

A Novel Cue based Picture Word Shape Character Password Creation Scheme

Kevin Curran, Ulster University, Londonderry, UK

Andrew Snodgrass, Ulster University, Londonderry, UK

ABSTRACT

The number of internet users is growing at a rapid rate and this means users now have to remember passwords for many different accounts. The side effects of this increase of user accounts is that users are putting password usability before password security in order to remember these passwords. This paper outlines a novel password creation scheme for creating strong, secure unique passwords that are easier for users to remember on multiple sites. The scheme includes features to more easily create a secure password and recall this password, whilst including multiple layers of security against a targeted attack by an adversary. Results showed that users who did not use a scheme had a much lower login success rate for their accounts than the users who used the created scheme. They also showed that the average password length for each group of users was the same meaning the created scheme passwords in this sample has no greater protection against brute-force attacks, but in terms of dictionary and hybrid attacks the scheme passwords generally seemed to have a lot more protection.

Keywords: Cybersecurity, Hacking, Password Security, Passwords, Security

1. INTRODUCTION

The rapid increase in the use of the internet means many users are new to the area of computing and are unaware of the intentions of hackers and of the risks or consequences of using insecure passwords. Many users may use multiple different password creation schemes and some do not use one at all. This means that after the number of accounts they have grows, they begin to have a mismatch of passwords for each account making password management very difficult. Even if password creation schemes are used, many of the schemes currently available give vague and sometimes contrasting instructions to users making it hard to choose a suitable scheme. These problems lead to the creation of weaker passwords and password reuse in order for users to regain control of their password management. Many users use passwords that are found in the dictionary or that contain easily retrievable information. Password attacks can easily crack passwords based solely from a word in the dictionary and a user's personal information can be

DOI: 10.4018/IJDCF.2015070103

found on many social networking sites online, meaning overall their user accounts are very insecure. Furthermore if users are reusing their passwords, when one user account is compromised the chances of the rest of a user's accounts being compromised is greatly increased. With many accounts containing personal information such as online banking sites, ecommerce sites, and social networks to name a few, password security should not be a compromise. In order to create a password that is more secure it must not be based on a dictionary word or contain personal information that can easily be found, but instead contain a number of words possibly including capital letters, numbers and special characters. This will give the created password the highest chance of standing up to the password attacks that can be used. The problem with this increase of security is that as the created password becomes more complex, the password becomes much harder to remember, therefore a balance is needed between password security and usability.

We propose that if characteristics from what are viewed as the best password creation schemes currently available are compared, evaluated, and combined in a number of different ways then new password creation schemes can be created. The security and usability of these newly devised schemes can be measured using various techniques to find the best scheme. The newly created scheme could allow users to create much more secure passwords that are much easier to remember. This in turn should allow users to use different passwords for every account increasing security even further. We therefore highlight the problem with password management and help solve this problem with the creation of a password creation scheme that allows users to create passwords that are more usable and secure, and easier to manage for multiple accounts.

2. PASSWORDS

A password "is a secret (typically a character string) that a claimant uses to authenticate its identity" (Scarfone and Souppaya, 2009). Passwords are the first line of defence in many information systems and this highlights the importance of password security, but Taneski et al. (2014) concluded that "the computer community has not made a very much-needed shift in password management for more than 35 years". Over thirty years ago, Morris and Thompson (1979) found that the majority of user's passwords were too short, contained only lower case letters or digits, and were easily found in dictionaries. Looking at information released recently it can be seen that little has changed. SplashData (2014), a company that develops password management applications, compiles a list of the most commonly used passwords and posts this annually. In 2013, the top password was "123456" followed closely by "password", "admin", and "monkey", to list a few other examples. These passwords provide little protection as they can be guessed very easily and despite increased advice on password creation with the multitude of user accounts today, users still tend to create weak passwords. With the large increase in internet connected users in recent years, the number of online accounts has grown rapidly. Many of these accounts such as email, social networks, and online banking contain personal information meaning password security should not be a compromise. As well as multiple online accounts, users may have to create and remember passwords for a growing number of devices such as their PCs, tablets, and phones.

In order to try and increase password security a number of different solutions have been implemented on some systems that ironically could be said to make the problem of managing passwords worse, while increasing security but decreasing usability. Password aging is used in order to make the user change their password after a certain amount of time. When a user logs into an account they will be forced to change their password before proceeding. This lack of warning may cause the user to create a password as quickly as possible to avoid the inconvenience, greatly decreasing the usability of the password as the probability of the user remembering the

password during their next login is greatly reduced. The large amount of passwords that users have to create, remember, manage, and sometimes change periodically, is the root cause of weak passwords and this also leads to a number of bad habits that reduce the security of these passwords even further. Some users re-use passwords for multiple accounts. This means if one account is compromised, a hacker could gain more information leading to other accounts being infiltrated. Some users even go as far as writing their passwords down. If this information got into the wrong hands, all the user's accounts could be compromised. These are growing problems as the numbers of user accounts increase and it does not seem to be getting solved. This is why a password creation scheme that is easy to use, creates secure passwords that are easier to remember, and creates a different password for each account, could be a big step in the right direction for account security.

2.1. Password Attacks

There are many different ways that passwords can be compromised and using these methods in order to retrieve a password is known as password cracking. "Password cracking is an attack that tries to guess a user's password by attempting hundreds, thousands, or millions of passwords" (Northrup, 2004). There are a number of password attacks outlined by Oriyano and Gregg (2010) in which the strength of a password can help reduce the chances of a successful attack or increase the length of time until a password is cracked.

Dictionary attacks use a list of words that are pre-defined in order to try and crack the password. If one of the words in the given list matches the password the attack will be successful. The way in which a pre-defined list is used means this attack may not always be successful. To protect against dictionary attacks users should try to use a creation scheme that does not involve choosing a simple common word but instead a combination of words with some character changes. This will mean the probability of a word in the list matching the password will be greatly reduced. If the attack is carried out actively online, there will usually be measures in place to limit the number of login attempts which will thwart the attack.

A brute-force attack will attempt to use all possible character combinations of a given character set, until the correct password is found. This means the attack will always be successful but there are some caveats. This attack will start with shorter passwords then increase password complexity as time goes by. If the chosen password is long and secure this attack could take a very long time, possibly years. As with dictionary attacks, if a brute-force is carried out on an active system there will usually be a limited amount of logins reducing the effectiveness of the attack.

Passwords for a system are generally stored in a database and to protect the confidentiality of these passwords, encryption and password hashing techniques are usually used. "Hash algorithms map binary values of an arbitrary length to small binary values of a fixed length, known as hash values. A hash value is a unique and extremely compact numerical representation of a piece of data" (Northrup, 2004). The method of using these hash values to store passwords is ideal as "you might never need (or want) to see a decrypted version. To authenticate, simply hash what the user types in and compare it to what's stored in the database" (Albahari and Albahari, 2012). This increases security when storing and transmitting passwords and due to the nature of hash algorithms, similar passwords will have no resemblance as a single-bit change in the source data will result in a significantly different hash value. The way in which these passwords are stored in a database of hash values leads to a number of attacks that can take place offline if a hacker can retrieve this database. An example of some passwords after being hashed using the SHA-1 hashing algorithm can be seen in Figure 1. It can be seen that all passwords no matter

what length return a hash value of the same size, and similar passwords show no resemblance in the hash value.

