An Investigation into Companies' Password Policies in Response to a Breach

Lukas Smith

CMP320: Ethical Hacking 3

BSc Ethical Hacking Year 3

2021/22

Note that information contained in this document is for educational purposes.

.

Abstract

This whitepaper aims to create a program that analyses leaked password lists to accurately predict the password policy enforced at the time of the breach. The Wayback machine and several external articles will be used to confirm the actual password policy just before the breach. Finally, the companys response to the breach will be discussed to note any particular changes due to the breach.

The resulting program was programmed in Python using PyCharm. It was created by using a class ‘password.py’, which takes the password list and outputs it and the statistics into a class object. The ‘main.py’ program then takes this class and outputs three main files with the objects:

- ‘output.txt’ – The predicated password policy and a summary of the password statistics to assist with the analysis
- ‘PasswordLenOutput.csv’ – A list of all of the password lengths and how many times that password length was used
- ‘PasswordOutput.csv’ – A list of the 100 most common passwords within the list (this was only used in the non-unique password lists)

In summary, the program was successful, as the only issues came from outlying passwords within the password lists used to test the program. The Hotmail password list was the most accurate, and the program accurately guessed the password policy for that site. The Hotmail password list proves the accuracy of the program when using the correct password list.

In terms of the password statistics, there were several common password phrases in use, such as ‘password’, ‘abc123’ and personal information such as names and ages. Additionally, the most common passwords always met the minimum requirements for the password policy, such as minimum length, number and symbols.

From further investigating the companies after the breach, it turns out that most companies did not change their password policies in response to a breach and instead opted for resetting leaked credentials, promoting secure passwords and promoting the use of 2FA.

.

+Contents

.

.

1 INTRODUCTION

1.1 BACKGROUND

Password policies are standard among businesses to promote secure password usage within their applications. Secure and unique passwords reduce the chance of them being brute-forced by an attacker. By enforcing specific criteria such as password length, prohibiting personal information and dictionary words, unique passwords and requiring a range of different characters, the user is less likely to use a simple password that could be found in an attacker's password list. With compromised credentials accounting for 61% of breaches last year (Verizon, 2021), password security is more prominent than ever.

There have been several discussions regarding password policy, mainly surrounding whether regularly resetting user passwords and enforcing complexity should be necessary. There is an argument to be made that regularly getting users to reset their passwords makes it harder for users to come up with unique passwords and then remember them. Instead, password resets should only be required when a breach has been discovered. Additionally, it is recommended that businesses encourage Multi-Factor Authentication (MFA) so that user accounts are significantly less likely to be compromised in the event of a password data breach.

57% of businesses globally make use of MFA (LastPass, 2019). Although this is an annually increasing statistic, there is still a large portion of businesses that only use passwords for authentication, leaving them significantly more vulnerable in the event of a data breach. Since the most extensive compiled password list, RockYou2021, contains over 8 billion unique passwords, it is even more critical to ensure that passwords are unique and secure.

When companies experience a breach, it is common to go through an Incident Response Plan. The last stage of this is to reflect on the incident and implement changes to prevent future incidents. In the case of a password data breach, it should be a typical response to assess the password policy to guarantee its resilience. Should the password policy be found to be outdated, it must be revaluated.

1.2 AIM

This paper aims to take password breaches and be able to accurately predict the password policy the company used at the time of the breach. In order to reach this aim, a python program will be developed to attempt to determine the password policy at the time of the password breach by using the publicly available compromised credentials. The password policy includes maximum and minimum character length, allowed characters, symbol enforcement, number enforcement and character casing enforcement. The program will also output interesting password statistics, such as the most common passwords and average password length.

The output of this project will then be discussed to analyse the password policy at the time against the current password policy, which will allow the author to identify any changes since the password breach. Additionally, the company’s response to the breach will be further discussed. Finally, the Wayback machine will be used to determine the program’s overall effectiveness by analysing the password policy that was actually enforced during the password breach.

2 Procedure

2.1 Overview of Procedure

The following procedure is split up into the follow sections to make it easier to follow:

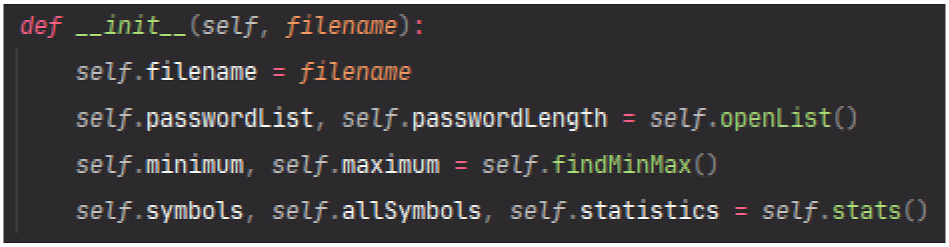
- Password.py – The class function created to parse the password lists and turn them into class objects
- Main.py – The main python program used to create the outputs
- Password Lists – The description of how the password lists were split up to anonymise the data used

In order to undertake this project, several tools and software were used:

- Excel – To create the resulting graphs
- Notepad++ –To analyse the password lists and outputs
- PyCharm – The Python IDE used to program
- Wayback Machine – To check the accuracy of the program after runtime

2.2 Password.py

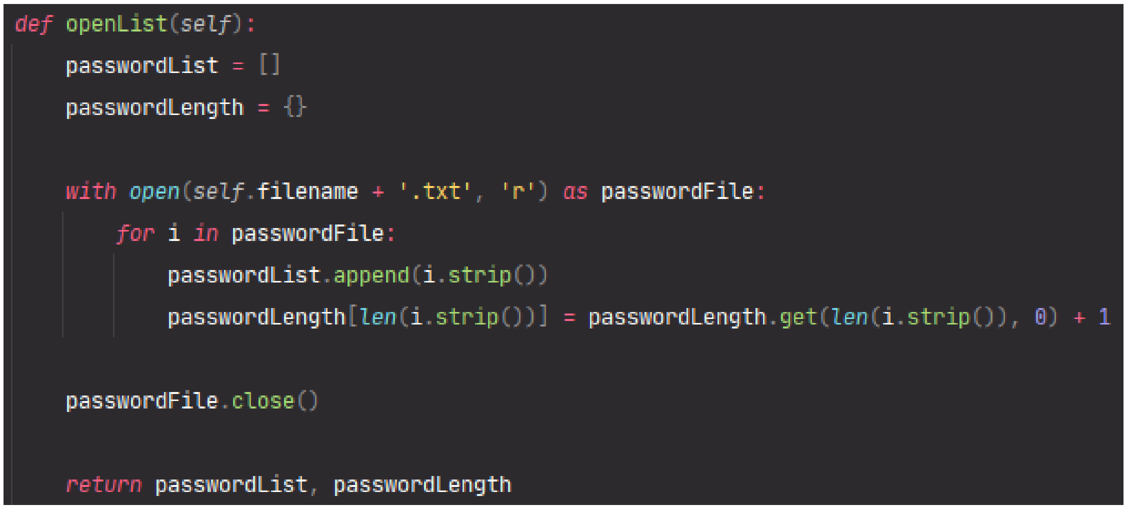
The first step in the programming was to create a python class that can take in a password text list and store its statistics. For this, several variables are needed. filename to take the file's name to make it easy to analyse multiple password lists and create a unique output file, passwordList to store the passwords in a list for analysis and passwordLength to store a collection of password lengths to find the average password length efficiently. For the statistics, minimum and maximum store the smallest and largest passwords whereas allSymbols is a collection noting how many times a symbol appears and symbols is a list of all unique symbols. Finally, the statistics list stores information regarding the format of the passwords, including if they are all lowercase, all uppercase, all alphamerical or all alphanumerical.



```python
def __init__(self, filename):
    self.filename = filename
    self.passwordList, self.passwordLength = self.openList()
    self.minimum, self.maximum = self.findMinMax()
    self.symbols, self.allSymbols, self.statistics = self.stats()
```

Figure 1 - All variables created on initialisation of object

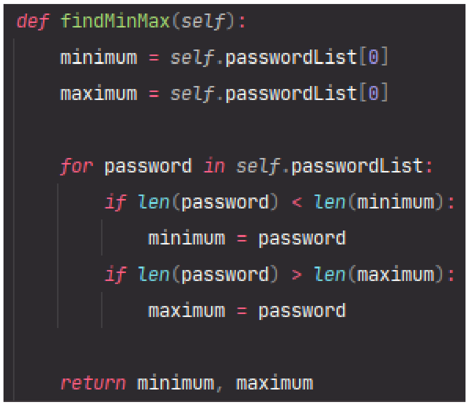
The first function openList, takes the filename and adds the txt extension and opens it, taking the password list line by line and stripping it of newline characters and end spaces. These characters must be removed due to the format of the password list. This stripped password is then appended to the passwordList variable. Additionally, the length of the password is added to the password length collection to analyse the most common password lengths later. The password length was initially stored using a dictionary; however, it was too slow with large password lists.



```python
def openList(self):
    passwordList = []
    passwordLength = {}

    with open(self.filename + '.txt', 'r') as passwordFile:
        for i in passwordFile:
            passwordList.append(i.strip())
            passwordLength[len(i.strip())] = passwordLength.get(len(i.strip()), 0) + 1

    passwordFile.close()

    return passwordList, passwordLength
```

Figure 2 - openList function

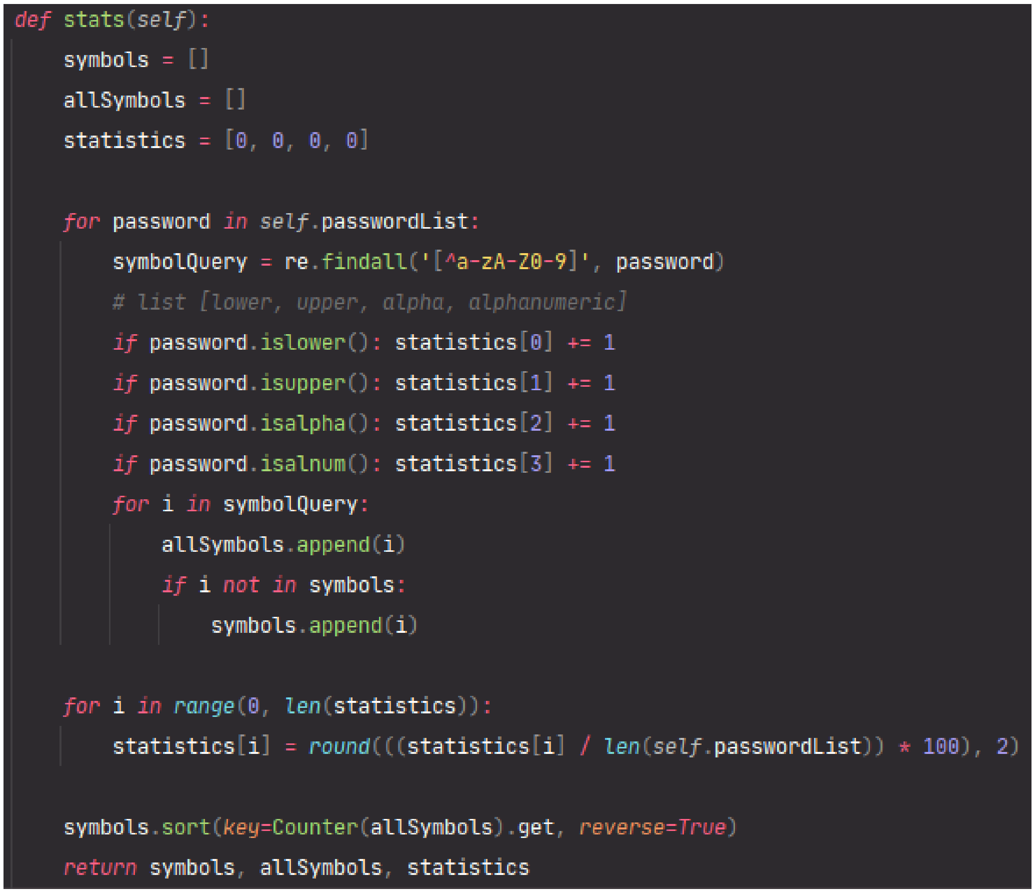
Next is the findMinMax function, which takes the passwordList that was created and filled earlier and loops through it to find the shortest and longest length passwords. It stores the password rather than its length to display the password in the final output.



```python
def findMinMax(self):
    minimum = self.passwordList[0]
    maximum = self.passwordList[0]

    for password in self.passwordList:
        if len(password) < len(minimum):
            minimum = password
        if len(password) > len(maximum):
            maximum = password

    return minimum, maximum
```

Figure 3 - findMinMax function

Finally, the stats function takes the password list and uses the regex library to search each password for non-alphanumeric characters. If the symbol is unique, it is added to the symbols list, and all symbols are added to the allSymbols collection to count how many times each symbol is used. Additionally, the statistic list is filled for each type of password. At the end of the function, the symbols list is sorted by the most used symbol from the allSymbols collection. The reverse variable ensures that the most popular symbol is at the top of the list. The sort function is from the counter library already in-built in Python from the collections structure.



```
def stats(self):
    symbols = []
    allSymbols = []
    statistics = [0, 0, 0, 0]

    for password in self.passwordList:
        symbolQuery = re.findall('[^a-zA-Z0-9]', password)
        # list [lower, upper, alpha, alphanumeric]
        if password.islower(): statistics[0] += 1
        if password.isupper(): statistics[1] += 1
        if password.isalpha(): statistics[2] += 1
        if password.isalnum(): statistics[3] += 1
        for i in symbolQuery:
            allSymbols.append(i)
            if i not in symbols:
                symbols.append(i)

    for i in range(0, len(statistics)):
        statistics[i] = round(((statistics[i] / len(self.passwordList)) * 100), 2)

    symbols.sort(key=Counter(allSymbols).get, reverse=True)
    return symbols, allSymbols, statistics
```

Figure 4 - stats function

The full code for this class can be found at ***Appendix B – Password.py***.

