Enhancement of Password Guessing Resistant Protocol With Cookie and Session Monitoring

Sindhu.M
PG Scholar
K.S.Rangasamy College Of Technology,
Namakkal
India
msindhume@gmail.com

Abstract—**Web applications and Secure Shell (SSH) logins are initiated with remote login Services. Remote login services are attacked with Brute force and dictionary attacks. Password Guessing attacks are initiated by Botnets. Automated Turing Tests (ATTs) is conducted to identify automated malicious login attempts. Pinkas and Sander (PS) and Van Oorschot and Stubblebine proposals(VS) are used to limit the online password guessing attack based on ATTs. The PS protocol reduces the number of ATTs sent to legitimate users. The VS protocol reduces the security overhead with the significant cost of usability. Security, usability and user interface factors are considered in the remote login process. Password Guessing Resistant Protocol (PGRP) is designed to restrict login attacks. PGRP limits the total number of login attempts from the unknown remote hosts. PGRP enforces ATTs after a few failed login attempts are made from unknown remote machines. PGRP allows a high number of failed login attempts from known machines without answering any ATTs. Known machines are systems with the successful login have occurred within a fixed period of time. White-listed IP address and client cookie are used to identify the known machines. PGRP supports both graphical user interface and character-based interfaces. User name and ip address are used to detect legitimate users. The Password Guessing Resistant Protocol (PGRP) is enhanced to control cookie thefts. Black lists are used to manage the attacker addresses under login verification. Compromised machine attacks are handled with the user name and IP address associations. Concurrent login verifications is applied with session details.**

Keywords—**Secure shell; White listed IP address; Black list; ATTs;**

I.INTRODUCTION

Online Guessing attacks on password-based systems are inevitable and commonly observed against web applications and SSH logins. In a recent report, SANS identified password guessing attacks on websites as a top cyber security risk. Interestingly, SSH servers that disallow standard password authentication may also suffer guessing attack, e.g., through the exploitation of a lesser known/used SSH server configuration called keyboard interactive authentication. However, online attacks have some inherent disadvantages compared to offline attacks; attacking machines must engage in a interactive protocol, thus allowing easier detection; and in most cases, attacker can try only limited number of guesses from a single machine before being locked out, delayed, or challenged to answer Automated Turing Tests. Consequently, attackers often must employ a large number of avoid detection or lock-out. On the other hand, as users generally choose common and relatively weak password and attackers currently control large botnets, online attacks are much easier than before.

One effective defense against automated online password guessing attacks is to restrict the number of failed trails without ATTs to a very small number, limiting automated programs as used by attacker to three free password guesses for a targeted account, even if different machined from a botnets are used.

Several othjer techniques are deployed in practice, including: allowing login attempts without ATTs from a different machine, when a certain number of a failed attempts occurs froma a given machine; allowing more attempts without ATTs after a time-out period; and time-limited account locking. Many existing techniques and proposals involve ATTs,with the underlying assumption that these challenges are sufficiently diffivult for bots and easy for most people. However, users increasingly dislike ATTs as these are perceived as an extra steps . Due to successful attcks which break ATTs without human solvers. ATTs perceived to be more difficult for bots are being deployed. As a consequence of this arms-race, present-day ATTs are becoming increasingly difficult for humans users, fueling a growing tension between security and usability of ATTs. Therefore, focus on reducing user annoyance by challenging users with fewer ATTs, while at the same time subjecting bot logins to more ATTs, to drive up the economic cost to attackers.

Two well-known proposals for limiting online guessing attacks using ATTs are Pinkas and Sander and Van Oorschot and Stubbline. The PS proposal reduces the number of ATTs sent t legitimate users but at some meaningful loss of security; for example, in an example setup PS allows attackers to eliminate 95 percent of the password space without answering any ATTs. The VS proposal reduces this but at a significant cost to usability; for example, VS may require all users to answer ATTs in certain circumstances. The proposal in the present paper, called password Guessing Resistant protocol (PGRP) significantly improves the security-usability trade-off, and can be more generally deployed beyond browser-based authentication.

PGRP builds on these two previous proposals. In articular, 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. We define known machines as those from which a successful login has occurred within a fixed period of time. These are identified by their Ip addresses saved on the login server as a white list, or cookies stored on client machines. A white-listed IP address and/or client cookie expire after a certain time.

PGRP accommodates both graphical user interface and character-based interfaces, while the previous protocols deal exclusively with the former, requiring the use of browser cookies. PGRP uses either cookies or IP addresses, or both for tracking legitimate users. Tracking users through their IP addresses also allows PGRP to increases 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 reduce the utility of IP address information, in practice, the use of IP addresses for client identification appears feasible. In recent years, the trend of logging in to online accounts through multiple personal devices is growing. When used from a home environment, these devices often share a single public IP address which makes IP-based history tracking more user friendly than cookies. For example cookie must be stored, albeit transparently to the user, in all devices used for login.

