
2) HoneyIndex: After creating fake accounts (Honeypots) then create two password files F1 and F2. F1 stores the username and honeyindexsets. F2 stores the each user index number and its corresponding password hash value.

B. User Registration:
After the initialization process, system ready for user registration. In this phase, a legacy-UI is preferred, i.e a username and passwords are required from the user to register the system. Honeyindexes are generated periodically honeyindex set of each account should be regenerated.

C. User Authentication:
It includes:
1) Honeypot Checking: If user is going to Login into the System. If password matches with the hash password then user can Login. If hacker login to the system. Here if hacker tries to access the system and if he enters any honeyword then the notification or alert message is given to the Actual user about someone is trying to access your account. Here user is going to Login into the System. If password matches with the hash password then user can Login.
2) Security Email Sender: Here hacker login to the system. Here if hacker tries to access the system and if he enters any honeyword then the notification or alert message is given to the Actual user.

6. ALGORITHM FOR USER AUTHENTICATION

Steps :
1. Input :Enter username and password
2. Output:Successful Authentication.

```
Begin
T fake user account created(honeypots). K-1 randomly select and assign to user account and honeyindex set is build.
Xi=(Xi1,Xi2|{Xik-1)
One element is sugar index Ci.
Two _les are created F1 and F2.
F1=(Username , honeyindex set)
F2=(Index number , hash of password)
End
```

Shamirs secret sharing algorithm:

Step1: Decide secret

Step2: Decide threshold

Step3: Create polynomial

Step4: Draw graph

Step5: Plot points on the graph

5. CONCLUSIONS

In existing system honey indexing and honey pots are not used so storage cost is more. So proposed system implement the Hybrid Technique using Honey pot and Honey index to analyzed the security of the begin authentication system by reducing storage of honey word. In this proposed system we have analyzed the security of the honey word system. System make diffcult to attacker to get password from honey words, which is based on honey indexing. System architecture is finalized by using secret sharing based Hybrid Technique.

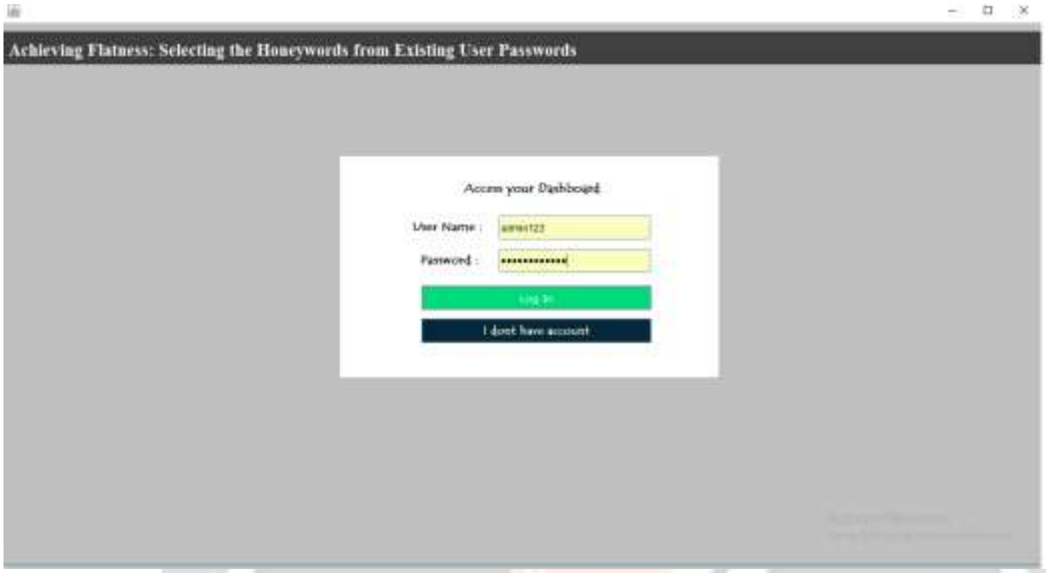
Future Work:

This work motivates by using parallel/distributed approach we can increase speed up of the system.