As before, dictionary attacks and brute-force attacks can be used offline against the hashed passwords. The dictionary attack will be very similar involving the comparison of hashed values from the list with the hashed values in the password database until a match is found or all possible values in the list have been tried. Similarly a brute-force attack will try all possible character combinations only this time hashing the password before comparing them. Another offline attack is known as a hybrid attack. These are similar to dictionary attacks but have a higher level of sophistication. The attack consists of two phases. First different combinations of words from the dictionary are tried. If this is unsuccessful the second phase of the attack will add characters and symbols, increasing password complexity in order to increase the chances of cracking the password.

Finally another offline attack known as a rainbow table attack computes the hashes of every possible character combination before carrying out the attack. A hacker can then compare a password hash with all the pre-computed hashes to find a match. The disadvantage of this method is again the amount of time required to pre-compute a large number of hashes for all possible passwords. Also if long passwords are used this time increases again as the longer that maximum password length is, the longer it will take to compute the hashes. As these attacks can be carried out offline there will be no limited logins or other methods to help protect the passwords but there is a technique that can be used to further increase the security of the retrieved hash values called salting. Before passwords are hashed, extra characters are added, changing the hash value but not the password. A hacker who retrieves the list of hashed passwords will have a much more difficult time cracking a password as they will then have to determine the password by reversing the hash or determining the text added to generate the password.

There are also a number of nontechnical methods a hacker can use to crack passwords and sometimes these “nontechnical methods can be as effective as technical methods at obtaining passwords” (Oriyano and Gregg, 2010). A hacker may retrieve passwords through the use of personal information of a targeted user. Silberschatz et al. (2008) notes that “all too frequently, people use obvious information (such as the names of their cats or spouses) as their passwords”. With the available information posted on social networks and other services online, and users basing their passwords on this information, a hacker can greatly narrow down possible passwords for certain accounts. Shoulder surfing is another term describing an attacker who obtains a password by observing the user entering their passwords. Generally to inhibit these attacks users should not base passwords solely on easily retrievable personal information and create a password that is complex enough that it is not as easily observed when being typed in. As can be seen, there are many different techniques attackers can use to compromise passwords and everyone has to

Figure 1. Hashing examples

password	5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8
password1	e38ad214943daad1d64c102faec29de4afe9da3d
test123	7288edd0fc3ffcbce93a0cf06e3568e28521687bc
longerpassword123?	b447dc6c61b93cceb266b2eab3a664ec283ad40a

be considered when devising a password creation scheme in order for the created passwords to stand-up as much as possible against these attacks.

2.2. Measuring Password Security and Usability

Many studies advocate using password entropy as an indicator for password quality (Wanli et al. 2010). Entropy is simply the lack of order and predictability of characters used in a password. Although entropy can be used as a measure of password quality, “it can provide no more than a very rough approximation of overall password strength” (Kelley et al. 2012). Since it is difficult to judge how secure one password is over another by looking at the characters used, Wanli et al. (2010) found that a better measure for the security of a password should also be “decided by the time required to crack the password”. Password usability can be measured by how easily a user can remember a password. A password creation scheme’s usability can be measured with a number of metrics such as the time taken and number of attempts taken to input the correct password for an account, rate of success for account logins and the amount of users that store their passwords by writing them down (Ur and Segreti, 2014).

The main problem with password security and usability is that there is a balancing act between the two. As noted by Zviran and Haga (1993), “the trade-off between memorability and safety poses a dilemma in the generation of passwords”. In order to make passwords stronger it is recommended and sometimes even required by some password policies to have both lower and upper case characters, including numbers and special characters. This is where the trade-off lies. As users add these characters the security of the password may increase but the usability will drop as the user will most likely forget the password or require multiple attempts when logging in next time. To counteract this trade-off some password creation techniques include the use of pictures, objects, and stories that users can recall when logging into an account. Some of these methods are described in the section below detailing some of the many password creation schemes that are available.

2.3. Password Managers

To help with the problems associated with password memorability and usability, a number of solutions are available for users. One increasingly popular solution is the use of a password manager. “Password managers work by saving users’ online passwords and later auto-filling the login forms on behalf of users” (Zhao et al. 2013). This solution provides a number of advantages in the area of password usability but it also has some disadvantages. The password manager will remember a users’ username and password for their chosen accounts and will fill these in automatically upon login, meaning users do not have to remember all of these passwords. This means longer, more complex, and more secure passwords can be chosen, which are different for each account. Some password managers even offer the functionality to generate a secure password for the user. This can greatly increase the security of user accounts. The downside is that the password manager is usually cloud based to allow users to login on many different devices and the account is usually protected with a master password. If this master password is compromised a hacker will have access to every account which is stored within the password manager. In conclusion it can be seen that tools like this greatly increase password usability and also allow unique, secure passwords for different accounts if features are utilised, but password security is still a very important factor. Every account stored within the password manager is still susceptible to the password attacks described previously including the password manager account itself, so the problems involved with creating a secure and usable password for these accounts still need to be addressed.




2.4. Password Creation Schemes

A password creation scheme is the process a user goes through to create a new password for an account. There are many different ways that users can create passwords and they all have pros and cons, and have a specific balance between usability and security. Wildenhain et al. (2012) lists a number of methods a user would find if researching about how to create their passwords. Within the list of creation schemes the National Institute of Standards and Technology (Scarfone and Souppaya, 2009) provides a number of different schemes in a guide to enterprise password management. Many of these schemes include the use of a base password with a certain modification. In order to create the base password a number of methods can be used. The mnemonic method were “a user selects a phrase and extracts a letter of each word in the phrase”, the altered passphrase method were “a user selects a phrase and alters it to form a derivation of that phrase”, and the combining words method were “a user can combine two or three unrelated words”. The created base password can then be modified with the addition of random characters or with the substitution of special characters. A problem with passwords created in this way is trying to remember the words and modifications chosen for multiple accounts in order to have a different password for each one. This leads to the creation of passwords using pictures, objects, and stories. Other schemes listed involve the use of pictures and objects to create the passwords. The user can select an image and use a selection of random words to describe a story taking place within the image. The user can then use this image as a cue when logging in helping them create a strong password but also making it a lot easier to remember the password. An example of a Person Action Object cue can be seen in Figure 2 below. Users using the Person Action Object scheme will use a number of these to create a memorable story then use this story as a cue, possibly making it easier to recall their password.

Wildenhain et al. (2012) notes that users are presented with a lot of different schemes, many with vague instructions, meaning users can have a difficult time gauging the security offered by each one and can have unfounded confidence in the security provided with their chosen scheme if using it incorrectly. Another problem faced when using a certain password creation scheme is the mismatch of password policies implemented on many different systems. Farrell (2008) states that “we must often adhere to different and possibly mutually incompatible password policies”. Some systems may have a minimum and maximum password length, and may have a limited set of special characters or none at all, meaning the password created using the creation scheme may not be accepted. A small workaround may allow the password to be accepted but this will reduce usability as during the next login the user will most likely not remember the change they had to make.

