Graphical Secure Password Method against Online Password Hackers(Guessing Attacks)

G.Sudheer Reddy[1],G.Venkata Prasad[2] ,V.Hari Prasad[3]

Abstract:
Passwords are a common form of authentication and are often the only barrier between a user and your personal information. There are several programs attackers can use to help guess or "crack" passwords, but by choosing good passwords and keeping them confidential, you can make it more difficult for an unauthorized person to access your information.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 (e.g., when attempts are made from known, frequently-used machines) can make several failed login attempts before being challenged with an ATT. We analyze the performance of PGRP with two real-world data sets and find it more promising than existing proposals.

KeyWords:
Online password, graphical secure password

Introduction:

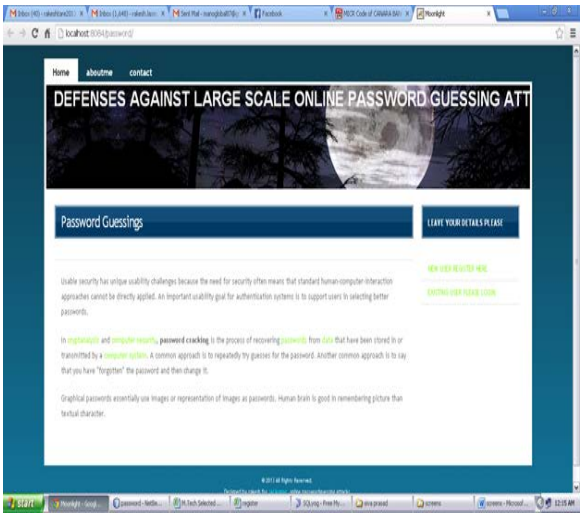
The common weakness in these hacks is the password. It's an artifact from a time when our computers were not hyper-connected. Today, nothing you do, no precaution you take, no long or random string of characters can stop a truly dedicated and devious individual from cracking your account. The age of the password has come to an end; we just
haven't realized it yet. Passwords are as old as civilization. And for as long as they've existed, people have been breaking them.
In 413 BC, at the height of the Peloponnesian War, theAthenian general Demosthenes landed in Sicily with5,000 soldiers to assist in the attack on Syracusae. Things were looking good for the Greeks. Syracusae, a key ally of Sparta, seemed sure to fall. But during a chaotic nighttime battle at Epipole, Demosthenes'
forces were scattered, and while attempting to regroup they began calling out their watchword, a prearranged term that would identify soldiers as friendly. The Syracusans picked up on the code and passed it quietly through their ranks. At times when the Greeks looked too formidable, the watchword allowed their opponents to pose as allies. Employing this ruse, theundermatched Syracusans decimated the invaders, and when the sun rose, their cavalry mopped up the rest. It was a turning point in the war.The first computers to use passwords were likely those in MIT's Compatible Time-Sharing System, developed in 1961. To limit the time any one user could spend on the system, CTSS used a login to ration access. It only took until 1962 when a PhD
student named Allan Scherr, wanting more than his four-hour allotment, defeated the login with a simple hack: He located the file containing the passwords and printed out all of them. After that, he got as much time as he wanted.
During the formative years of the web, as we all went online, passwords worked pretty well. This was due largely to how little data they actually needed to protect. Our passwords were limited to a handful of applications: an ISP for email and maybe an ecommerce site or two. Because almost no personal information was in the cloud—the cloud was barely a wisp at that point
—there was little payoff for breaking into an individual's accounts; the serious hackers were still going after big corporate systems.
So we were lulled into complacency. Email addresses morphed into a sort of universal login, serving as our username just about everywhere. This practice persisted even as the number of accounts—the number of failure points—grew exponentially. Web-based email was the gateway to a new slate of cloud apps. We began banking in the cloud, tracking our finances in the cloud, and doing our taxes in the cloud. We stashed our photos, our documents, our data in the cloud.
Eventually, as the number of epic hacks increased, we started to lean on a curious psychological crutch: the notion of the "strong" password. It's the compromise that growing web companies came up with to keep people signing up and entrusting data to their sites. It's the Band-Aid that's now being washed away in a river of blood.
One proposal to reduce problems related to text passwords is to use password managers. These typically require that users remember only a master password. They store (or re-generate) and send on behalf of the user, to web sites hosting user accounts, the appropriate passwords. Ideally the latter are generated by the manager itself and are stronger than user-chosen passwords. However, implemen-tations of password managers introduce their own usability issues [Chiasson et al. 2006] that can exacerbate security problems, and their centralized