A.Contributions

1) Strict but user-friendly att-based scheme. The proposed PGRP scheme is more restricted against attackers than commonly used countermeasures and two earlier proposals. At the same time, PGRP requires answering fewer ATTs t attempts to recall a password.

2) First reported empirical analysis of attbased schemes. We compare PGRP's performance and usability to previous such schemes, using two data sets from a university environment.

3) Applicagilitytoweb and text logins. PGRP is not limited to web only login, as it uses IP address and/or other methods to identify a remote machine in addition to optionally using cookies. By using text-based ATTs, SSH login can be adapted to use PGRP.

II.RELATED WORK

Although online password guessing attacks have been known since the early days of the Internet, there is little academic literature on prevention techniques. Account locking is a customary mechanism to prevent the adversary from attempting multiple password for a particular username, the adversary can mount a DoS attack by making enough failed login attempts to lock a particular account. Delaying server response after receiving user credentials, whether the password is correct or incorrect, prevents the adversary from attempting a large number of passwords in a reasonable amount of time for a particular username. However, for adversaries with access to a large number of machines, this mechanism is ineffective. Similarly, prevention techniques that rely on requesting the user machine to perform extra nontrivial computation prior to replying to the entered credentials are not effective with such adversaries.

As discussed in Section 1, ATT challenges are used in some login protocols to prevent automated programs from brute force and dictionary attacks. Pinkas and Sander presented a login protocol based on ATTs to protect against online password guessing attacks. It reduces the number of ATTs that legitimate users must correctly answer so that a user with a valid browser cookie will rarely be promoted to answer an ATT. A deterministic function of the entered user credentials is used to decide whether to ask the user an ATT. To improve the security of the PS protocol, Van Oorschot and Stubblebine suggested a modified protocol in which ATTs are always require once the number of failed login attempts for a particular username exceeds a threshold ; other modifications were introduced the effects of cookie theft.

For both Ps and VS protocols, the decision function AskATT()) requires careful design. He and Han pointed out that a poor design of this function may make the login protocol vulnerable to attacks such as the “known function attacks” and “changed password attack”. The authors proposed a secure nondeterministic keyed hash function as AskATT() so that each username is associated with one key that should be changed whenever the corresponding password is changed. The proposed function requires extra server-side storage per username and at least one cryptographic hash operation per login attempt.

III. PASSWORD GUESSING RESISTANT PROTOCOL

In this section, we present the PGRP protocol, including the goals and design choices.

A. Protocol Goals

Our objective for PGRP include the following

- The login protocol should make brute force and dictionary attacks ineffective even for adversaries with access to large botnets.
- The protocol should not have any significant impact on usability. for example: for legitimate users, any additional steps besides entering login credentials should be minimal effect in decreasing the login usability.

The protocol should be easy to deploy and scalable, requiring minimum computational resources in terms of memory, processing time, and disk space.

B. Protocol Overview

The general idea behind PGRP is that except for the following two cases, all remote host must correctly answer an ATT challenge prior to being informed whether access is granted or the login attempt is unsuccessful; 1) when the number of failed login attempts for a given username is very small, and 2) when the remote host has successfully logged in using the same user name in the past. In contrast to previous protocol, PGRP uses either IP addresses, cookies, or both to identify machines from which users have been successfully authenticated.

The decision to require an ATT challenge upon receiving incorrect credentials is based on the received cookie and/or the emote host's IP address. In addition, if the number of failed login attempts for a specific username is below the threshold, the user is not required to answer an ATT challenge even if the login attempt is from a new machine for a first time.

C. Data Structures

PGRP maintains three data structures:

1. W. A list of {source IP address, username} pairs such that for each, a successful login from the source IP address has been initiated for the username previously.
2. FT. Each entry represents the number of failed login attempts for a valid username, u_n . A maximum of K_2 failed login attempts are recorded. Accessing a nonexisting index returns 0. FS. Each entry represents the number of failed login attempts for each pair of (srcIP, un). Here, srcIP is the IP address for a host in W or a host with a valid cookie, and un is a valid username attempted from srcIP.

A maximum of K1 failed login attempts are recorded; crossing this threshold may mandate passing an ATT. An entry is set to 0 after a successful login attempt. Accessing a nonexisting index returns 0.

Each entry in W, FT, and FS has a "write-expiry" interval such that the entry is deleted when the given period of time has lapsed since the last time the entry was inserted or modified. There are different ways to implement write-expiry intervals.

A simple approach is to store a timestamp of the insertion time with each entry such that the timestamp is updated whenever the entry is modified. At anytime the entry is accessed, if the delta between the access time and the entry timestamp is greater than the data structure write-expiry interval the entry is deleted.

