Enhanced Security Solution to Prevent Online Password Guessing Attacks

Nikitha Bhasu[1], Raju. K. Gopal[2]

[1](Computer Science and Engineering, Mar Baselios College of Engineering and Technology/Kerala University, India)

[2](Computer Science and Engineering, Mar Baselios College of Engineering and Technology / Kerala University, India)

ABSTRACT : *Brute force and dictionary attacks on password-only remote login service are very common and it is on the rise. Preventing such attacks by hackers is a complex problem. Automated Turing Test is an effective solution to identify and prevent automated malicious login attempts with minimal difficulties to users. In this paper, we discuss the insufficiency of existing and proposed login protocol designed to address extensive online dictionary attacks. We propose a new Enhanced Password Guessing Resistant Protocol (EPGRP), derived from revisiting prior proposals designed to restrict such attacks. While EPGRP limit the total number of login attempts from unknown remote hosts to as low as single attempt before being challenged with ATT. For enhancing the security, a One Time Password is also used in addition to ATT.*

Keywords - *online password guessing attacks, brute force attacks, dictionary attacks, ATTs, OTP.*

1. INTRODUCTION

The online password guessing attacks are unavoidable and it is increasing day by day. In the web applications and SSH logins, this attacks are usually observed. According to current report of SANS [6], a top cyber security problem is password guessing attacks on websites. However, the standard password authentication disallowed by SSH servers also suffer guessing attacks, e.g., keyboard interactive authentication [5]. When compared to offline attacks the online attacks have some disadvantages: easier detection is allowed when attacking systems must involve in an interactive protocol, and in many cases, before being locked out to answer Automated Turing Tests (ATTs, e.g., CAPTCHAs [4,5,7]) from a single machine, the attackers can try only limited number of guesses. As a result, to avoid detection or lock-out, attackers often must employ a huge number of systems called botnets. However by some cases, online attacks are much easier than before when the users normally choose common and relatively inadequate passwords, and attackers presently control huge botnets.

The valuable defense against automated online password guessing attacks is to avoid the number of failed trials to a very minimal without using ATTs. This will limit the bots used by attackers to few free password guesses for a targeted account. Nevertheless, this troubles the authentic user who then must answer an ATT on the next login attempt.

Other defense mechanisms for the online password guessing attacks are: allow logins without ATTs even if there are a given number of failed attempts; once the account is locked, allow login without ATTs after a time-out period. As a result of successful attacks which crack ATTs without human solvers, ATTs observed to be tougher for bots are being deployed. Also, we focus on decreasing user annoyance by challenging users with lesser ATTs, at the same time with more ATTs for bot logins. For limiting online guessing attacks using ATTs, there are two well-known proposals named Pinkas and Sander [1] (denoted PS), and van Oorschot and Stubblebine [2] (denoted VS) and the PGRP (Password Guessing Resistant Protocol) [8]. The PS proposal limits the number of ATTs sent to authentic users, but at some loss of security. The VS proposal reduces this but at a substantial cost to usability. The PGRP increases the security level as compared with PS and VS proposals. The proposal in the current paper, called Enhanced Password Guessing Resistant Protocol (EPGRP), significantly improves the security and usability, and can be extended beyond browser-based authentication.

EPGRP is developed originally from PGRP proposal. The PGRP is based on PS and VS proposals. In particular, to limit attack coming from a large botnets (e.g., comprising hundreds of thousands of bots), EPGRP enforces ATTs after a few (e.g., 4) failed login attempts. EPGRP imposes One Time Password for the registered user for

ensuring more protection. On the other hand, EPGRP allows a high number (e.g., 4) of failed attempts from known systems without answering any ATTs. We define known systems as those from which a successful login has occurred within a fixed duration of time. These are detected by their IP addresses saved in the login server as a white list, or cookies stored on client systems. A white-listed IP address and/or client cookie expires after a definite time.

EPGRP accommodates both graphical user interfaces (e.g., browser-based logins) and character-based interfaces (e.g., SSH logins). EPGRP uses either cookies or IP addresses, or both for marking authentic users. EPGRP increases the number of ATTs for password guessing attacks by tracking users through their IP addresses and decreases the number of ATTs if it is from a genuine login. In recent years, the tendency of logging into online accounts through multiple personal devices (e.g., PCs, laptops, smart phones) is raising. When used from a home setting, these devices often share a single public IP address (i.e., a simple NAT address) which makes IP-based history tracking more comfortable than cookies.