Manuscript received August 5, 2014
Manuscript revised August 20, 2014

architecture in-troduces a single point of failure and attractive target: attacker access to the master password provides control over all of the user's managed accounts.
When text password users resort to unsafe coping strategies, such as reusing pass-words across accounts to help with memorability, the decrease in security cannot be addressed by simply strengthening, in isolation, the underlying technical secu-rity of a system. Usability issues often significantly impact its real-world security. User interface design decisions may unintentionally sway user behaviour towards less secure behaviour. Successful authentication solutions must thus also include improved usability design based on appropriate research taking into account the abilities and limitations of the target users. In graphical passwords, human mem-ory for visual information is leveraged in hope of a reduced memory burden that will facilitate the selection and use of more secure or less predictable passwords, dissuading users from unsafe coping practices.
Early surveys of graphical passwords are available [Monrose and Reiter 2005; Suo et al. 2005]. More recent papers briefly summarize and categorize 12 schemes [Hafiz et al. 2008], and review numerous graphical password systems while offering usability guidelines for their design [Renaud 2009a]. In this paper we provide a comprehensive review of the first twelve years of published research on graphical passwords, and reflect on it. It is now clear that the graphical nature of schemes does not by itself avoid the problems typical of text password systems. However, while proposals in this first period of research exhibit some familiar problems, we see signs that an emerging second generation of research will build on this knowledge and leverage graphical elements in new ways to avoid the old problems.
As will be seen, early graphical password systems tended to focus on one par-ticular strength, for example being resistant to shoulder-surfing, but testing and analysis showed that they were vulnerable to one or more other types of attacks. Except in very specific environments, these would not provide adequate security.

Security:

An authentication system must provide adequate security for its intended environ-ment, otherwise it fails to meet its primary goal. A proposed system should at minimum be evaluated against common attacks to determine if it satisfies security requirements. A brief introduction is provided here
We classify the types of attacks on knowledge-based authentication into two general categories: guessing and capture attacks. In successful guessing attacks, attackers are able to either exhaustively search through the entire theoretical pass-word space, or predict higher probability passwords (i.e., create a dictionary of likely passwords) so as to obtain an acceptable success rate within a manageable number of guesses. Guessing attacks may be conducted online through the intended login interface or offline if some verifiable text [Gong et al. 1993] (e.g., hashes) can be used to assess the correctness of guesses. Authentication systems with small the-oretical password spaces or with identifiable patterns in user choice of passwords are especially vulnerable to guessing attacks.
Password capture attacks involve directly obtaining the password, or part thereof, by capturing login credentials when entered by the user, or by tricking the user into divulging their password. Shoulder-surfing, phishing, and some kinds of malware are three common forms of capture attacks. In shoulder-surfing, credentials are cap-tured by direct observation of the login process or through some external recording device such as a video camera. Phishing is a type of social engineering attack where users are tricked into entering their credentials at a fraudulent website that records users' input. Malware uses unauthorized software installed on client computers or servers to capture keyboard, mouse, or screen output, which is then parsed to find login credentials.

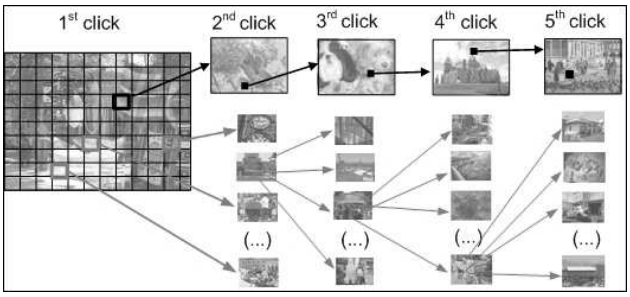


Graphical password or graphical user authentication (GUA):

A graphical password is an authentication system that works by having the user select from images, in a specific order, presented in a graphical user interface (GUI). For this reason, the graphical-password approach is sometimes called graphical user authentication (GUA).

A graphical password is easier than a text-based password for most people to remember. Suppose an 8-character password is necessary to gain entry into a particular computer network. Instead of w8KiJ72c, for example, a user might select images of the earth (from among a

screen full of real and fictitious planets), the country of France (from a map of the world), the city of Nice (from a map of France), a white stucco house with arched doorways and red tiles on the roof, a green plastic cooler with a white lid, a package of Gouda cheese, a bottle of grape juice, and a pink paper cup with little green stars around its upper edge and three red bands around the middle.



Graphical passwords may offer better security than text-based passwords because many people, in an attempt to memorize text-based passwords, use plain words (rather than the recommended jumble of characters).