It can be seen that there are a number of different factors that need to be taken into account when creating passwords and finding the best balance between security and usability is key. The created passwords have to have a high level of entropy possibly including a number of words with special characters in order to hold up as much as possible against the passwords attacks listed. The downside to this is that as the password gets more complex the usability greatly decreases. This project proposes that if the best password creation schemes that are currently available are evaluated, the characteristics that help increase security and usability can be found. These characteristics can then be used to devise a new password creation scheme that has a higher level of security and usability allowing for better account security and easier password management overall. With an increasing number of passwords to remember, users are taking risks by creating and reusing weak passwords that may be easy to remember but have little or no protection against password attacks. It can be seen that some of the password creation schemes that are currently available give vague and sometimes contrasting instructions to the user making them hard to

Figure 2. Person action object example (Blocki, 2012)

Cue	Action	Object
person	kissing	piranha
		

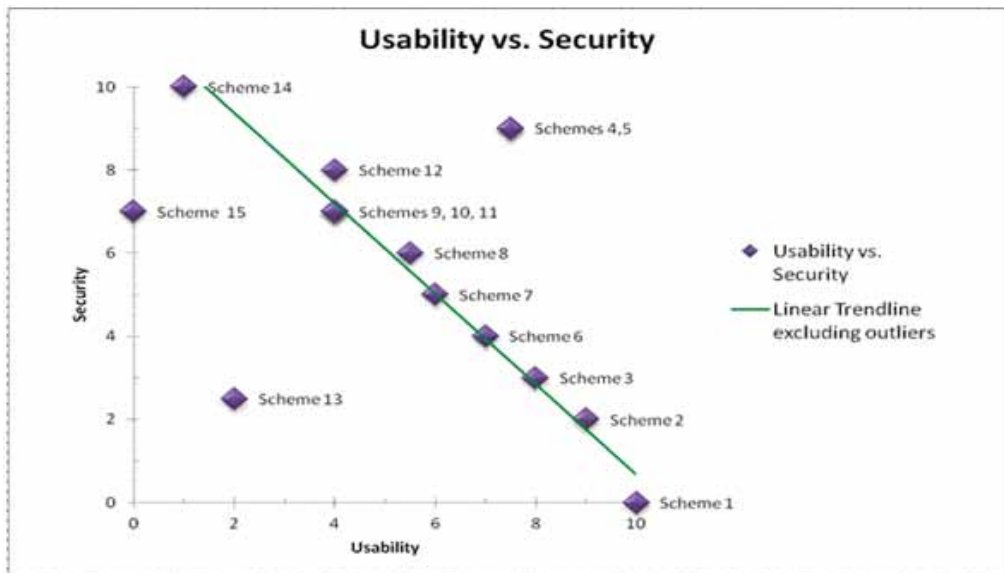
follow and hard for users to choose the best scheme. This paper outlines a password creation scheme that provides clear instructions for the user, helps create secure, unique passwords for each user account, and helps make these passwords memorable with the use of certain cues. If this can be achieved, the security of user's accounts who use the scheme would be greatly increased and the frustration of resetting password that have been forgotten would be reduced

3. PASSWORD CREATION SCHEME DESIGN

Current password creation scheme characteristics were evaluated in order to determine which characteristics will benefit the newly created scheme. The main aspects that needed to be focussed on were the security of the generated passwords and the usability of the scheme. This is the inherent difficulty with password creation. As the password becomes more complex the security of the password is greatly increased although unfortunately this in turn greatly reduces the usability as users can no longer remember their chosen password. To counteract this decrease in usability the newly created scheme aimed to include features to allow the password to be adequately complex but also help the user recall the complex password more easily. Wildenhain et al. (2012) compiled a list of the most popular password schemes and assigned a value using "gut feeling" for the security and usability of each scheme. These values can be seen in Figure 3. This graph shows a graphical representation of a number of the most popular password creation schemes and their security and usability score out of ten.

The schemes with the highest level of security and usability (scheme 4/5) were the Base Person Action Object scheme, and the Base Picture Scheme. The Base Person Action Object scheme involves the user selecting three people that they know and storing their names. The user then selects a random action and a random object and creates a story involving the names, action and object selected for each account. When the user logs into an account, they will look up the names stored for that account, remember the action and object used, and subsequently use this to recall their password in order to login to the site. The Base Picture scheme involves

Figure 3. Password schemes usability vs. security (Wildenhain et al., 2012)



the user selecting and storing a picture that is interesting. The user will then select four words from sixteen randomly selected dictionary words. These words will then be used to create a story within the picture. When the user logs in, they will retrieve the picture for their account, remember the corresponding words, and recall their password.

These schemes have some advantageous traits over some of the other schemes listed in order to provide these levels of security and usability. First of all, users are provided with a cue for each account password. A person or a picture is used to help them recall a story, subsequently leading them to recreating their chosen password. The login can be rehearsed over time, meaning if a user logs into an account frequently, eventually they may no longer need the cue. The use of pictures and random actions or objects provides a lot of different references to use in order to create the password. This allows the user to create a longer more random password increasing the effectiveness against brute force and dictionary attacks. Finally if an adversary retrieves the users pictures, or list of people, or even knows the scheme used, the password will still not be compromised. Although these schemes can be used to create stronger more memorable passwords they have quite a few disadvantages. The users have to store their password cues and be able to view these cues when they need access to an account. This can mean a user may be unable to login to their account temporarily if they do not have access to their cues or may even have to reset their account password if they lose their cues. The creation process for these schemes is quite inconvenient. Users have to look for a picture and use computer program or website to select random dictionary words, or think which people to use for each account before creating their password. Eventually the user may go back to their old habits due to these inconveniences. In terms of the Person Action Object scheme, a user will most likely run out of people to use as the number of accounts increases. This may mean they will begin to re-use people's names which may start to cause interference between different accounts, affecting their ability to recall the correct password. Finally these schemes provide no incentive to add any special characters meaning the level of entropy is not as high as it could be.

From studying the characteristics of the advantages and disadvantages of what are thought to be the best schemes, a clear list of traits can be compiled. If a password creation scheme is produced including these traits, theoretically it should provide a higher level of usability and security. The created scheme should try and incorporate the following:

- A cue to help the user create and subsequently recall their password, while allowing the user to remember as little as possible.
- A cue that if seen by an adversary, still does not compromise the user's password.
- A scheme that helps users create passwords containing references from multiple words.
- A scheme that helps users incorporate a number of special characters to further increase entropy.
- A scheme that is convenient, meaning everything the user needs to create the password is provided, allowing the user to create their password with minimal effort.
- A scheme that provides a cue that is very dynamic with a "limitless" cue, so as the user has more accounts, cues never overlap or cause interference.
- A scheme with easy to understand instructions so users will not feel overwhelmed, while making it complex enough to create strong passwords.

Implementing all of these traits into a password scheme provides a number of challenges. Providing a cue while allowing the user to remember as little as possible, but keeping the password secure if an adversary sees the cue is quite difficult. A scheme based on a cue makes it very difficult to help the user integrate any special characters while still allowing them to recall these characters in the future. Providing the user with a scheme that is quick, easy to use and convenient while maintaining password strength requires a very different solution. Finally, again it is about keeping a balance between usability of the scheme and the strength of the created passwords.

The first main area of focus was the implementation of a cue for users. The main aim was for the cue to be able to allow the user to include a mixture of multiple words, numbers, and special characters within their password. A single cue e.g. a person or a picture, would not allow for all of these variables. It was decided that the user would be provided with multiple cues in order to create their password. The cue would consist of multiple images. One image would contain a picture, one image would contain a random dictionary word, another would contain a shape of various different colours, and another would contain a number of special characters. The user could then create a "mini" scheme for each, remembering what they do for the picture, word, shape, and characters in order to create their password. When the user goes to login to their account again, they will remember in what order they use the images and what they do for each, recreating their password. This would allow a password consisting of multiple words and special characters, increasing the entropy of the password meaning increasing the strength, while also maintaining usability as the user only has to remember their small schemes. An example of the scheme layout and examples of each image category and possible passwords are shown in Figures 5 and 6.





