University of Derby

Department of Computing & Mathematics

A project completed as part of the requirements for the BSc (Hons) Computer Forensic Investigation

entitled

Forensic Analysis of the Cloud-Based Password Manager ‘LastPass’ on Windows

by

Luke Michael Acott
l.acott1@unimail.derby.ac.uk
in the years 2012-2016
Abstract

In this project, the popular cloud-based password manager LastPass is analysed and tested to see what artefacts are left behind by the application from usage that could be potentially useful in a digital forensic investigation. An in depth study into why password managers are becoming a necessity in the modern computing landscape was conducted and experiments were designed to create realistic usage data for the application. Several potentially useful forensic artefacts were found and discussed in detail and recommendations were provided on how to obtain useful data from the password manager.
Acknowledgements

I would like to express my gratitude to all family, friends and academics who have helped me through the duration of things project, without their support in terms of advice or just simple motivation this project would not have been fully achievable.

Firstly I would like to thank my parents (Mike and Jane Acott) for the continued motivational support throughout this project

Thank you to my sister (Emily Acott) for taking the time to go through my literature with me several times to ensure that there were no mistakes and that the text flowed nicely.

Thank you to all of my course mates and friends from the University for providing continued support throughout this processes and helping to keep my sanity in check.

Finally I would like to thank Dr Virginia N. L. Franqueira for her supervision throughout this project, without this the project would not be the form that it is in today.
Table of Contents

Abstract ........................................................................................................................................... 1

Acknowledgements ....................................................................................................................... 2

Table of Contents ......................................................................................................................... 3

Table of Figures .......................................................................................................................... 7

Table of Tables ............................................................................................................................... 9

1. Introduction .................................................................................................................................. 10

   1.1. Project Rationale .................................................................................................................. 10

   1.2. Project Aim and Objectives ............................................................................................... 10

   1.3. Project Roadmap .................................................................................................................. 11

2. Literature Review ....................................................................................................................... 12

   2.1. Introduction .......................................................................................................................... 12

   2.2. Password usability ............................................................................................................... 12

      2.2.1. Password memorability ............................................................................................... 14

      2.2.2. Password entropy ......................................................................................................... 15

      2.2.3. Alternative password complexity calculations .......................................................... 16

      2.2.4. Password strength meters .......................................................................................... 17

   2.3. Password Security ............................................................................................................... 19

      2.3.1. Types of password attacks ........................................................................................... 19

      2.3.2. Password expiration policies ....................................................................................... 20

      2.3.3. Two-Factor Authentication ......................................................................................... 20
7.1.1. Aim and Objectives ........................................................................................................56

7.1.2. Improvements and future work ....................................................................................58

8. Bibliography ....................................................................................................................59

9. Appendices ......................................................................................................................65

  9.1. Resources used during testing and forensic analysis ..................................................65

  9.2. Virtual machine baseline setup notes .......................................................................67

  9.3. Simulated usage data creation notes: Experiments 1-4 .............................................87

  9.4. Forensic analysis testing notes ..................................................................................122
Table of Figures

Figure 1: Comparison of forensic process models (Casey, 2004) ........................................... 30

Figure 2: Project Phase Diagram .................................................................................................. 31

Figure 3: Screenshot of the advanced Options screen visible during the LastPass setup on virtual machine 1. ........................................................................................................... 39

Figure 4: Screenshot of the account creation screen for LastPass on virtual machine 1........... 40

Figure 5: Screenshot of the LastPass vault in Google Chrome from virtual machine 1 with the anomaly of the duplicate Reddit account. ......................................................................................... 40

Figure 6: Screenshot of the LastPass password vault on virtual machine two ....................... 41

Figure 7: Screenshot of the LastPass vault running in offline mode after virtual machine two had been restored to a previous snapshot. ......................................................................................... 42

Figure 8: Screenshot of test credit card information in the LastPass vault on virtual machine one while disconnected from the internet ......................................................................................... 42

Figure 9: Screenshot of the shared folder within the LastPass vault on Virtual Machine 1 ....... 43

