International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Design New Security Protocol against Online Password Guessing Attacks

Nitin. R. Shinde[1], S. K. Sonkar[2]

[1, 2]Department of Computer Engineering, Amrutvahini College of Engineering, Sangamner (MH), India

Abstract: ***Know a day's large number of transaction is done through online e-commerce services. As online transaction increases, the security for that transaction also increases because there will be large number of attacks on password can done by remote login, Especially Brute force attacks and Dictionary attacks. In this paper, we proposed new security protocol called as Password Guessing Resistant Protocol (PGRP), to restrict the dictionary and brute force attacks. PGRP limit total number of login attempts from user known machine. For known user PGRP allow as low as single attempt per user name and those valid user, it allow several failed login attempts before challenged with an ATT (Automated Turing Test).***

Keywords: Brute force attacks, Dictionary attacks, PGRP, ATT

1. Introduction

Online idea attacks on password-based systems square measure inevitable and username remarkably determined against net applications and SSH logins. During a recent report, SANS known word idea attacks on websites as a top cyber security risk. As example of SSH password guessing attacks, one experimental UNIX honeypot setup has been reported to suffer on the average 2,805 SSH malicious login tries per laptop per day curiously, SSH servers that forbid customary password authentication may suffer idea attacks, e.g. through the exploitation of a lesser known/used SSH server configuration known as keyboard interactive authentication. However, online attacks have some inherent disadvantages compared to offline attacks: assaultive machines should interact in interactive protocol, thus allowing easier detection; and in most cases, attackers will try solely restricted range of guesses from one machine before being fast out, delayed, or challenged to answer Automated Turing Tests (ATTs, e.g., CAPTCHAs). Consequently, attackers typically should use an outsized range of machines to avoid detection or lock-out. On the opposite hand, as users usually choose common and comparatively weak words and attackers presently management massive botnets, online attacks square measure a lot of easier than before.

One effective defense against automated online password guessing attacks is to limit the amount user of failing trials while not ATTs to an awfully tiny variety (e.g., three), limiting machine controlled programs (or bots) as utilized by attackers to three free parole guesses for a targeted account username, even if completely different machines from a botnet are used. However, this inconveniences the legitimate user who then should answer an ATT on succeeding login try.

Several different techniques are deployed in observe, including: permitting login tries while not ATTs from a different machine, once a particular variety of failing rise occur from a given machine; permitting additional tries without ATTs when a time-out period; and time-limited account protection. Several existing techniques and proposals involve ATTs, with the username delaying assumption that these challenges are sufficiently troublesome for bots and straightforward for most individuals. However, users progressively dislike ATTs as these are perceived as an (username necessary) further step; Yan and Ahmad for usability problems associated with normally used CAPTCH. Owing to productive attacks that break ATTs while not human solvers (e.g. ATTs perceived to be tougher for bots are being deployed. As a consequence of this arms-race, contemporary ATTs are becoming progressively troublesome for human users, fueling a growing tension between security and value of ATTs. Therefore, we have a tendency to specialize in reducing user annoyance by challenging users with fewer ATTs, here as at a similar time subjecting large logins to additional ATTs, to near the economic price to attackers.

Two well-known proposals for limiting on-line dead reckoning attacks victimization ATTs are Pinkas and smoother (herein denoted PS), and van Oorschot and Stubblebine (herein denoted VS). For convenience, The notation proposal reduces the amount of ATTs sent to legitimate users, however at some meaning loss of security; as an example, in associate example setup (with p 1⁄4 0:05, the fraction of incorrect login {attempts |makes associate attempt |tries} requiring an ATT) notation allows attackers to eliminate 95% of the word space while not respondent any ATTs. The VS proposal reduces this however at a big price to usability; as an example, VS might need all users to answer ATTs in circumstances. The proposal within the gift paper, known as Password Guessing Resistant Protocol (PGRP), significantly improves the security-usability trade-off, and can be a lot of usually deployed on the far side browser-based authentication. PGRP builds on these 2 previous proposals. In particular, PGRP enforces ATTs once many (e.g., three) username successful login tries are made up of username known machines. On the opposite hand, PGRP permits a high range (e.g.30) of username successful tries from noted machines while not respondent any ATTs. We define noted machines as those from that a surefire login has occurred among a set of your time. These are identified by their IP address saved on the login server asa white list, cookies hold on on consumer machines. A white listed IP address and/or consumer cookie expire once a definite time.