This figure shows an example from each image category used in the scheme. A number of possible password fragments are shown for each and finally a number of example passwords are shown, just some of the endless combinations a user could create.

This cue solution added multiple layers to security. An adversary would have to look at the four images provided then try to work out what the user did for each and in which order they used the images. This means it would be very difficult to crack the password but not impossible if the adversary puts in the effort over a long period of time in a targeted attack. The next step was to increase the level of security against an adversary further. Instead of showing the user their four

Figure 4. Created scheme layout

Picture	Word	Shape	Characters
---------	------	-------	------------

Figure 5. Example passwords created with the scheme

Image Category	Possible Passwords
Picture 	light traffic stop right go
Word 	detacitsihpos mature SoPhIsTiCaTeD
Shape 	rstary sixredyel startenry
Characters 	To be included in chosen area of the password or throughout the password
Possible Combinations	<*light*<maturerstary goSoPhIsTiCaTeDrstart<*< <*rightmaturestartenry*< <*<trafficmaturesixredyel

images for their account, it was decided that the user would see ten images. These ten images would be indexed from zero to nine. When the user first uses the scheme they will create a four digit key. This key will be kept private and will remain the same for all of the user’s accounts. The user will use this four digit key in order to view the four images they need to create their password. This increased the level of the cue security greatly while barely reducing usability as the user only has to remember an extra four digits. An adversary will now see ten images

Figure 6. Password creation scheme instructions

- If this is your first time using this scheme, create a four digit key (digits 0 – 9). You can choose any combination, but combinations containing all of the same digits (e.g. 8888), your year of birth (e.g. 1992), or an easy to remember combination (e.g. 1234), are not recommended. Make sure you keep this key private as it will be the same for all of your accounts.
- When creating the password for your account you will be presented with ten images numbered from zero to nine. The ten images presented will consist of four different categories. Each image will either be a picture, a random dictionary word, a shape containing a number of colours, or some special characters.
- You will use your four digit key to locate the images that you will use to create your overall password. You can do as you wish with each image but it is recommended you keep a consistent scheme for each image category e.g. for a picture you may wish to choose something it reminds you of, or the first thing that comes to mind when you view it. For a shape you may use the number of sides or vertices, along with the colour etc. Try not to be too distinct. The image is only a cue. Do not make it too easy for someone else to guess.
- When you log back into your account you will be presented with the same ten images. You will use your four digit key to locate the images you used, and you will use these images to recall your password.

consisting of pictures, words, shapes, and characters. Each digit in the user's key does not have to be unique (e.g. 1122 would be acceptable), although a key containing all of the same digits is not recommended due to a decrease in security of the final password. This means there would be ten thousand combinations of images the adversary would have to try to get the correct sequence of images, and even then, without knowing what the user did with certain images, they would need to try many more password combinations using these images, meaning the final result is a cue that is very secure even if the user's key was compromised. It should be noted that the images chosen from the list using the user's key may not necessarily contain an image from all four categories. As well as the security of the user's cue, the final password has the possibility of including multiple words, numbers, and special characters, while still being easy for the user to recall. This should provide a high level of protection against dictionary and hybrid attacks. Furthermore with the scheme consisting of four images the password has the possibility of being quite long, decreasing the effectiveness of brute force attacks.

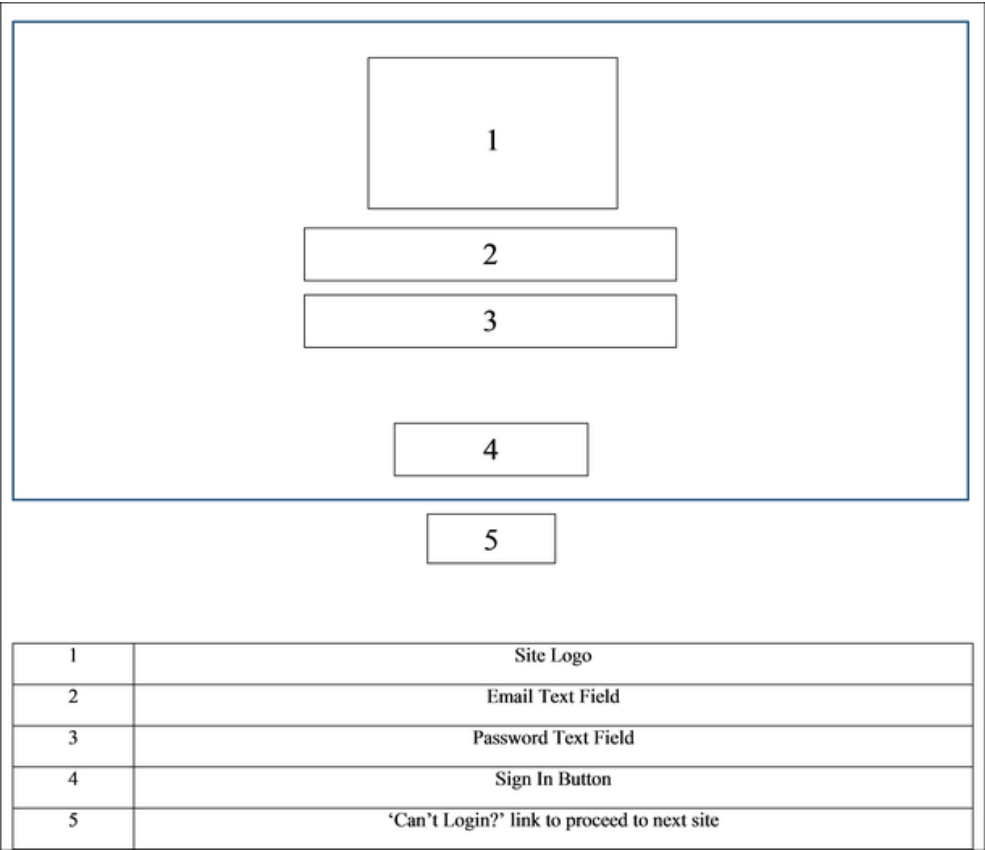
Another disadvantage noted with previous schemes was lack of proper instructions or instructions that were difficult to understand. If the user is not provided with enough information they may use the scheme incorrectly, but given too much information they may choose to avoid the scheme as they may see it as too inconvenient or too difficult to implement. The given instructions aim to provide clear and concise guidance for the users, so they can find it easy to understand and be confident enough to start using the scheme straight away. The instructions for the newly created scheme are shown in Figure 7 which shows the instructions a user would see when first using the scheme.

The website design for non-scheme users is shown in Figure 8

The website design for scheme users is shown in Figure 9.

The sites were designed to have a minimal look and feel, and to include only what the user needed, to reduce any possible variables affecting the results. When creating their accounts, the user will input their designated email address and their chosen password. Some validation will occur checking that the username and password have been supplied and making sure the user does not have an account for the site already. If the validation succeeds the user will be presented with the next site and so on until all twenty accounts have been created. When logging

Figure 7. Example website design for non-scheme users



in a similar series of steps will occur. The user will enter their email address and their password. This time validation will check that it has been at least one week since the account was created and make sure the input password matches the stored password. If the password is correct the user will proceed to the next site, while if it is incorrect the user will be shown an alert stating that the password was incorrect. The user has as many attempts as they wish to gain access to their account but if they cannot recall their password they can click the 'Can't Login?' link at the bottom of the page to proceed to the next site.