Figure 10: Screenshot of the shared Reddit account in the LastPass vault on virtual machine 3 while the network was disconnected. ......................................................................................... 44

Figure 11: Screenshot of LastPass history within the LastPass vault on virtual machine 1..... 44

Figure 12: Screenshot from Encase 7 showing options used to process each forensic image. 45

Figure 13: Screenshot of the History files for Internet Explorer on virtual machine 1 in Encase 7 ......................................................................................................................................... 46

Figure 14: Screenshot of the LastPass vault tutorial from virtual machine 1 in Encase 7 ...... 46

Figure 15: Screenshot of the installed applications on virtual machine 1, as found by the System Info Parser option in Encase 7 ........................................................................................................... 47
Figure 16: Screenshot of the SQLite database for the LastPass Google Chrome extension on virtual machine 1 viewed using the transcript window in Encase 7...

Figure 17: Screenshot of the LastPassData table within the SQLite database found on virtual machine 1, in OS Forensics 3.3...

Figure 18: Screenshot of the LastPassData table within the SQLite database found on virtual machine 2, in OS Forensics 3.3...

Figure 19: Screenshot of the LastPassPreferences table within the SQLite database found on virtual machine 1, in OS Forensics 3.3...

Figure 20: Screenshot of the LastPassSavedLogins table within the SQLite database found on virtual machine 1, in OS Forensics 3.3...

Figure 21: Screenshot of the LastPassSavedLogins2 table within the SQLite database found on virtual machine 1, in OS Forensics 3.3...

Figure 22: Screenshot of the LastPassSavedLogins2 table within the SQLite database found on virtual machine 2, in OS Forensics 3.3...

Figure 23: Screenshot of the LastPassData table within the SQLite database found on virtual machine 3, in OS Forensics 3.3...

Figure 24: Screenshot of the LastPassSavedLogins2 table within the SQLite database found on virtual machine 2, in OS Forensics 3.3...

Figure 25: Screenshot of keyword search result for “100243363” from virtual machine 1 in Encase 7...

Figure 26: Screenshot of keyword search result for “testingpassword” across all virtual machines in Encase 7...
Table of Tables

Table 1: Screenshots of the different password strength meters used by multiple popular websites, with the password “Password1” entered to show password advice………………………….17

Table 2: Specifications of Baseline virtual machine to be used during testing…………………………33

Table 3: Experiment 1 tests and expected results.........................................................................34

Table 4: Experiment 2 tests and expected results.........................................................................35

Table 5: Experiment 3 tests and expected results.........................................................................36

Table 6: Experiment 4 tests and expected results.........................................................................37
1. Introduction

1.1. Project Rationale

Website and application developers are constantly increasing the security requirements for user passwords (Fahl et al., 2013). Users often have multiple accounts across different online services and should be using strong, unique passwords to ensure their data is kept secure. To tackle the problem of remembering these complex passwords many users are starting to use password managers built into popular web-browsers such as Google Chrome and Firefox. However these built-in password managers have been shown to have significant security vulnerabilities (Zhao and Yue, 2014). To address this third party applications such as LastPass have been created (LastPass, 2015). While security research has been done into third-party password managers (Zhao, Yue, and Sun, 2014) there is not currently sufficient research specifically on the forensic analysis of the password managers which is why the project will focus on this area.

LastPass is an increasingly popular password manager (Henry, 2015) that is being recommended and used by various large organisations (Roskelley and Chianis, 2013). However the data security of LastPass has recently come into question due to a security outbreak in which data was taken from LastPass servers (Siegrist, 2015). One of the main features that makes cloud-based password managers, like LastPass, so popular is account synchronisation, which allows users to easily access their accounts from multiple devices.

1.2. Project Aim and Objectives

The aim for the project is to perform forensic analysis on the cloud-based password manager LastPass and evaluate the forensic value of artefacts left on the local machine when using selected features.

The aim will be met by achieving the following objectives:

1. Determine the current state of research being performed on password managers from both a forensics and a security standpoint.

2. Design experiments to create usage data for the forensic analysis of cloud-based password managers
3. Capture data using an image of the output from the experiments in the form of forensic image files.