PGRP accommodates each graphical user interfaces (e.g, browser-based logins) and character-based interfaces (e.g., SSH logins), whereas the previous protocols deal solely with

the previous, requiring the employment of browser cookies. PGRP uses either cookies or IP, or each for tracking legitimate users. Pursuit users through their IP addresses additionally permits PGRP to extend the amount of ATTs for word dead reckoning attacks and meantime to decrease the amount of ATTs for legitimate login tries. Although NATs and Internet proxies might (slightly) cut back the utility of IP address data, in apply, the employment of IP addresses for consumer identification seems possible [4]. In recent years, the trend of work in to on-line accounts through multiple personal devices (e.g., PCs, laptops, smart phones) is growing. once used from a home atmosphere ,these devices typically share one public IP address (i.e. a simple NAT address) that makes IP-based history tracking a lot of user friendly than cookies. as an example, cookies should be hold on, albeit transparently to the user, in all devices used for login.

2. Related Work

Although on-line word shot attacks are known since the first days of the web, there's very little academic literature on bar techniques. Account locking could be a customary mechanism to stop an opponent from making an attempt multiple passwords for a selected username. A DoS attack by creating enough username successful login makes an attempt to lock a selected account. Delaying server response when receiving user credentials, whether or not the password is correct or incorrect, prevents the opponent from making an attempt an oversized range of passwords during a reasonable quantity of your time for a selected username. However, for adversaries with access to an oversized range of machines (e.g. a botnet), this mechanism is ineffective. Similarly, prevention techniques that think about requesting the user machine to perform further nontrivial computation before replying to the entered credentials aren't effective with such adversaries.

ATT challenges are employed in some login protocols to stop machine-controlled programs from brute force and dictionary attacks. Pinkas and sander presented a login protocol (PS protocol) supported ATTs to protect against on-line word shot attacks. It reduces the number of ATTs that legitimate users should properly answer so a user with a so username browser cookie (indicating that the user has antecedently logged in successfully) can seldom be prompted to answer an ATT. A deterministic operate (AskATT) of the entered user credentials is employed to come to a decision whether or not to raise the user an ATT. to boost the protection of the notation protocol, van Oorschot and Stubblebine urged a changed protocol during which ATTs are invariably needed once the number of username successful login makes an attempt for a selected username exceeds a threshold; alternative modifications were introduced to reduce the on sequences of cookie stealing.

He and Han dynasty pointed out that a poor style of this operate might build the login protocol at risk of attacks like the "known operate attack" (e.g. if an easy cryptanalytic hash operate of the username and therefore the word is employed as AskATT) and "changed word attack" (i.e. an opponent mounts a dictionary attack before and when a word amendment event initiated by a legitimated user). In future, we planned a secure nondeterministic keyed hash operate as AskATT() so each username is related to one key that ought to be changed whenever the corresponding word modified. The planned operate needs further server-side storage per username and a minimum of one cryptanalytic hash operation per login try.

3. Proposed System Architecture

The main security goal is to restrict an attacker who is launching the attack from large botnet on single or multiple accounts. In terms of usability, we want to reduce the number of ATT send to the legitimated user. The proposal called Password Guessing Resistant Protocol (PGRP), significantly improves the security-usability trade-off, and can be more generally deployed beyond browser based authentication. In particular, to limit attackers in control of a large botnet, PGRP enforces ATTs after a few failed login attempts are made from unknown machines. On the other hand, PGRP allows a high number of failed attempts from known machines without answering any ATTs.

