Volume 3, Issue 8, August 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Prevention of Tool Based Online Password Guessing Attacks

Santhosh Kumar Samudrala
Dept of Computer Science & Eng
M.Tech, Software Engineering
Kakatiya Institute Of Tech & Sci
Kakatiya university
Warangal, India

Venkatramulu Sunkari
Dept of Computer Science & Eng
Associate Professor
Kakatiya Institute Of Tech & Sci
Kakatiya University
Warangal, India

Dr. CV Guru Rao
Dept of Computer Science & Eng
Professor and Head of CSE
SR Engineering College
Jawaharlal Nehru Tech
University Hyderabad, India

Abstract— Incredible force along with dictionary attacks on password-only remote login services at the moment are widespread and rising. Enabling effortless login regarding legitimate customers while preventing such attacks is often a difficult trouble. Automated Turing Assessments (ATTs) continue to be an powerful, easy-to-deploy procedure for identify automated malicious login attempts having reasonable charge of trouble to customers. In this particular paper, we talk about the inadequacy of existing along with proposed login protocols made to address significant scale online dictionary attacks e. grams., from some sort of botnet of tens of thousands of nodes. We propose a brand new Password Estimating Resistant Method (PGRP), derived on revisiting earlier proposals made to restrict this kind of attacks. While PGRP limits the total number of login tries from unfamiliar remote hosts to as low as a solitary attempt for each username, legitimate users in most cases e. grams., when attempts are made of known, frequently-used machines can make several unsuccessful login tries before becoming challenged by having an ATT.

***Keywords*— Password Guessing, 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. As an example of SSH password guessing attacks, one experimental Linux honeypot setup has been reported to suffer on average 2,805 SSH malicious login attempts per computer per day. Interestingly, SSH servers that disallow standard password authentication may also suffer guessing attacks, 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 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. Consequently, attackers often must employ a large number of machines to avoid detection or lock-out. On the other hand, as users generally choose common and relatively weak passwords thus allowing effective password dictionaries, 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, 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. 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 time-out 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. However, users increasingly dislike ATTs as these are perceived as an unnecessary extra step for usability issues related to commonly used CAPTCHAs. Due to successful attacks 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 human users, fueling a growing tension between security and usability of ATTs. Therefore, we 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.

II. Previous Work

Only recently, two-factor authentication systems based on mobile devices have started to gather some interest within the research community. An authentication mechanism is presented which requires both a Web and a GPRS connection. The end user enters userid/password credentials via the web-based interface and receives a OTP via SMS on his mobile phone, which he must then type in to be granted access to the system. The GPRS connection can be inconvenient for the user since it can be very costly and network quality of service (including availability of network coverage) is not always satisfactory. In addition, security of the scheme relies on information (image) related to the user,

but the underlying rationale needs to be expanded with further arguments. The work of contributions in the context of mobile payments and offer similar approaches to mobile user authentication. Messages are routed to the mobile device through the GSM-SMS service and rely on the phone number as a means of authentication. the researchers develop a stronger authentication mechanism based on information stored in a Subscriber Identity Module (SIM) card and the Authentication Centre of the subscriber's carrier. One drawback of this approach is the necessity for the financial service provider to enter into a prior agreement with the network carrier.