6. RESULTS AND ANALYSIS

1. Login Form:

Figure 6.1 shows the screenshot of Login page of the system.


Achieving Flatness: Selecting the Honeywords from Existing User Passwords
Access your Dashboard
Log In
I dont have account

User Name : admin123

Password : ••••••••••••

Fig-6.1.Login page of the system

2. Main Screen :

Figure 6.2 shows screen shot of Main Screen of system.

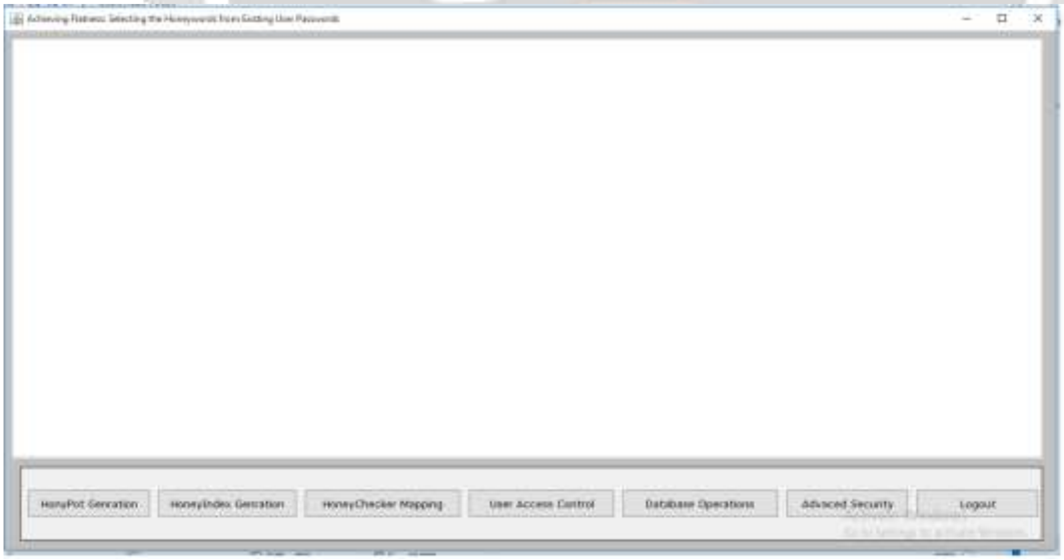
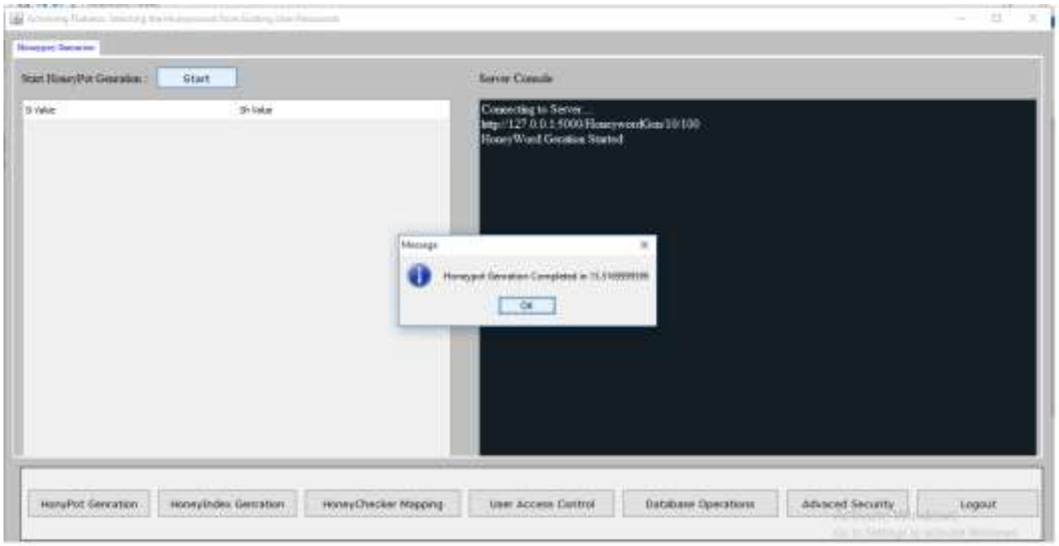