Figure 1 Represents the entire architecture of the PGRP Protocol

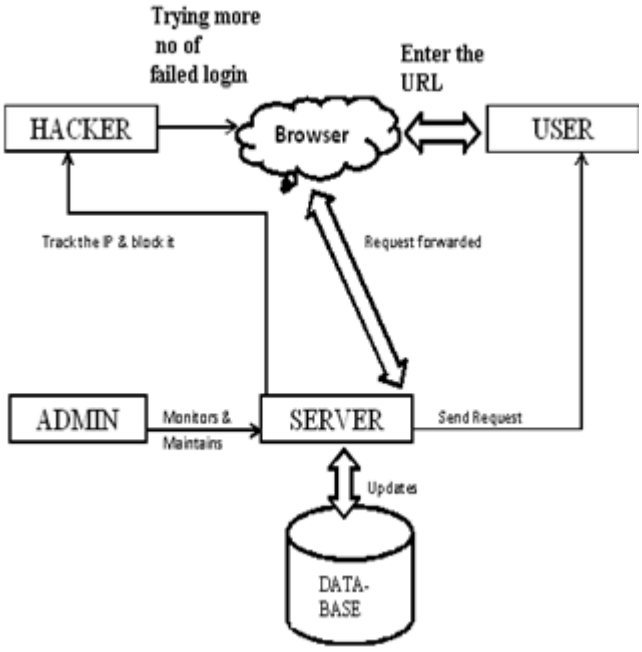

Figure 1: Proposed System Architecture

A. Hacker:
When there is more of failed login attempts for a particular account than that user is been traced using the IP address. This method find the user's IP instead of the user browser's cookie since cookie can be easily modified and deleted. The use of IP address is also a tedious process when the request if from a large botnet. Since it involves the process of network address translation. The hacker must be traced carefully when requesting for the resources in the network.

B. Generate CAPTCHA:
CAPTCHA is the completely Automated Public Turing Test to tell Computers and Humans Apart. When the number of attempts made to login increases beyond three limits a CAPTCHA will be generated. The user must undergo this

ATT challenge. This is used as a validation method to verify whether the user is a valid user based on the time taken to complete the challenge. The generated CAPTCHA will be dynamic (i.e.,) new CAPTCHA will be generated for each transaction performed by the user. In this protocol the CAPTCHA generated are the ATTs which will be generated when the user has failed 3 login attempts. This provides a convenient method for the valid user.

C. Forwarding New Password:

This performs the password generation, which generates new passwords for each transaction so that the account password cannot be traced out by anyone (i.e.,) unauthorized users. This operation is performed after the verification of the user (i.e.,) after the user undergoes the ATT challenge. If the verification is success the generator will generate and forward the new password to the valid user.

D. Blocking IP:

The users are traced using the IP addresses which are been assigned to the system. If the user's attempt made to login fails even after the new password which is generated then that particular IP address which attempts more failed attempts will be traced and blocked for that particular username. The blocking of IP address is based on time out scheme which makes it convenient to the legitimate users and stop the hackers from guessing the pass-words of the user. This makes the user's password more secured from the unauthorized user's access.

The system architecture depicts that the user must undergo an ATT only after a limited number of failed at-tempts made to the login. A captcha will be generated after a three failed login attempts. When the user enters the captcha, the server will collect the details of the particular user and will validate it.

The functional requirements of the system is to resist the online guessing attacks over the passwords which are been achieved using the password guessing resistant protocol. The requirements are to enter the user name and password for checking authorized user or not. If the user name is correct then the User will be successfully logged in. The Server monitors all details during the communication. If the User misbehaves any Login attempt it will be identified and the misbehaved user will be blocked in the network.

4. Proposed PGRP Algorithm

Input:

t1 (default=30d), t2 (default=1d), t3 (default=1d), k1 (default=30), k2 (default=3) // The keyword "default" denotes the default parameter value and'd' denotes day, k1,k2>= 0

username, password, cookie //username, password, and remote host's browser cookie if any

W (global variable, expires after t1) //white list of IP addresses with successful login

FT (global variable, default=0, expires after t2) //table of number of failed logins per username

FS (global variable, default=0, expires after t3) //table of number of failed logins indexed by (sourceIP, username) for hosts in W or hosts with valid cookies