Overview of authentication mechanisms: for home banking systems Knowledge-based authentication techniques are still common but, instead of relying on the user entering only a password at the login prompt, they introduce other mechanisms to make it extremely difficult for attackers to steal and reuse authentication information. In some systems, implementations protection against brute-force and key-logger attacks is achieved by means of drop-down menus to enter passwords characters. This method is used by a number of banks in the UK, including Barclays, Lloyds TSB Group and the Royal Bank of Scotland. In others cases, a challenge composed of the image of a character string is displayed; users then respond with the characters corresponding to their personal identification numbers (PIN). Vendors offering this solution include Swivel Technologies. A more sophisticated challenge/ response mechanism displays the image of a keypad, with the keys in random positions, and asks users to point-and-click on each number corresponding to their PIN. Virtual keypads can help prevent keyboard-logger-type attacks, but they are vulnerable to special purpose Trojans that capture screenshots on each mouse-click. In use by some banks, they do not afford sufficient protection against phishing attacks. Several banks have distributed thousands of security tokens to their customers to promote stronger two-factor authentication mechanisms. Many are based on stand-alone tokens (e.g. RSA Security's RSASecurID) while others (less frequently), on devices that plug into Universal Serial Bus (USB) ports. However, the cost of such offerings is not insignificant. Alternative authentication mechanisms are often based on the ``something you possess'' paradigm. In its simplest form, a bank provides the customer with a list of OTPs or transaction numbers (TANs) on a card. This may be a simple printed list, a booklet of vouchers or a scratch card. Each time a user accesses the Bank's online services, the next number in the list is used. In combination with a PIN, this method is sometimes called PIN and TAN, but administration is slow and cumbersome, discouraging banks from its use. As an alternative, Germany's Postbank uses TAN via the cellular phone Short Message Service (SMS). Although this represents a simple form of two-factor authentication, it is vulnerable to phishing. A spoofed online bank site can prompt the customer for userid, password and TAN. The bogus site responds that the TAN has already been used, so customers are convinced they forgot to scratch the number off the list. Now the phisher has a good TAN. To defeat this attack, some banks prompt customers for TANs at random, rather than sequentially. Displaying the serial number of a TAN list enables the bank to issue a new one to customers, well before previous lists have expired. This also provides (weak) authentication of the bank's web site to users, defending against many kinds of phishing attacks. When properly implemented, TAN lists can provide robust authentication, but experience has shown that they are expensive to administer and inconvenient for customers. Another way to prevent attackers from reusing authentication information is to allow customer access only from a registered device (PC, personal digital assistant, phone, etc). Devices may be assigned an ID in the form of a secured cookie, shared flash object or other unique “signature” derived from device characteristics bound to an IP address, geocoding information, client software or hardware configurations. Because device identification (www.passmarksecurity.com, www.safe3w.com) usually limits user access to registered devices, it may reduce mobility, which remains an attractive feature of Internet banking. The challenge is to devise a convenient and secure device registration process. In addition, there is no way of preventing fraudulent access by a family member or colleague with physical access to the device. Hence, device identification can significantly strengthen authentication mechanisms in general, but it needs a robust device registration process. For bank customers possessing a mobile device (e.g. a GSM phone), SMS messaging offers a secure out-of-band service that can represent an authentication system that is as secure as any dedicated hardware token, but at a lower cost and with greater convenience. Typically, after entering valid credentials in a web site, users receive a OTP in an SMS message delivered to their mobile phone or personal digital assistant. This solution is attractive for both the bank and the customer given that there is no additional token to issue or carry. However, SMS coverage can be patchy in some areas, and high network latency can delay OTPs for several minutes. Also, most users will be charged for the SMS message every time they access the bank. This mechanism provides increased protection against phishing since it provides mutual authentication (albeit in a weak form) of parties: only the bank knows the registered mobile phone number to which to send the SMS message. This mechanism has been adopted by National Australia Bank (NAB), New Zealand's ASB Bank and BankDirect Internet Bank service. Interestingly, NAB is also providing 70,000 hardware tokens for online customers, presumably for those who live in areas without SMS coverage do not possess SMS phones or, simply do not want to use SMSs. SMS-based OTP is a relatively mature method of providing robust customer authentication and significant protection against phishing, but may be severely limited by mobile network quality of service.

At a high level of abstraction the interaction between User UA and Bank BK is modeled as a sequence of two sessions following the initialization phase: initialization: in this phase user UA registers with the Bank and is assigned a profile including a set of login credentials to access the Home Banking System. UA is assigned a signature/verification key pair and a digital certificate signed by a certification authority CA managed by the Bank. UA's profile will also contain an additional credential keyUA used as the basis for the two-factor authentication mechanism. In a typical usage scenario this cryptographic key can be automatically generated, assigned to the users' profile and stored in a database system hosted by the Bank we stress that this occurs only in the initialization phase or at some later time for key renewal. Alternatively, key keyUA may be a low entropy password chosen by the user or generated automatically by the Bank. The key is actually embedded in the authentication software e.g. Java which is uploaded to the portable device TB either

from the terminal TA we assume that terminal has already downloaded the software in a secure SSL session via a mode 3 Bluetooth session or from some hardware token e.g. USB memory device. Protection of the key stored in the mobile phone e.g. in case of loss may be afforded by a simple and common 4/6-digit PIN or by stronger biometric mechanisms;

session-1: principal UA establishes a secure channel with her bank BK . To this end, UA runs a web-based SSL/TLS application on her portable computer TA to interact with the Home Banking System. This session uses digital certificates and access is granted to the user on presentation of the login credentials. The actual two-factor authentication is implemented through an automatic mechanism realised with a Bluetooth-enabled mobile phone session-2;