Fig6.2. Main screen of the system

3.Honeypot Generation :
Figure 7.4 shows screen shot of Honeypot generation of system.

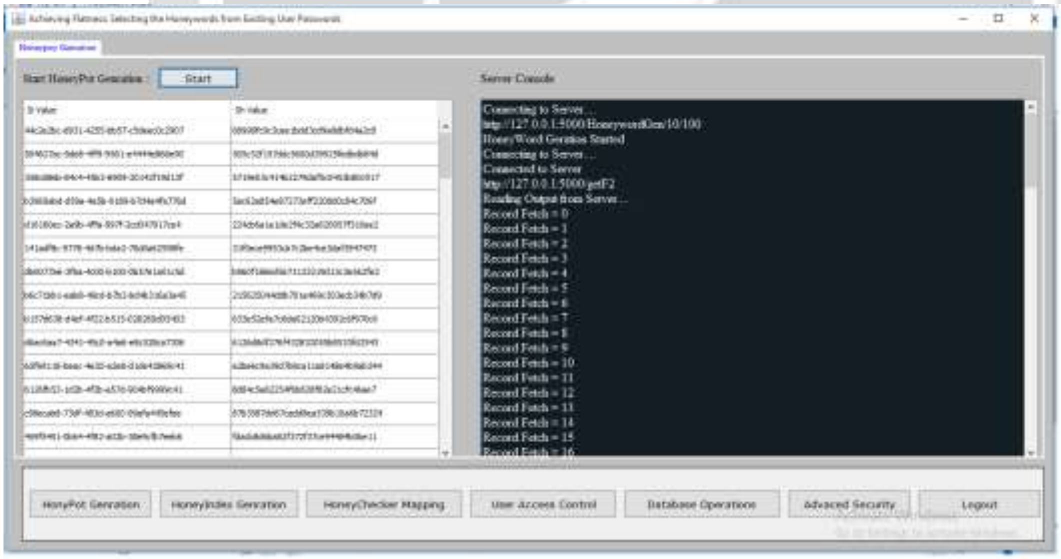


```
Connecting to Server....
http://127.0.0.1:5000/HoneywordGen/10/100
HoneyWord Genation Started
```

Honeypot Genration Completed in [illegible]

Fig6.3. Honeypot generation of the system

4.Honeypot Generation with Si and Sh value :
Figure 6.4 shows screen shot for Honeypot Generation with Si and Sh value.