When a user creates a password online they usually have no guidance. They are sometimes given a certain description of what a secure password should consist of and sometimes some basic rules are enforced. The site may contain a password strength meter which fills up depending on if certain rules are met and the user may be unable to submit their password until it is deemed acceptable, but in most cases the user can choose as they wish. In order to test the effectiveness of the newly created scheme it was decided that the scheme would be tested against this current method. Ten people were selected to take part. The ten users were required to create a password for twenty popular sites/accounts and login to these accounts one week after they were created. Five were given the current method of password creation and the other five were given the new

Figure 8. Example website design for scheme users

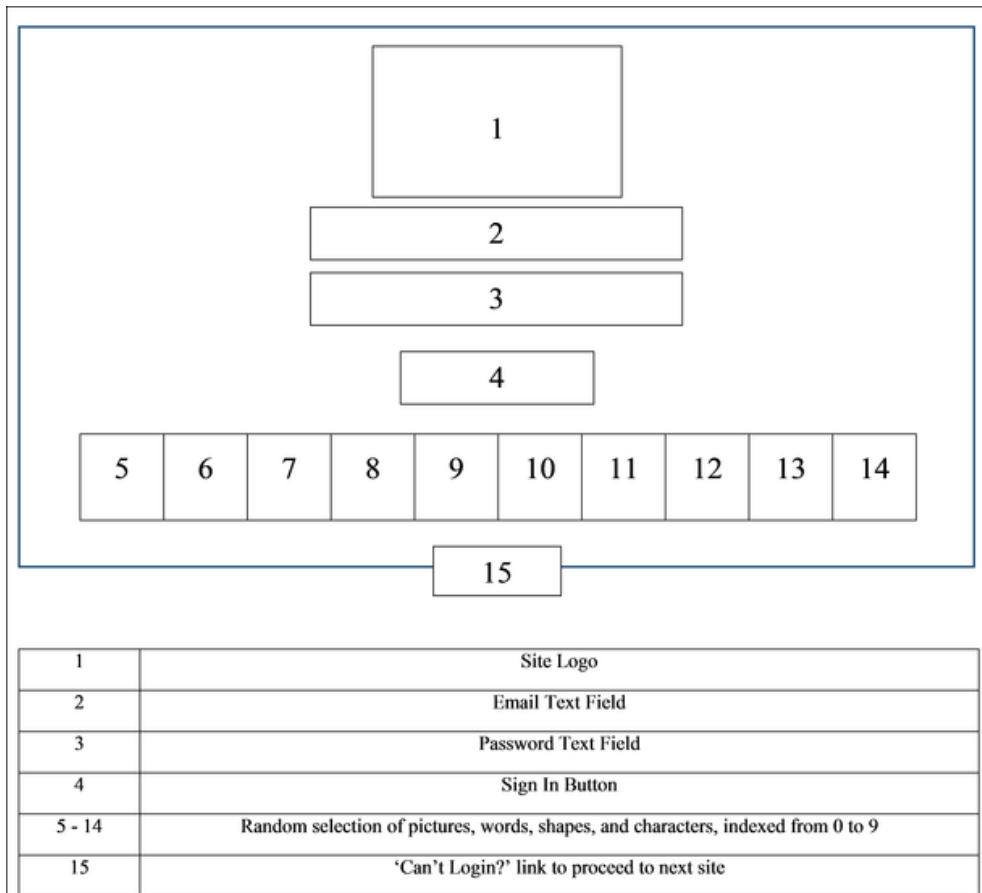


Figure 9. Non-scheme site sample



creation scheme. The users without a scheme would create passwords that they thought were secure as possible while maintaining memorability as much as possible, and the users with the scheme would create their passwords using the instructions provided.

First of all some research was carried out to compile a list of the twenty most popular or well-known websites/accounts. This would mean most of the sites would be familiar to the users as these sites would be implemented for the user testing. The sites chosen are shown in Table 5 below.

A total of forty sites were created using HTML and JavaScript. Twenty sites would be used as non-scheme sites providing no guidance for the user and twenty sites would implement the password creation scheme to assist the user. Each site consisted of the site logo, an email text field for the user's assigned email, a password text field for the user's created password, and a sign-in button to submit the information, with the addition of the scheme images for the twenty scheme sites. A comparison between a scheme and non-scheme site can be seen in Figures 10 and 11.

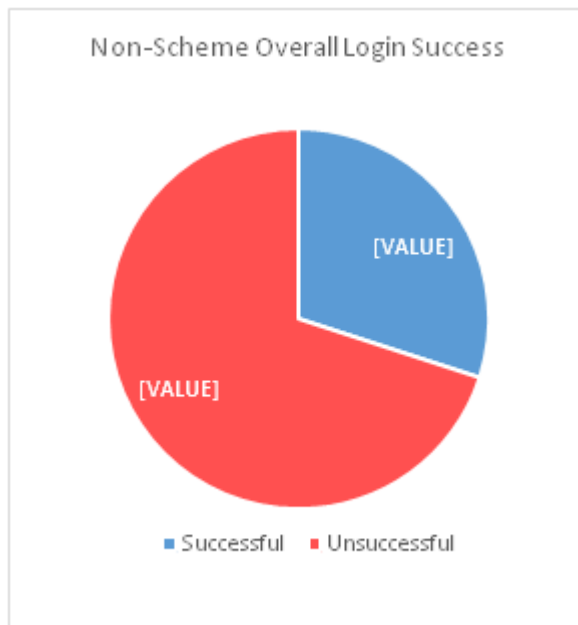
A number of PHP scripts were created to handle the input from the users. While creating accounts the scripts made sure the user supplied their email address and password and that the user had not already created an account for the site already. If the input was valid the scripts then created the appropriate records in a database. When logging into accounts the scripts made sure it was at least one week since the account had been created and checked to see if the input password matched the stored password. These PHP scripts interfaced with a MySQL database to store all the user data. All users were free to choose whatever passwords they thought were suitable. They had no limitations and no rules were enforced. This meant the tests would reveal a good measure of how the users adapted to the scheme and how secure the passwords they generated were without "forcing" the passwords to be secure. The five non-scheme users were first told that they had to create a password for twenty accounts and they would have to login to these accounts one week later. They were advised to create a secure password for each account and told that a secure password should usually consist of no less than eight characters, a number of words, and special characters. They were also told that there were no strict rules enforced and the password should be secure as possible while still enabling them to gain access to their accounts. The scheme users were given the same information although they were also given the scheme instructions.

Table 1. Selected sites/accounts for user testing

Selected Sites/Accounts	
Amazon	Dropbox
eBay	Facebook
Google	Gumtree
iCloud	Imgur
Instagram	LinkedIn
Netflix	Outlook
PayPal	Plex
Skype	Spotify
Tumblr	Twitch
Twitter	Wikipedia

Figure 10. Scheme site sample

Figure 11. Overall non-scheme login success rate



The users would input their designated email address and their chosen password then click 'Submit'. As the users created their accounts, their passwords were stored for each site, as well as their designated email address, the scheme used, the password length, the date and time the account was created, and a generated SHA256 password hash value. These values would later be used to validate the user logins and gather final statistics. When the users were finished creating their passwords for all accounts, they were asked to verify them. One week later all ten users were asked to log back into their accounts. This time the database would store the number of login attempts for each site and if the site login was successful or not. If the user had trouble logging into a site, after a few attempts they could proceed to the next site. This would be stored as a failed login. The data stored would eventually allow for a number of statistics to be calculated.