2.3 MAIN.PY

The primary python function takes the created class object and outputs it into a text file. The output is created by first asking the user for the file's name, which will make it easier to test multiple files in quick succession. From there the 'report' function is run, this first creates and opens the output file, first writing the estimated password policy. This consists of first writing the password length range, then all the non-alphanumeric symbols, using the textwrap library to wrap the symbols instead of them all being written in one line. Then it is written if character case enforcement is in place by checking the statistics regarding all uppercase and all lowercase passwords.

The program then goes on to write about general statistics, first the shortest password and its length then the longest password and its length. It then writes the top 10 most popular passwords and password lengths above 5% of the total passwords. Furthermore, the statistics list is written, including the total number of passwords. Finally, the allSymbols collection displays the top 10 most common symbols.

Initially, all off the passwords and password lengths would be displayed. As this makes it significantly harder to interpret the results, the program was changed to display the most common results instead.

The rest of the results are placed into two CSV files, one containing the 100 most common passwords and the other containing all the password lengths.

After initial testing, it was determined that having three files for each password list made it inconvenient to view the results, so the code was changed to create a folder with the same name as the password file. The resulting files are written into these directories instead of within the main directory. This was done using the OS library within Python.

See the full code at ***Appendix A – Main.py***.

2.4 Password Lists

Finally, to use the program, password lists have to be obtained. In this case, https://haveibeenpwned.com/ was used to find publicly available password lists. As these lists often contained other irrelevant information, they had to be reformatted only to include the passwords. The first one was the ‘000webhost’ leak, which contained email addresses, usernames and IP addresses. This is also important as we want to anonymise the data. A quick script was written to parse the list just to include passwords. Sometimes information was missing from the list, in which case try and except lines were written to allow for this, and in other cases, the lines were blank, so if statements check for this and pass over them. The list is then sorted into alphabetic order.

```
output = open('000webhostunsorted.txt', 'wb')
error = open('error.txt', 'wb')

with open('000webhost.com.txt', 'r', encoding="utf-8") as passwordFile:
    for line in passwordFile:
        line = line.strip()
        if line:
            try:
                if len(line.split(':', 3)[3].strip() + '\n') != 0:
                    output.write(line.split(':', 3)[3].strip() + '\n')
            except:
                try:
                    if len(line.split(':', 3)[3].strip() + '\n') != 0:
                        output.write(line.split(':', 3)[3].strip() + '\n')
                except:
                    error.write(line)

output.close()
sort = open('000webhost.txt', 'w')
with open('000webhostunsorted.txt', 'r') as r:
    for line in sorted(r):
        sort.write(line.strip() + '\n')

sort.close()
error.close()
```

Figure 5 - textparse.py

This was also done for the gmail and Fortinet leaks, the python scripts for these can be found at ***Appendix D – fixlist.py (gmail)*** and ***Appendix C – fixfortinet.py***, respectfully.

3 RESULTS

Before displaying the results, it should be noted that the 000webhost and the Fortinet password lists were more comprehensive, meaning that duplicate passwords could be used to analyse the most common passwords. When moving on to the other three password lists however, it was noted that these lists were unique, meaning that any duplicate passwords had been removed (most likely to prepare the list for brute-forcing). Having unique password lists means that these lists could not be analysed for most common passwords; however still worked for identifying password policies, which is why they are still included.

The following password lists were used:

- 000webhost
- Fortinet
- Gmail
- Hotmail
- MySpace

3.1 000WEBHOST

The full raw data created by this password list are set out within ***Appendix E – 000webhost*** and are sub-headed appropriately.

3.1.1 Output.txt

There was a total of 15,271,083 passwords in this password list. The smallest password found was one character long, and the longest password found was 255 characters. As more than 1% of passwords were all lowercase and 81% of passwords were uppercase, it is unlikely that there was forced character casing. The most common password was 'abc123' which was found 24,928 times. Additionally, the most common password length was 8 characters long. Interestingly, 93% of passwords were alphanumerical (no symbols), but in passwords that did use symbols, the most common symbol was the @ symbol which was used 340,857 times. On top of regular symbols used within passwords, there were also many strange characters, such as control characters (for example, the ESCAPE character).

3.1.2 PasswordLenOutput.csv

As stated before, the most common password length was eight characters long. Most of the passwords are between 6 and 17 characters long, with less than 0.02% of the passwords being below six characters.

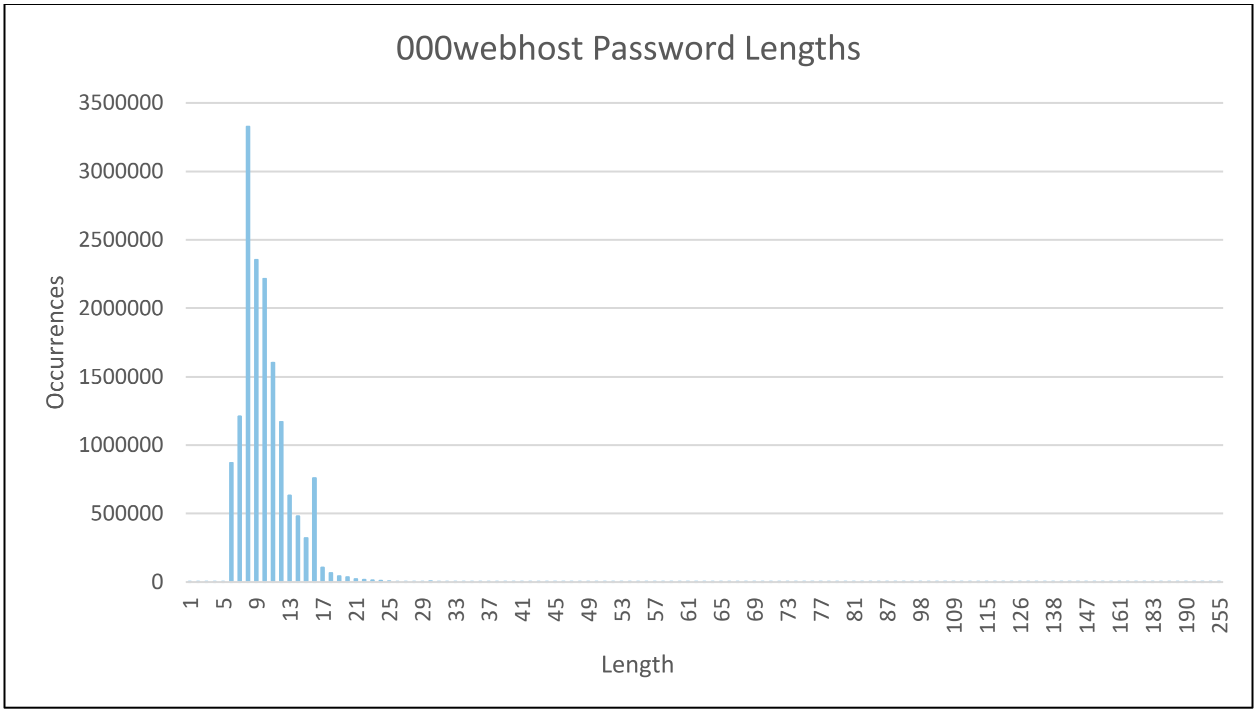


Figure 6 - 000webhost Password Lengths

The above graph shows that many of the passwords are within the smaller range, with very few passwords above 21 characters and below 6 characters.

3.1.3 PasswordsOutput.csv

Within this file are the most common passwords within this password list, with the most common password being 'abc123', which was used 24,928 times. Many of these passwords contain common patterns such as '123' or 'abc'.

A graph with the top 100 passwords would be too large to fit into this document; however the following graph shows the top 50 passwords.

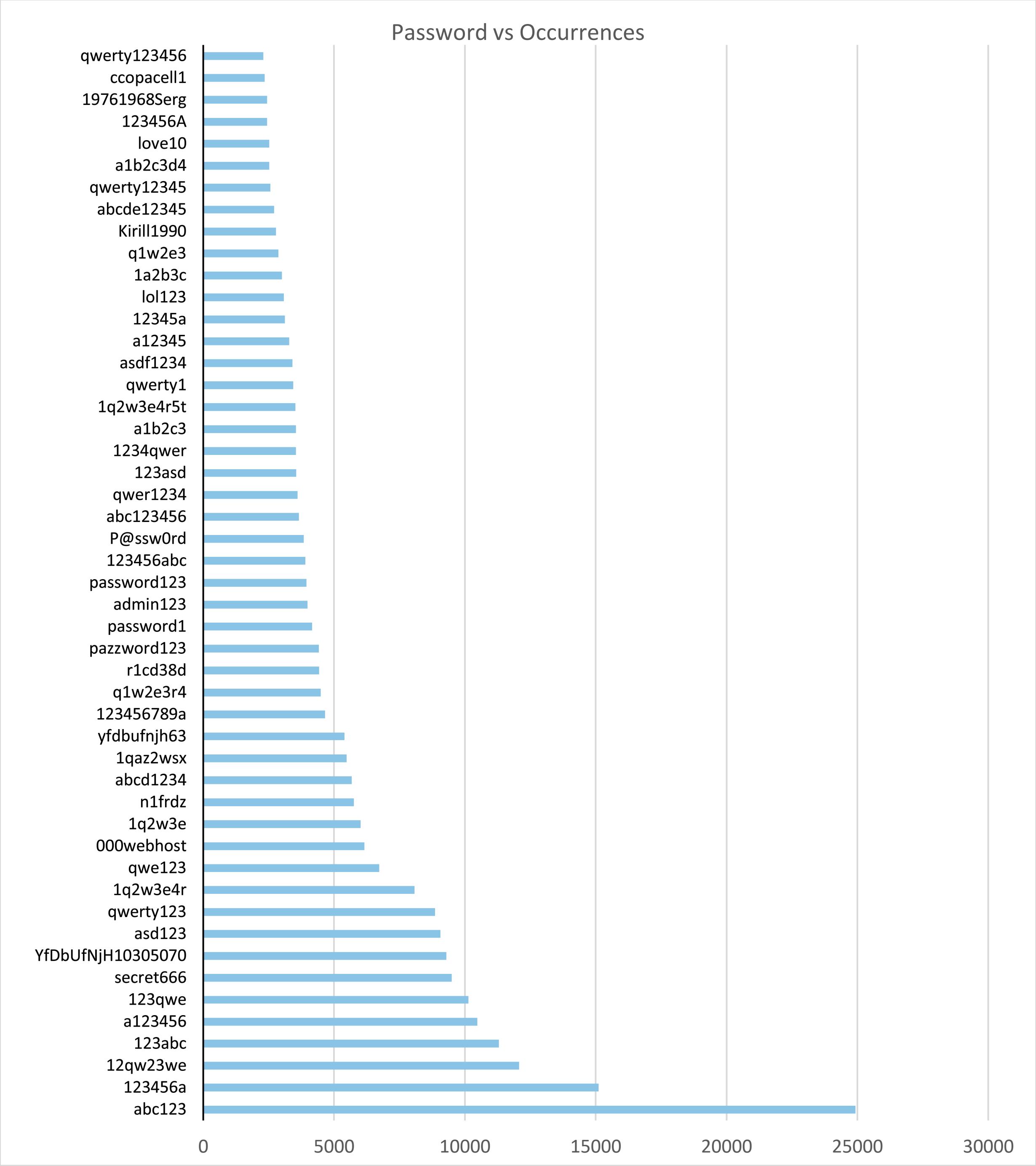


Figure 7 - Password Vs Occurrences Graph

As can be seen in the above graph, many of the passwords are dictionary words and contain common password patterns. Additionally, the site name is the 13th most common password. Many passwords are just numbers and letters, with no change in the casing and no symbols.