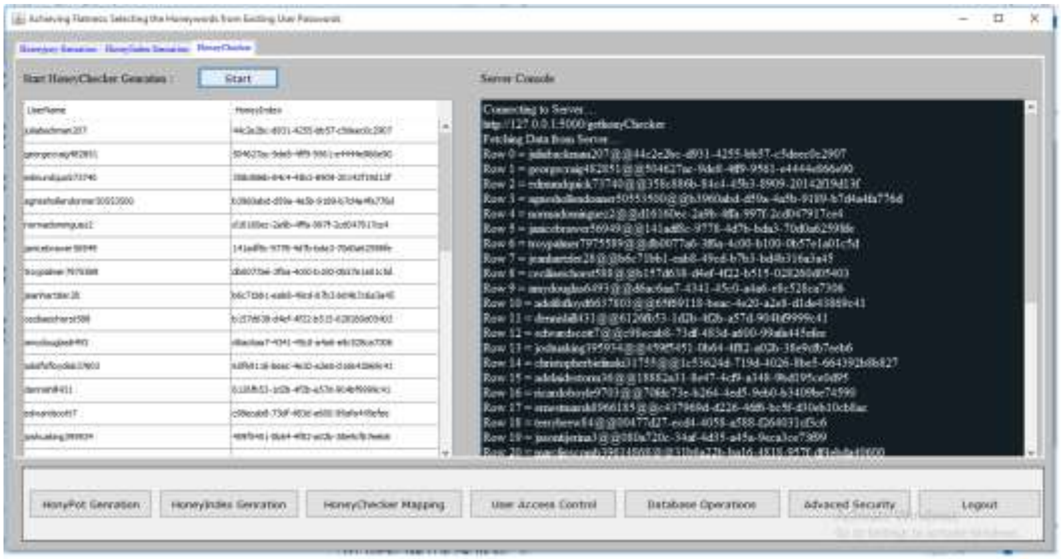
Si Value	Sh Value
44c2a2bc-d931-4255-8b57-c5deac0c2907	[illegible]
594627ac-5de8-4ff8-9301-e444ed60e90	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]
[illegible]	[illegible]

```
Connecting to Server....
http://127.0.0.1:5000/HoneywordGen/10/100
HoneyWord Genation Started
Connecting to Server....
Connected to Server
http://127.0.0.1:5000/getF2
Reading Output from Server....
Record Fetch = 0
Record Fetch = 1
Record Fetch = 2
Record Fetch = 3
Record Fetch = 4
Record Fetch = 5
Record Fetch = 6
Record Fetch = 7
Record Fetch = 8
Record Fetch = 9
Record Fetch = 10
Record Fetch = 11
Record Fetch = 12
Record Fetch = 13
Record Fetch = 14
Record Fetch = 15
Record Fetch = 16
```

Fig6.4. Honeypot Generation with Si and Sh value.

5.HoneyChecker Generation:

Figure 6.5 shows screen shot of HoneyChecker Generation.