1) Start
2) ReadCredential (username, password, cookie) // login prompt to enter username/password pair
3) IfLoginCorrect (username, password) then // username/password pair is correct
4) If(((Valid(cookie, username,k1,true) V ((sourceIP, username)ϵW) /\ (FS[sourceIP, username] < k1)) \/ (FT[username]< k2)) then
5) FS [sourceIP, username] =0
6) Add sourceIP to W // add source IP address to the white list
7) GrantAccess (username, cookie) // this username also sends the cookie if applicable
8) Else
9) If (ATTChallenge () = Pass) then
10) FS [sourceIP, username] =0
11) Add sourceIP to W
12) GrantAccess (username, cookie)
13) Else
14) Message ("The answer to ATT Challenge is incorrect")
15) Else
16) If (((Valid (c0okie, username, k1, false) V ((sourceIP, username) ϵ W)/\ (FS [sourceIP, username] < k1)) then
17) FS [sourceIP, username] = FS [sourceIP, username] +1
18) Message ("username and password is incorrect")
19) Else if (ValidUsername (username) ^ (FT (username) <K2)) then
20) FT [username] = FS [username] +1
21) Message ("username and password is incorrect")
22) Else
23) If (ATTChallenge () = Pass) then
24) Message ("username and password is incorrect")
25) Else
26) Message ("The answer to ATT Challenge is incorrect")
27) End

5. Password Guessing Resistant Protocol

In this section, we have a tendency to gift the PGRP protocol, together with the goals and style selections.

A.1Goals and Operational Assumptions

A.1.1 Protocol Goals

Our objectives for PGRP embody the following:

1) The login protocol ought to create brute force and dictionary attacks ineffective even for adversaries with access to giant botnets (i.e. capable of launching the attack from several remote hosts).
2) The protocol mustn't have any important impact on usability (user convenience). For instance: for legitimate users, any further steps besides entering login credentials ought to be taken. Increasing the security of the protocol should have token impact in decreasing the login usability.
3) The protocol ought to be simple to deploy and climbable, requiring minimum process resources in terms of memory, interval, and disc space.

A.1.2 Assumptions

We assume that adversaries will solve little share of ATTs, e.g. through machine-controlled programs, brute force mechanisms, and low paid staff (e.g. Amazon Mechanical Turk). Incidents of attacker's exploitation IP addresses of

known machines and cookie stealing for targeted parole guessing also are assumed to be taken. Ancient password-based authentication isn't appropriate for any untrusted setting (e.g. a keylogger might record all keystrokes, together with passwords in a very system, and forward those to a distant attacker). We have a tendency to don't stop existing such attacks in untrusted environments, and therefore basically assume any machines that legitimate users use for login are trustworthy. The information integrity of cookies should be protected.

A.1.3 Overview

The general plan behind PGRP is that apart from the following 2 cases, all remote hosts should properly answer an ATT challenge before being wise to whether or not access is granted or the login try is unsuccessful:

1) Once the number of failing login tries for a given username is very small.
2) Once the remote host has with success logged in exploitation a similar username within the past (however, such a host should pass an ATT challenge if it generates additional failing login tries than a prespecified threshold).

In distinction to previous protocols, PGRP uses either IP addresses, cookies, or each to spot machines from which users are with success documented. The decision to need an ATT challenge upon receiving incorrect credentials relies on the received cookie (if any) and/or the remote host's IP address. additionally, if the number of failing login tries for a particular username is below a threshold, the user isn't needed to Answer an ATT challenge notwithstanding the login try is from a replacement machine for the primary time (whether the provided username password combine is correct or incorrect).

B.2. Data Structure and Function Description

B.2.1 Data Structures

PGRP maintains three Data structures:

1) **W:** an inventory of {source IP address, username} pairs such that for every combine, a in login from the supply IP address has been initiated for the username previously.
2) **FT:** every entry during this table represents the amount of failed login tries for a username. A maximum of k2 failing login tries are recorded. Accessing a nonexistent index returns zero.
3) **FS:** every entry during this table represents the amount of failed login tries for every combine of (sourceIP, username). Here, sourceIP is that the IP address for a host in W or a host with a valid cookie, and username could be a valid username attempted from sourceIP . A most of k1 failing login attempts are recorded; crossing this threshold might mandate passing an ATT (e.g.,reckoning on FT [username]). An entry is about to zero when a in login attempt. Accessing a nonexistent index returns zero.

