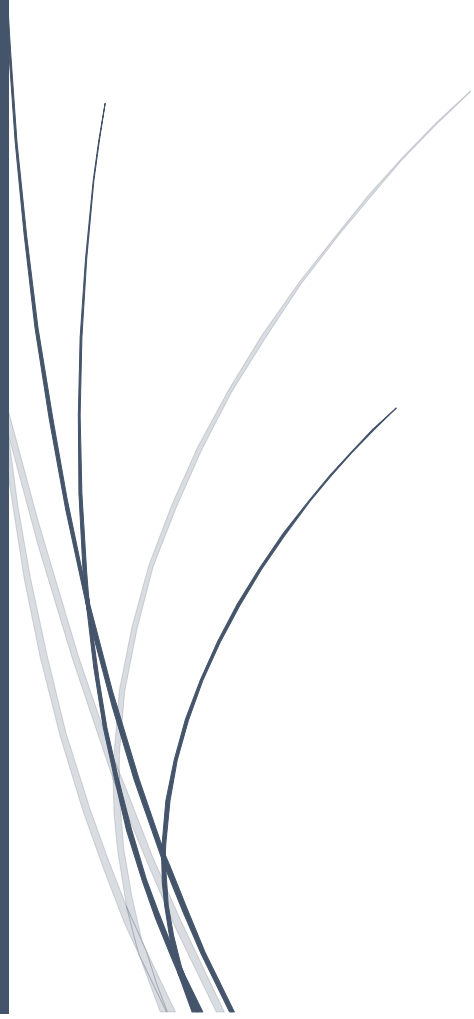
21/10/2016

Your PassWORD is Outdated

An Upgrade to PassSENTENCE is advised!

Terms of Reference



Dale Stubbs - 14024149
MANCHESTER METROPOLITAN UNIVERSITY

Course Specific Learning Outcomes

Upon completion of this project the student will:

- Develop an understanding of the scope and theoretical underpinnings of forensic computing including its technical, professional, legal and ethical aspects;
- Develop a critical and analytical approach to problem-solving in the forensic domain.
- Independently plan, manage and successfully complete a project of substantial size in an area that is relevant to your Degree programme;
- Demonstrate that you have the capacity to gain new skills and knowledge independently of teaching;
- Critically reflect and evaluate existing work and your own work;
- Integrate the learning obtained from other units taken on your Degree programme.

Project Background:

Passwords are everywhere and used on a daily basis by most, if not all, of today's population that have access to a computer, phone tablet or Smart Technology. SearchSecurity defines a password as:

> *"A password is an unspaced sequence of characters used to determine that a computer user requesting access to a computer system is really that particular user. Typically, users of a multiuser or securely protected single-user system claim a unique name (called a user ID) that can be generally known. In order to verify that someone entering that user ID really is that person, a second identification, the password, known only to that person and to the system itself, is entered by the user." (SearchSecurity, 2016)*

Anyone that has to create an online account will most likely be asked to provide a password as a way of securing the data held within the account being created by only allowing those who can provide the password with access to the data being held. The concept of the password system is fundamentally valid, however, the variety of rules regarding password creation from different sources creates a frustrating situation when creating passwords. Moreover, it increases the difficulty in memorising the multitude of passwords that any one person will need to create throughout their lifetime. For example when a user creates a new Apple account they need to follow these rules:

"Passwords must be at least 8 characters, including a number, an uppercase letter, and a lowercase letter. Don't use spaces, the same character 3 times in a row, your Apple ID, or a password you've used in the last year." (Passrequirements.com, 2016)

Whereas someone making a new Microsoft account have these rules to follow:

"Passwords must have at least 8 characters and contain at least two of the following: uppercase letters, lowercase letters, numbers, and symbols." (Passrequirements.com, 2016)

An attempt has already been introduced to overcome the need to create and remember multiple passwords for the multitude of accounts that a single person may require in their lifetime. This ‘solution' is the use of a password manager. There are several password managers in existence to date and they range in price from free to $40 (£31.45) per year for the premium services provided.

The theory behind these passwords managers is, again, a fundamentally valid theory but it appears to be, for the most part, a double edged sword. These programs generate and store your unique passwords for all of the accounts you create which is a valuable asset to have but all of them require a 'master password' to keep your passwords safe and this is their single biggest problem. It is the one single point of failure on each of the password managers; if someone 'cracks' your master password then they have all of your passwords.

