ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Volume 1, Issue 10, December 2013

Secured Password Hacking Process Using Multi Authentication Process

P.Kavitha

Associate Professor, Dept. of IT, Bharath University, Chennai, TamilNadu, India.

ABSTRACT: Online Guessing attacks on Password Based Systems are inevitable and commonly observed against Web Applications. Server Verifies User Name from the Cookie of the User's Machine, System IP, Catcha, Password of the User, Number of Failure Attempts by the User, Web Browser. This Process of Verification is called as Automated Turing Tests (ATT). Authentication of User will start by asking Secret Questions which was answered during the Registration Phase.Brute force and dictionary attacks on password-only remote login services are now widespread and ever increasing. Enabling convenient login for legitimate users while preventing such attacks is a difficult problem. Automated Turing Tests (ATTs) continue to be an effective, easy-to-deploy approach to identify automated malicious login attempts with reasonable cost of inconvenience to users. Inadequacy of existing and proposed login protocols designed to address large scale online dictionary attacks .we propose a new Password Guessing Resistant Protocol (PGRP), derived upon revisiting prior proposals designed to restrict such attacks. While PGRP limits the total number of login attempts from unknown remote hosts to as low as a single attempt per username, legitimate users in most cases can make several failed login attempts before being challenged with an ATT. We analyze the performance of PGRP with two real-world datasets.

1.INTRODUCTION

Online guessing attacks on password-based systems are inevitable and commonly observed against web applications and SSH logins. SANS identified password guessing attacks on websites as a top cyber security risk. Online attacks have some inherent disadvantages compared to offline attacks: attacking machines must engage in an interactive protocol, thus allowing easier detection; and in most cases, attackers can try only limited number of guesses from a single machine before being locked out, delayed, or challenged to answer Automated Turing Tests. One effective defence against automated online password guessing attacks is to restrict the number of failed trials without ATTs to a very small number, limiting automated programs as used by attackers to three free password guesses for a targeted account, even if different machines from a botnet are used. Several other techniques are deployed in practice, including: allowing login attempts without ATTs from a different machine, when a certain number of failed attempts occur from a given machine; allowing more attempts without ATTs after a timeout period; and time limited account locking. Many existing techniques and proposals involve ATTs, with the underlying assumption that these challenges are sufficiently difficult for bots and easy for most people.

Human-memorable passwords are a mainstay of computer security. To decrease vulnerability of passwords to brute-force dictionary attacks, many organizations enforce complicated password-creation rules and require that passwords include numerals and special characters. Human-memorable passwords are a mainstay of computer security. To decrease vulnerability of passwords to brute-force dictionary attacks, many organizations enforce complicated password-creation rules and require that passwords include numerals and special characters.

An alphabetical password generated by a human, even if it is not a dictionary word, is unlikely to be uniformly distributed in the space of alphabet sequences. In fact, if asked to pick a sequence of characters at random, it is likely that an English-speaking user will generate a sequence in which each character is roughly equidistributed with the frequency of its occurrence in English text. Analysis of password database reveals a significant number of alphabetical passwords which are neither dictionary words, nor random sequences attackers often must employ a large number of machines to avoid detection or lock-out., a user generally choose common and relatively weak passwords 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 trials without ATTs to a very small number (e.g., three). However, this inconveniences the legitimate user who then must answer an ATT on the next login attempt.
Several other techniques are deployed in practice, including: allowing login attempts without ATTs from a different machine, when a certain number of failed attempts occur from a given machine; allowing more attempts without ATTs after a timeout period; and time limited account locking. Many existing techniques and proposals involve ATTs, with the underlying assumption that these challenges are sufficiently difficult for bots and easy for most people. Enable the authentication by allowing the user to answer various secret questions and comparing those answers with those already stored in the cookie. Main aim in the project is to reduce the inconvenience caused to the legitimate user due to unauthorized access.

Passwords are the most common method of authenticating users, and will most likely continue to be widely used for the foreseeable future, due to their convenience and practicality for service providers and end-users. Although more secure authentication schemes have been suggested in the past, using smartcards or public key cryptography, none of them has been in widespread use in the consumer market.