The password length would provide an average password length comparison between the scheme and non-scheme accounts. The number of login successes/failures would allow for a comparison of how successful each scheme was in the area of usability and the generated hash files would be used to run a number of attacks including a dictionary and hybrid attack. This would allow a comparison of security between passwords created with or without the scheme.

4. EVALUATION

A total of ten users took part during the testing process. Five users used no scheme and the other five users used the newly created scheme. Seven males and three females, with an age range of fourteen to forty-seven were distributed between the scheme and non-scheme groups as evenly as possible. Each group included users that had a range of technical abilities but all users had a fair understanding of accounts and passwords from previous experience. A total of five users created twenty accounts each without guidance from the new scheme. Figure 13 illustrates the overall login success rate for these users. Overall there was a total of one hundred accounts created by these users. This shows that only thirty percent of the one hundred non-scheme logins were successful compared to seventy percent that were unsuccessful. This means that many of the passwords created without the scheme could not be recalled by the users meaning they could not access their accounts. Again, five users created twenty accounts each, this time using the newly created scheme. Each account displayed ten images containing pictures, words, shapes, and characters. The users created a four digit key and used this key for each of their accounts to create their passwords.

Figure 14 illustrates the overall login success rate for the users who used the new scheme. There was a total of one hundred accounts created. It shows that sixty-seven percent of logins

Figure 12. Overall scheme login success rate

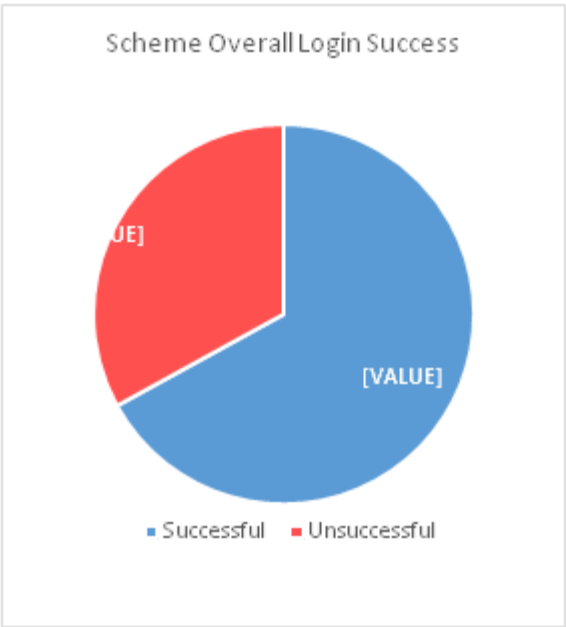
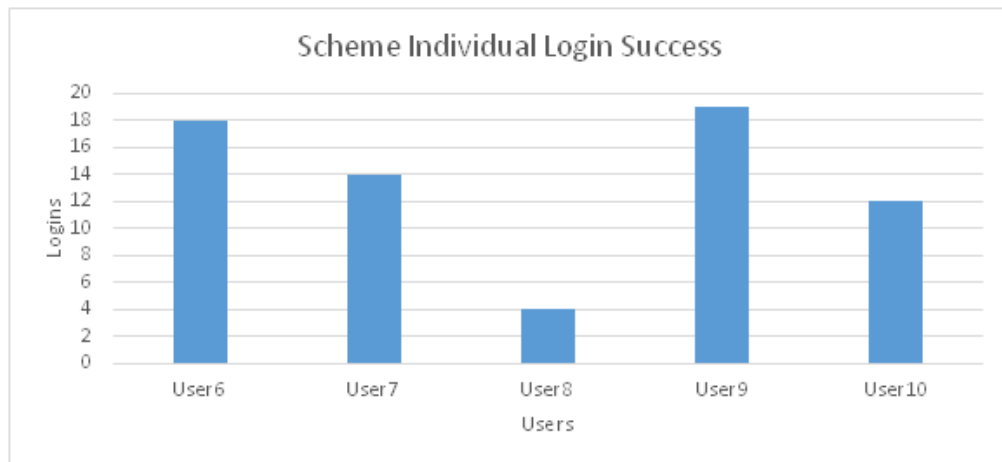


Figure 13. Individual non-scheme login success rate

were successful compared to thirty-three percent that were unsuccessful, quite a difference compared to the non-scheme results. This shows that the users were able to recall many more passwords and access their accounts when the scheme was implemented into the sites. Looking at the individual logins for each user provides more insight into the results. These results are illustrated in the charts (Figures 15 and 16).

Figure 15 shows the number of successful logins for each individual non-scheme user. Each user created twenty accounts. The chart shows that user 1 and user 4 had sixteen and fourteen successful logins respectively, while user 2, user 3, and user 4 failed recall their correct passwords to login to any of their accounts.

This chart illustrates the large variance that occurred between different users who created passwords without help from a creation scheme. Some users were able to recall a significant amount of their passwords while others could not recall any. This could be due to a number of reasons. The users who successfully logged in may have used a more consistent password creation method throughout helping them to remember more passwords but the passwords that were created may not have been very secure.

Figure 16 shows the number of successful logins for each user who used the new scheme. Again, each user created twenty passwords. It shows user 6, user 7, user 9, and user 10 successfully logged into eighteen, fourteen, nineteen, and twelve accounts respectively, with the worst performing user being user 8 having only 4 successful logins. This shows that the number of successful logins per user is much higher for users that used the scheme compared to the users who did not, although there are still a number of users who did not fare as well as others.

4.1. Usability

When looking at the overall login success rate for both login methods (Figures 13 and 14 above) a clear difference can be seen between both, with a 66% success rate using the scheme compared to just 30% success rate without it. Looking at Figure 15, a number of users not using the scheme actually fared quite well when logging back into their accounts while user 2, user 3, and user 5 could not access any of their accounts. Table 6 below shows a sample of passwords from a non-scheme user who successfully logged in to a number of their accounts and from a non-scheme