3.2 FORTINET

The raw data produced for this password list can be found within ***Appendix F – Fortinet*** and has been sub-headed appropriately.

3.2.1 Output.txt

Once the whitespace and usernames were removed, 398,826 passwords were recorded within this password list. First, the shortest password, 'b', is one character long, whilst the longest password, which appears to be a series of hex bytes broken up by spaces, is 599 characters long. There were 33 unique symbols, all relatively common, with no control or non-Latin characters. 26% of passwords were all uppercase and 5% were lowercase. The most common password was 'Temporal2020', with it appearing 2328 times within the list, and the most common password length was eight characters, with 21% of passwords being this long. 4.67% of passwords only contain characters from the alphabet, and 46.74% of passwords do not contain symbols. The most common symbol was @, which was found 81,765 times within password lists.

3.2.2 PasswordLenOutput.csv

The bar graph below shows that the most popular password lengths are between 8-11 with these character lengths accounting for just under 60% of the total passwords.

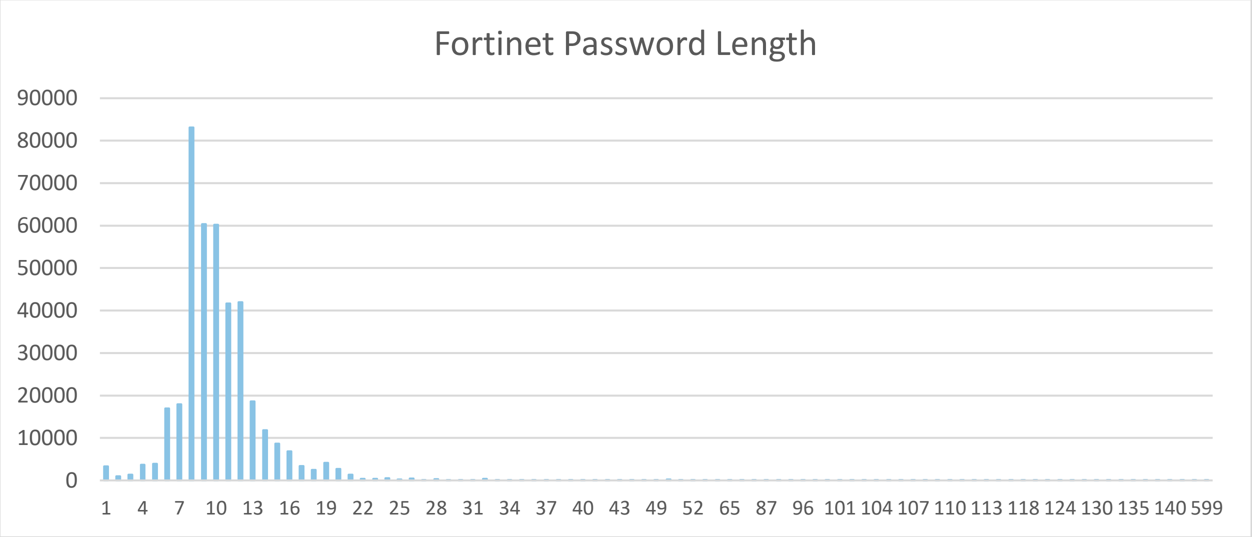


Figure 8 - Fortinet Password Length Graph

3.2.3 PasswordsOutput.csv

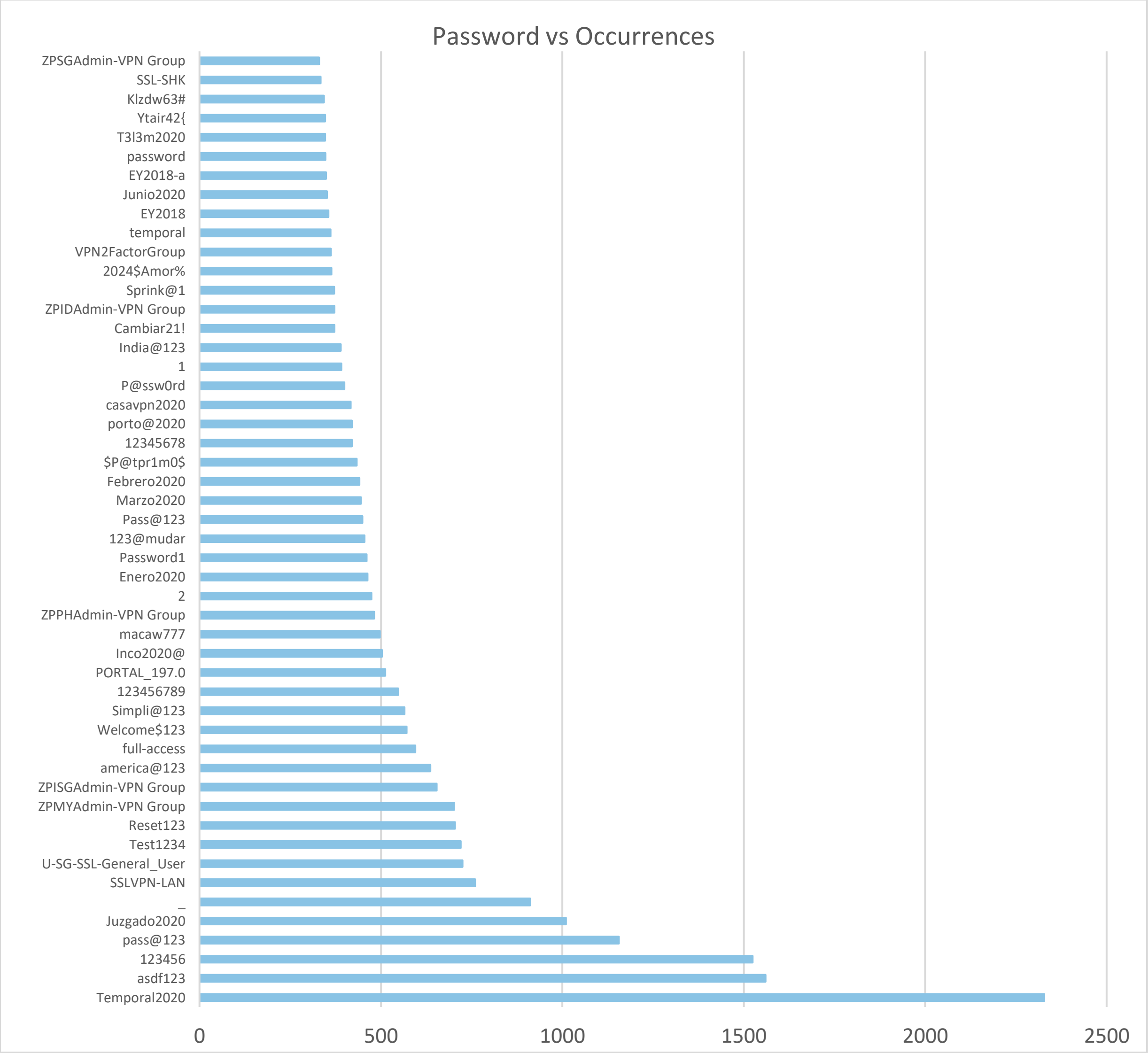


Figure 9 - Fortinet Password Occurrences

The above graph (which only includes the top 50 passwords due to the size of the graph) presents the most common passwords, with Temporal2020 being used 2328 times. The word 'VPN' appears in several of the most common passwords, and most of the passwords involve dictionary words. Additionally, a large portion of the most common passwords are not alphanumerical.

3.3 GMAIL

The raw data produced for this password list can be found within ***Appendix G – Gmail*** and has been sub-headed appropriately. This is the first list that doesn't include common passwords, as the list only includes unique passwords rather than the full database.

3.3.1 Output.txt

Within this password list, there were 3,132,000 passwords. The first shortest password was one character long and was ` whilst the longest password was 91 characters long and is a series of 'g's. Again, there are no 'most popular' passwords due to the password list being unique characters. The most common password length was 8 characters long, accounting for 29% of the passwords. 85% of the passwords were all uppercase, and 0% of the passwords were all lowercase. Whilst 29.63% of the passwords are only alphabet characters and 97% of the passwords do not contain symbols. The list of symbols contains mostly regular symbols with no non-latin characters appearing; however, it does contain four control characters. Finally, the most popular symbol in the password list is . and it was found 23,487 times.

3.3.2 PasswordLenOutput.csv

As stated earlier, the most common password length is 8 characters, and it was used for 909,480 passwords. The majority of the passwords are between 6 and 10 characters.

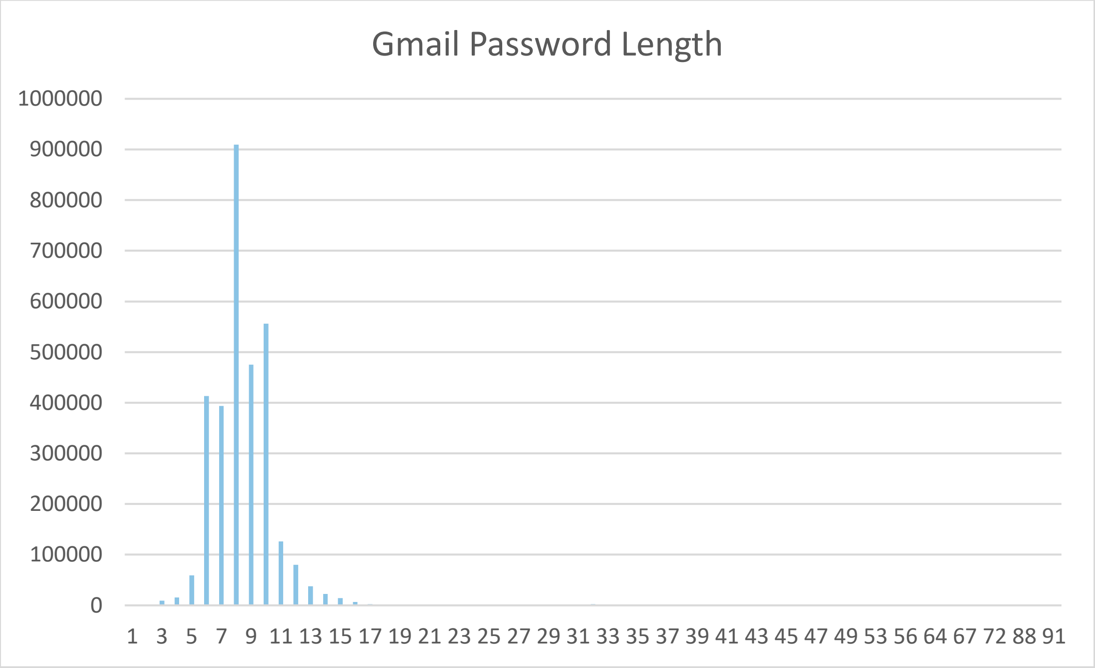


Figure 10 - Gmail Password Lengths Graph

As the graphs shows very few of the passwords are outside the 6 to 10 range.

3.4 HOTMAIL

The raw data produced for this password list can be found within ***Appendix H – Hotmail*** and has been sub-headed appropriately. Again, this password list does not contain any common password statistics as the passwords within the list are all unique.

3.4.1 Output.txt

This password list was significantly smaller than the others, with only 8,566 passwords. The first shortest password in the list is six characters long and was '123456' whilst the longest password in the list is 16 characters long and is ‘supermariojavith’. The most common password length is 6 characters long, and this accounts for 21.27% of the passwords. In terms of character casing, 73.44% of the passwords are all uppercase, whereas 5.05% of the passwords are all lowercase. Regarding types of characters, 45.62% of the passwords contain only characters from the alphabet, and 93.11% contain numbers and letters. All of the symbols within the list are standard symbols and do not contain any non-latin characters or control characters. Finally, the most common symbol found within the password list was . and it appears 230 times.

3.4.2 PasswordLenOutput.csv

The most common password length was 6 characters, and it was used 1,822 times. Furthermore, most of the passwords were between 6 and 9 characters long, with these lengths accounting for 70% of the total passwords.