An attacker that is interested in breaking into any account in the system, rather than targeting a specific account. Attacker can try many login attempts in parallel and circumvent the timing measure using the fact that user logins are typically handled by servers that can handle many login sessions in parallel. For example, the attacker can send a login attempt every

10 milliseconds, obtaining a throughput of 100 login attempts per second, regardless of how long the server delays the answers to the login attempts.
Markov models are commonly used in natural language processing, and are the heart of speech recognition systems . A Markov model defines a probability distribution over sequences of symbols. It allows sampling character sequences that have certain properties. A Markov models have been used before in the context of passwords.
Number of login attempts observed during each attack varied widely across the honeypots, from 1 or 2 up to hundreds or even thousands of attempts. The largest number of attempts observed during a single attack session was 9,311. This attack, observed on the honeypot located on the residential DSL connection, lasted for 117 minutes and accounted for nearly half of the login attempts observed on this honeypot.
Pinkas and Sander introduced the topic with a Strawman login protocol that requires answering an ATT challenge first before entering the username, password pair. Failing to answer the ATT correctly prevents the user from proceeding further. This protocol requires the adversary to pass an
ATT challenge for each password guessing attempt, in order to gain information about correctness of the guess.

II. PREVIOUS RESEARCH

[1]Dictionary attacks are the best known threats on the password-based authentication schemes. Based on Reverse Turing Test , some usable and scalable authentication schemes are proposed to defeat online dictionary attacks mounted by automated programs. However it is found that these authentication schemes are vulnerable to various online dictionary attacks. In this paper, a practical decision function is presented, based on which RTT authentication schemes are constructed and shown to be secure against all the known online dictionary attacks. After formally modeling of the adversary, the static and dynamic security of the authentication schemes are proved formally.
In online dictionary attacks, an adversary verifies whether a password is correct by interacting with the server. There are some measures devised to defeat online dictionary attacks in practice but not suitable for networking
service because some shortcomings persist as described in the following in more detail. The account is locked after a predefined number of failed login attempts. In this way, an adversary cannot verify a large number of passwords in a short time. This method is acceptable to single computer environment. But in network services, the drawbacks of this method are denial of service and vulnerable to global online dictionary attacks.

[2]Online services often use IP addresses as client identifiers when enforcing access-control decisions. Academic community has typically eschewed this approach, due to the effect that NATs, proxies, and dynamic addressing have on a server’s ability to identify individual clients. It is unclear to what extent these edge technologies actually impact the utility of using IP addresses as client identifiers. This paper provides some insights into this phenomenon. Mapping

out the size and extent of NATs and proxies, and characterizing the behavior of dynamic addressing. Using novel measurement techniques based on active web content, we present results gathered from 7 million clients over seven months. Most NATs are small, consisting of only a few hosts, while proxies are much more likely to serve many geographically distributed clients. Server can generally detect if a client is connecting through a NAT or proxy, or from a prefix using rapid DHCP reallocation. From measurement experiences developed and implemented a methodology by which a server can make a more informed decision on whether to rely on IP addresses for client identification or to use more heavyweight forms of client authentication.
With the exception of this subsection, this paper only discusses using IP addresses to enforcing access controls. IP addresses are used by servers in a variety of other settings, including geolocation and fraud detection. Websites use geolocation in both content personalization and access control contexts.