A dictionary search can often hit on a password and allow a hacker to gain entry into a system in seconds. But if a series of selectable images is used on successive screen pages, and if there are many images on each page, a hacker must try every possible combination at random. If there are 100 images on each of the 8 pages in an 8 image password, there are 1008, or 10 quadrillion (10,000,000,000,000,000), possible combinations that could form the graphical password! If the system has a built-in delay of only 0.1 second following the selection of each image until the presentation of the next page, it would take (on average) millions of years to break into the system by hitting it with random image sequences

dictates the next image. If they dislike the resulting images, they could create a new password involving different click-points to get different images. We envision that CCP fits into an authentication model where a user has a client device (which displays the images) to access an online server (which authenticates the user). We assume that the images are stored server-side with client communication through SSL/TLS. For further discussion, see Section 6.For implementation, CCP initially functions like PassPoints. During password creation, a discretization method (e.g., see [1]) is used to determine aclick-point's tolerance square and corresponding grid. For each click-point in a subsequent login attempt, this grid is retrieved and used to determine whether the click-point falls within tolerance of the original point. With CCP, we furtherneed to determine which next-image to display.

Similar to the PassPoints studies, our example system had images of size 451x331 pixels and tolerance squares of 19x19 pixels. If we used robust dis- cretization [1], we would have 3 overlapping candidate grids each containing approximately 400 squares and in the simplest design, 1200 tolerance squares per image (although only 400 are used in a given grid). We use a function f(username, currentImage, currentT oleranceSquare) that uniquely maps each tolerance square to a next-image. This suggests a minimum set of 1200 images required at each stage. One argument against using fewer images, and having multiple tolerance squares map to the same next-image, is that this could po-tentially result in misleading implicit feedback in (albeit rare) situations where users click on an incorrect point yet still see the correct next-image.

Password Guessing Resistant Protocol (PGRP) Code:

```
import java.awt.Color; import java.awt.Dimension; import
java.awt.Graphics; import java.awt.event.*; import
java.awt.event.*;
import java.awt.image.BufferedImage;
import java.io.File; import java.util.Random; import
java.util.*;
import java.util.Vector;
import javax.imageio.ImageIO;
import javax.swing.JOptionPane;
import java.io.*;
import java.sql.*;
public LoginImagePanel()
{
initComponents();

loginimagePanel.addMouseListener(new
MouseListener(){
public void mouseClicked(java.awt.event.MouseEvent
evt)
{
System.out.println("mouse clicked"+evt.getX()+"
"+evt.getY());
x = evt.getX();
y = evt.getY(); try
{ Class.forName("com.mysql.jdbc.Driver");
con =
DriverManager.getConnection("jdbc:mysql://localhost:33
06/
graphicalpassword","root","root");
st = con.createStatement();
rs = st.executeQuery("select sound from imageselection
where username="'+LoginFrame.loginID+"' and
xcordinate="'+x+"' and ycordinate="'+y+"'");
if(rs.next())
{
//  String[]
roseindia={"Dolphin","Duck","Lion","Rhino","Bird"};
//Random randomGenerator = new Random();
```

```
System.out.println(LoginFrame.loginID);
// for(int i=0;i<5;i++)
//{
//System.out.println(roseindia[i]);
//int randomInt = randomGenerator.nextInt(5);
musicfile = rs.getString(1);}
else
{
String[]
roseindia={"Dolphin","Duck","Lion","Rhino","Bird"};
Random randomGenerator = new Random();
//  System.out.println("mouse clicked"+evt.getX()+"
"+evt.getY());
// x = evt.getX();
//  y = evt.getY();
System.out.println(LoginFrame.loginID);
// for(int i=0;i<5;i++)
//{
//System.out.println(roseindia[i]);
int randomInt = randomGenerator.nextInt(5);
musicfile = roseindia[randomInt];
//rd=req.getRequestDispatcher("/failure.jsp");
// out.println("welcome");
}
//rd.forward(req,res);
} catch(Exception e2)
{
//System.out.println("Exception : "+e2.toString());
//out.println(e2);
}
```

Future Work:

The proposed Cued Click Points scheme shows promise as a usable and memorable authentication mechanism. By taking advantage of users' ability to recognize images and the memory trigger associated with seeing a new image, CCP has advantages over PassPoints in terms of usability. Being cued as each image is shown and having to remember only one click-point per image appears easier than having to remember an ordered series of clicks on one image. In our small comparison group, users strongly preferred CCP.We believe that CCP offers a more secure alternative to PassPoints. CCP increases the workload for attackers by forcing them to first acquire image sets for each user, and then conduct hotspot analysis on each of these images. Furthermore, the system's flexibility to increase the overall number of images in the system allows us to arbitrarily increase this workload. Future work should include a thorough assessment of the viability of CCP as an authentication mechanism, including a long term study of how these passwords work in practice and whether longer CCP passwords would be usable.The security of CCP also deserves closer examination, and should address how attackers might exploit the emergence of hotspots.