D. Functions

PGRP uses the following functions:

- ReadCredential (OUT: un, pw, cookie). Shows a login prompt to the user and returns the entered username and password, and the cookie receive from the user's browser (if any).
- LoginCorrect (IN: un, pw; OUT: true/false). If the provided username-password pair is valid, the function returns true; otherwise, it returns false.
- GrantAccess (IN: un, cookie). The function sends the cookie to the user's browser and then enables access to the specified user account.
- Message (IN: text). Shows a text message.
- ATTchallenge (OUT: Pass/Fail). Challenges the user with an ATT and returns "pass" if the answer is correct; otherwise, it returns "Fail".
- Valid Username. If the provider username exists I the login system, the function returns true; otherwise, it returns "Fail".
- Valid. First, the function checks the validity of the cookie where it is considered invalid in the following cases: 1) the login username does not match the cookie username; 2) the cookie is expired; or 3) the cookie counter is equal to or greater than K1.

The function returns true only when a valid cookie is received. If state-true, a new cookie is created including the following information: username, expiry date, and a counter of the number of failed login attempts.

E. Cookies versus Source IP addresses

Similar to the previous protocols, PGRP keeps track of user machine from which successful logins have been initiated previously. Browser cookies seem a good choice for this purpose if the login server offers a web-based interface. Typically, if no cookie is sent by the user browser to the login server, the server sends s cookie to the browser after a successful login to identify the user on the next login attempt. However, if the user uses multiple browsers or more than one OS on the same machine, the login server will be unable to identify the user in all cases. Cookies may also be deleted by user or automatically as enabled by the private browsing mode of most modern browsers. Moreover, cookie theft might enable an adversary to impersonate a user who has been successfully authenticated. In addition, using cookies requires a browser interface.

Consequently, we choose to use both browser cookies and source IP address in PGRP to minimize user inconvenience during the login process. Also, by using IP addresses only, PGRP can be used in character-based login interfaces such as SSH. An SSH server can be adapted to use PGRP using text based ATTs. For example, a prototype of a text-based CAPTCHA for SSH is available as a source code patch for OpenSSH.

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 an ATT challenge is imposed. For identification through a source IP address, the condition FS[srcIP;] un<k_1 limits the number of failed login attempts an identified user can make without answering ATTs. Also, as the function valid updates a counter in the received cookie in which the cookie is considered invalid once this counter hits or exceeds k_1. This function is also to check the counter in case of failed attempt.

F. Decision Function for Requesting ATTs

Below we discuss issues related to ATT challenges as provided by the login server.

The decision to challenge the user with an ATT depends 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.

IV COMPARISION WITH OTHER ATT-BASED PROTOCOLS

In this section, we analyze the security, usability, and required system resources of PGRP as compared to a strawman protocol and the PS and VS protocols. This section also provides a comparative summary of major limitations in each protocol.

A. Security Analysis

Following the previous analysis of PS, assume a fixed password space of cardinality N, assume password are equi-probable, and that the delay between when the {username, password} pair is entered and the ATT challenge is presented to the user is identical whether or not he credentials are correct. Also assume that cookie theft, and adversaries using legitimate user's IP addresses occur rarely.

B. Usability Comments on ATT Challenges

Our main security goal is to restrict an attacker who is in control of a large botnet from launching online single-account or multiaccount password dictionary attacks. In terms of usability, we want to reduce the number of ATTs sent to legitimate users as much as possible. A user receives ATTs when the total number of failed attempts exceeds threshold k_2, and the login attempt is initiated from 1) an unfnown machine or 2) a known machine from which the user has already failed k_1 times. This happens for both cases of correct and incorrect username-password pairs, assuming the provided username is valid. Below we discuss different login scenarios and the extra effort as required from users by PGRP. The analysis below indicates that only limited usability impact may be expected from our proposal; the same can also be inferred from our real-world data analysis.

C. System Resources

No lists are maintained in the PS protocol, thus no extra memory overhead is improved on the login server. In the VS protocol, only FT is maintained. The number of entries in this list grows linearly with unique usernames used in failed login attempts. An attacker may try to exhaust a login server's memory by failed login attempts for many usernames. For any cookie-based login protocol, the login server may also need to store information regarding each generated cookie to ameliorate cookie theft attacks. Note that neither the PS nor VS protocol uses IP addresses. The most expensive server operation in PS, VS, and PGRP is generating an ATT.