1.1 Contributions

i) User friendly ATT Based Scheme

Compared to commonly used methods such as the two earlier proposals and PGRP [1,2,8], the proposed EPGRP scheme is more obstructive against attackers. At the same time, EPGRP challenges with fewer ATTs for authentic users which include those who require multiple attempts to recall a password.

ii) Secure OTP Based Scheme

An OTP (One Time Password) is used as additional security authentication during the login process.

iii) Applicability to Web and Text Logins

EPGRP is not restricted to web only login (unlike proposals exclusively relying on browser cookies), and it uses IP address and/or other methods to detect a remote system in addition to cookies. By using text-based ATTs (e.g., textcaptcha.com), SSH login can be improved to use EPGRP.

2. LITERATURE SURVEY

2.1 Securing passwords against dictionary attacks

Our protocols may use several tests that attempt to distinguish between a human user and a computer program. These tests should be easy for human users to pass, yet be hard for automated programs. They were first suggested by Naor, and were denoted as reverse Turing tests (RTTs) [1]. Alternative names for these tests are CAPTCHAs, and mandatory human participation protocols.

A reverse Turing test should satisfy the following requirements:

(i) Automated generation: It should be easy to generate many instances of the test in an automated way.

(ii) Easy for humans: A generated test should be easy for human users to solve.

(iii) Hard for machines: An automated adversary is a program that cannot interact with a human user after receiving its input. We require that any automated adversary cannot answer the test correctly with probability that is significantly better than guessing. The input of the adversary can be of two types: (1) It can include a complete description of the algorithm that generates the RTTs. In this case the adversary can generate by itself many instances of the RTT together with their solutions. (2) A weaker notion of security is against adversaries that receive up to m examples of RTT instances and their solutions, where m is a parameter.

(iv) Small probability of guessing the answer correctly: We require that the probability that a random guess produces a correct answer to the test is small.

2.2 On countering online dictionary attacks with login histories and humans-in-the-loop

In this paper, we modify the protocol of Pinkas and Sander, presenting a new history-based protocol with ATTs. The new protocol offers opportunities for improved security and user-friendliness (e.g., fewer ATTs to legitimate users) and greater flexibility (e.g., allowing protocol parameter customization for particular situations and users). Also note that many ATT-based protocols, including that of Pinkas and Sander[2], are vulnerable to an ATT relay attack: ATT challenges may be relayed to unsuspecting ones, who produce responses, which are then transmitted back to the challenger. We explore this threat and

mechanisms to address it, and propose additional (orthogonal) enhancements to Pinkas-Sander type protocols. We discuss complementary techniques to address such attacks and to augment the security of the original protocol.

Here we modify the original protocol, intending to both progress the user experience and improve security, e.g., to improve the percentage of time that an adversary is challenged with an ATT, without further inconveniencing legitimate users. A major feature of our new protocol is the additional flexibility and configurability, including failed login thresholds and potentially lower ATT challenge probabilities. This allows the protocol to be tailored to match particular settings, classes of users, and applications; while determining the finest parameters for specific user profiles appears nontrivial. Another new aspect is storing cookies only on trustworthy machines. The new protocol can be parameterized to give the original protocol as a special case. A significant improvement of our protocol over prior work concerns protecting against relay attacks by forcing an ATT challenge on all login attempts after the number of failed logins reaches a threshold. Later work enabled an important fraction of the password space to be eliminated with an automated attack. Per-user failed login counts also provide protection against sweatshop attacks and ATT relay attacks, especially such attacks targeting a particular account.

2.3 How good are humans at solving captchas?

Completely Automated Public Turing tests to tell Computers and Humans Apart (CAPTCHAs) are widely used by websites to distinguish abusive programs from actual human users. Captchas normally present a user with a easy test like reading digits or listening to speech and then ask the user to type in what they saw or heard. The image or sound is usually distorted in various ways to make it difficult for a machine to execute the test. When efffective, captchas can stop a wide variety of abuses, such as incorrect account invention and spam comments on blogs and forums [3]. Captchas are intended to be easy for humans to perform, and difficult for machines to perform.

However, most recent research has focused only on making them hard for systems. In this paper, we present what is to the best of our experience the first huge scale evaluation of captchas from the human view point, with the goal of assessing how much friction captchas present to the average user. For the target of this study we have asked workers from Amazon's Mechanical Turk and an underground captcha- breaking service to solve more than 318,000 captchas issued from the 21 most popular captcha schemes (13 images schemes and 8 audio scheme). Analysis of the resulting data shows that captchas are often complex for humans, with audio captchas being mostly problematic. We also find some demographic styles indicating, for example, that non-native speakers of English are slower in typical and less exact on English-centric captcha schemes. Confirmation from a week's worth of eBay captchas (14,000,000 samples) suggests that the solving accuracies found in our study are close to real-world values, and that refining audio captchas should become a priority, as nearly 1% of all captchas are delivered as audio rather than images. Finally our study also shows that it is more useful for an attacker to use Mechanical Turk to solve captchas than an underground service.