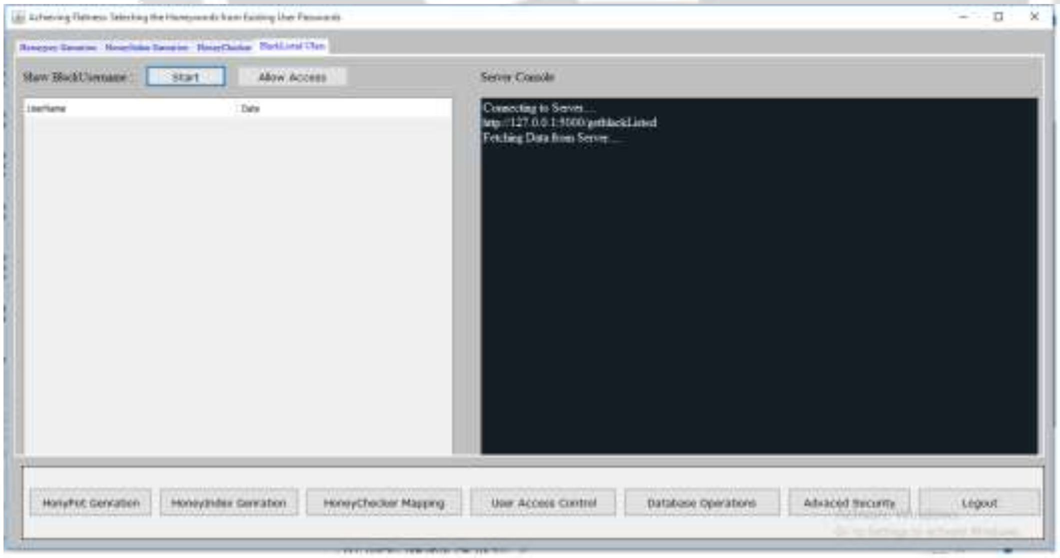
UserName	HoneyIndex
juliabackman207	44c2e2bc-d931-4255-bb57-c5deec0c2907
georgecraig482851	504627ac-9de8-4ff9-9561-e4444e866e90
edmundquick73740	358c886b-84c4-45b3-8909-20142f19d13f
agneshollendonner50553500	b3960abd-d59a-4a5b-9109-b7d4a4fa776d
normadominguez2	d16160ec-2a9b-4ffa-997f-2cd047917ce4
janicebrauer56949	141adf8c-9778-4d7b-bda3-70d0a6259fde
troypalmer7975589	db0077a6-3f6a-4c00-b100-0b57e1a01c5d
jeanhartzler28	b6c71bb1-eab8-49ed-b7b3-bd4b316a3a45
ceciliaschoerst588	b157d638-d4ef-4f22-b515-028280d05403
amydouglas6493	d6ac6aa7-4341-45c0-a4a6-e8c528ca7306
adolfofloyd6637803	6f489118-beac-4e20-a2e8-d1de43889c41
damianlall431	61206f53-1d2b-4f2b-a57d-904fd9999c41
edwardscott7	c98ecab8-73df-483d-a600-99afa445efec
joshuaking395934	459f5451-0b64-4f82-a02b-38e9db7eeb6

```
Connecting to Server...
http://127.0.0.1:5000/gethoneyChecker
Fetching Data from Server...
Row 0 = juliabackman207@@44c2e2bc-d931-4255-bb57-c5deec0c2907
Row 1 = georgecraig482851@@504627ac-9de8-4ff9-9561-e4444e866e90
Row 2 = edmundquick73740@@358c886b-84c4-45b3-8909-20142f19d13f
Row 3 = agneshollendonner50553500@@b3960abd-d59a-4a5b-9109-b7d4a4fa776d
Row 4 = normadominguez2@@d16160ec-2a9b-4ffa-997f-2cd047917ce4
Row 5 = janicebrauer56949@@141adf8c-9778-4d7b-bda3-70d0a6259fde
Row 6 = troypalmer7975589@@db0077a6-3f6a-4c00-b100-0b57e1a01c5d
Row 7 = jeanhartzler28@@b6c71bb1-eab8-49ed-b7b3-bd4b316a3a45
Row 8 = ceciliaschoerst588@@b157d638-d4ef-4f22-b515-028280d05403
Row 9 = amydouglas6493@@d6ac6aa7-4341-45c0-a4a6-e8c528ca7306
Row 10 = adolfofloyd6637803@@6f489118-beac-4e20-a2e8-d1de43889c41
Row 11 = damianlall431@@61206f53-1d2b-4f2b-a57d-904fd9999c41
Row 12 = edwardscott7@@c98ecab8-73df-483d-a600-99afa445efec
Row 13 = joshuaking395934@@459f5451-0b64-4f82-a02b-38e9db7eeb6
Row 14 = christopherbelmonte31755@@1c53624d-719d-4026-8be5-664392b8b827
Row 15 = adeladestorma36@@18882a31-8e47-4cf9-a148-9bd195ce0df95
Row 16 = ricardoboyle9703@@70fdc73e-b264-4ed5-9eb0-b3409be74590
Row 17 = ernestmarsh8966185@@c437969d-d226-46f6-bc5f-d30eb10cb8ac
Row 18 = terryherre84@@00477d27-ecd4-4058-a588-f264031d5c6
Row 19 = jasonbjerina3@@081a720c-34af-4d35-a45a-9eca3ce73899
Row 20 = [illegible]
```

Fig-6.5.HoneyChecker Generation

6.User Access Control :

Figure 7.8 shows screen shot for User Access Control.