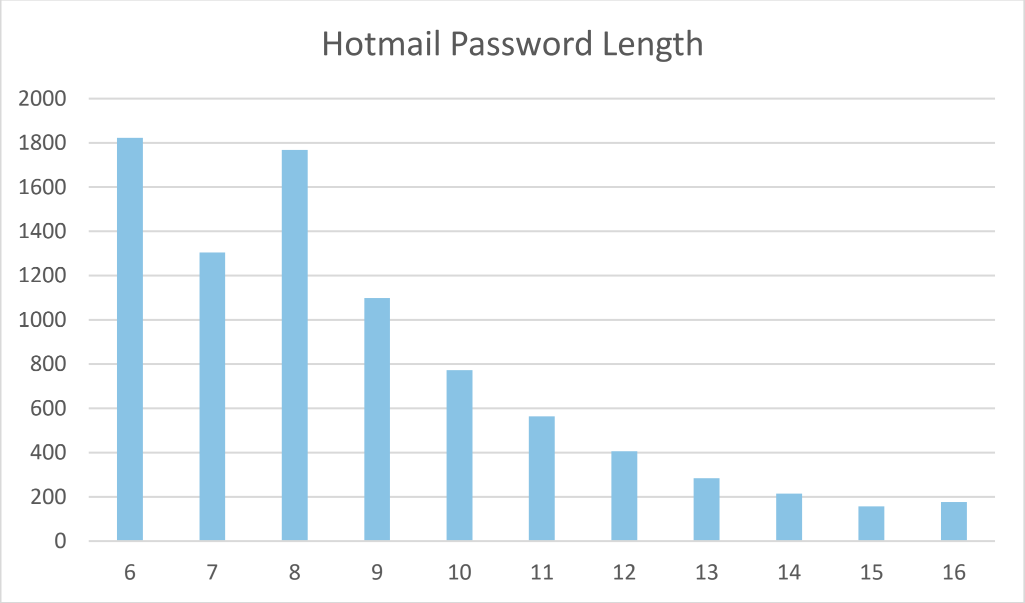


Figure 11 - Hotmail Password Length Graph

The graph shows the most common password lengths within the Hotmail password list.

3.5 MySpace

The raw data produced for this password list can be found within ***Appendix I – MySpace*** and has been sub-headed appropriately.

3.5.1 Output.txt

There were a total of 37,126 passwords within the MySpace password list. The first shortest password within the list was one character long and was) whilst the longest password was a 6,341 characters long and it was a repetition of ‘abcdefgh’. The most common password length was 8 characters, accounting for 22.92% of the passwords. For character casing, 92.43% of the passwords were all uppercase, and 3.54% were lowercase. Regarding character types, 7.35% of the passwords only contained characters from the alphabet and 89.31% of the passwords contained both alphabet characters and numbers but no symbols. The symbols contained within the list were majorly common, however, they also include several unknown symbols. Finally, the most common symbol used within the list is # which was found 5,477 times inside the list.

3.5.2 PasswordLenOutput.csv

As shown in the graph below, the most common password lengths are between 6 and 10, with these lengths accounting for 92% of the password lengths. Very few passwords are outside of this range, and the most common password length is 8 with 8,509 occurrences, closely followed by 7 with 8,397 occurrences.

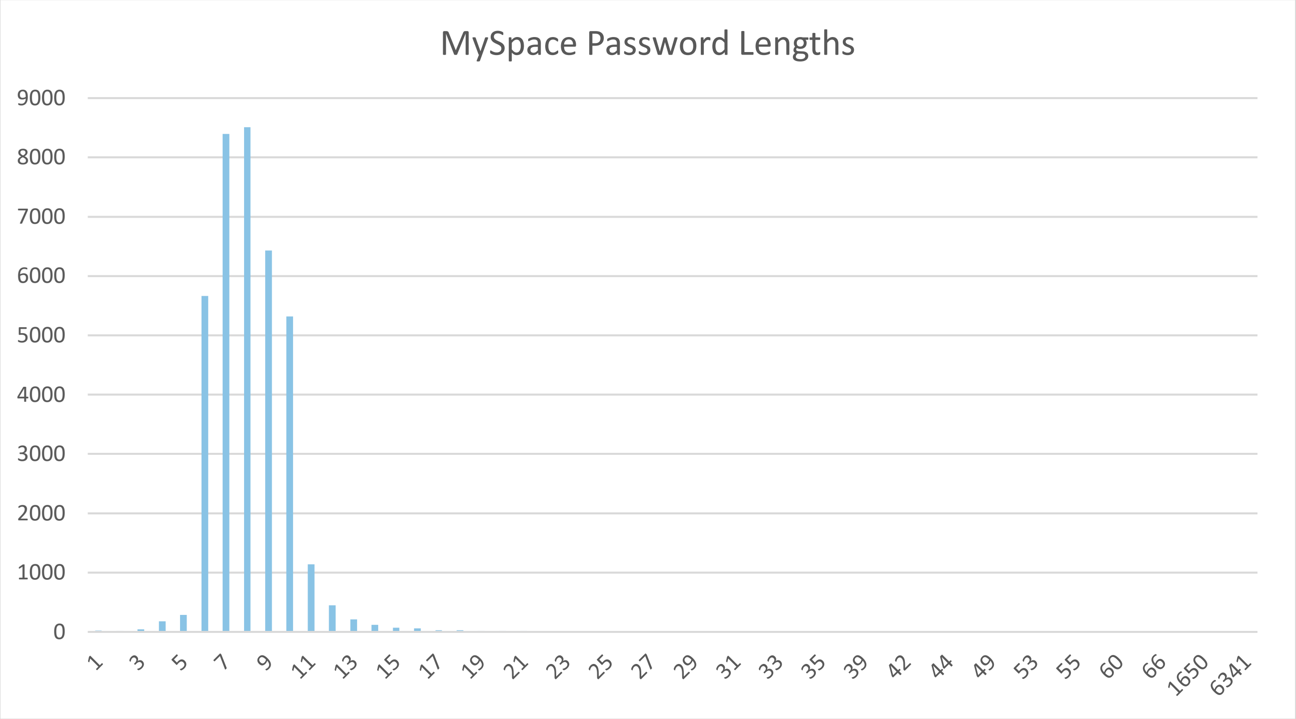


Figure 12 - MySpace Password Length Graph

4 DISCUSSION

4.1 GENERAL DISCUSSION

To evaluate the program’s effectiveness, the Wayback Machine is used for each site just before the breach occurred to determine the actual password policy and then compare that against the program’s results.

4.1.1 000webhost

The 000webhost breach occurred in March 2015, when 13 million records were leaked due to a PHP vulnerability within the site.

As this is the most extensive password list used, it was the one with the most outliers. It is safe to say the max character limit is 255 as there were no passwords greater than this length, and it is one less than $2^{8,}$ suggesting that that was the number of bits assigned to the password field. Furthermore, two passwords are 255 characters long, and the first password is cut off, further suggesting that this is the max character limit. As the lowest password length was 1, it would be understandable to assume that this is the smallest password length, however, the giant spike in passwords greater than or equal to 6 suggests that this could be the actual character limit. Additionally, the passwords with less than 6 characters make up 0.014% of the total passwords. These outlying passwords could be due to malicious users exploiting the site and avoiding the character limit.

Due to the significant number of non-Latin characters and control characters, it is safe to say there is no character filtering. Additionally, the number of all lowercase and all uppercase passwords suggests no character casing enforcement on the site. The large portion of alphanumerical passwords shows that there are no symbol requirements in place on the site.

When accessing the signup page ‘order.php’ on the Wayback Machine (000webhost, 2015), it is shown that the password policy requires at least 6 ‘symbols’ (it can be assumed that symbols are being used instead of characters in the case due to the nature of the leak). Additionally, numbers and letters are both required.

From this information, it is safe to assume that the password list isn’t 100% accurate as there were passwords with less than 6 characters. The passwords in the list are likely due to how the password list was parsed by the textparse.py program (see ***Figure 5 - textparse.py***). The program also appears to have predicted the password policy accurately.

Webhosts’ response to the breach was first to reset all members’ passwords and suggest that users change their passwords on sites where they may use the same password. On creating a new account on their site, the minimum password length has been upped to 8 characters; however, symbols are still not required. They did instead change the encryption method they use for their passwords. (000webhost, 2015)

4.1.2 Fortinet

Fortinet is a corporation which sells its own developed cybersecurity solutions. The Fortinet breach occurred between May 2019 and June 2021. It leaked a list of just under 500,000 VPN credentials which threatened organisations’ security as the VPNs allow users to access their systems remotely. The breach resulted from a path traversal vulnerability on out-of-date devices that allowed Fortinet credentials on the vulnerable device to be leaked.

The Fortinet password list was not as sizable as the 000webhost leak; however, it still held a significant number of credentials (398,826). 3312 passwords are one character long, suggesting this is the lower character limit. The highest character limit was initially thought to be 599; however, after investigating the passwords further, it turns out that passwords above 128 characters were errors caused by parsing the password list. Removing these entries showed the max character limit to be 128 characters. As 47% of passwords were alphanumeric, it is unlikely that symbols were required, and the number of all lowercase and uppercase passwords suggests there was no character case enforcement. Additionally, the number of only alphabet passwords suggests that numbers were not required.

After investigating Fortinet’s password policy at the time of the breach, it was discovered that password policies were dependent on the company’s own password policy; however, there were max and minimum limits that the program did find, such as passwords not being larger than 128 characters, that case enforcement can be disabled, and that symbol enforcement can be disabled. (FortinetGURU, 2019)

There were a few outliers again with the program, but this was due to the parsed password list rather than the policy analyser.

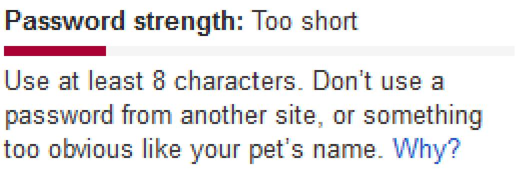
Fortinet did not make any active changes in response to the breach, but a statement was released explaining the next steps for companies using their systems. As the password policy is defined company by company rather than by Fortinet, Fortinet made no changes to password policies directly. The statement does suggest several next steps, such as disabling the VPNs until they have been patched and updating systems to ensure they are not still vulnerable. (Windsor, 2021)

4.1.3 Gmail

In September 2014, 5 million Gmail and Passwords were published online, although Google claimed that its systems were not compromised. It turns out that this list came about from a collection of several other leaks, and Google states that less than 2% of the credentials were up to date (Google, 2014); however, that is still a substantial number of credentials.

The first shortest password within this list was one character long and was ‘`’ whilst the longest password was 91 characters long and was a series of repeating 9s. Other than 4 control characters, all other symbols were common symbols, suggesting that the control characters may be outliers. There were also no non-latin characters. As there are no all-lowercase passwords, character casing is likely enforced and requires a mixture of lowercase and uppercase characters. Symbol enforcement is also very unlikely due to 97% of passwords being alphanumeric.

It is a little harder to get Google’s password policy due to how the site is set up and because the Wayback machine has gmail.com as a banned URL; however, a stack overflow article states that the password length is set between 8 and 100 characters, which is different from the results. The difference could be because the password list is not as accurate as initially thought. There does not seem to be any specific policy other than the password must be ‘secure’.



Password strength: Too short

Use at least 8 characters. Don't use a password from another site, or something too obvious like your pet's name. Why?

Figure 13 - Gmail Password Policy

Other than the minimum character limit and the control characters, however, the password policy the program predicted was correct.

On discovering the breach, Google claimed that their systems were not breached but noted that they had contacted individuals whose credentials were in the dump. This blog also gave several tips, including a link on creating a strong password and setting up 2FA on a Gmail account. (Google, 2014)

4.1.4 Hotmail

In 2009 10,000 Hotmail credentials were leaked onto a third-party site, and the malicious actors claimed that many more accounts were comprised. Microsoft stated that this was not from an internal breach but instead due to a phishing scheme. Microsoft requested the site remove the credentials, and an investigation was launched to “determine the impact to customers”. (NZ Herald, 2009)

The password policy program identified the password length as 6 to 16 characters long. There were no non-latin characters and all of the symbols used were relatively common, suggesting that character filtering was in place. As for character casing, due to the number of passwords being all lowercase and all uppercase, it is improbable that there is character casing enforcement. Since 93% of the passwords were alphanumerical, symbol enforcement would not have been in place and 45% of passwords were only alphabet characters, meaning numbers were not required.

The Wayback Machine does not have an archive of the site before the breach, however, an article describing how to reset a Hotmail password created in October 2009 describes the password policy as a minimum of six characters. The website also states that the max character limit is 16, as Hotmail will ignore anything above this limit (Free Email Tutorials, 2009). According to this, the password policy program correctly identified the password policy.

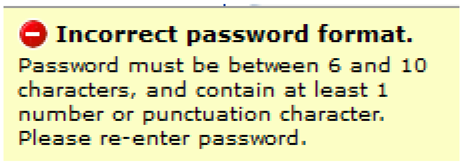
Hotmail did not make any password policy changes following the breach and instead opted to contact customers whose leaked credentials. The lack of action is concerning as it gives potential malicious actors a more significant opportunity to attempt to use the stolen credentials. They instead advised users to change their account passwords as soon as possible.