session-2: TA laptop and TB Smartphone engage in a application level conversation by invoking an authentication protocol -Encrypted Key Authentication Protocol. The setting is in a symmetric trust model since there is no non-repudiation of origin for exchanged messages. Authentication is achieved by both principals proving knowledge of the shared key without revealing any information on the key itself to a malicious third party. On receipt of, principal TB can decrypt this message if she knows; when no error condition occurs TB recovers, otherwise TB aborts the protocol; computes key from the exclusive-or of and or some other mask generation function. TA verifies the received tag by construction of key and aborts the protocol in case of failure; Both and TB accept and mutually prove possession of the password keyUA to each other. The protocol allows a user possessing a Bluetooth enabled phone terminal TB to automatically authenticate to the system home banking client TA and thus to Bank BK if the correct key keyUA is stored on it. A software agent running on terminal host, retrieves the password from the user's remote profile in session-1 and engages in the protocol execution with terminal TB. After successful authentication a session-id is generated that expires within a limited time frame in which case the authentication process must be repeated, provided the user has previously entered the correct credentials in the Web-based front-end application running on TA. One advantage of this mechanism is that the user could use a common set of credentials to logon to multiple network applications and use a different keyUA in each profile, with no need to remember it to authenticate via the mobile device.

III. Propsed System

A. Initialization

We assume that adversaries can solve a small percentage of ATTs, e.g., through automated programs, brute force mechanisms, and low paid workers. Incidents of attackers using IP addresses of known machines and cookie theft for targeted password guessing are also assumed to be minimal. Traditional password-based authentication is not suitable for any untrusted environment e.g., a key logger may record all keystrokes, including passwords in a system, and forward those to a remote attacker. The data integrity of cookies must be protected e.g., by a MAC using a key known only to the login server. 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 (however, such a host must pass an ATT challenge if it generates more failed login attempts than a pre specified threshold. In contrast to previous protocols, 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 (if any) and/or the remote host's IP address. In addition, if the number of failed login attempts for a specific username is below a threshold, the user is not required to answer an ATT challenge even if the login attempt is from a new machine for the first time.

B. Defence Functions

The main defense function include 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. A FT Each entry in this table represents the number of failed login attempts for a valid username, un. A maximum of k2 failed login attempts are recorded. Accessing non existing index returns 0. FS Each entry in this table 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 nonexistent 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 any time 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.

C. Decision Function requesting ATTs

These function are for ATT challenges as provided by the login server. 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. Upon 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, FS, is less than k1 over a time period determined by t3. 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, is less than k1 over a time period determined by t3; The number of failed login attempts from any machine for that username. The last case enables a user who tries to login 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.

D. Secure User support functions

The USF is most effectively implemented as a Browser Plugin, thus having the ability to access the Public Key of the web-site as well as rendering the CAPTCHA. Additionally, it needs to implement a local security mechanism for e.g. a Username/Password to ensure secure access to the Image List and Bit Position Map database. USF Browser Plugin could be appropriately modified to read the database. There may be a potential security risk here, as the Browser on the Public computer could be compromised to make a local copy of the User-specific Image List and Bit Position Map, which would enable a human-assisted attack for the next login attempt. To prevent this we secure the database server which has the images.

E. Exchange Image List

The initial Image List and Bit Position Map can be suitably formatted and sent within the confirmation email, which is sent to Users when they register for an account with the authentic web-site. As a general approach, the Image List and Bit Position Map could be sent as a hidden parameter using within the HTML page which is rendered to the user, after successful login. The Browser Plug-in would implement the additional parsing logic to read and store it. This mechanism also has the advantage that it can be used uniformly after initial User registration at the website, as well as after every successful login. The website could also have a hyperlink to enable the User to create and receive a new Image List and Bit Position Map. Such, a User-controlled remapping enables flexibility in choosing the right time for renewing the database. As a result, if any Public Key-embedded CAPTCHA-related data was compromised at the Public computer, it is essentially nullified via the generation of a new Image List and Bit Position Map.

F. Recovering from Identity Attacks

In this module we check if the User is attacked for example due to access from a Public Computer, then they need a backup mechanism to recover the right to their username. To enable this, the USF within the User's trusted identity may collect and store the various CAPTCHA based challenges that it receives over a period of time. Now, using the response we can Identity Attack, it could present to the authentic website, to stake it's claim to a particular Username/Password. The website would check whether the CAPTCHAs are indeed valid, whether they contained the correct Image and corresponding sub-part of the Public Key, at those respective time-instances, as per its history of various Image Lists and Bit Position Maps for the Username.

IV. Results

The concept of this paper is implemented and different results are shown below, The proposed paper is implemented in Java technology on a Pentium-IV PC with minimum 20 GB hard-disk and 1GB RAM. The propose paper's concepts shows efficient results and has been efficiently tested on different Datasets.