4. Examine the output using the selected forensic analysis methodology to see what artefacts are left behind.

5. Analyse the final results obtained in terms of potential use in Forensic Investigations.

1.3. **Project Roadmap**

This Chapter provides an introduction to the project and the initial aims and objectives that this project sets to achieve. Below is a brief description on all chapters to outline the structure of the project.

Chapter 2 will be a literature review of current research being performed on Password Managers, both from a security and a forensics standpoint. It will also provide an in depth study to the various password manager features that will be tested along with research as to why password managers are becoming more important as the current usability and security of password authentication is assessed and compared with alternatives.

Chapter 3 will evaluate and select a methodology to be used for the design and implementation phases of the project, and will also evaluate and select a methodology to be used for the forensic analysis. It will also go into detail on how the methodologies will be applied to the different sections – Design & Implementation, and the Forensic Analysis stages.

Chapter 4 will show detail on how the experiments used to create usage data for LastPass will be created, and how the subsequent forensic analysis will be conducted on the data collected.

Chapter 5 will present and analyse the results of the experiments and forensic analysis, showing findings of forensic artefacts.

Chapter 6 will discuss the implications of the artefacts found, showcasing how useful they would be in a forensic investigation.

Finally Chapter 7 will evaluate how successful the project has been by individually assessing how the aim and objectives of the project has been met and will then provide recommendations for improvements and potential work once the project has finished.
2. Literature Review

2.1. Introduction

Computer users generally have multiple accounts for different web applications such as email, social media and online accounts. The majority of these services require a password for authentication. Passwords have been the primary choice of human-computer authentication since the 1960s and have been identified as a weak point in security since the 1970s (Bonneau, 2012). Users are strongly advised to have unique passwords for these services. In most cases each individual service has a slightly different set of password requirements based on the guidelines listed in SP 800-118. Multiple applications are available to assist with the task of remembering and creating these passwords, such as password generators and password managers. Most modern web-browsers contain a form of password manager, allowing users to save their password to make access to services easier.

This literature review will look into issues with password usability and memorability, specifically highlighting the recent trends on increasingly complex password requirements and the deployment of different entropy checks and password strength meters on forms. Then the current trends with password security in general will be researched including a brief review of popular password policies and different methods that are being deployed to enhance password security such as two-step authentication. Finally, various different types of password managers will be compared from a feature and security standpoint.

2.2. Password usability

There have been multiple studies into the usability of passwords for end-users with case studies agreeing that while strong passwords are an inconvenience from user perspective they are an integral part of modern computer security. It is very difficult for a potential replacement for password authentication to make grounds in the industry due to the high cost of change (Jeff et al., 2004; Vu et al., 2007; Inglesant and Sasse, 2010; Boothroyd and Chiasson, 2012).

A study into the design of textual password authentication for children was conducted by Read and Cassidy (2012). Young children were questioned on what criteria they thought makes a strong password and were assessed on password creation under their criteria. These children believed that a good password should be simple, easy to remember and hard to guess.
The majority of younger children created passwords between 0-5 characters while majority of older children who also participated in the study created passwords between 6-10 characters. Simplicity was assessed by checking the ratio between the number of characters in whole words in the password against the character length of the password. Results showed that 46% of younger children had 100% simplicity while 48% of older children had 100% simplicity. Memorability was assessed by counting the amount of login attempts required before the children gained access to their system between 15 minutes and an hour after creation. Only 7 children overall were unable to recall their passwords, with 6 of these being young children. Guessability in this context was assessed by measuring the difference between the username and the password, checking to see if the password was guessable by the username provided. 96% of younger children had a rating of 0 while 74% of older children had a rating of 0. This means that the younger children’s passwords were harder to guess.

The recommendations from this study for password requirement design were that the length requirements should be between 4-8 characters with a 4-character minimum for young children. This would mean that these passwords would be fairly easy for a brute force attack to crack and it is suggested that because of this alternative authentication to passwords is recommended for children. A mixture of lower and upper cases as well as numbers or symbols is also not recommended for children to prevent memorability issues. Several warning prompts were also suggested to be used during password creation including where the password contains an incorrect spelling of a word or where the password is too similar to the username (Read and Cassidy, 2012).