4.1.5 MySpace

MySpaces credentials were obtained initially in June 2013 but were sold on the dark web in May 2016. A security researcher bought this database and put it up to download for free. The original database contained the credentials of 360 million users; however, the password list used for this project is a stripped list only containing unique passwords.

The MySpace dump had a password length of between 1 and 6341, with the majority of passwords being 8 characters long. There were a couple of control characters which could be due to the way the password list was parsed as the rest of the symbols are relatively common symbols. With the percentage of all lowercase and uppercase passwords, the password policy is unlikely to enforce character casing. It is possible that numbers were not required due to the number of the alphabet only passwords, and it is highly likely that symbols were not enforced due to a large number of alphanumerical passwords.

The issue with this list is that the passwords date back to accounts created in 2008, and MySpace has changed its password policy over the years. For example, in late 2008, the password policy required passwords to be between 6 and 10 characters and contain a number or a punctuation character (See Figure 14 - Late 2008 MySpace password policy). Interestingly, when accessing the site in early 2008, it does not seem to have any enforcement, with the site correctly redirecting when entering the password '1'. This would explain the large number of outlying characters and differences regarding password security.



Incorrect password format.
Password must be between 6 and 10 characters, and contain at least 1 number or punctuation character. Please re-enter password.

Figure 14 - Late 2008 MySpace password policy

In May of 2016, Time Inc (the company that owned MySpace) confirmed that the site had been breached. They noted that they contacted all affected users and invalidated all affected users. They also noted they were monitoring any suspicious activity in case anyone tried to use these credentials. At the current time, their password policy has been updated so that one lowercase, one uppercase, one number and one special character is now required. (MySpace, n.d.)

4.1.6 General Overview

Overall, the password policy program was successful, except for outlying passwords within the used password lists. The returned password policy was accurate. The password policy was accurately predicted in the one entirely accurate dataset (Hotmail). These results further show that the main issues came from the password lists rather than the python program. The password statistics were particularly interesting as there was a common theme among all the password lists, such as all lowercase passwords and the minimum numbers, symbols, or length required. Common phrases were found in the lists such as 'password', 'abc123' and personal information such as names of people or companies.

The companies did not change their password policies in direct response to a password breach in the examples above. In the cases that the password policies are different now, this was usually done years after the initial breach. Instead, most companies opted to reset the passwords of users with stolen credentials and make them aware of safer password practices, such as how to create a secure password and enable 2FA.

4.2 Conclusions

This project aimed to create a program that analysed leaked password lists to accurately predict the password policy enforced during a breach. The Wayback machine and several external articles were used to confirm the actual password policy at the time of the breach. The companys response to the breach was also discussed, including if passwords were reset or if any tips were communicated regarding secure password usage.

Overall, the program was successful, as the only issues came from outlying passwords within the password lists used to test the program. The Hotmail password list was the most accurate, and the program accurately guessed the password policy for that site. The Hotmail password list proves the accuracy of the program when using the correct password list.

Interestingly, most companies did not change their password policies in response to a breach and instead opted for resetting leaked credentials, promoting secure passwords and promoting the use of 2FA.

Regarding password statistics, the most common passwords always met the minimum of the password policy requirements. All are lowercase, only including the minimum number of symbols or numbers and being the smallest length possible. There were also several prevalent phrases such as ‘password’, ‘abc123’ or personal information such as names or locations.

4.3 Future Work

Further research could be undertaken to improve the accuracy of the program regarding false data, so that the password policy can accurately be identified even if there are a couple of outlying passwords. Additionally, more statistics could be generated at runtime to further analyse the passwords within the list, such as the most common dictionary words.

To further prove the current accuracy of the program, more accurate password lists could be produced, which have been more precisely cleaned up. Furthermore, more recent password breaches could be used to investigate companies’ responses to password breaches more effectively.

Finally, given more storage space, more comprehensive password lists could be used to investigate re-occurring password statistics further, as this was a limiting factor for the author when wanting to investigate the ‘rockyou’ password list.

References

000webhost, 2015. *000webhost Database Hacked Data Leaked.* [Online]
Available at: https://www.000webhost.com/000webhost-database-hacked-data-leaked
[Accessed 24 May 2022].

000webhost, 2015. *Order Free Web Hosting [Archived].* [Online]
Available at: https://web.archive.org/web/20150107025332/http://www.000webhost.com/order.php
[Accessed 24 May 2022].

FortinetGURU, 2019. *Password Policy.* [Online]
Available at: https://www.fortinetguru.com/2019/04/password-policy/
[Accessed 24 May 2022].

Free Email Tutorials, 2009. *Change Hotmail password.* [Online]
Available at:
http://www.freeemailtutorials.com/windowsLiveHotmail/configureHotmailSettingsOptions/changingYourAccountPassword.php
[Accessed 25 May 2022].

Google, 2014. *Cleaning up after password dumps.* [Online]
Available at: https://security.googleblog.com/2014/09/cleaning-up-after-password-dumps.html
[Accessed 24 May 2022].

LastPass, 2019. *The 3rd Annual Global Password Security Report.* [Online]
Available at: https://www.lastpass.com/-/media/10aa2f653c774e428aa4cc6732734828.pdf
[Accessed 17 May 2022].

MySpace, n.d. *Join & Sign in on Desktop.* [Online]
Available at: https://help.myspace.com/hc/en-us/articles/202546100-Join-Sign-in-on-Desktop
[Accessed 25 May 2022].

NZ Herald, 2009. *Hotmail, msn users hit in 'phishing attack'.* [Online]
Available at: https://www.nzherald.co.nz/technology/hotmail-msn-users-hit-in-phishing-attack/2IHUJOPXF26FNEDFOEYQEYUYDA/
[Accessed 25 May 2022].

Verizon, 2021. *2021-data-breach-investigations-report.pdf.* [Online]
Available at: https://www.verizon.com/business/resources/reports/2021/2021-data-breach-investigations-report.pdf
[Accessed 17 May 2022].

Windsor, C., 2021. *Malicious Actor Discloses FortiGate SSL-VPN Credentials.* [Online]
Available at: https://www.fortinet.com/blog/psirt-blogs/malicious-actor-discloses-fortigate-ssl-vpn-credentials
[Accessed 24 May 2022].

APPENDICES

APPENDIX A – MAIN.PY

```
from Password import Password
from collections import Counter
import textwrap
import os


def report(run):
    os.makedirs(os.path.dirname(run.filename + '/'), exist_ok=True)
    output = open(run.filename + '/Output.txt', 'w')

    output.write('Estimated Password Policy \n')
    output.write('Length: ' + str(len(run.minimum)) + '-' + str(len(run.maximum)) + '\n')
    output.write('None Alphanumeric Symbols: \n')
    wrapper = textwrap.TextWrapper(width=50)
    string = wrapper.fill(text=''.join(map(str, run.symbols)))
    output.write(string + '\n')
    if run.statistics[0] > 0.5 and run.statistics[1] > 0.5:
        output.write('Char Case enforcement unlikely\n')
    else:
        output.write('Char case enforcement likely\n')

    output.write('\nStatistics\n')

    output.write('Shortest Length : ' + str(len(run.minimum)) + '\n')
    output.write(run.minimum + '\n\n')
    output.write('Longest Length : ' + str(len(run.maximum)) + '\n')
    output.write(run.maximum + '\n\n')

    output.write('Most Popular Passwords \n')
    passwords = Counter(run.passwordList)
    for i in range(1, len(passwords.most_common(11))):
        output.write(str(i) + '. ' + passwords.most_common(11)[i-1][0] + ', ' +
str(passwords.most_common(11)[i-1][1]))
        output.write('\n')
    output.write('\n')

    with open(run.filename + '/PasswordsOutput.csv', 'w') as csvOutput:
        csvOutput.write('password, occurrences\n')
        for i in range(1, len(passwords.most_common(101))):
            csvOutput.write(passwords.most_common(101)[i-1][0] + ', ' +
str(passwords.most_common(101)[i-1][1]))
            csvOutput.write('\n')
    output.write('Password Lengths\n')

    for value in sorted(run.passwordLength, key=run.passwordLength.get, reverse=True):
        if round((run.passwordLength[value] / len(run.passwordList) * 100), 2) > 5:
            output.write(str(value) + ' : ' + str(round((run.passwordLength[value] / len(run.passwordList) *
100), 2)) +
                         '%\n')
    output.write('\n')

    with open(run.filename + '/PasswordLenOutput.csv', 'w') as csvOutput:
        csvOutput.write('length, occurrences\n')
        for key, value in run.passwordLength.items():
            csvOutput.write(str(key) + ', ' + str(value) + '\n')

    output.write('Percentages\n')
    output.write('Number of passwords: ' + str(len(run.passwordList)) + '\n')
    output.write('All Uppercase: ' + str(run.statistics[0]) + '%\n')
    output.write('All Lowercase: ' + str(run.statistics[1]) + '%\n')
    output.write('All Alphabet: ' + str(run.statistics[2]) + '%\n')
    output.write('All Alphanumerical: ' + str(run.statistics[3]) + '%\n')
    output.write('\n')

    count = Counter(run.allSymbols)
    for i in range(1, len(count.most_common(11))):
        output.write(str(i) + '. ' + count.most_common(11)[i-1][0] + ' : ' + str(count.most_common(11)[i-
1][1]))
        output.write('\n')

    output.close()


def main():
    print("What's the file name?")
    run = Password(input())

    report(run)

    print('Output: ' + run.filename + 'Output.txt')


main()
```

APPENDIX B – PASSWORD.PY

```
import re
from collections import Counter


class Password:
    def __init__(self, filename):
        self.filename = filename
        self.passwordList, self.passwordLength = self.openList()
        self.minimum, self.maximum = self.findMinMax()
        self.symbols, self.allSymbols, self.statistics = self.stats()

    def openList(self):
        passwordList = []
        passwordLength = {}

        with open(self.filename + '.txt', 'r') as passwordFile:
            for i in passwordFile:
                passwordList.append(i.strip())
                passwordLength[len(i.strip())] = passwordLength.get(len(i.strip()), 0) + 1

        passwordFile.close()

        return passwordList, passwordLength

    def findMinMax(self):
        minimum = self.passwordList[0]
        maximum = self.passwordList[0]

        for password in self.passwordList:
            if len(password) < len(minimum):
                minimum = password
            if len(password) > len(maximum):
                maximum = password

        return minimum, maximum

    def stats(self):
        symbols = []
        allSymbols = []
        statistics = [0, 0, 0, 0]

        for password in self.passwordList:
            symbolQuery = re.findall('[^a-zA-Z0-9]', password)
            # list [lower, upper, alpha, alphanumeric]
            if password.islower(): statistics[0] += 1
            if password.isupper(): statistics[1] += 1
            if password.isalpha(): statistics[2] += 1
            if password.isalnum(): statistics[3] += 1
            for i in symbolQuery:
                allSymbols.append(i)
                if i not in symbols:
                    symbols.append(i)

        for i in range(0, len(statistics)):
            statistics[i] = round(((statistics[i] / len(self.passwordList)) * 100), 2)

        symbols.sort(key=Counter(allSymbols).get, reverse=True)
        return symbols, allSymbols, statistics
```

APPENDIX C – FIXFORTINET.PY

```
file2 = open('fortinet.txt', 'w', encoding="UTF-8", errors="ignore")
with open('fortinet-2021.txt', 'r', encoding="UTF-8", errors="ignore") as file:
    for line in file:
        if len(line.strip().split(':')) == 2:
            if line.strip().split(':')[1] != '':
                file2.write(line.strip().split(':')[1] + '\n')

file2.close()
```

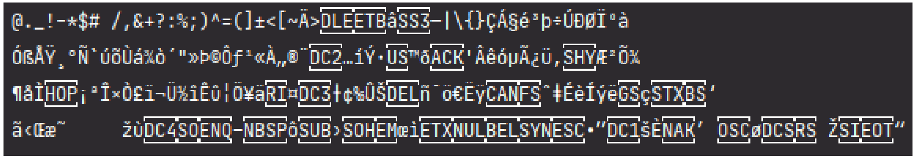
APPENDIX D – FIXLIST.PY (GMAIL)

```
file2 = open('gmailpass.txt', 'w', encoding="UTF-8", errors="ignore")
with open('alleged-gmail-passwords.txt', 'r', encoding="UTF-8", errors="ignore") as file:
    for line in file:
        if line.strip() != '':
            file2.write(line.strip() + '\n')

file2.close()
```