2.4 Revisiting defenses against large-scale online password guessing attacks

The proposal in the current paper, called Password Guessing Resistant Protocol (PGRP), significantly improves the security and usability, and can be more normally deployed beyond browser-based certification. PGRP builds on proposals called PS and VS proposal. This new proposal has no use of captcha's and it shows more secure ATTs than captchas. In particular, to limit attackers in control of a large botnet (e.g., comprising hundreds of thousands of bots), PGRP enforces ATTs after a few (e.g., 3) failed login attempts are made from unknown systems. However, PGRP allows a large number (e.g., 30) of failed attempts from known machines without answering any ATTs. We define known systems as those from which a successful login has occurred within a fixed duration of time. These are specified by their IP addresses saved on the login server as a white list, or (as in PS [1]) cookies stored on client systems. A white-listed IP address and/or user cookie expires after a certain time.

PGRP accommodates both graphical user interfaces (e.g., browser-based logins) and character-based interfaces (e.g., SSH logins), while

the previous protocols deal exclusively with the previous, requiring the use of cookies. PGRP uses cookies or IP addresses, or both for tracking legitimate users [8]. Tracking users through their IP addresses also allows PGRP to increase the number of ATTs for password guessing attacks and meanwhile to decrease the number of ATTs for legitimate login attempts. Although NATs and web proxies may (slightly) reduce the utility of IP address information, in practice, the handling of IP addresses for client identification appears feasible [7]. In recent years, the trend of logging in to online accounts through multiple personal devices (e.g., PCs, laptops, smart phones) is rising. When used from a home setting, these devices often share a single public IP address (i.e., a simple NAT address) which makes IP-based history tracking more user friendly than cookies. For example, cookies must be stored, although clearly to the user, in all devices used for login.
The objectives for PGRP include the following:

1. The login procedure should make brute force and dictionary attacks ineffective even for adversaries with access to large botnets (i.e., capable of launching the attack from many remote hosts).
2. The protocol should not have any significant impact on usability (user suitability). For example: for genuine users, any additional steps besides entering login authorization should be lesser. Increasing the security of the protocol must have minimal effect in limiting the login usability.
3. The protocol should be comfortable to deploy and scalable, requiring lesser computational resources in conditions of memory, space etc.

3. EXISTING SYSTEM

Nowadays some protocols are used to avoid the password attacks from adversaries. Many existing methods and schemes involve ATTs, with the fundamental assumption that these challenges are sufficiently difficult for bots and easy for most people. However, users increasingly disgust ATTs as these are perceived as extra steps. Due to successful attacks which break ATTs without human solvers, ATTs perceived to be tougher for bots are being deployed. As a significance of this arms-race, present-day ATTs are becoming complicated for human users, fueling an increasing tension between security and usability of ATTs. Therefore, we focus on decreasing user annoyance by challenging users with minimum ATTs, while at the similar time subjecting bot logins to more ATTs, to drive up the financial cost to attackers.
(i) Many existing techniques and proposals involve ATTs, with these challenges are sufficiently difficult for bots.
(ii) Present day ATTs consist of several steps, make it difficult to user for quick login purposes.
(iii) The existing techniques like CAPTCHAs [4, 5, and 7], ATTs are becoming complex for human users.
(iv) PS proposal: It reduces the number of ATTs sent to original users, but at some loss of security.
(v) VS proposal: This proposal may require all users to answer ATTs in certain circumstances.
(vi) PGRP: It enforces ATT after less failed login efforts from unknown systems and allows a high number of failed attempts from known systems without answering any ATT.

4. PROPOSED SYSTEM

4.1 Goals
The objectives for EPGRP include the following:
i) To make login process secure for authentic users and to make it tough for adversaries.
ii) The protocol should guarantee much usability.
iii) The protocol should be scalable and requiring less system assets in terms of memory, processing time, and disk space.
4.1.1. Assumptions

We suppose that opponents can solve a less percentage of ATTs e.g., through bots, brute force mechanisms & less paid workers. Instances of attackers using IP addresses of known systems and cookie theft for targeted password guessing are supposed to be minimal. For any untrusted situations the traditional password-based authentication is not suitable. (e.g., a key logger may record all keystrokes, including passwords in a machine, and direct those to a remote attacker). We can't avoid such existing attacks in any untrusted environments. Also the data integrity of cookies must be confirmed.
4.1.2 Outline

The overall idea behind EPGRP (see Fig.1) is that except for the below two cases, an ATT challenge is presented during the login

process:1) when the number of failed login attempts for a given username is very least; and 2) when the user has successfully logged previously.