Inglesant and Sasse (2010) conducted a case study on the usability of password policies in an organisational environment specifically looking at the cost of unusable password policies to both end users and the organisation itself. The study investigated password use at two major organisations to see how their users were impacted by their password policy. It specifically explored aspects of the password policy that cause problems for users, what strategies are used to cope with these problems, how these coping strategies affect productivity and security and if there are unexpected password issues that are not covered by password policy. The main observation in terms of usability made was that when users cannot cope with the demands of a strict password policy it reduces productivity and can lead to users adopting coping strategies such as writing down their passwords which can drastically reduce security.

In conclusion the study found that the security policies shouldn’t be created with just a focus on maximising password strength alone and instead should be designed with human computer
interface principles to help users create passwords that are appropriately strong for their specific context of use.

The user cost of password policies was also looked at, split into factors that help mitigate the load for users as well as the factors which place load on users. Factors that help mitigate the load for users include the deployment of single sign-on services (see section 2.3.4) and password managers (see section 2.4). Factors that place load on the user include the use of strong passwords in terms of length and character sets, frequent password change and non-reuse of passwords which are shown to only be of marginal benefit to security by Herley (2009). Another important factor to consider when creating password policies is that for users password use is a secondary task that interrupts a user’s primary task of using their computer system (Inglesant and Sasse, 2010).

### 2.2.1. Password memorability

Password memorability is becoming a growing area for concern within the computer security industry. Studies have shown that passwords are more commonly becoming harder to remember and easier for a computer to guess so users are advised to have unique passwords for each account they possess (Bonneau, 2012; Kelley et al., 2012). Each unique password has to meet the security requirements of the intended service. Password strength is generally assessed by the length and complexity, which is determined by the unpredictability of character sets. A keyspace is the total number of possible values that a key (password) could have. This increases as the length and amount of character sets used increases (Scarfone and Souppaya, 2009). However there has been no actual consensus on a standard for measuring password strength leading to different services having different guides and password requirements. Commonly used password requirements have a minimum and maximum length, along with the requirement of one or two character sets such as upper case or a symbol.

Password reuse is also becoming a common occurrence as users acquire more accounts for different services. This is a large security concern since if one account is compromised which uses the same credentials as other services then multiple accounts could be compromised (Kelley et al., 2012). The best way to circumvent this is to ensure that each account for each service has a strong yet unique password. The obvious downside to this is that these unique passwords will be difficult to remember.
Jeff et al. (2004) researched several techniques used to provide advice on password security, specifically on four requirements with different testing groups:

1. Advised to use at least seven characters and a number
2. Advised to choose a completely random password off a sample paper by closing their eyes and pointing to eight separate characters
3. Advised to create a passphrase using 1 letters of each word in the phrase
4. A comparison sample which had no advice given to them

The first and third group were shown to be the least secure with the passphrase group being most secure.

Kelley et al. (2012) analysed the impact that different password polices have on password strength, and found that passwords created under stricter requirements are generally more resistant to automated cracking but were found to be harder to remember and create. This caused other security issues to occur with users often writing down a password instead of memorising it (Vu et al., 2007; Proctor et al., 2002).

On the other hand Boothroyd and Chiasson (2012) explored the implications on the security of users writing down their passwords after creation to see if it improved the strength of the passwords chosen by the users. The study involved two hypotheses: “Users who are encouraged to write down their password will formulate more secure passwords than those users advised against writing down their password” and “Users who wrote down their password will have more successful logins than those users who did not write down their password”. Conversely their results showed that users who write down their password did not always have stronger passwords, and that passwords with higher entropy had the least amount of successful logins due to the complexity of the passwords.

2.2.2. Password entropy

Password entropy is a commonly used method for assessing how secure a password is. There are several different types of entropy checks, the most popular of these being the original: Shannon’s Entropy (Shannon, 1948). It works by using an algorithm to show how easy a password would be to guess based from the characters it is made up from. The National Institute of Standards and Technology (NIST) definition of entropy is the following: “Entropy in an information system is the measure of the disorder or randomness in the system.”
standard then states “Passwords that do not have sufficient entropy are more likely to be recovered through brute force attacks.” (Scarfone and Souppaya, 2009).