Each entry in W, FT, and FS contains a "write-expiry" interval such the entry is deleted once the given amount of time (t1 , t2 , or t3) has been expire since the last time the entry was inserted or changed. There are alternative ways to implement write-expiry intervals (e.g. hashbelt). A simple approach is to store a timestamp of the insertion time with every entry such the timestamp is updated whenever the entry is changed. At any time the entry is accessed, if the delta between the interval and therefore the entry timestamp is bigger than the information structure write-expiry interval (i.e. t1, t2, or t3), the entry is deleted.

C.2. Functions

PGRP uses the subsequent functions (IN denotes input and OUT denotes output):

- **ReadCredential(OUT: username, password, cookie):** Shows a login prompt to the user and returns the entered username and password, and therefore the cookie received from the user's browser (if any).
- **LoginCorrect(IN: username,password; OUT: true/false):** If the provided username and password combine is valid, the function returns true; otherwise, it returns false.
- **GrantAccess(IN: username,cookie):** The perform sends the cookie to the user's browser and so permits access to the desired user account.
- **Message(IN:text):**text message shows. ATT Challenge(OUT: Pass/Fail): Challenges the user with an ATT and returns "Pass" if the solution is correct; otherwise, it returns "Fail."
- **ValidUsername(IN: username; OUT: true/false):** If the provided username exists within the login system, the perform returns true; otherwise, it returns false.
- **Valid(IN:cookie,username,k1,state;OUT:cookie,true/false**): First, the function checks the validity of the cookie (if any) where it is considered invalid in the following cases:
  1) The login username does not match the cookie username.
  2) The cookie is expired.
  3) The cookie counter is equal to or greater than k1.

The perform returns true only if a valid cookie is received. If state=true (i.e. the entered user credentials are correct, as in line four of algorithm), a new cookie is formed (if cookies are supported within the login system) together with the subsequent information: username, end date, and a counter of the amount of failed login tries (since the last in login; initialized to 0). Notice that if state= true, the function doesn't send the created cookie to the user's browser. Rather, the cookie is distributed later by the GrantAccess() perform. If state= false (i.e., the entered user credentials are incorrect, as in line sixteen of algorithm) and a valid cookie is received, the cookie counter is incremented by one and therefore the cookie is distributed back to the user's browser. No action is performed for all the opposite cases.

D.3 Cookies versus Source IP Addresses

Similar to the previous protocols, PGRP keeps track of user machines from that in logins are initiated previously. Browser cookies appear a decent selection for this purpose if the login server offers a web-based interface. Typically, if no cookie is distributed by the user browser to the login server, the server sends a cookie to the browser when a successful login to spot the user on ensuing login try. However, if the user uses multiple browsers or quite one OS on a similar machine, the login server is going to be unable to identify the user all told cases. Cookies may additionally be deleted by users, or mechanically as enabled by the personal browsing mode of newest browsers. Moreover, cookie stealing (e.g., through session hijacking) may alter an person to impersonate a user

World Health Organization has been with success documented in the past. Additionally, an exploitation cookie needs a browser interface (which, e.g., isn't applicable to SSH).

Alternatively, a user machine is often known by the source IP address. wishing on supply IP addresses to trace users might end in inaccurate identification for numerous reasons, including: 1) a similar machine could be allotted different IP addresses over time (e.g., through the network DHCP server and dial-up Internet); and 2) affects security since some users/attackers may not be asked to answer an ATT challenge even though they have not logged in successfully from those machines in the past.