[3]In its Top-20 Security Risks report for 2007, the SANS Institute called brute-force password guessing attacks against SSH, FTP and telnet servers “the most common form of attack to compromise servers facing the Internet.” A recent study also suggests that Linux systems may play an important role in the command and control networks for botnets. Defending against brute-force SSH attacks may therefore prove to be a key factor in the effort to disrupt these networks. In this paper, we report on a study of brute-force SSH attacks observed on three very different networks: an Internet-connected small business network, a residential system with a DSL Internet connection, and a university campus network. The similarities observed in the methods used to attack these disparate systems are quite striking. The evidence suggests that many brute-force attacks are based on pre-compiled lists of usernames and passwords, which are widely shared. Analysis of the passwords used in actual malicious traffic suggests that the common understanding of what constitutes a strong password may not be sufficient to protect systems from compromise. Study data are also used to evaluate the effectiveness of a variety of techniques designed to defend against these attacks.
The armies of compromised computer robots, known as botnets, have received a lot of attention over the past few years. To date, most of that attention has been focused on the compromised Windows machines thought to populate the ranks of botnet armies. Until the results of eBay’s recent study of internal security threats were publicized last fall, little attention was paid to the role compromised Linux systems might play in supporting botnets.
[4]The use of passwords is a major point of vulnerability in computer security, as passwords are often easy to guess by automated programs running dictionary attacks. Passwords remain the most widely used authentication method despite their well-known security weaknesses. User authentication is clearly a practical problem. From the perspective of a service provider this problem needs to be solved within real-world constraints such as the available hardware and software infrastructures. From a user’s perspective user friendliness is a key requirement. In this paper we suggest a novel authentication scheme that preserves the advantages of conventional password authentication, while simultaneously raising the costs of online dictionary attacks by orders of magnitude. The proposed scheme is easy to implement and overcomes some of the difficulties of previously suggested methods of improving the security of user authentication schemes. Key idea is to efficiently combine traditional password authentication with a challenge that is very easy to answer by human users, but is infeasible for automated programs attempting to run dictionary attacks. This is done without affecting the usability of the system. The proposed scheme also provides better protection against denial of service attacks against user accounts.

[5]Human-memorable passwords are a mainstay of computer security. To decrease vulnerability of passwords to brute-force dictionary attacks, many organizations enforce complicated password creation rules and require that passwords include numerals and special characters. The paper demonstrate that as long as passwords remain human-memorable, they are vulnerable to “smart-dictionary” attacks even when the space of potential passwords is large. The first insight is that the distribution of letters in easy-to-remember passwords is likely to be similar to the distribution of letters in the users’ native language. Using standard Markov modeling techniques from natural language processing, this can be used to dramatically reduce the size of the password space to be searched. The second contribution is an algorithm for efficient enumeration of the remaining password space. This allows application of time-space tradeoff techniques, limiting memory accesses to a relatively small table of “partial dictionary” sizes and enabling a very fast dictionary attack evaluated the method on a database of real-world user password hashes. Our algorithm successfully recovered 67.6% of the passwords using a 2×109 search space. This is a much higher percentage than Oechslin’s “rainbow” attack, which is the fastest currently known technique for searching large keyspaces. These results call into question viability of human-memorable character-sequence passwords as an authentication mechanism.

III. PROPOSED SYSTEM

In the Proposed Model of implementation, user's input of several Parameters is verified by the server. User will be giving his / her User ID which is verified and compared with the User ID of the same user in the Cookies. This process of Server Authentication id called as Automated Turing Tests (ATT). Comparison of user Name from the user with the User name in the Cookies of the user's Machine. User's System IP of login is verified with the System IP of the same user when registered. User's captcha is also verified.
The general idea behind PGRP is that except for the following two cases, all remote hosts must correctly answer an ATT challenge prior to being informed whether access is granted or the login attempt is unsuccessful: when the number of failed login attempts for a given username is very small and when the remote host has successfully logged in using the same username in the past.

3.4.1 USER REGISTRATION:

In the online accessing system we have to register the user with certain details for his future retrieval process. Without registering, a user can't access the further details. For registering, the user should give the User name, Captcha, and password. Once a user registered his details he can access the online facilities further. Each user will be identified by a unique username, password, System IP, Web browser which are stored in cookies.

3.4.2 SERVER:

A server is a computer program running to serve the requests of other programs, the "clients". Thus, the "server" performs some computational task on behalf of "clients". The clients either run on the same computer or connect through the network. Here the Server acts as the main resource for the client. Server is responsible for maintaining all the client information and further used for the Authentication process when the user reenters the online process.

3.4.3 COOKIES VERIFICATION:

The details got from the user during registration are stored in cookies and then sent to the server. When the user reenters the online process, for e.g.: banking process the cookie will verify for the details such as Username, System IP, captcha, Password, Web browser and the number of attempts of the user and then allows only the authorized user.

3.4.4 SYSTEM IP & CATCHA:

The System IP is nothing but the IP address of the System which was used by the user during online registration process. Catcha is the verification text displayed inside the box. These two verification will help the online processing to be more secure.