This has led to an attempt to keep the master password safe by replacing it with biometric security features. Fingerprint scanners are one such feature. On the face of it, this would appear to solve the issue, however, in reality, the cost of installing or distributing fingerprint scanners could be expensive. They are also not as infallible as most people believe as they can give an increased amount of false negatives (denying the legitimate user access) for reasons such as wet/damp or dirty fingers. Also, research at Yokohama National University in 2002 found that 'gelatin (or "Gummi") fingers were successful around 80% of the time' *(Matsumoto, 2002).*

Another biometric feature I will discuss the 'futuristic' Retinal Scanner. Microsoft has announced that the Microsoft Lumia 950 will have a feature called "Windows Hello" which will grant fast access to the user's phone after scanning the user's iris. The scanner shines a shines a red light into your eye that reviewers of the product found to be 'quite dazzling after multiple uses which is far from ideal if you are a frequent user of the phone.' *(Hall, 2016)*

This leads to a lack of trust lack of proficiency when using biometric features in security. Some people don't trust the master password system or biometric tools whereas most just don't like the idea of not knowing what their passwords. Users are looking for easier ways to overcome these issues and have done so by turning to a less secure but easier way to create passwords for various accounts – Password Reuse. A study by Ofcom, the UK communications watchdog, revealed that 55% of the 1805 adult internet users questioned used password reuse *(Ofcom.org.uk, 2013).*

Password reuse, as the name suggests, is where rather than making a brand new password and attempting to memorise it and what account it is used for, a user, will use the same password with slight variances to it in order to meet the specified criteria. This creates a potentially dangerous situation for those who wish to protect their personal information as if one password is cracked by a computer criminal then there's a higher risk of them cracking all of the user's passwords.

Project Overview:

I intend to create a program that, on the surface, will be a password manager but one that does not generate random passwords in the way that most others do. My program will ask the user to provide a 'PassSentence' (an easy to remember sentence) that only the user will ever have access to. My program will then generate a series of numbers that will represent the placement of characters within the sentence and which order they are to be used.

For example, if I were to use the sentence
'Manchester_Metropolitan-University!Is&Great'
My program would produce an outcome that would look like '6, 21, 42, 11, 25' which would mean the actual password for the account would be 'eta_U'.

This eliminates the requirement to remember a lot of passwords and replaces it with the need to only remember your selected sentence. The passwords generated from this sentence will

be randomly generated allowing for highly secure passwords. However, unlike current password managers, the password characters will never be stored, only the character placements and order will be stored (6, 21,42,11,25 from the example above). This means that even if a hacker acquires access to your password database they would only see a database of numbers thus removing the single point of failure.

The single biggest issue I will come up against is the lack of a single generic set of rules for creating a new password for a specific account. I will need to research the password requirements on as many sites as deemed necessary in order to obtain an average set of requirements and use this information to create a set of guidelines for my passwords generator to follow.

Based on the information obtained I will then need to grant a certain level of control to the user in order to generate a suitable password for certain accounts (numeric characters and/or non-alphanumeric characters).

This will not affect the security of the passwords created or stored but it will pose a challenge when attempting to write the code for the program and it will also add in extra areas where human error can occur.

I am also unsure of which language to write the program in as I am most comfortable when using Java but I do not believe this will provide me with a good enough program. As a result of this, I will be attempting to enhance my understanding of the Python programming language and creating the program using Python.

I will be creating a stand-alone program to use as my prototype which will be written in Python and then using this prototype as the basis for creating my browser plug-in (extension). I will attempt to create the extension using the Kango platform which itself uses JavaScript for the creation of browser extensions.

Aim:

The aim of this project is to create a new type of password manager that has no single point of failure. If a hacker was to obtain your passwords from the password manager they would only have a set of numbers and would be deemed useless by the hacker.

Objectives:

In order to achieve the required aims, I will:

- Produce a fully functioning prototype of the program to begin my testing of the product in order to determine how many of the passwords need to be cracked before the entire sentence becomes compromised. This will require a deeper understanding of Probability Theory.
- Determine a minimum character length for the PassSentence itself to ensure the amount of passwords that may be required can be generated with a minimal amount of crossover of characters. This too will require a deeper understanding of Probability Theory.
- Generate a minimum requirement for each of the generated passwords in order to ensure compatibility of the password to the specified account. I will need to give the user the ability to request a password that contains (or does not contain) capital letters, numbers and/or non-alphanumeric characters depending on the accounts password rules.

Problems:

Incorrect algorithm – The inability to create the necessary algorithm for the creation of every password requirement is the most important problem I may face. If this occurs I will need to give the user the option of entering their own password into the program and storing it appropriately.

Cross-Platform Compatibility – There will be a high probability that I will not be able to ensure my product will work with all currently available web browsers. I will need to aim my product at a selection of the most popular browsers on the market today to achieve the largest scope of coverage possible.