Drawbacks of characteristic a user by means that of either a browser cookie or a supply IP address include: 1) failing to identify a machine from that the user has documented successfully within the past; and 2) wrong characteristic a machine the user has not documented before. Case 1) decreases usability since the user could be asked to answer an ATT challenge for each correct and incorrect login credentials. On the other hand, case 2) affects security since some users attackers might not be asked to answer an ATT challenge even though they need not logged in with success from those machines within the past. However, the likelihood of launching a dictionary or brute force attack from these machines seems to be low. First, for identification through cookies, a directed attack to steal users' cookies is needed by an person. Second, for identification through IP addresses, the person a valid cookie is received from the user machine must have access to a machine within the same subnet because the user.

Consequently, we elect to use each browser cookies of failing login tries from the user machine's and supply IP address in PGRP to reduce user inconvenience throughout than k1 over a period of time determined by t3 ;the login method. Also, by exploitation IP addresses only. The user machine's IP address is within the whitelist W can be utilized in character-based login interfaces like SSH. and therefore the variety of failing login tries from this An SSH server are often tailored to use PGRP exploitation text-based IP address for that username, FS[sourceIP ; username] < k1). For example, a model of a than k1 (line 16) over a period of time determined by t3: text-based CAPTCHA for SSH is accessible as a ASCII text file three.

The security implications of mistakenly treating a machine as one that a user has previously successfully logged in from is limited by a threshold such that after a specific number of failed login attempts (k1 in Fig. 1), an ATT challenge is imposed. For identification through a source IP address, the condition FS½sourceIP; username_ < k1 in line 4 (for correct credentials) and in line 16 (for incorrect credentials) limits the number of failed login attempts an identified user can make without answering ATTs (see Fig. 1). Also, the function Valid (cookie, username, k1,true) in line 4 updates a counter in the received cookie in which the cookie is considered invalid once this counter hits or exceeds k1. This function is also called in line 16 to check this counter in case of a failed login attempt.

E.4 Decision Function for Requesting ATTs

Below we have a tendency to discuss problems associated with ATT challenges as PGRP shows completely different messages just in case of incorrect provided by the login server in Fig. 1. The choice to combine (lines twenty one and 24) and incorrect challenge the user with AN ATT depends on 2 factors: answer to the given ATT challenge (lines fourteen and 26). While 1) whether or not the user has documented with success from the showing a personality's that the entered same machine previously; and 2) the overall variety of failing combine is inaccurate, an automatic program unwilling to login tries for a particular user account. For definitions of answer the ATT challenge cannot ensure whether or not it's W, FT, and F S. the combine or the ATT that was incorrect. However, while this is additional convenient for legitimate users, it offers additional info to the assailant concerning the answered ATTs. PGRP are often changed to show just one message in lines fourteen, 21, 24, and 26 (e.g., "login fails" as within the note and VS protocols) to stop such info leak.

E.4.1 Username-Password pair Is Valid

As within the condition in line four, upon coming into an accurate username-password combine, the user won't be asked to answer an ATT challenge within the following cases:

1) A valid cookie is received from the user machine (i.e., the perform Valid returns true) and therefore the variety of failing login tries from the user machine's IP address for that username, FS [sourceIP ; username] is less than k1 over a period of time determined by t3 ;
2) The user machine's IPaddress is within the whitelist W and the variety of failing login tries from this IP address for that username, FS[sourceIP ; username] is less than k1 over a period of time determined by t3.
3) The amount of failing login tries from any machine for that username, FT [username], is below a threshold k2 over time period determine by t2.

The last case permits a user who tries to login from a replacement machine/IP address for the primary time before k2 is reached to proceed while not an ATT. However, if the amount of failing login tries for the username exceeds the edge k2 (default 3), this may indicate a approximation attack and therefore the user should pass AN ATT challenge.

E.4.2 Username-Password pair Is Invalid

Upon coming into an incorrect username-password combine, the user won't be asked to answer an ATT challenge within the following cases:

1) A valid cookie is received from the user machine (i.e., the function Valid returns true) and the number of failed login attempts from the user machine's IP address for that username, FS[sourceIP; username] is less than k1 (line 16) over a time period determined by t3;
2) The user machine's IP address is in the whitelist W and the number of failed login attempts from this IP address for that username, FS[sourceIP, username], is less than k1 (line 16) over a time period determined by t3;
3) The username is valid and the number of failed login attempts (from any machine) for that username, FT[username], is below a threshold k2 (line 19) over a time period determined by t2.