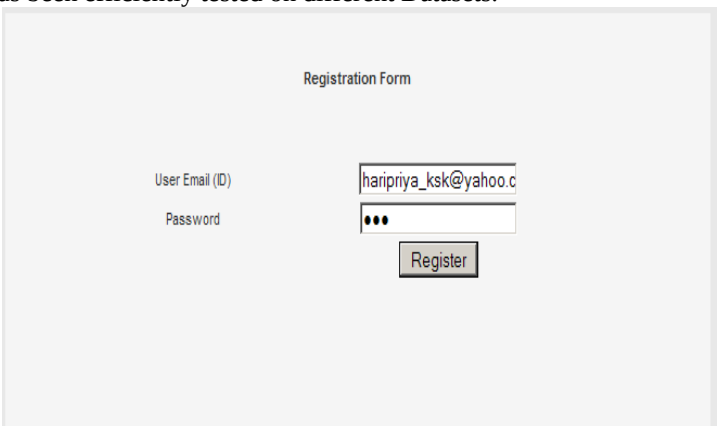

Fig. 1 Initial Login form.

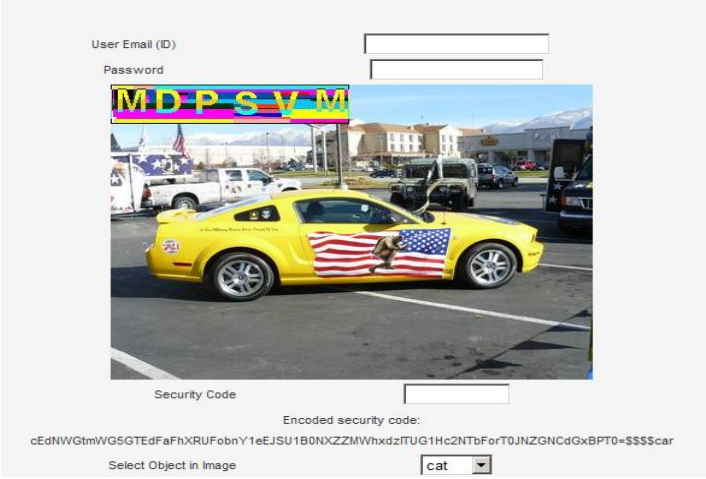

Fig. 1 Enhanced ATTs Embedded Captcha

Fig. 3 Enhanced ATTs Embedded Captcha

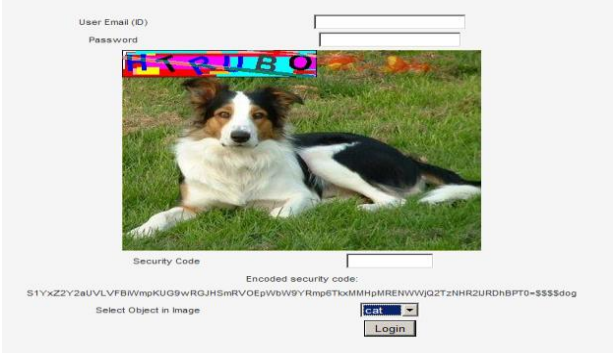

Fig. 4 Enhanced ATTs Embedded Captcha

VI. 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 e.g., less ATTs and other requirements. In contrast, PGRP is more restrictive against brute force and dictionary attacks while safely allowing a large number of free failed attempts for legitimate users. Our experiments 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.

References

[1] M. Casado and M.J. Freedman, "Peering through the Shroud: The Effect of Edge Opacity on Ip-Based Client Identification," Proc. Fourth USENIX Symp. Networked Systems Design and Implementation (NDSS '07), 2007.

[2] 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.

[3] 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.

[4] 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.

[5] D. Florencio, C. Herley, and B. Coskun, "Do Strong Web Passwords Accomplish Anything?," Proc. USENIX Workshop Hot Topics in Security (HotSec '07), pp. 1-6, 2007.

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

[7] 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.

[8] C. Nampempre and M.N. Dailey, "Mitigating Dictionary Attacks with Text-Graphics Character Captchas," IEICE Trans. Fundamentals of Electronics, Comm. and Computer Sciences, vol. E90-A, no. 1, pp. 179-186, 2007.

[9] 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.

[10] M. Weir, S. Aggarwal, M. Collins, and H. Stern, "Testing Metrics for Password Creation Policies by Attacking Large Sets of Revealed Passwords," Proc. 17th ACM Conf. Computer and Comm. Security, pp. 162-175, 2010.

[11] 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.

[12] 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.