Timetable and Deliverables:

Week Number	Week Commencing	Work To Be Done	Deliverable
Week 0	19/09/2016	Python Programming Code Academy Tutorial Videos	
Week 1	26/09/2016	Python Programming Lynda.com Tutorial Videos	
Week 2	03/10/2016	Draft 1 of ToR Complete first draft of ToR.	Draft 1 of ToR
Week 3	10/10/2016	Draft 2 of ToR Make necessary modifications to ToR	Draft 2 of ToR
Week 4	17/10/2016	ToR and Ethics Form Final Draft of ToR and Ethics Form	Completed ToR Uploaded
Week 5	24/10/2016	Literature Review (LR) Find and evaluate 8 – 10 papers on passwords and password managers	Draft 1 of LR and Ethics Form Uploaded
Week 6	31/10/2016	Second Draft of LR Make necessary amendments to LR	Completed LR Uploaded
Week 7	07/11/2016	Probability Theory Research Calculate the number of possible passwords that can be generated from a specific password length.	Minimum Length of PassSentence Confirmation
Week 8	14/11/2016	Probability Theory Research Calculate the rate of character crossover within the password generator	Quantity of compromised passwords needed to crack PassSentence Determined
Week 9	21/11/2016	Algorithm design Construct the pseudocode for the password generation algorithm along with the user interface	Product Pseudocode

Week 10	28/11/2016	Algorithm design and Construction of Prototype Convert the pseudocode into Python code and construct the prototype.	Prototype Constructed
Week 11	05/12/2016	Product Design Create a detailed document of how my product will work and instructions of use.	Product Design and Guidelines of Use
Week 12	12/12/2016	Prototype Debugging Use the PassSentence Prototype for debugging purposes.	Debugging Complete
Week 13	09/01/2017	JavaScript Pseudocode Create pseudocode of the Python program for conversion into JavaScript	JavaScript Pseudocode
Week 14	16/01/2017	JavaScript Code Convert the pseudocode into JavaScript code	JavaScript Code
Week 15	23/01/2017	ED Complete the evaluation design of the product	First Draft of ED
Week 16	30/01/2017	ED Draft 2 Make the necessary amendments to the ED	Complete ED
Week 17	06/02/2017	Product Construction Build the PassSentence browser plugin and begin testing.	Final Product and Test Results
Week 18	13/02/2017	Report Outline (RO) Complete the first draft of the report.	First draft of RO
Week 19	20/02/2017	RO Draft 2 Make the necessary amendments to the RO	Complete RO
Week 20	27/02/2017	Presentation Slides Produce the necessary slides for the presentation	Presentation Slides Draft 1
Week 21	06/03/2017	Presentation Slides Complete any and all amendments to the slides ready for the practise presentation	Presentation Slides Draft 2
Week 22	13/03/2017	Presentation Practise Practise the presentation and achieve the correct timing and materials (slides) required	Practise Presentation

Week 23	20/03/2017	Presentation Modification Make necessary modifications to the presentation materials/script	Final Presentation Slides and Script
Week 24	27/03/2017	Project Completion Ensure all work has been completed and product is fully operational	All work Completed.
Week 25	24/04/2017	Prepare Report, Product and Presentation slides for submission	Complete Presentation and Submit Competed Project

Required Resources:

As the project only requires the use of either Java or Python in order to create the working model I will require no additional resources when creating this program. I will only require the use of a notepad variety program (such as Sublime Text or Notepad++) to write the code and the computer terminal to compile the program.

Bibliography

SearchSecurity. (2016). *What is password? - Definition from WhatIs.com*. [online] Available at: http://searchsecurity.techtarget.com/definition/password [Accessed 14 Oct. 2016].

Passrequirements.com. (2016). *Password requirements for itunes | passrequirements.com*. [online] Available at: http://passrequirements.com/passwordrequirements/itunes [Accessed 30 Sep. 2016].

Passrequirements.com. (2016). *Password requirements for microsoft | passrequirements.com*. [online] Available at: http://passrequirements.com/passwordrequirements/hotmail [Accessed 30 Sep. 2016].

Matsumoto, T. (2002). *Importance of Open Discussion on Adversarial Analyses for Mobile Security Technologies---A Case Study for User Identification---*. 14th ed. [ebook] Seoul. Available at:http://perso.telecom-paristech.fr/~chollet/Biblio/Articles/Domaines/BIOMET/spoof-fp5p4.pdf [Accessed 14 Oct. 2016].

Murgia, M. (2016). *Biometrics will replace passwords, but it's a bad idea*. [online] The Telegraph. Available at: http://www.telegraph.co.uk/technology/2016/05/26/biometrics-will-replace-passwords-but-its-a-bad-idea/ [Accessed 14 Oct. 2016].

Hall, C. (2016). *Microsoft Lumia 950 review: The dawn of Windows 10 Mobile - Pocket-lint*. [online] Pocket-lint.com. Available at: http://www.pocket-lint.com/review/136018-microsoft-lumia-950-review-the-dawn-of-windows-10-mobile [Accessed 14 Oct. 2016].

Ofcom.org.uk. (2016). *UK adults taking online password security risks - Ofcom*. [online] Available at: https://www.ofcom.org.uk/about-ofcom/latest/media/media-releases/2013/uk-adults-taking-online-password-security-risks [Accessed 14 Oct. 2016].