A failed login attempt from a user with a valid cookie or in the whitelist W will not increase the total number of failed login attempts in the FT table since it is expected that legitimate users may potentially forget or mistype their password (line 16-18). Nevertheless, if the user machine is identified by a cookie, a corresponding counter of the failed login attempts in the cookie will be updated. In addition, the FS entry indexed by the {source IP address, username} pair will also be incremented (line 17). Once the cookie counter or the corresponding FS entry hits or exceeds the threshold k1 (default value 30), the user must correctly answer an ATT challenge.

E.4.3 Output Messages

PGRP shows different messages in case of incorrect {username, password} pair (lines 21 and 24) and incorrect Answer to the given ATT challenge (lines 14 and 26). While showing a human that the entered {username, password} pair is incorrect, an automated program unwilling to answer the ATT challenge cannot confirm whether it is the pair or the ATT that was incorrect. However, while this is more convenient for legitimate users, it gives more information to the attacker about the answered ATTs. PGRP can be modified to display only one message in lines 15, 22 25, and 27 (e.g., "login fails" as in the PS and VS protocols) to prevent such information leakage.

E.4.4 Why to not Black-List Offending IP Addresses

We choose to not produce a blacklist for IPaddresses creating many failing login tries for the subsequent reasons:

1) This list might consume sizable memory.
2) Legitimate users from blacklisted IP addresses might be blocked.
3) Hosts exploitation dynamic IP addresses appear additional engaging targets (compared to hosts with static IP addresses) for adversaries to launch their attacks from (e.g., spammers).
4) If the cookie mechanism isn't accessible for the login server, PGRP will operate by exploitation solely supply IP addresses to keep track of user machines.

6. Conclusion and Future Enhancement

In previous ATT based login protocol, their exists security tradeoff with respect to the number of failed login attempt done by user. In opposite of that, PGRP is more secure against Brute force attack and Dictionary Attacks. PGRP providing less number of challenges to the legitimated user and more challenges to the hacker. This also provides a secured login to the valid users by generating new passwords and forwarding it to their mobile phones. Blocking IP is an added advantage which is used to overcome the account locking system.

The further enhancement can be done by encrypting the password which is been generated and forwarded to the valid user. Even the encrypted password can be a onetime password which is been generated by the server. This method will be more authenticated which may avoid the password modification or the thefts when it is been send from the browser to the valid user.

References

[1] Mansour Alsaleh, Mohammad Mannan, and P.C. van Oorschot,"Revisiting Defenses against Large-Scale Online Password Guessing Attacks"IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 9, NO. 1,JANUARY/FEBRUARY 2012

[2] 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.

[3] S. Chiasson, P.C. van Oorschot, and R. Biddle, "A Usability Study and Critique of Two Password Managers", Proc. USENIX Security Symp., pp. 1-16, 2006.

[4] A. Narayanan and V. Shmatikov, "Fast Dictionary Attacks on Human Memorable Passwords Using Time-Space Tradeoff", Proc. ACM Computer and Comm. Security (CCS 05), pp. 364-372, Nov. 2005..

[5] B. Pinkas and T. Sander, "Securing Passwords against Dictionary Attacks", Proc. ACM Conf. Computer and Comm. Security (CCS 02), pp. 161-170, Nov. 2002.

[6] J. Jayavasanthi Mabel, Mr. C. Balakrishnan,"RESISTING PASSWORD BASED SYSTEMS FROM ONLINE GUESSING ATTACKS",International Journal of Emerging Technology and Advanced Engineering,Volume 3, Special Issue 1, January 2013)

[7] Harshitha.H.K And Sreedevi.N,"ImprovingE_ciency of Password Security Against Large Scale Online Attacks",The International Journal Of Engineering And Science (IJES) Volume-2 Issue 6 2013

[8] Ms.Vaishnavi J. Deshmukh, Prof. Sapna S. Kaushik"Convenient Approach From Online Password Guessing Attacks",International Journal For Engineering Appli-Cations And Technology,Issue,2(1):10 oct,2013