3.4.5 PASSWORD & WEB BROWSER:

The user should enter the password and the type of web browser during registration itself. So that it will be stored and it is used with other details for the authentication process. If the user needs to change the web browser or the System IP then he need to answer the secret Question.

3.4.6 VERIFICATION:

In this project, Verification is done by means of Cookies. Using cookies, the data already entered by the user is compared with the currently given data by the user. If the Username, System IP, catcha, Password and Web browser matches, then he will be allowed for the further processing or else access is denied. Among the above details if the user needs to change the System IP or Web browser then he need to answer the secret question and catcha. After authentication, updations will be done.

3.5 DECISION FUNCTION FOR REQUESTING ATTS

The decision to challenge the user with an ATT depends on two factors whether the user has authenticated successfully from the same machine previously and the total number of failed login attempts for a specific user account. As in the condition in entering a correct username-password pair, the user will not be asked to answer an ATT challenge in the following cases a valid cookie is received from the user machine and the number of failed
login attempts from the user machine's IP address for that username is less than k1 over a time period determined by t3 the user machine's IP address is in the white list W and the number of failed login attempts from this IP address for that

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Volume 1, Issue 10, December 2013

username is less than k1 over a time period determined by t3 the number of failed login attempts from any machine for that username, is below a threshold k2 over a time period determined by t2. The last case enables a user who tries to log in from a new machine/IP address for the first time before k2 is reached to proceed without an ATT. However, if the number of failed login attempts for the username exceeds the threshold k2 , this might indicate a guessing attack and hence the user must pass an ATT challenge.

3.6 PASSWORD GUESSING RESISTANT PROTOCOL (PGRP)

The objectives for PGRP include the login protocol should make brute-force and dictionary attacks ineffective even for adversaries with access to large botnets. The protocol should be easy to deploy and scalable, requiring minimum computational resources in terms of memory, processing time, and disk space.

3.6.1 DATA STRUCTURE AND FUNCTION DESCRIPTION

W: A list of source IP address, username pairs such that for each pair, a successful login from the source IP address has been initiated for the username previously. FT: Each entry in this table represents the number of failed login attempts for a valid username, un. A maximum of *k*2 failed login attempts are recorded. FS: Each entry in this table represents the number of failed login attempts for each pair 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.

3.6.2 ALGORITHM

PGRP ALGORITHM

```
 Begin ReadCredential(un, pw, cookie)
if LoginCorrect(un, pw) then
if (((V alid(cookie, un,k1,true) ∨ ((srcIP, un) ∈ W)) ∧ (FS[srcIP, un] < k1))  (FT[un] < k2)) then
FS[srcIP, un]  0
 Add srcIP to W
GrantAccess(un, cookie)
 else if (ATTChallenge() = Pass) then
FS[srcIP, un]  0
 Add srcIP to W
 GrantAccess(un, cookie)  else
 Message('The answer to the ATT challenge is incorrect')
 else
if ((Valid(cookie, un,k1,false)  ((srcIP, un)  W))  (FS[srcIP, un] < k1)) then
 FS[srcIP, un] FS[srcIP, un] + 1
 Message('The username or password is incorrect')
 else if (ValidUsername(un)  (FT[un] < k2)) then
 FT[un]  FT[un] + 1
 Message('The username or password is incorrect')
 else
 if (ATTChallenge() = Pass) then
 Message('The username or password is incorrect')
 else
 Message('The answer to the ATT challenge is incorrect')
 end
```

User gives all the personal details to the server and server stores all the details in the cookies. When the user enters the username password cookie checks it and if it is valid user name and password checks the IP address. If IP address is correct then att passes the user for entering the web browser and can authenticate. If the answer is incorrect att will ask to enter the valid user name and password. If user details is wrong then Att will send a message that "the user name or password is incorrect". FS[srcIP, un] FS[srcIP, un] + 1 else if it is a valid username and password (un) (FT[un] < k2)) then FT[un] FT[un] + 1 and the att challenge passes.