APPENDIX E – 000WEBHOST

E1 – Output.txt

Estimated Password Policy
Length: 1-255
None Alphanumeric Symbols:
Char Case enforcement unlikely



```
@._!-*$# /,&+?:%;)^=(]±<[~Ä>DLEETBâSS3–|\{}ÇÁ§é³þ÷ÚÐØÏºà
ÓßÅŸ¸°Ñ`úõÙá¾ò´"»Þ®Ôƒ¹«À„®¨DC2…íÝ·US™ðACK'ÂêóµÃ¿ü,SHYÆ²Õ¼
¶åÌHOP¡ªÎ×Ò£ï¬Ü½îÊû¦Ö¥äRI¤DC3†¢‰ÛŠDELñ¯ö€ËÿCANFSˆ‡ÉèÍýëGSçSTXBS‘
ã‹Œæ˜      žùDC4SOENQ–NBSPôSUB›SOHEMœìETXNULBELSYNESC•”DC1šÈNAK’ OSCøDCSRS ŽSIEOT“
```

Statistics
Shortest Length : 1

Longest Length : 255
Panel de control (servidor web) El panel de control de un servidor web es un software que provee una interfaz grßfica para la gesti¾n de usuarios y la administraci¾n de los servicios del servidor. Generalmente son en sistemas operativos tipo unix, tal com

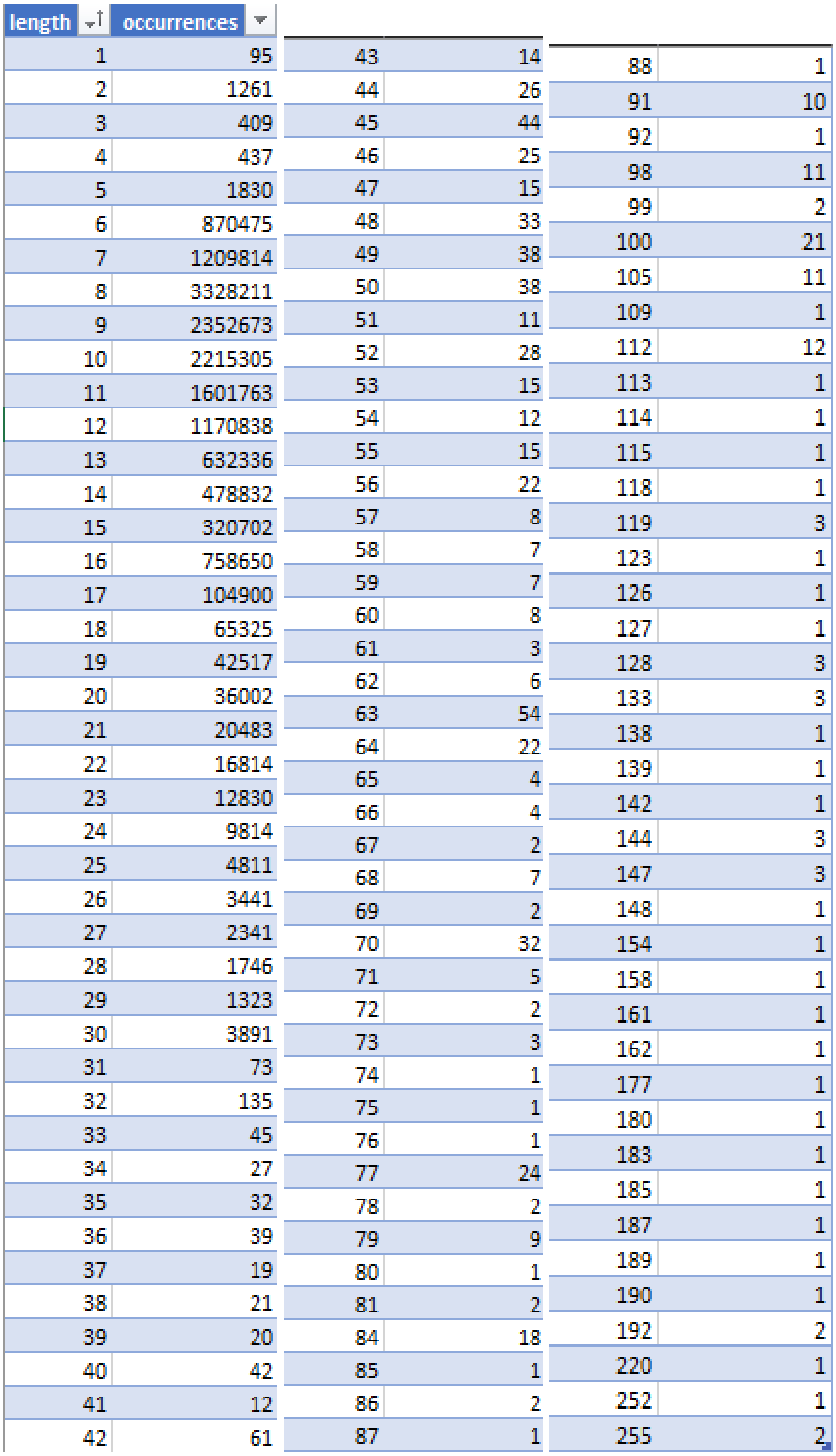
Most Popular Passwords
1. abc123, 24928
2. 123456a, 15115
3. 12qw23we, 12073
4. 123abc, 11294
5. a123456, 10471
6. 123qwe, 10133
7. secret666, 9494
8. YfDbUfNjH10305070, 9295
9. asd123, 9064
10. qwerty123, 8854

Password Lengths
8 : 21.79%
9 : 15.41%
10 : 14.51%
11 : 10.49%
7 : 7.92%
12 : 7.67%
6 : 5.7%

Percentages
Number of passwords: 15271083
All Uppercase: 80.59%
All Lowercase: 1.54%
All Alphabet: 0.27%
All Alphanumerical: 93.01%

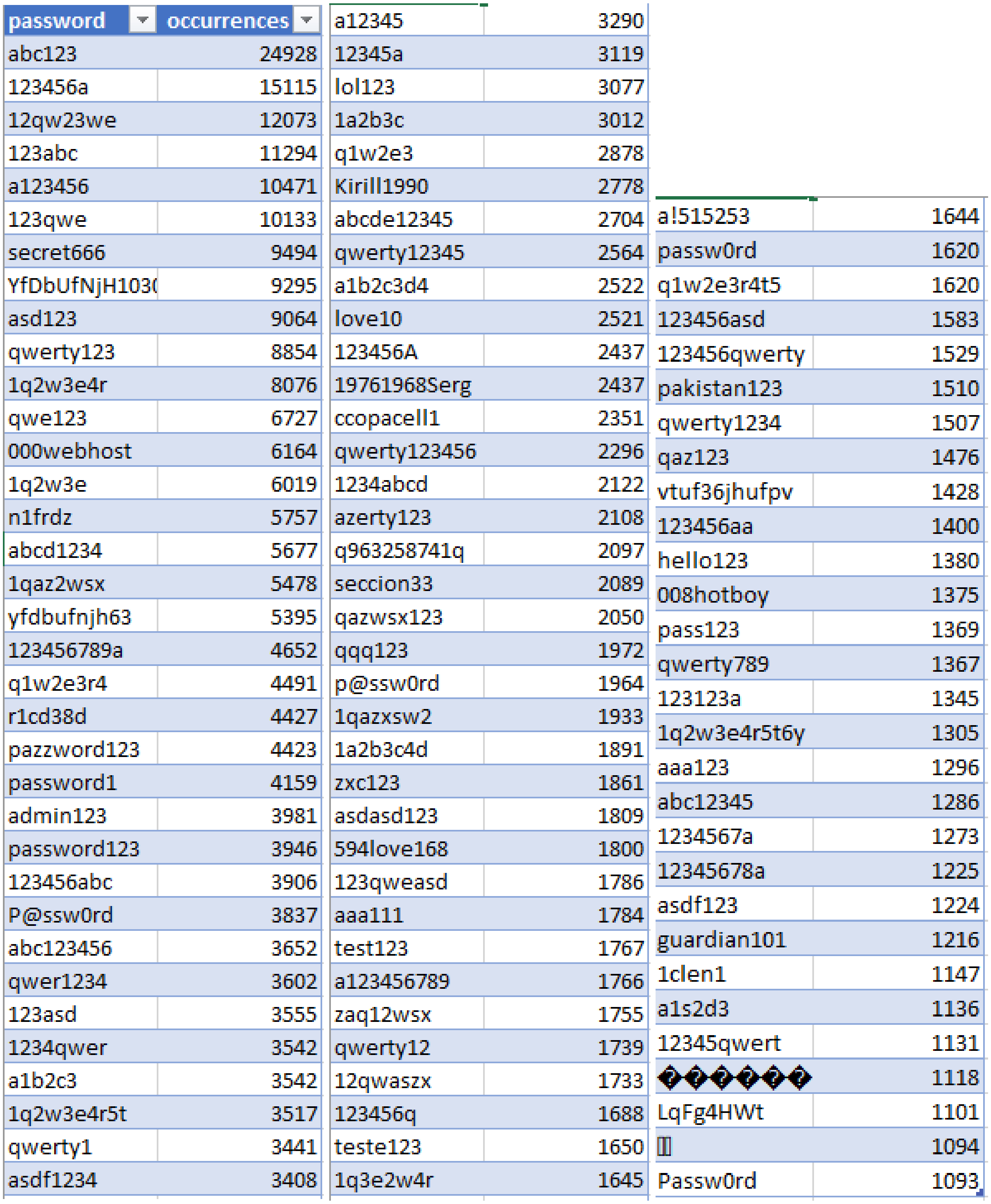
```
1. @ : 340857
2. . : 294474
3. _ : 134297
4. ! : 133406
5. - : 101697
6. * : 90581
7. $ : 64195
8. # : 63218
9.   : 61915
10. / : 33740
```

E2 – PasswordLenOutput.csv



length	occurrences
1	95
2	1261
3	409
4	437
5	1830
6	870475
7	1209814
8	3328211
9	2352673
10	2215305
11	1601763
12	1170838
13	632336
14	478832
15	320702
16	758650
17	104900
18	65325
19	42517
20	36002
21	20483
22	16814
23	12830
24	9814
25	4811
26	3441
27	2341
28	1746
29	1323
30	3891
31	73
32	135
33	45
34	27
35	32
36	39
37	19
38	21
39	20
40	42
41	12
42	61
43	14
44	26
45	44
46	25
47	15
48	33
49	38
50	38
51	11
52	28
53	15
54	12
55	15
56	22
57	8
58	7
59	7
60	8
61	3
62	6
63	54
64	22
65	4
66	4
67	2
68	7
69	2
70	32
71	5
72	2
73	3
74	1
75	1
76	1
77	24
78	2
79	9
80	1
81	2
84	18
85	1
86	2
87	1
88	1
91	10
92	1
98	11
99	2
100	21
105	11
109	1
112	12
113	1
114	1
115	1
118	1
119	3
123	1
126	1
127	1
128	3
133	3
138	1
139	1
142	1
144	3
147	3
148	1
154	1
158	1
161	1
162	1
177	1
180	1
183	1
185	1
187	1
189	1
190	1
192	2
220	1
252	1
255	2

E3 – PasswordsOutput.csv



password	occurrences
abc123	24928
123456a	15115
12qw23we	12073
123abc	11294
a123456	10471
123qwe	10133
secret666	9494
YfDbUfNjH103(	9295
asd123	9064
qwerty123	8854
1q2w3e4r	8076
qwe123	6727
000webhost	6164
1q2w3e	6019
n1frdz	5757
abcd1234	5677
1qaz2wsx	5478
yfdbufnjh63	5395
123456789a	4652
q1w2e3r4	4491
r1cd38d	4427
pazzword123	4423
password1	4159
admin123	3981
password123	3946
123456abc	3906
P@ssw0rd	3837
abc123456	3652
qwer1234	3602
123asd	3555
1234qwer	3542
a1b2c3	3542
1q2w3e4r5t	3517
qwerty1	3441
asdf1234	3408
a12345	3290
12345a	3119
lol123	3077
1a2b3c	3012
q1w2e3	2878
Kirill1990	2778
abcde12345	2704
qwerty12345	2564
a1b2c3d4	2522
love10	2521
123456A	2437
19761968Serg	2437
ccopacell1	2351
qwerty123456	2296
1234abcd	2122
azerty123	2108
q963258741q	2097
seccion33	2089
qazwsx123	2050
qqq123	1972
p@ssw0rd	1964
1qazxsw2	1933
1a2b3c4d	1891
zxc123	1861
asdasd123	1809
594love168	1800
123qweasd	1786
aaa111	1784
test123	1767
a123456789	1766
zaq12wsx	1755
qwerty12	1739
12qwaszx	1733
123456q	1688
teste123	1650
1q3e2w4r	1645
a!515253	1644
passw0rd	1620
q1w2e3r4t5	1620
123456asd	1583
123456qwerty	1529
pakistan123	1510
qwerty1234	1507
qaz123	1476
vtuf36jhufpv	1428
123456aa	1400
hello123	1380
008hotboy	1375
pass123	1369
qwerty789	1367
123123a	1345
1q2w3e4r5t6y	1305
aaa123	1296
abc12345	1286
1234567a	1273
12345678a	1225
asdf123	1224
guardian101	1216
1clen1	1147
a1s2d3	1136
12345qwert	1131
��������	1118
LqFg4HWt	1101
▯	1094
Passw0rd	1093