In distinction to earlier protocols, EPGRP uses IP addresses, cookies, or both to detect systems from which users have been effectually validated. The evaluation need an ATT challenge, upon receiving invalid authorization, and it is based on the received cookie and/or the users IP address. Also, if the number of failed login attempts for a specific username is lower than a threshold, the user is not demand to answer an ATT challenge even if the login attempt is from a new system for the first time.

4.2 Data Structures and Function Explanation

4.2.1 Data Structures

EPGRP maintains three data structures:

1. W: It contains a list of {source IP address, username} pairs. For each pair, a effective login from the source IP address has been introduced for the username in the earlier.
2. FT: In this table, each entry represents the number of failed login attempts for a valid username, un. It records the maximum of k2 failed login attempts. Approaching a non-existing index returns 0.
3. FS: In this table, each entry represents the number of failed login attempts for each pair of (srcIP, un). Here, for a host in W or a host with a valid cookie, srcIP is the IP address and un is a valid username attempted from srcIP. It records a maximum of k1 failed login attempts; crossing this threshold may pass an ATT. After a successful login attempt an entry is set to 0. Approaching a non-existing index returns 0. There is a "write-expiry" interval for each entry in W, FT, and FS. It shows that the entry is deleted when the given duration of time (t1, t2, or t3) has lapsed since the final time the entry was inserted or modified.

4.2.2 Functions

EPGRP uses the following functions (IN stands for input and OUT stands for output):

1. ReadCredential (OUT: un, pw, cookie). It represents a login prompt. It returns the entered username and password, and the cookie obtained from the user's browser (if any).
2. LoginCorrect (IN: un,pw; OUT: true/false). If the available username-password pair is correct, the function yield true; else, it yield false.
3. GrantAccess (IN: un,cookie). To the user's browser, the function sends the cookie and then enables access to the identified user account.
4. Message (IN: text). Display a text message.
5. OTPChallenge (OUT:Pass/Fail). Encounters the user with an ATT and returns "Pass" if the answer is correct; else, it yield "Fail."
6. ATTChallenge (OUT:Pass/Fail). Encounters the user with an ATT and returns "Pass" if the answer is correct; else, it returns "Fail."
7. ValidUsername (IN: un; OUT: true/false). If the granted username exists in the login system, the function yield true; else, it yields false.

```
Input:
t1, t2, t3, k1 (def=2), k2 (def=2)
//k1,k2 ≥ 0
un, pw, cookie // username, password, and user's
browser cookie if any
W (global variable, expires after t1) // white list of
IP addresses with successful login
 FT (global variable, expires after t1) // table of
number of failed logins per username
FS (global variable, expires after t1) // table of
number of failed logins indexed by (srcIP,
username) for hosts in W or hosts with valid
cookies
1.  begin
2.  ReadCredential (un, pw, cookie)
3.  if LoginCorrrect (un, pw) then
4.   if(( Validcookie, un, k1, true)□((srcIP, un)
    ∈W)) □ (FS[srcIP,un<k1)) □ (FT[un]<k2))
    then
5.      FS[srcIP,un] = 0
6.      Add srcIP to W
7.    GrantAccess( un, cookie)
8.   else
9.    if(OTPChallenge()=Pass) then
10.       FS(srcIP, un] = 0
11.       Add srcIP to W
12.       GrantAccess(un, cookie)
13.     else
14.        Message('OTP challenge is improper')
15.  else
16.  if(( Validcookie, un, k1, false)□((srcIP, un)
    ∈W)) □ (FS[srcIP,un<k1)) then
17.        FS[srcIP, un] = FS[srcIP, un] + 1
18.        Message('username or password is
    improper')
19. else if ( ValidUsername(un) □ (FT[un]<k2))
    then
20.     FT[un] = FT[un] +1
21.     Message(' username or password is
    improper');
22.  else
23.     if (ATTChallenge()=Pass) then
24.       Message(' username or password is
    improper')
25.  else
26.       Message('ATT challenge is improper')
```

Fig.1 Enhanced Password Guessing Resistant Protocol

8. Valid (IN:cookie,un,k1,state; OUT: cookie, true/false). First, the function checks the validity of the cookie (if any) where it is considered incorrect in the following cases: 1) the login username does not equivalent the cookie username; 2) the cookie is expired/time-out; or 3) the cookie counter is equal to or higher than k1. Only when a authorized cookie is received, the function returns true. If state= true (i.e., the entered user information are valid, as in line 4 of Fig. 1), a new cookie is created (if cookies are supported in the login system) including the following data: username, expiry date, and a counter of the number of failed login attempts. Observe that the function does not send the created cookie to the user's browser then the state= true. Rather, the cookie is sent later by the GrantAccess() function. If state= false (i.e., the entered user information are invalid, as in line 23 of Fig. 1) and a authorized cookie is obtained, the cookie counter is increased by one and the cookie is sent back to the user's browser.