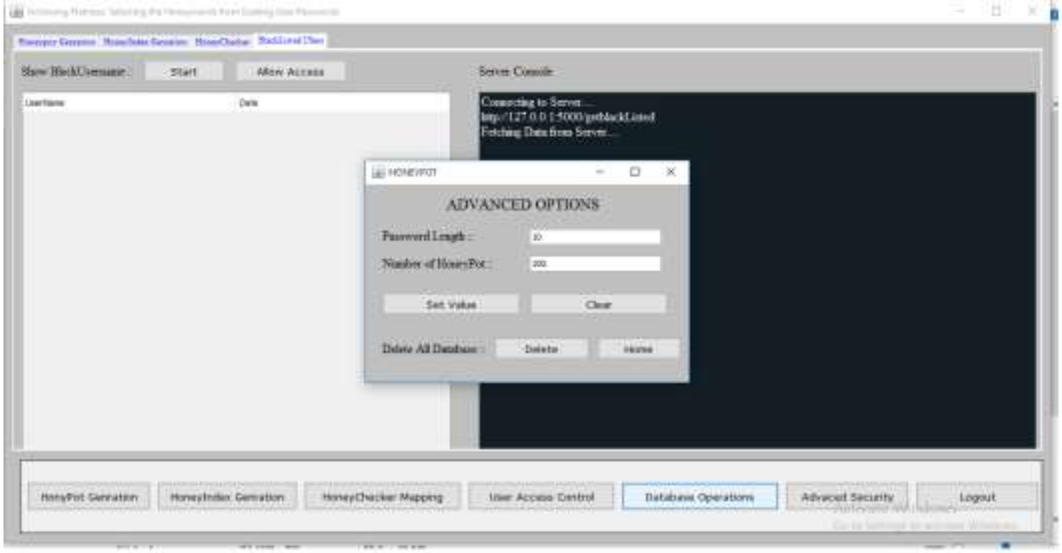


```
Connecting to Server...
http://127.0.0.1:5000/getblackListed
Fetching Data from Server...
```

Fig.6.6. User Access Control

7.Database Operation :

Figure 7.9 shows screen shot for databse operation .



```
Connecting to Server....
http://127.0.0.1:5000/getblackListed
Fetching Data from Server....
```

ADVANCED OPTIONS

Password Length : 10

Number of HoneyPot : 200

Fig-6.7: Clicking database operation show advance options

7. REFERENCES

[1]. A. Juels and R. L. Rivest, Honeywords: Making Passwordcracking Detectable, in Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security, ser. CCS 13. New York, NY, USA:
ACM, 2013, pp. 145160.

[2] D. Mirante and C. Justin, Understanding Password Database Compromises, Dept. of Computer Science and Engineering Polytechnic Inst. Of NYU, Tech. Rep. TR-CSE-2013-02, 2013.

[3] M.Weir, S. Aggarwal, B. de Medeiros, and B. Glodek, Password Cracking Using Probabilistic Context-Free Grammars, in Security and Privacy, 30th IEEE Symposium on. IEEE, 2009, pp. 391405

[4] F. Cohen, The Use of Deception Techniques: Honeypots and Decoys, Handbook of Information Security, vol. 3, pp. 646655, 2006.

[5] A. Juels and R. L. Rivest, Honeywords: Making Passwordcracking Detectable, in Proceedings of the 2013 ACM SIGSAC Conference on Computer Communications Security, ser. CCS 13. New York, NY, USA:
ACM, 2013, pp. 145160.

[6] C. Herley and D. Florencio, Protecting financial institutions from bruteforce attacks, in SEC08, 2008, pp. 681685.

[7] A. Pathak, An Analysis of Various Tools, Methods and Systems to Generate Fake Accounts for Social Media, Ph.D. dissertation, Northeastern
University Boston, 2014.

[8] D. Nagamalai, B. C. Dhinakaran, and J. K. Lee, An In-depth Analysis of Spam and Spammers, arXiv preprint arXiv:1012.1665, 2010.

[9] C. Biever, Project Honeypot to Trap Spammers, New scientist, no. 2485,