However, Kelley et al. has shown Shannon’s Entropy (Shannon, 1948) to be an inadequate way of assessing password security since the design of the algorithm is not at all appropriate for use of assessing password strength. Whilst it is useful at providing a statistic metric of password strength, it only provides a rough correlation with how guess resistant a password is and is unable to correctly predict differences in guessability between password sets (Kelley et al., 2012).

Wanili et al. (2010) argues that the problem with entropy is that it uses a Markov process to attempt to guess, or list how many guesses would be required to “crack” a password. Password guessing is not a Markov process; it implies that through guesswork the password can be found by finding the starting characters and forming patterns, e.g. if the first letter is found to be ‘t’ the second letter is likely to be ‘h’ and the third ‘e’. Individual characters on passwords cannot be confirmed this way – the only way to tell if the characters are correct is to have the password in full, meaning it’s an all or nothing method and that it is meaningless to calculate password entropy based on the characters that make up the password.

2.2.3. Alternative password complexity calculations

Research has shown that there are several alternatives suggested for use instead of or alongside password entropy checks. Wanli et al. (2010) suggests using a new method called a password quality indicator (PQI) which give users a better idea as to how secure the password they are creating actually is. This method assumes that the quality of a password is assessed on how long it would take to crack a password. This is determined by calculating the editing distance of the password to a base dictionary word as well as what the passwords effective length is which can be calculated by how many different character sets are being used.

Taha et al. (2013) suggests that neither PQI nor an entropy meter alone are sufficient in assessing the strength of a password. PQI, dictionary based checking and entropy checks were tested against each other, with the conclusion that all three should be incorporated using a proposed algorithm to resolve the consistency issue with password advice.
2.2.4. Password strength meters

Another method of helping to inform users of how strong a password is at creation is by deploying a password strength meter. High traffic websites including Facebook, Google, Yahoo and Dropbox use a password strength meter. However the password strength meter algorithms often vary between different services to the extent that some allow weak passwords such as ‘Password1’ whereas other services flag this as insecure as shown in Table 1. These differences are not helpful for users as it could end up giving potentially insecure and conflicting advice.

<table>
<thead>
<tr>
<th>Screenshot of Password Strength Meter</th>
<th>Website Name, URL and Steps taken to get to the Password Strength Meter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook.com: navigate to <a href="https://www.facebook.com/settings?tab=account&amp;amp;section=password&amp;amp;view">https://www.facebook.com/settings?tab=account&amp;amp;section=password&amp;amp;view</a></td>
<td>Facebook.com: navigate to <a href="https://www.facebook.com/settings?tab=account&amp;amp;section=password&amp;amp;view">https://www.facebook.com/settings?tab=account&amp;amp;section=password&amp;amp;view</a></td>
</tr>
<tr>
<td>Yahoo.com: navigate to <a href="https://uk.mail.yahoo.com/">https://uk.mail.yahoo.com/</a> click ‘Sign Up’</td>
<td>Yahoo.com: navigate to <a href="https://uk.mail.yahoo.com/">https://uk.mail.yahoo.com/</a> click ‘Sign Up’</td>
</tr>
</tbody>
</table>

Table 1: Screenshots of the different password strength meters used by multiple popular websites, with the password “Password1” entered to show password advice.