4.3 Cookie versus Source IP Addresses

If the login server offers a web-based interface, browser cookies have a better choice for this process. If no cookie is sent by the user browser to the login server after a successful login, the server sends a cookie to the browser to identify the user on the next login attempt. However it is difficult for the login server to identify the remote user, if the user uses multiple browsers or more than one OS on the same machine. Cookies may be deleted automatically as enabled by the browser, or deleted manually by the users. Also, cookie theft (e.g., through session hijacking) might enable an opponent to impersonate a user who has been successfully authenticated in the past. However, for using cookies the user interface must be a browser (which, e.g., is not applicable to SSH).

Alternatively, a user system can be identified by the source IP addresses to trace users may result in invalid recognition for several reasons, including: 1) same system might be assigned with different IP addresses over time; and 2) a group of systems might be represented by a smaller number or even a single public IP address, if a NAT mechanism is in place.

The drawbacks of identifying a user by means of either a browser cookie or a source IP address include:1) failing to specify a system from which the user has authenticated successfully in the former; and 2) wrongly identifying a system that the user has not authenticated before. Case 1) reduces usability since the user might be asked to answer an ATT challenge for both valid and invalid login authorization. On the other hand, case 2)

affects security since some users/ attackers may not be asked to answer an ATT challenge whereas they have not logged in successfully from those systems in the past. However, the chance of launching a dictionary or brute force attack from these systems appears to be minimal. First, for identification through cookies, a directed attack to giveaway users' cookies is required by an opponent. Second, for identification through IP addresses, the opponent must have access to a system in the similar subnet as the user.

In EPGRP to minimize user inconvenience during the login process, we choose to use both browser cookies and source IP address (or only one of them if the other is not applicable). Also, by using IP address only, EPGRP can be used in character-based login boundary such as SSH. If EPGRP using text-based ATTs, an SSH server can be adapted to use (e.g., textcaptcha.com). For example, a pattern of a text- based CAPTCHA for SSH is accessible as a source code patch for OpenSSH .

4.4 Decision Function for Requesting ATTs

Below we discuss issues related to ATT challenges as provided by the login server in Fig.1. To challenge the user with an ATT belongs on two factors: 1) whether the user has authenticated successfully from the same machine previously; and 2) the total number of failed login attempts for a specific user account. For definitions of W, FT, and FS, see Section 4.2

4.4.1 Valid Username-Password Pair

According to the condition in line 4, upon entering a valid username-password pair, the user will not be requested to answer an ATT challenge in the following cases:

1. A valid cookie is received from the user system if the function Valid returns true and the number of failed login attempt s from the user system's IP address for that username, FS[srcIP,un] is less than k1 over a time duration decided by t3.
2. The user systems IP address is in the white list W and the number of failed login attempts from this IP address for that username, FS[srcIP,un], is less than k1 over a time duration decided by t3;
3. The number of failed login attempts from any machine for that username, FT[un], is below a threshold k2 over a time duration decided by t2.

A user tries to login from a new system/IP address for the initial time before k2 is reached to proceed without an ATT are shown in the last case. On the other hand, if the number of failed login attempts for the username exceeds the threshold k2, indicate a guessing attack. Hence the user must pass an ATT Challenge.

4.4.2 Invalid Username-Password Pair

Upon entering a invalid username-password pair, the user will not be requested to answer an ATT challenge in the following cases:

1. A valid cookie is received from the user system if the function Valid returns. Also the number of failed login attempts from the user systems IP address for that username, FS[srcIP,un] is less than k1 over a time duration decided by t3.
2. The user systems IP address is in the white list W and the number of failed login attempts from this IP address for that username, FS[srcIP,un], is less than k1 over a time duration decided by t3.
3. The number of failed login attempts from any machine for that username, FT[un], is below a threshold k2 over a time duration decided by t2.

From a user a failed login attempt with a valid cookie or in the white list W will not increase the total number of failed login attempts in the FT table since it is expected the genuine users may possibly forget or mistype their password (line 16-18). However, if the user machine is identified by a cookie, an equivalent counter of the failed login attempts in the cookie will be updated. Also, the FS entry indexed by the {source IP address, username} pair will also be incremented (line 17). Once the cookie or the equivalent FS entry hits or go beyond the threshold k1 (default value 2), the user must correctly answer an ATT Challenge.

4.4.3 Messages Showing Output

EPGRP displays different messages in case of invalid {username, password} pair & invalid answer to given ATT Challenge. While showing a human that the entered {username, password} pair is invalid, an automated program reluctant to answer the ATT Challenge cannot approve whether it is the pair or the ATT that was incorrect. On the other hand, while this is more suitable for genuine users, it gives more information to the attacker about the answered ATTs.