Figure 14. Individual scheme login success rate

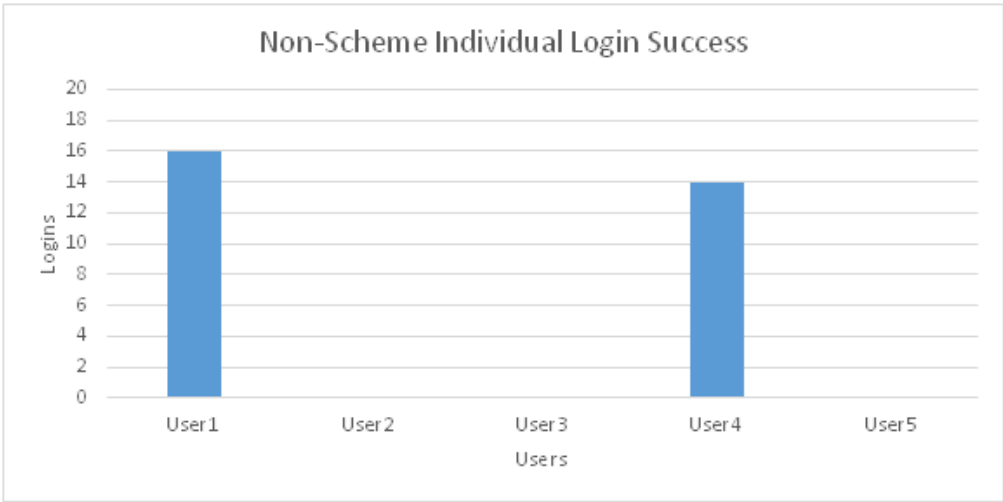


Figure 15. oclHashcat Dictionary Attack Status

```
Session.Name...: cudaHashcat
Status.....: Exhausted
Input.Mode.....: File <../../../../realuniq.lst>
Hash.Target....: File <../../../../fyphash.txt>
Hash.Type.....: SHA256
Time.Started...: Sun Apr 19 13:32:23 2015 <2 mins, 27 secs>
Time.Estimated.: 0 secs
Speed.GPU.#1...: 1050.7 kH/s
Speed.GPU.#2...: 1113.5 kH/s
Speed.GPU.#*...: 2164.2 kH/s
Recovered.....: 6/200 <3.00%> Digests, 0/1 <0.00%> Salts
Progress.....: 1196843344/1196843344 <100.00%>
Skipped.....: 0/1196843344 <0.00%>
Rejected.....: 15136951/1196843344 <1.26%>
HWMon.GPU.#1...: 0% Util, 26c Temp, N/A Fan
HWMon.GPU.#2...: 0% Util, 27c Temp, N/A Fan

Started: Sun Apr 19 13:32:23 2015
Stopped: Sun Apr 19 13:34:51 2015
```

user who could not access any of their accounts. This table shows a comparison of a sample of passwords between two non-scheme users.

From looking at the sample of passwords from both users it can be seen that they both based their passwords on the sites logo or on the service the site provided although there are differences. User 1 kept a consistent scheme throughout, were as user 5’s passwords contained more symbols and were less consistent. This most likely explains the increase in login success rate for some of the users. User 1’s consistent scheme allowed them to recall many of their passwords but at the same time it could be said that the passwords created may not hold up as well against a hybrid attack. User 5’s approach created passwords with a slightly higher level of entropy but

this came at a consequence of the user being unable to recall their passwords a week later. This clearly shows the importance of a well implemented scheme to support users in making their passwords more secure while maintaining usability.

The scheme results show a much higher success rate, and looking at a sample of passwords, they seem to contain a much higher level of entropy. Even so, there are some users who did not fare too well compared to others. Table 7 below shows a sample of passwords from two scheme users. This table shows a comparison of a sample of passwords between two scheme users.

From these scheme samples it is harder to tell why one user did not do as well as another but again, it seems to be down to consistency. User 8 seems to have included only a number of the special characters provided in some accounts, and in others there seems to be no characters implemented at all. Again user 8 also seems to alternate between and add capital letters in certain parts of their password. User 9 seems to have a much more consistent scheme for each image category leading to a much higher success rate. This may be the reason to explain the reduced login success rate from user 8.

4.2. Security

In order to test the security of the created passwords it was decided that a number of factors would be looked at. First of all the average password length for each scheme would be calculated. This would give an idea of how well the creation scheme passwords would hold up against a brute-force attack compared to the non-scheme passwords. Secondly a dictionary attack and a number of hybrid attacks would be used against the collated list of password hashes. This would show how well the scheme and non-scheme passwords would protect against these attacks. To protect against a brute-force attack, the longer the password is, the better. The average non-scheme password was 12.79 characters, and the average scheme password turned out to be 12.65 characters. Looking at the individual average password length for each user in Tables 8 and 9 below, it can be seen in some cases the scheme password was slightly longer, but overall in this sample there was no more or no less protection against this type of attack. Tables 8 and 9 show the average password length for each of the five individual test subjects for each creation method.

To protect against dictionary and hybrid attacks, it is best to have passwords containing references to parts of multiple words including special characters and capital letters, basically

Table 2. Non-scheme sample passwords

User 1 Samples (16/20 accounts accessed)	User 5 Samples (0/20 accounts accessed)
friends&family?	facebk@123
clothesshopping?	chrome1501
chewinggum?	treegum@15
gloomyweather?	cloudblue123
imagirl?	rugmi015
rainbowcam?	instagm@201
linkedinblue?	lin@in21
stardustfilm?	flixnet5678?
discletter?	lookout4?
peterpiper?	yap15@pal?

anything that is not found in a dictionary. In order to test the password security strength against these attacks a dictionary attack was used, followed by a number of hybrid attacks. CrackStation (2015), is an online password hash cracker, allowing users to submit password hashes to be cracked. The site also provides a wordlist that contains “every wordlist, dictionary, and password database leak” that could be found on the internet after a lot of time searching, and also every word stored in the Wikipedia databases retrieved in 2010. Overall the wordlist is 15GB in size and contains 1,493,667,782 words. oclHashcat is a GPGPU-based (General-purpose computing on graphics processing units) multi-hash cracker providing many different attack modes that can be used against multiple password hashes at a time. It utilises the power of a systems graphics processing unit (GPU) which is much more efficient for this type of workload. This tool along with the wordlist was used to carry out the attack on the collected password hashes.

The attacks were run on a high-end gaming system that contained two GPUs. With this consumer grade hardware, oclHashcat was able to compare hashes at speeds of up to 1500 MH/s while working with SHA256 hash values during dictionary attacks. First of all a simple dictionary attack was carried out using the supplied list and the two hundred password hashes. The test completed in less than two and a half minutes, unveiling six of the two hundred passwords. Figure below shows the final status of the dictionary attack, illustrating how many hashes can be compared in such a short amount of time.

Next a number of hybrid attacks were used. This uses the list in a similar way to a dictionary attack but also implements brute-force style behaviour by adding different characters to the dictionary words. The attacks carried out were ones which completed in a reasonable time frame. After around twelve hours of running various attacks a total of forty passwords were retrieved. The passwords that were retrieved are listed in Tables 10 and 11.

These tables show a list of some passwords that were cracked using a number of password attacks for each scheme.

Although salting will most likely be used to further protect these stored hashes, this is still a useful test to see how the created passwords cope against this attack. From the list it can be seen that from the forty passwords cracked many of them were from the non-scheme users. Clearly many of these cracked passwords contain dictionary words, names, date of births, and very low levels of entropy. These results demonstrate that the created scheme can allow users to create passwords with higher levels of entropy while maintaining usability under certain circumstances.