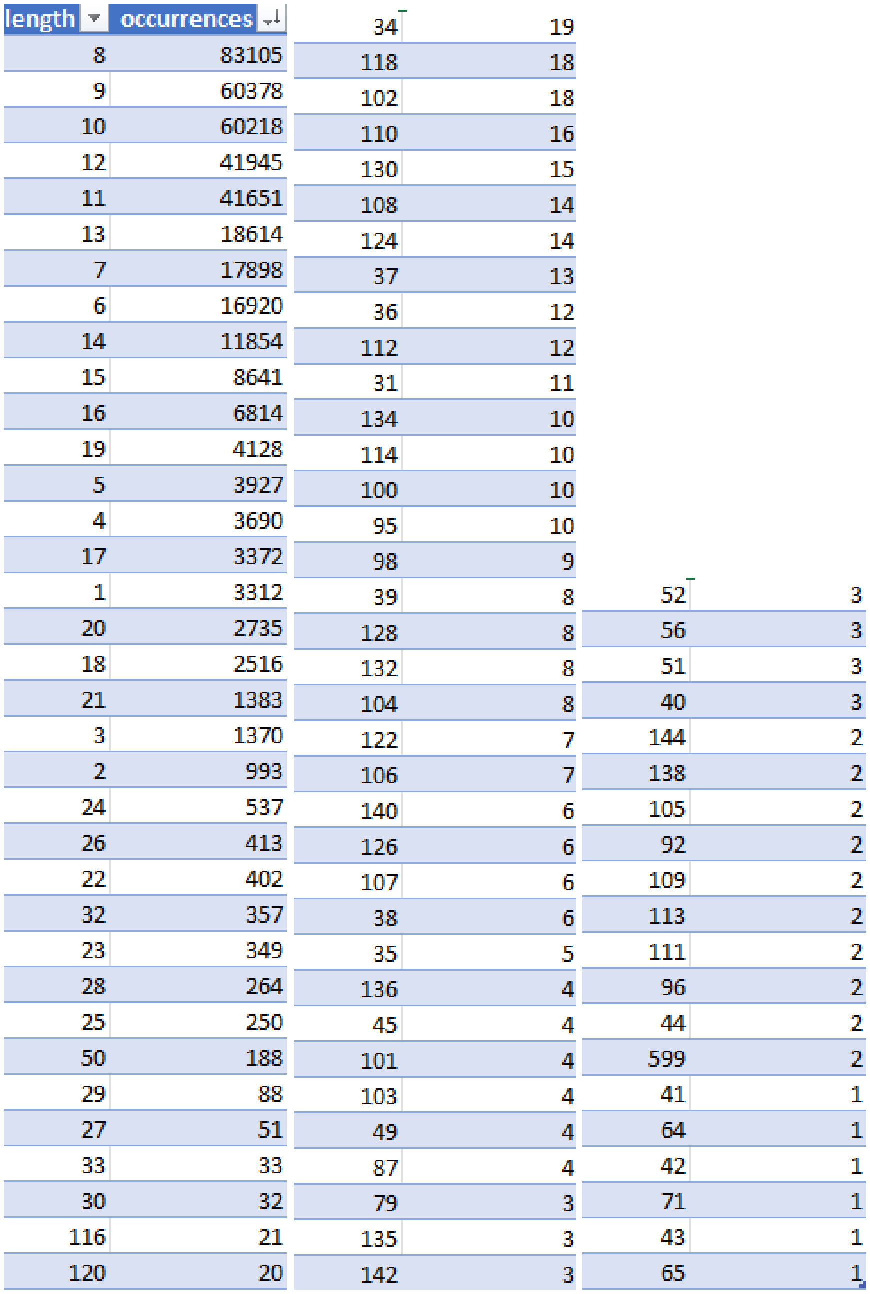
APPENDIX F – FORTINET

F1 – Output.txt

```
Estimated Password Policy
Length: 1-599
None Alphanumeric Symbols:
@!*$-#._ %+&=,?/^(){\;[~]">}<'`|
Char Case enforcement unlikely

Statistics
Shortest Length : 1
b

Longest Length : 599
01 00 00 00 d0 8c 9d df 01 15 d1 11 8c 7a 00 c0 4f c2 97 eb 01 00 00 00 0c 58 13 5d 28 8d ae 48 af 75
c0 5c 10 07 7c 8d 00 00 00 00 18 00 00 00 57 00 49 00 4e 00 49 00 4e 00 45 00 54 00 43 00 72 00 65
00 64 00 00 00 03 66 00 00 a8 00 00 00 10 00 00 00 ec c2 03 64 c5 05 31 88 55 ce 02 22 f2 f3 f9 76 00
00 00 00 04 80 00 00 a0 00 00 00 10 00 00 00 24 81 bf 59 c6 47 b0 a9 52 15 6e d1 12 bc ee 14 28 00 00
00 a7 64 9e 75 28 c5 00 9f 47 9d 6a e0 1e 73 d4 7c 11 02 9f b2 8a da a9 3f 11 4b ba ba 93 bd 35 59 5c
b3 6b 32 96 62 ea 84 14 00 00 00 ad 1b a1 c5 5b c0 10 8f 74 5a dc 50 21 5d 12 da e0 e0 bd 51

Most Popular Passwords
1. Temporal2020, 2328
2. asdf123, 1560
3. 123456, 1524
4. pass@123, 1156
5. Juzgado2020, 1010
6. _, 911
7. SSLVPN-LAN, 760
8. U-SG-SSL-General_User, 725
9. Test1234, 720
10. Reset123, 704

Password Lengths
8 : 20.84%
9 : 15.14%
10 : 15.1%
12 : 10.52%
11 : 10.44%

Percentages
Number of passwords: 398826
All Uppercase: 25.95%
All Lowercase: 4.94%
All Alphabet: 4.67%
All Alphanumerical: 46.74%
```

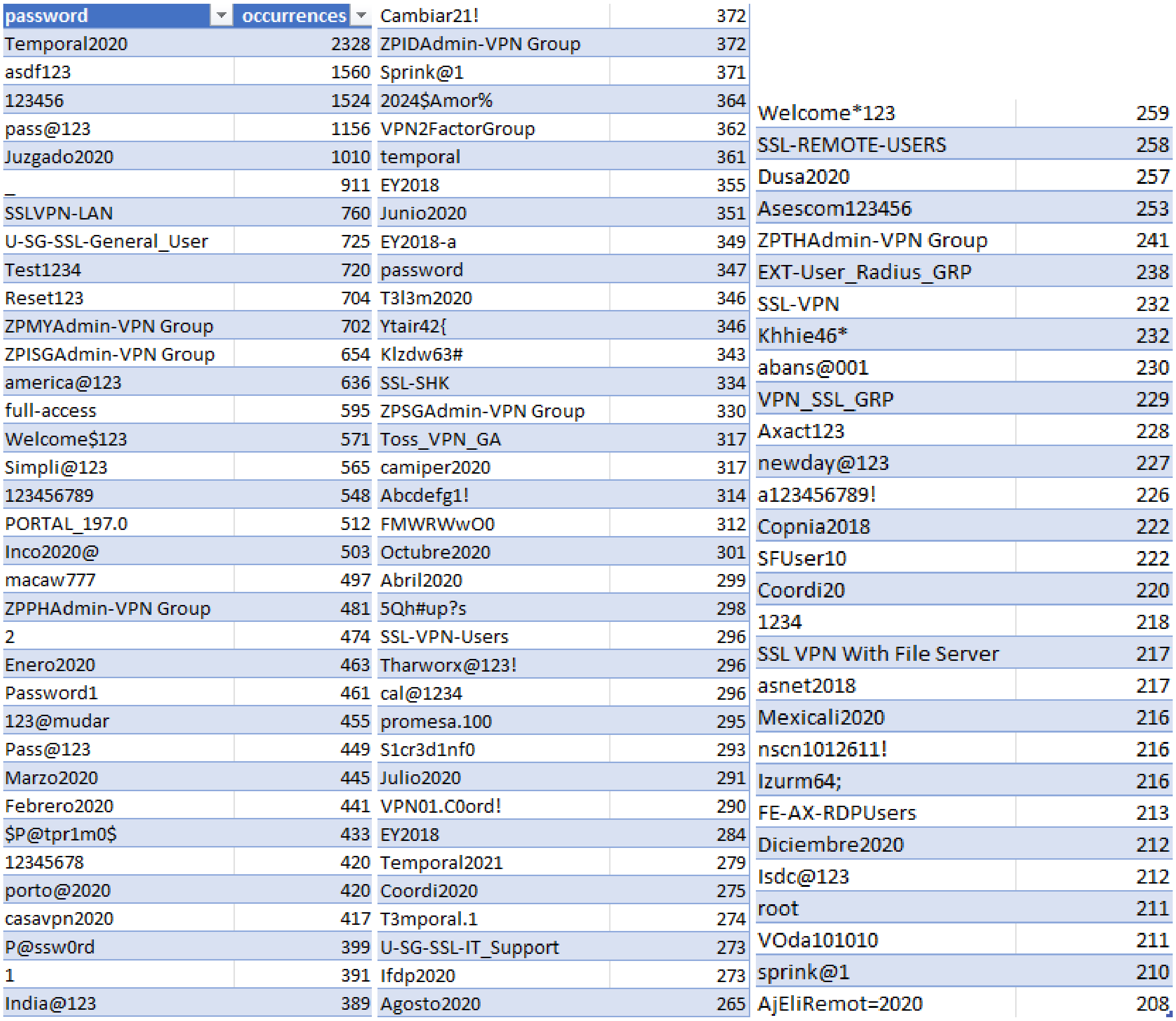
```
1. @ : 81765
2. ! : 40584
3. * : 27436
4. $ : 26065
5. - : 24555
6. # : 23970
7. . : 23657
8. _ : 19799
9.   : 13052
10. % : 9204
```

F2 – PasswordLenOutput.csv



length	occurrences
8	83105
9	60378
10	60218
12	41945
11	41651
13	18614
7	17898
6	16920
14	11854
15	8641
16	6814
19	4128
5	3927
4	3690
17	3372
1	3312
20	2735
18	2516
21	1383
3	1370
2	993
24	537
26	413
22	402
32	357
23	349
28	264
25	250
50	188
29	88
27	51
33	33
30	32
116	21
120	20
34	19
118	18
102	18
110	16
130	15
108	14
124	14
37	13
36	12
112	12
31	11
134	10
114	10
100	10
95	10
98	9
39	8
128	8
132	8
104	8
122	7
106	7
140	6
126	6
107	6
38	6
35	5
136	4
45	4
101	4
103	4
49	4
87	4
79	3
135	3
142	3
52	3
56	3
51	3
40	3
144	2
138	2
105	2
92	2
109	2
113	2
111	2
96	2
44	2
599	2
41	1
64	1
42	1
71	1
43	1
65	1

F3 – PasswordsOutput.csv



password	occurrences
Temporal2020	2328
asdf123	1560
123456	1524
pass@123	1156
Juzgado2020	1010
_	911
SSLVPN-LAN	760
U-SG-SSL-General_User	725
Test1234	720
Reset123	704
ZPMYAdmin-VPN Group	702
ZPISGAdmin-VPN Group	654
america@123	636
full-access	595
Welcome$123	571
Simpli@123	565
123456789	548
PORTAL_197.0	512
Inco2020@	503
macaw777	497
ZPPHAdmin-VPN Group	481
2	474
Enero2020	463
Password1	461
123@mudar	455
Pass@123	449
Marzo2020	445
Febrero2020	441
$P@tpr1m0$	433
12345678	420
porto@2020	420
casavpn2020	417
P@ssw0rd	399
1	391
India@123	389
Cambiar21!	372
ZPIDAdmin-VPN Group	372
Sprink@1	371
2024$Amor%	364
VPN2FactorGroup	362
temporal	361
EY2018	355
Junio2020	351
EY2018-a	349
password	347
T3l3m2020	346
Ytair42{	346
Klzdw63#	343
SSL-SHK	334
ZPSGAdmin-VPN Group	330
Toss_VPN_GA	317
camiper2020	317
Abcdefg1!	314
FMWRWwO0	312
Octubre2020	301
Abril2020	299
5Qh#up?s	298
SSL-VPN-Users	296
Tharworx@123!	296
cal@1234	296
promesa.100	295
S1cr3d1nf0	293
Julio2020	291
VPN01.C0ord!	290
EY2018	284
Temporal2021	279
Coordi2020	275
T3mporal.1	274
U-SG-SSL-IT_Support	273
Ifdp2020	273
Agosto2020	265
Welcome*123	259
SSL-REMOTE-USERS	258
Dusa2020	257
Asescom123456	253
ZPTHAdmin-VPN Group	241
EXT-User_Radius_GRP	238
SSL-VPN	232
Khhie46*	232
abans@001	230
VPN_SSL_GRP	229
Axact123	228
newday@123	227
a123456789!	226
Copnia2018	222
SFUser10	222
Coordi20	220
1234	218
SSL VPN With File Server	217
asnet2018	217
Mexicali2020	216
nscn1012611!	216
Izurm64;	216
FE-AX-RDPUsers	213
Diciembre2020	212
Isdc@123	212
root	211
VOda101010	211
sprink@1	210
AjEliRemot=2020	208