4.4.4 Why Not to Black-List Offending IP Addresses

We choose not to create a blacklist for IP addresses making many failed login attempts for

the following reasons: 1) this list may consume considerable memory; 2) genuine users from blacklisted IP addresses could be blocked (e.g., using compromised machines); and 3) hosts using dynamic IP addresses seem more attractive targets (compared to hosts with static IP addresses for opponents to launch their attacks from (e.g., spammers [16]). If the cookie mechanism is not available for the login server, PGRP can manage by using only source IP addresses to keep track of user systems.

5. METHODOLOGY

The Fig.2 shows the operation done by the new proposal. This is an authentication system. This trusted system shows the conditions for genuine users and hackers. The genuine users are secured by using the EPGRP algorithm. This algorithm displays the use of ATT and OTP.

The ATT (Automated Turing Test) used are the secret questions and their answers. It creates more complexity for hackers. The OTP (One Time Password) creates authorization to the legitimate users. This OTP is sent to the mobile of the genuine users while attempt to login. This OTP protect the users from hackers. For hackers the ATT and OTP are unknown. So these two challenges create more and more hurdles to the genuine users.

Fig. 2 Working of Proposed System

6. IMPLEMENTATION DETAILS

The implementation done on this new proposal is given below:

6.1 Cookie Management

Read cookie from request. Then check if login username and cookie username are same. The condition is true then checks if the cookie is expired or not. If the condition is false then return false. If the cookie is expired then return false. If cookie is not expired then check if the cookie counter is greater than or equal to k1 (no: of failed login attempt) and then the condition is true then return false. If it is false then return true.

6.2 IP Address Checking

Check if the current IP Address is in the White list or not. If the login attempts are from new 5.3 Base Application

Banking application is shown as demo. The employee registration is shown and customer addition and approval is done here. The account addition and approval is shown.

6.3 ATT Challenge

The secret questions and answers are set as ATT Challenge at the time of registration of the user. If login from registered IP the password incorrect for four times then ATT Challenge is given.

6.4 OTP Challenge

The user try to login from a new IP and the username and password is correct. Then One Time Password is send to the registered mobile number of the user. For a genuine user he/she knows the OTP and successfully login. For a hacker it is difficult to answer the OTP and failed to login. For incorrect OTP, next time also ask the OTP.

6.5 EPGRP Algorithm implementation

Enhanced Password Guessing Resistant Protocol (EPGRP) algorithm is implemented here. Integration of all the mechanisms is done in this section. The working of algorithm is done in this main part. The implementation section is completed in this part.

7. RESULTS

The new proposed EPGRP algorithm is demonstrated by using an application. I use a banking application for showing the operations of this new proposal. This authentication system demonstrates the login experience of both genuine users and hackers. For genuine users this trusted system provides a hassle-free login experience without compromising the security. In the case of

hackers this system creates more and more complexity during the login process. The system keeps track of the IP addresses of the users and validates it during authentication process. If there is any doubt regarding the IP address then it challenges with an OTP. Otherwise it asks ATT for the incorrect password/username. The cases shown in the result section are as follows:

7.1 Genuine user login with his/her home PC

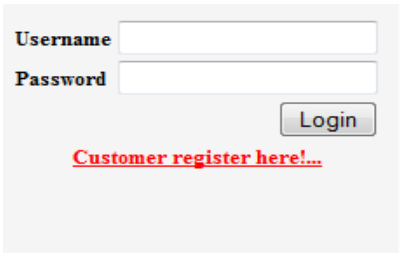

Fig.3 Login Page

The fig.3 shows that if a trusted user login with correct username and password by his/her PC then he/she is successfully login

7.2 User login with public PC

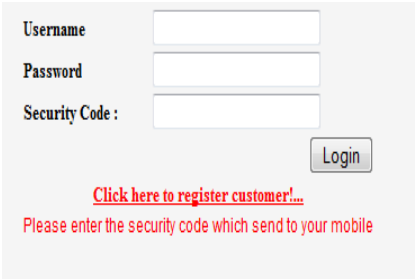

Fig. 4 Display of security code

The fig.4 shows that if a user login with public PC then at the first login attempt a security code is asked as One Time Password. If the user successfully answered this security code then he/she is allowed to login.