Carnavalet and Mannan (2015) conducted extensive research into different password strength meters used by popular websites and password managers over a period of 18 months. Advice was sent to the service providers during this period on how to improve their password strength meters to provide consistent results, and then the password strength meters were re-tested. Over 9.5 million passwords were tried on the strength meters from 13 different dictionaries including that of the LinkedIn password leak (Whittaker, 2012). Only Dropbox and Keypass explained the design and the logic behind their password strength meters.
Password managers are an alternative to users having to remember multiple passwords for multiple accounts. Different types of password managers are outlined in section 2.4. Four password manager password strength meters were also analysed: 1Password, LastPass, KeePass, and RoboForm. It is even more important that these password strength meters are accurate in their security assessment since they are used to create master passwords for the password vaults, which store the passwords for all the user’s other services. From this it was alarming to find that the LastPass password strength meter assessed the password ‘Password1’ at around 70%, seeing this as a good secure password even though it is one of the most popular passwords. The reason for this is that the password strength meter did not take dictionary attacks into account. It just looked at the assessment of if the password contains the username in which points were deducted, the password length and the variation of different character sets used (Carnavalet and Mannan, 2015).

Kim et al. (2014) developed a web-browser extension named ‘YourPassword’ that provides feedback to users on their passwords for multiple accounts. This helps users to create different passwords for each account by comparing password similarity and reducing the security score on similar passwords. The goal was to help encourage users to create unique and strong passwords for different websites. The password strength meter adjusted score according to individual password strength, password re-use, website sensitivity, password encryption on transmission and password similarity.

To test ‘YourPassword’ users were given a list of passwords that needed to be changed to be more secure for several accounts. Some of the passwords were either repeated or very similar to those used on other accounts. The testing had three main groups: a control group that received no advice, a group that used Microsoft Password Checker to evaluate password strength and finally a group that used ‘YourPassword’. Results found that while the passwords created by the control group and the Microsoft Password Checker group were more secure they were not unique, with the same password often being used on multiple accounts. The ‘YourPassword’ group however had uniquely individual passwords created for each account (Kim et al., 2014). While the passwords generated might be more secure, it is worth noting that there are still usability issues due to the burden of having to remember multiple unique passwords as mentioned in section 2.2.1.
2.3. **Password Security**

The security of using passwords for authentication has been well discussed in the computing industry. While it is agreed that a long, strong password using multiple character sets can be considered secure, there are still critical vulnerabilities to using password authentication not including the previously discussed usability issues that strong passwords present.

2.3.1. **Types of password attacks**

One types of attack on password authentication is a brute force attack. Brute force attacks involve an adversary attempting to gain access to a system by entering every possible combination of password into the password field until access is granted. The more complex a password is, the more attempts needed before an adversary could gain access. One method to try and block this type of attack is by locking user accounts after a certain amount of guesses have been made which prevents ‘Online’ brute force attacks. ‘Offline’ attacks involve an adversary trying the different combinations on an offline dataset, such as stolen password database from a service, for example the LinkedIn password leak of 2012 (BBC, 2012). These datasets are normally encrypted, however they could be eventually broken by a powerful computer after a long period of time depending on how exactly the passwords are stored.

How secure a password is against a brute force attack depends entirely on the encryption or hashing algorithm used by the service that stores the authentication information. Ducklin (2013) overviewed two different algorithms used for encrypting passwords used to secure Zip files. One method of encryption, known as PBKDF2 was used by Posey (2013) to advise that an 8-character password is reasonably secure with an estimate of 5 years would be needed to break the password. However other algorithms were not taken into account during this test such as PKZIP which Ducklin showed that the password could have been cracked in around two and a half hours. Basically the lesson learned from this is that unless a service provides detailed information on the algorithm used to secure a password, and an end user actually understands this information it is best to assume the worst and choose an overly secure password, a 14-character password using a mixture of cases including digits and symbols was recommended for use by Ducklin to try to circumvent brute force attacks.

Another type of attack on password authentication is a dictionary attack which involves an adversary systematically entering different words or alterations of words from a dictionary as a password to attempt to gain access to an account. This method can be circumvented if
passwords either contain no dictionary words at all, or are heavily edited with substituted characters from different character sets to increase the time it would take for the password to be cracked. Dictionaries come in different forms, including previously leaked or cracked password database or general language dictionaries (Rouse, 2005).

2.3.2. Password expiration policies

Password expiration is one method that is commonly used by system administrators to try and keep password authentication secure (Scarfone and Suratose, 2015). This policy involves user passwords expiring after a set period of time, and prevents the re-use of these passwords for either a fixed amount of new passwords e.g. the previous password cannot be used for the next six passwords, or a set period of time.