Table 3. Scheme sample passwords

User 8 Samples (4/20 accounts accessed)	User 9 Samples (19/20 accounts accessed)
redc\$number	@@<!kn?!!ab
franchscurveface	:?:?nu!!**em
Carsshapeas	gifl**!!li
Whiteshapesuns	bewa))(((pa
redsulta	fl!@@@nug!
starAsworld	**!!@@@!stta
Benonenumbersa	*?!<))((cima
Wdis4shape	su<<<>>>hewh
Sasholidays	(:?!clstre
sSHAPESHAPEHH	hecobonu

Table 4. Individual non-scheme user average password lengths

Non-Scheme	Average Password Length
User 1	11.95
User 2	13.75
User 3	12.45
User 4	14.05
User 5	11.75

Table 5. Individual scheme user average password lengths

Scheme	Average Password Length
User 6	15.00
User 7	15.20
User 8	12.50
User 9	10.45
User 10	10.10

Table 6. List of cracked passwords from non-scheme users

Non-Scheme
1c5a17e1183d17f61f48e6e8c526d3a5ba0200a6f3ac6a757802933fe299f631:christopher1993
2be7b8ce086c86fb257405abe15be7ccbe2d7d2c957c7f1300960cb215e986b2:christopher20
00ea20cfea2457e2473a33bd5090515c98494ea7bd5f3ec43c0420afbcf3da13:facetoface
f5194275eb531425edb5d97ab16903ecf865b4025e691e57e6c1217f4ba15fe8:intheblue
a0b5b16ae8b966950ea49b137d54985d5eb991ba5e6970bd8ab5d1e88e5f22e4:password1993
65e84be33532fb784c48129675f9eff3a682b27168c0ea744b2cf58ee02337c5:qwerty
a7b459edbfb4db53748bf9f986a05e73b8585afc201a3a36f8872d163459ac2:password2015

Table 7. List of cracked passwords from scheme users

Scheme
00e5a14ebbf0ebb9b82c595f658ae43d5570c9cbd15f11e5c72310ecf03054a2:@rc422
6fc5d47103f1815b919ba49aba8264165e99d62d5cf63316605dc0a9102c5237:procbohe
033379f1788cc34d759d6e788850835b74957f25b175544a49beacc649570ca4:m@arrowred
32dbe091f921c8e80021dcea098464b2c8e3bc0717e9eaaac9b5c8ae8f227555:glcohemax
f63ffd477969c03cfe8df5dccc4029b1368c01f864293cae2868bb84f15dae870:gaeinsteinh
66354b7172f9ecca8c34b596156cd6686b0f2d9dc9bf61fbb11f82d5fc7d510f:4wihex
bffb0993d687b984baebada1669e789ab24534a19a4a04a7a2b854ca939d48d:Sasholidays
cd1636574c4cf20e83ea06737366fb69ca840b5072cade013a19e53531308706:flsonust
6a65544be6872b358a3f06e45f32c2b029633716d842698784e7de308c28bee5:*!starai

Although the scheme passwords stood up well against the attacks, some were compromised. This could point to areas with the scheme which could possibly be improved.

There are a number of variables which could have affected the final results involving the testing methodology and the users involved. In order to test the schemes effectiveness, users were required to create twenty passwords. This allowed the schemes to be tested sufficiently but this is also quite an unrealistic situation in the real world. Although users would usually have many more than twenty sites, it is unrealistic to expect them to create these all in one go while remembering them a week later, although this does illustrate how the cues provided are successful with the high success rate from most scheme users. Users attitudes towards the scheme used could have also had an effect. Some of the scheme users were quite reluctant to include all of the supplied characters, or to make their password too long. This may have been because they were worried in case they would not remember the password. Ironically changing what was provided too much, cutting out characters, and not sticking to their decided schemes made some users forget their passwords. This means when the scheme is used in a real world situation and the user becomes more comfortable with it, it may provide even better results.

It is also worth noting that this scheme is still sufficiently robust even in the event of a potential attacker being aware of the password generation additional security pin code which selects associated images. We view this as additional entropy but not a key piece of the system. This adheres to Kerckhoffs's principle.

5. CONCLUSION

This proposed system allows users to create passwords with a high level of entropy helping to protect them against dictionary/hybrid password attacks while also maintaining a high login success rate, even compared to users who were creating weak non-scheme passwords. Any disadvantages with the scheme seem to be with how it is implemented with different users. Protection against brute-force attacks was not increased a lot in the sample of collected passwords as the average password length remained the same between the scheme and non-scheme accounts. A solution to this could be created with some alterations to the scheme. The scheme could possibly include the addition of a static base password added to all the user's account or possibly adding the user's four digit key throughout their password. This could greatly increase the length of the created passwords, greatly increasing the level of security. Although changes like this could be made, more testing would have to be carried out to make sure these additions do not alter the balance between security and usability too much. This could be a possible area for future study.

REFERENCES

- Albahari, J., & Albahari, B. (2012). *C# 5.0 in a Nutshell* (5th ed.). O'Reilly Media, Inc.
- Blocki, J. 2012. *Person Action Object Stories*. Carnegie Mellon School of Computer Science, Available from: <http://www.cs.cmu.edu/~jblocki/personActionObject.htm>
- CrackStation. 2015. *CrackStation's Password Cracking Dictionary*. CrackStation, <https://crackstation.net/buy-crackstation-wordlist-password-cracking-dictionary.htm>
- Dawson, C. W. (2009). *Projects in Computing and Information Systems: A Student's Guide* (2nd ed.). Pearson Education.
- Farrell, S. (2008). Password Policy Purgatory. *Practical Security*, 12, 84–87.
- Kelley, P. G., Komanduri, S., Mazurek, M. L., Shay, R., Vidas, T., & Bauer, L. 2012. *Guess Again (and Again and Again): Measuring Password Strength*. *Security and Privacy (SP), 2012 IEEE Symposium on*. 523-537.
- Morris, R., & Thompson, K. (1979). Password Security: A Case History. *Communications of the ACM*, 22(11), 594–597. doi:10.1145/359168.359172
- Northup, A. (2004). *MCAD/MCSD Self-Paced Training Kit: Implementing Security for Applications with Microsoft® Visual Basic®. NET and Microsoft Visual C#®. NET*. Microsoft Press.
- Oriyano, S. P., & Gregg, M. (2010). *Hacker Techniques, Tools, and Incident Handling*. Jones & Bartlett Learning.
- Scarfone, K., & Souppaya, M. (2009). *Guide to Enterprise Password Management*. National Institute of Standards and Technology.
- Silberschatz, A., Galvin, P. B., & Gagnene, G. (2008). *Operating System Concepts* (8th ed.). John Wiley & Sons.
- SplashData. 2014. "Password" unseated by "123456" on our annual "Worst Passwords" list. SplashData. Available from: <https://www.splashid.com/blog.php>
- Taneski, V., Hericko, M., & Brumen, B. 2014. Password security — No change in 35 years? *In: Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014 37th International Convention on*. 1360-1365.
- Ur, B., & Segreti, S. 2014. *The continuing quest for secure and usable passwords*. Carnegie Mellon University, https://passwordscon.org/wp-content/uploads/2014/08/Ur_Segreti.pdf
- Wanli, M., Campbell, J., Tran, D., & Kleeman, D. 2010. Password Entropy and Password Quality. *In: Network and System Security (NSS), 2010 4th International Conference on*. 583-587.
- Wildenhain, A., Blocki, J., Datta, A., & Blum, M. 2012. *Comparison of Usability and Security of Password Creation Schemes*. Carnegie Mellon School of Computer Science. http://www.cs.cmu.edu/~jblocki/Anne_Wildenhain_2012.htm
- Zhao, R., Yue, C., & Sun, K. 2013. A Security Analysis of Two Commercial Browser and Cloud Based Password Managers. *In: Social Computing (SocialCom), 2013 International Conference on*. 448-453. doi:10.1109/SocialCom.2013.70
- Zviran, M., & Haga, W. J. (1993). A Comparison of Password Techniques for Multilevel Authentication Mechanisms. *The Computer Journal*, 36(3), 227–237. doi:10.1093/comjnl/36.3.227