7.2.1 Incorrect login with public PC

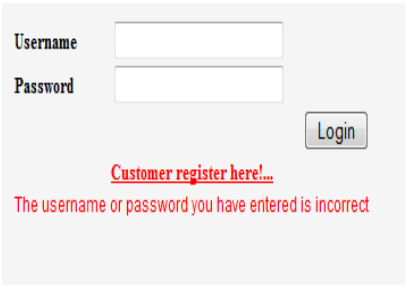

Fig. 5 Display of incorrect username & password

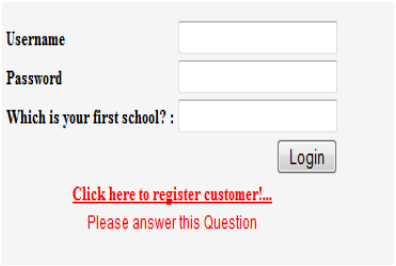

Fig. 6 Display of ATT Challenge

Fig 5 and fig.6 shows that, while a hacker tries to login, he/she guess many passwords during the login attempt. If the password is incorrect for two consecutive times, an ATT question is asked and an incorrect answer results in unsuccessful login.

7.3 Login page details

(i) Home Page

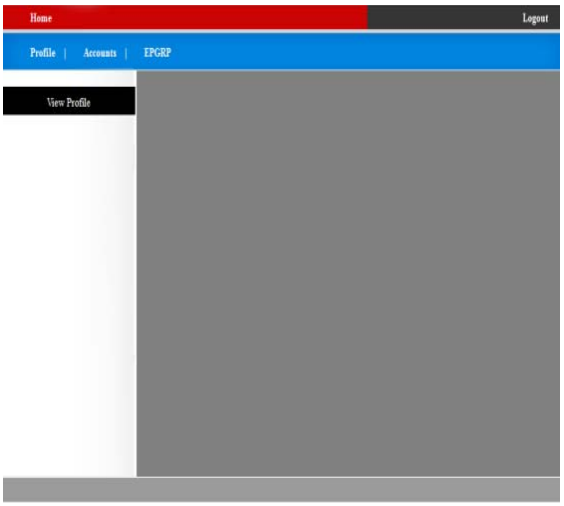

Fig.7 Display of Home Page

The fig.7 shows the details of profile, account and EPGRP algorithm based data.

(ii) EPGRP Page

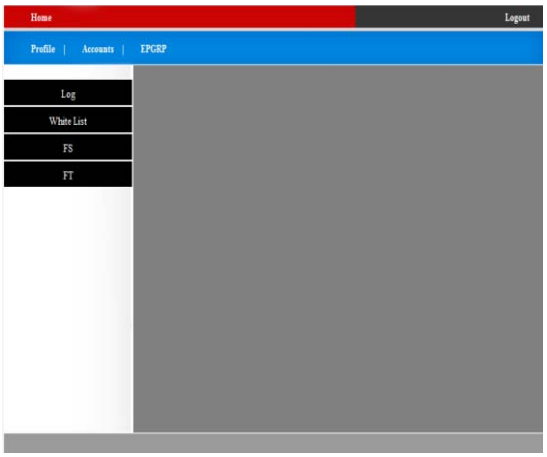

Fig. 8 Details of EPGRP Page

The fig.8 shows the log, white list, FS and FT.

(iii) Log Page

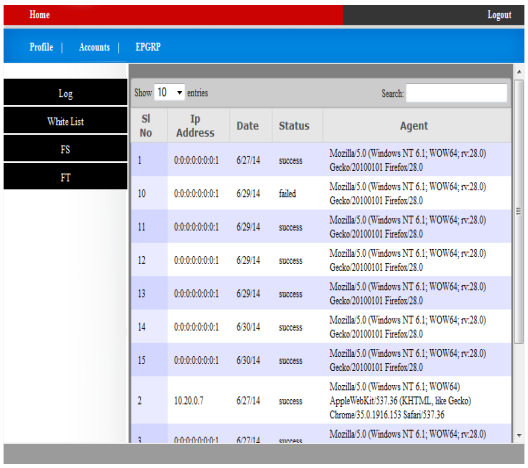

Fig.9 Display of log page

The fig.9 shows Log details. This log page contains the IP addresses, date, status (success/failed), agent.

(iv) White List Page

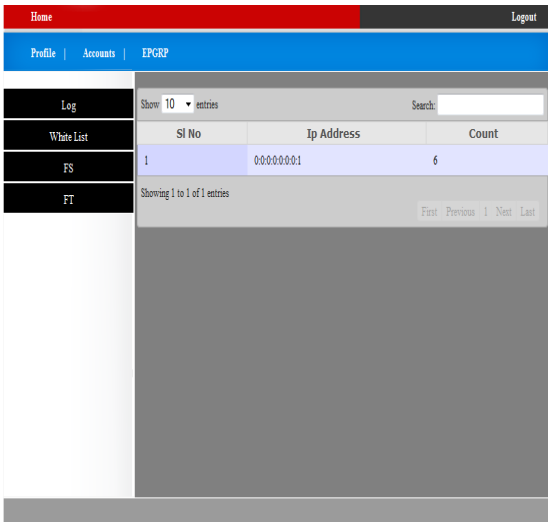

Fig. 10 Display of white list Page

The fig.10 shows the details of IP address of users and the count.