Conclusion:

Our tour of graphical password research reveals a rich palette of ideas, but few schemes that deliver on the original promise of addressing the known problems with text passwords. Indeed, review of the first era of graphical password schemes indicates that many of the same problems continue to re-surface. For graphical passwords to advance as a serious authentication alternative, we believe research must be conducted and presented in a manner allowing systematic examination and comparison of each scheme's main characteristics, showing how each meets the usability and security requirements of specific target environments.

Authenticating humans to computers remains a notable weak point in computer security despite decades of effort. Although the security research community has explored dozens of proposals for replacing or strengthening passwords, they appear likely to remain entrenched as the standard mechanism of human- computer authentication on the Internet for years to come. Even in the optimistic scenario of eliminating passwords from most of today's authentication protocols using trusted hardware devices or trusted servers to perform federated authentication, passwords will persist as a means of "last-mile" authentication between humans and these trusted single sign-on deputies.

In assessing usability, an apples-to-apples comparison requires comparing schemes of equivalent security (Figure 10). It is less meaningful to compare the usability of schemes o ffering vastly diff erent security propositions; if done, this should be explicitly acknowledged. For example, in terms of the size of theoretical password spaces, that of many recognition-based systems is comparable to 4-digit PINs, while for recall and cued-recall systems it is more comparable to text passwords of 8-

characters-or-more. Somewhat longer login times may be acceptable for password-level systems than for PIN-level systems, if they provide greater security.

References:

[1] Amazon Mechanical Turk. https://www.mturk.com/mturk/, June 2010.

[2] S.M. Bellovin, "A Technique for Counting Natted Hosts," Proc. ACM SIGCOMM Workshop Internet Measurement, pp. 267-272.

[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] 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.
[5] 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.
[6] 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.
[7] K. Fu, E. Sit, K. Smith, and N. Feamster, "Dos and Don'ts of Client Authentication on the Web," Proc. USENIX Security Symp., pp. 251-268, 2001.
[8] P. Hansteen, "Rickrolled? Get Ready for the Hail Mary Cloud!," http://bsdly.blogspot.com/2009/11/rickrolled-get-ready-for- hail-mary.html, Feb. 2010.
[9] 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.
[10] T. Kohno, A. Broido, and K.C. Claffy, "Remote Physical Device Fingerprinting," Proc. IEEE Symp. Security and Privacy, pp. 211-225
[11] 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.
[12] C. Namprempre and M.N. Dailey, "Mitigating Dictionary Attacks with Text-Graphics Character Captchas," IEICE Trans. Fundamen- tals of Electronics, Comm. and Computer Sciences, vol. E90-A, no. 1, pp. 179-186, 2007.
[13] 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[14] Nat'l Inst. of Standards and Technology (NIST), Hashbelt. http://www.itl.nist.gov/div897/sqg/dads/HTML/hashbelt.html,
[14] "The Biggest Cloud on the Planet Is Owned by... the Crooks," NetworkWorld.com., http://www.networkworld.com/ community/node/58829, Mar. 2010.
[15] J. Nielsen, "Stop Password Masking," http://www.useit.com/alertbox/passwords.html, June 2009.
[16] B. Pinkas and T. Sander, "Securing Passwords against Dictionary Attacks," Proc. ACM Conf. Computer and Comm. Security (CCS '02), pp. 161-170, Nov. 2002.
[17] D. Ramsbrock, R. Berthier, and M. Cukier, "Profiling Attacker Behavior following SSH Compromises," Proc. 37th Ann. IEEE/IFIP Int'l Conf. Dependable Systems and Networks (DSN '07), pp. 119-124, June 2007.

G.Sudheer Reddy is pursuing M.Tech (CSE) in Sphoorthy Engineering College, JNTU Hyderabad. He has completed his B.Tech (CSE). His area of interests includes cloud computing and network security.

G.Venkata Prasad working as an Associate Professor in Sphoorthy Engineering College, Hyderabad. He has completed M.Tech in St.Mary's College,Nalgonda His interests are Data Mining and database systems.

V.Hariprasad working as Head Of The Department (CSE & IT) in Sphoorthy Engineering College, Hyderabad. He is pusuing Ph.D in JNTU Kakinada and has completed his M.Tech(CSE) from JNTU Hyderabad and B.Tech(CSE) from JNTU Anantapur . His interests are Bio Infomatics, Temporal Databases and Mining Techniques.