V.PGRP WITH COOKIE AND SESSION MANAGEMENT

The Password Guessing Resistant Protocol (PGRP) is designed to handle attacks under web shell (SSH) remote login processes. The web process under the cookies for password entry process under the client side. The cookies based storage model is handled by the web server under the client environment. The client account details are maintained under the cookie files. The attackers can easily capture the passwords details under the cookies. The Password Guessing Resistant Protocol is enhanced to control cookie thefts. The user IP addresses are maintained in two ways. They are black list and white list. The black list maintains the attacker system addresses. The white list maintains the legitimate user system address values. The PGRP protocol manages the white list only. The white list is used for the legitimate user verification process. Black lists are used to manage the attacker addresses under login verification. Time and frequently values are used in the address list management process.

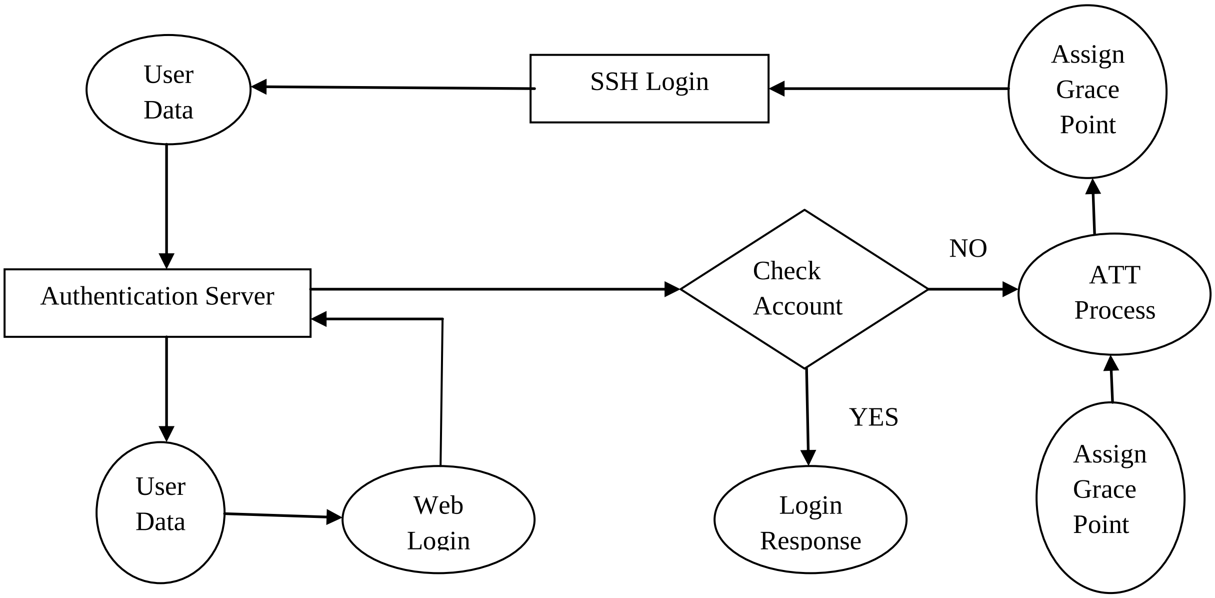

Fig 1.1: Flow Chart

The remote login requests are submitted from different machines. The compromised machine requests model initiated the login requests from one machine with other machine's address under the same network environment.

Compromised machine attacks are handled with the username and IP address associations. The username and their requests sequences are monitored to detect these types of attacks. The session information are maintained by the server to allow the user access request during the login verification is applied with session details. Simultaneously user logins are prevented by the system.

VI. CONCLUSION

Remote login schemes are used to access system resources on the Intranet and Internet environment. Password Guessing Resistant Protocol is used to manage password attacks. The PGRP is enhanced to protect cookie theft based attacks. The system handles single account attacks and multi account attacks. Graphical and console based login interfaces are supported by the system. The system usability is controlled. The system provides high security on remote login applications.

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] C. Namprempre and M.N. Dailey, "Mitigating Dictionary Attacks with Text-Graphics Character Captchas," IEICE Trans. Fundamentals of Electronics, Comm. and Computer /sciences, pp. 179-186, 2007.

[3] E. Bursztein, S. Betthard, 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.

[4] Y. He and Z.Han, "User Authentication with Provable Security against Online Dictionary Attacks. "J. Networks, vol. 4, no. 3, pp, 200-207, May 2009.

[5] "Botnet pierces Microsoft Live through Audio Captchas," TheRegister.co.uk, Mar. 2010.

[6] SANS.org, "Important Information: Distributed SSH Brute Force attacks, "SANS internet Storm Center Handler's Diary, 2010.

[7] M. Motoyama, K. Levchenko, C. Kanich, D. Mccoy, G.M. Voelker, and S. Savage, "Re: CAPTCHAs Understanding CAPTCHA Solving Services in an Economic Context, "Proc. USENIX Security Symp., Aug
2010.

[8] P. Hansteen, "Rickrolled? Get Ready for the Hail Mary Cloud!, "Feb.2010.