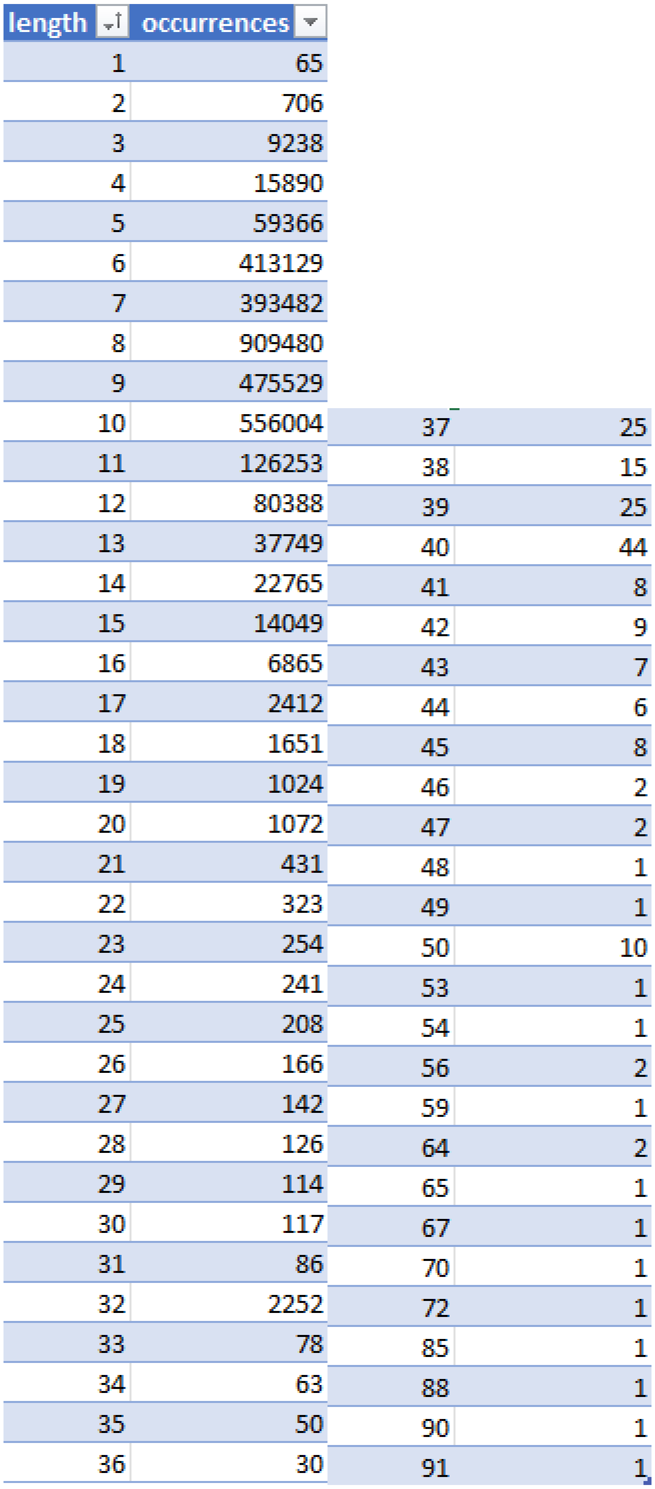
APPENDIX G – GMAIL

G1 – Output.txt

```
Estimated Password Policy
Length: 1-91
None Alphanumeric Symbols:
.@:_-*$#+,&/%)\(='^~<[]`">|{}
Char case enforcement likely

Statistics
Shortest Length : 1
`


Longest Length : 91
gggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggggg
ggg

Most Popular Passwords
1. :, 2
2. .., 2
3. 0, 2
4. 000000, 2
5. 00040, 2
6. 007007, 2
7. 012012, 2
8. 012345, 2
9. 014789, 2
10. 026026, 2

Password Lengths
8 : 29.04%
10 : 17.75%
9 : 15.18%
6 : 13.19%
7 : 12.56%

Percentages
Number of passwords: 3131976
All Uppercase: 84.82%
All Lowercase: 0.0%
All Alphabet: 29.63%
All Alphanumerical: 97.23%

1. . : 23487
2. @ : 22776
```

```
3. : : 14312
4. _ : 8646
5. - : 7606
6. * : 6816
7. $ : 6363
8. # : 6037
9. + : 2777
10. , : 2590
```

G2 – PasswordLenOutput.csv



length	occurrences
1	65
2	706
3	9238
4	15890
5	59366
6	413129
7	393482
8	909480
9	475529
10	556004
11	126253
12	80388
13	37749
14	22765
15	14049
16	6865
17	2412
18	1651
19	1024
20	1072
21	431
22	323
23	254
24	241
25	208
26	166
27	142
28	126
29	114
30	117
31	86
32	2252
33	78
34	63
35	50
36	30
37	25
38	15
39	25
40	44
41	8
42	9
43	7
44	6
45	8
46	2
47	2
48	1
49	1
50	10
53	1
54	1
56	2
59	1
64	2
65	1
67	1
70	1
72	1
85	1
88	1
90	1
91	1

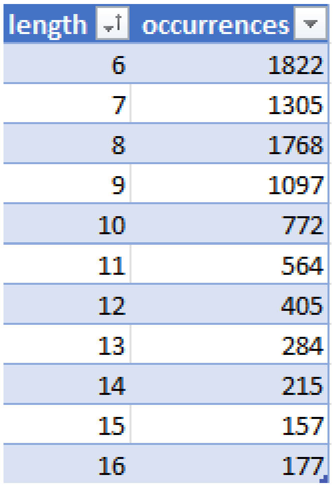
APPENDIX H – HOTMAIL

H1 – Output.txt

```
Estimated Password Policy
Length: 6-16
None Alphanumeric Symbols:
._-+*@/,$"\=!&#?()'%][^} <`>{
Char Case enforcement unlikely

Statistics
Shortest Length : 6
123456

Longest Length : 16
supermariojavith

Most Popular Passwords
1. adidas, 2
2. 407078811amacall, 2
3. 123456, 1
4. 123456789, 1
5. alejandra, 1
6. 111111, 1
7. tequiero, 1
8. alejandro, 1
9. alberto, 1
10. 12345678, 1

Password Lengths
6 : 21.27%
8 : 20.64%
7 : 15.23%
9 : 12.81%
10 : 9.01%
11 : 6.58%

Percentages
Number of passwords: 8566
All Uppercase: 73.44%
All Lowercase: 5.05%
All Alphabet: 45.62%
All Alphanumerical: 93.11%

1. . : 230
2. _ : 113
3. - : 111
4. + : 93
```

5. * : 80
6. @ : 67
7. / : 64
8. , : 32
9. $: 20
10. " : 20

H2 – PasswordLenOutput.csv



length	occurrences
6	1822
7	1305
8	1768
9	1097
10	772
11	564
12	405
13	284
14	215
15	157
16	177

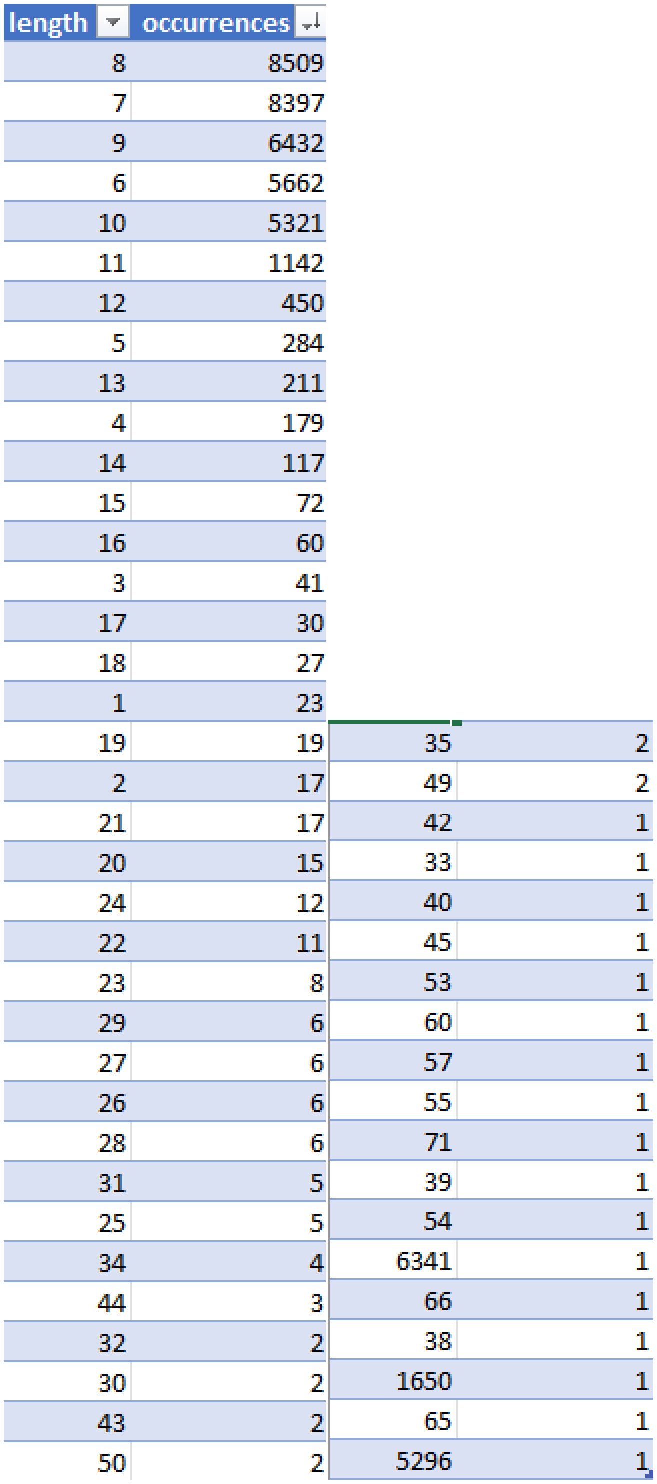
Appendix I – MySpace

I1 – Output.txt

```
Estimated Password Policy
Length: 1-6341
None Alphanumeric Symbols:
#!.*_-@$'?&,;/~=`+)\]([%^>"¬�{}♶����
Char Case enforcement unlikely

Statistics
Shortest Length : 1
)

Longest Length : 6341
{Cannot Display Due to Length}

Most Popular Passwords
1. password1, 1
2. abc123, 1
3. fuckyou, 1
4. monkey1, 1
5. iloveyou1, 1
6. myspace1, 1
7. fuckyou1, 1
8. number1, 1
9. football1, 1
10. nicole1, 1

Password Lengths
8 : 22.92%
7 : 22.62%
9 : 17.32%
6 : 15.25%
10 : 14.33%

Percentages
Number of passwords: 37126
All Uppercase: 92.43%
All Lowercase: 3.54%
All Alphabet: 7.35%
All Alphanumerical: 89.31%

1. # : 5477
2. ! : 1577
3. . : 1006
4. * : 476
5. _ : 309
```

6. - : 302
7. @ : 165
8. $: 123
9. ' : 110
10. ? : 105

I2 – PasswordLenOutput.csv



length	occurrences
8	8509
7	8397
9	6432
6	5662
10	5321
11	1142
12	450
5	284
13	211
4	179
14	117
15	72
16	60
3	41
17	30
18	27
1	23
19	19
2	17
21	17
20	15
24	12
22	11
23	8
29	6
27	6
26	6
28	6
31	5
25	5
34	4
44	3
32	2
30	2
43	2
50	2
35	2
49	2
42	1
33	1
40	1
45	1
53	1
60	1
57	1
55	1
71	1
39	1
54	1
6341	1
66	1
38	1
1650	1
65	1
5296	1