IV. RESULT ANALYSIS

Table 4.1 compares PGRP with the PS and VS protocols. The $c \geq 2$ in Q4 for the VS protocol. The answer to Q1 depends on the threshold *k*2. The adversary can eliminate only *k*2 passwords without answering ATTs. Likewise, for Q2, the expected number of ATTs the adversary must answer to correctly guess a password is one-half of the remaining passwords of the password space after subtracting the number of login attempts that do not require ATTs. PGRP is comparable to the VS protocol in multi-account attacks, PS seems slightly better than PGRP but only for login systems with a large number of users as in equation .

	Strawman protocol	Ps	Vs		PGRP
			Owner	Non-owner	
Q1	O	O	O	mb_1/N	mk_2/N
Q2	c/N	c/Pn	$\leq$ min(cp , b_2 + c)/N	(mb_1 + c)/N	(mk_2 + c)/N

Table 4.1 Security Analysis

$$(m \,.\, k_2 + c)/ N \;>\; c /(pN)$$

$$m >(c/k_2)\{(1/p_2) – 1\}$$

PGRP protocol, an adversary may be able to guess a subset of the valid usernames which is undesirable in certain cases .The *FT* list is not updated if the username is invalid, thus an ATT will be requested for each login attempt with an invalid username. Therefore, the adversary could generate a list of valid usernames as follows: if an attempted username requires an ATT for the first login attempt, the username is considered invalid; otherwise, the username is valid.

VS protocol maintained the number of entries 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. The PS nor VS protocol uses IP addresses. The most expensive server operation in PS, VS, and PGRP is generating an ATT. In PGRP three tables must be maintained. First, the whitelist, W is expected to

grow linearly with the number of users. At any given time, W contains a list of pairs that have been successfully authenticated in the last t1 units of time. The number of entries in FT increases by one whenever a remote host makes a failed login attempt using a valid username, if the username is not

already in FT, and the remote host's IP address is not in W . Therefore the VS protocol, the total number of valid usernames in the login server puts an upper bound on the number of entries in FT.

V. CONCLUSION

Online password guessing attacks on password only systems have been observed for decades Present-day attackers targeting such systems are empowered by having control of thousand to million node botnets. In previous ATT-based login protocols, there exists a security-usability trade-off with respect to the number of free failed login attempts versus user login convenience. PGRP is more restrictive against brute force and dictionary attacks while safely allowing a large number of free failed attempts for legitimate users. Our empirical experiments on two datasets gathered from operational network environments show that while PGRP is apparently more effective in preventing password guessing attacks (without answering ATT challenges), it also offers more convenient login experience, e.g., fewer ATT challenges for legitimate users even if no cookies are available. PGRP appears suitable for organizations of both small and large number of user accounts. The required system resources (e.g., memory space) are linearly proportional to the number of users in a system. PGRP can also be used with remote login service where cookies are not applicable .

REFERENCES

[1] M. Weir, S. Aggarwal, M. Collins, and H. Stern. Testing metrics for password creation policies by attacking large sets of revealed passwords. (2010)
[2] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses. (2007)
[3] J. Yan and A. S. E. Ahmad. A low-cost attack on a Microsoft CAPTCHA. In ACM Computer and Communications Security .(2008)
[4] M. Casado and M. J. Freedman. Peering through the shroud:The effect of edge opacity on ip-based client identification. (2007)
[5] S. Chiasson, P. C. van Oorschot, and R. Biddle. A usability study and critique of two password managers. (2006)
[6] D. Florˆencio, C. Herley, and B. Coskun. Do strong web passwords accomplish anything? In USENIX workshop on Hot topics in security. (2007)
[7] K. Fu, E. Sit, K. Smith, and N. Feamster. Dos and don’ts of client authentication on the web. (2001)
[8] P. Hansteen. Rickrolled? Get Ready for the Hail Mary Cloud! (2010)
[9] Y. He and Z. Han. User authentication with provable security against online dictionary attacks. (2009)
[10] T. Kohno, A. Broido, and K. C. Claffy. Remote physical device fingerprinting. (2005)
[11] M. Motoyama, K. Levchenko, C. Kanich, D. Mccoy, G. M. Voelker, and S. Savage. Re: CAPTCHAs understanding CAPTCHAsolving services in an economic context. (2010)
[12] C. Namprempre and M. N. Dailey. Mitigating dictionary attacks with text-graphics character CAPTCHAs. (2007)