(v) FS Count Page

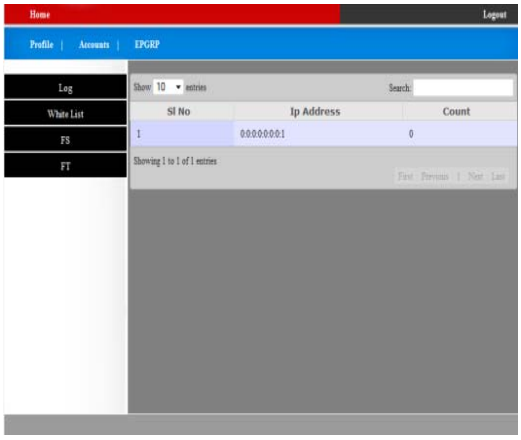

Fig.11 Display of FS count page

The fig.11 shows the number of failed login attempt per {source IP, username}.

(vi) FT Count Page

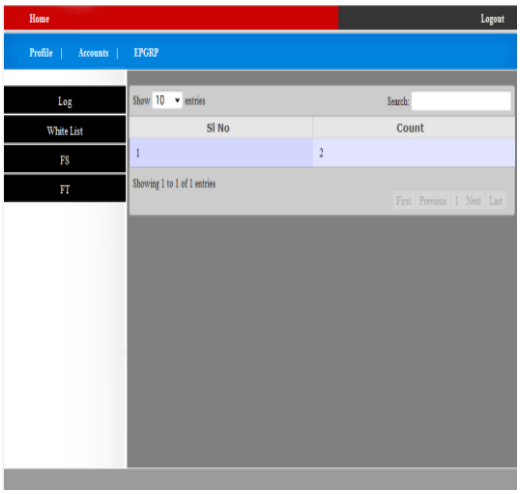

Fig.12 Display of FT count page

The fig.12 shows the number of failed login attempt per username.

8. FUTURE ENHANCEMENT

The future enhancement applicable to this proposal is MAC addresses also can be added in addition to the IP addresses of the hosts.

9. CONCLUSION

The online password guessing attacks on password-only systems have been observed for decades. Nowadays, attackers targeting such systems are empowered by having control of thousand to million botnets. In former ATT-based login protocols, there exists security-usability reciprocity with respect to number of free failed login attempts contrasted with user login suitability. In difference, while safely attempts a large number of free failed attempts for legitimate users, EPGRP are more preventive against brute force and dictionary attacks. In preventing password guessing attacks the EPGRP is more efficient and it also provides a hassle-free login experience for the legitimate user.

EPGRP is suitable for all organizations with varying number of user accounts with different security and sensitivity levels. EPGRP can also be used with remote login services where cookies are not relevant.

REFERENCES

[1] B. Pinkas and T. Sander, "Securing Passwords against Dictionary Attacks," Proc. ACM Conf. Computer and Comm. Security (CCS '02), pp. 161-170, Nov. 2002.

[2] P.C. van Oorschot and S. Stubblebine, "On Countering Online Dictionary Attacks with Login Histories and Humans-in-the- Loop," ACM Trans. Information and System Security, vol. 9, no. 3, pp. 235-258, 2006.

[3] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How Dynamic Are IP Addresses?," SIGCOMM Computer Comm. Rev., vol. 37, no. 4, pp. 301-312, 2007.

[4] J. Yan and A.S.E. Ahmad, "Usability of CAPTCHAs or Usability Issues in CAPTCHA Design," Proc. Symp. Usable Privacy and Security (SOUPS '08), pp. 44-52, July 2008.

[5] J. Yan and A.S.E. Ahmad, "A Low-Cost Attack on a Microsoft CAPTCHA," Proc. ACM Computer and Comm. Security (CCS '08), pp. 543-554, Oct. 2008.

[6] "The Top Cyber Security Risks," SANS.org, http://www.sans. org/top-cyber-security-risks/, Sept. 2009.

[7] E. Bursztein, S. Bethard, J.C. Mitchell, D. Jurafsky, and C. Fabry, "How Good Are Humans at Solving CAPTCHAs? A Large Scale Evaluation," Proc. IEEE Symp. Security and Privacy, May 2010.

[8] Mansour Alsaleh, Mohammad Mannan, and P.C. van Oorschot, Member, IEEE, "Revisiting Defenses against Large-Scale Online Password Guessing Attacks",January/February 2012.