However Zhang, Monrose, and Reiter (2010) conducted a large scale study on the success of password expiration at meeting its intended purpose. The paper investigated the concept of putting previously used passwords generated with a password expiration policy through an algorithm to guess or crack a user’s next password. It found that on average 41% of new passwords could be cracked offline in an offline attack with an expected effort of under 3 seconds from a 2.67GHz processor using the algorithm. 17% of accounts on average could be broken in an online attack with fewer than five guesses, with the assumption that after five guesses the user account would be locked.

2.3.3. Two-Factor Authentication

Authentication can involve three factors: 1) something a user knows, 2) something a user has, 3) something a user is. Password authentication uses factor one (Scarfone and Souppaya, 2009). To help enhance the security of password authentication many services now offer two-factor authentication in the form of factor one and two by using a token which displays an additional passcode that expires after a certain time period (Google, 2016a). This is typically entered after a user has input their password but this may change from system to system. This token is either a physical hardware device that a user must keep on their person to authenticate, or a software token which can be deployed to a mobile phone or laptop (RSA Security, 2016).

The reason this is more secure than standard password authentication is because in order to access a user account an adversary would need to know both the targeted users password and have access to the token application or physical device.
2.3.4. Single sign-on

Single sign-on (SSO) is a form of authentication which allows a user to sign in using one set of credentials to multiple websites or services. There are two main types of SSO currently in use: 1) Enterprise systems - involves entering organisational credentials to access multiple services used internally at an organisation. Typically this is seamless, meaning that the services can be accessed immediately after a user has entered the credentials to log into a work station, but occasionally it means that the user has to re-enter their credentials on the service portal depending on implementation. 2) Consumer systems – organisations such as Facebook have tools to allow their credentials to be used to authenticate with third party. However a study has shown that that the current adoption of SSO is limited and presents several privacy issues specifically with consumer systems since the organisations responsible for the SSO would be able to see where and when a user is logging in to (Hayashi and Hong, 2015; Horsch, Hülsing, and Buchmann, 2015).

2.4. Password managers

Password Managers are one method of overcoming the increasing burden of remembering multiple unique passwords. The user only has to remember the master password for the manager, then all other application and service passwords are then stored by the manager. This means that the passwords generated can be made more complex since the end user will not need to remember them. One downside to using this method depending on how the application is setup is that if the master password is not secure or is forgotten then a user could lose access to all their accounts and need to manually reset their passwords (Boothroyd and Chiasson, 2012).

There are several different types of password managers available, which will be classified as the following: Built-In Browser Password Managers (BBPM), Local Password Managers (LPM) and Cloud Based Password Managers (CBPM). All password managers contain the basic functions of storing and retrieving user passwords, while offering additional functionality such as password generators, account synchronisation and various methods to protect the passwords including using a master-password system.

2.4.1. Built-In Browser Password Managers

Whenever a user types in a username and password into a service using a modern web-browser a prompt is normally shown to save the credentials into the browser, allowing
quicker login in the future. (Note: this is different to ticking the “Remember me” button on the service website). While this makes it easier for users to remember their passwords it can create security concerns, especially if the browser does not prompt the user to create a master password to encrypt the data or provide another means of keeping the data secure.

Internet Explorer offers to save passwords for users by default, meaning when a user enters account information in a web page the browser will prompt to save this information (Microsoft, 2016b). The account details are stored and encrypted using an API function built into Windows. This function makes it difficult for an adversary to gain access to the data if it was taken while the user was not logged into the computer. However if the user is logged in the information is already accessible and could be taken (Zhao and Yue, 2014). Internet Explorer does not offer synchronisation or master password protection of passwords.

Microsoft has developed a new web-browser for Windows 10 called Microsoft Edge. The password manager used by this web-browser works a little differently from the one used by Internet Explorer. It allows synchronisation across different Windows 10 computers and management of the passwords saved. The account details are offered to be stored by default but this can be disabled (Microsoft, 2016a). Little academic or security research has been conducted on where the passwords are stored and encrypted and Microsoft have not released the information from this yet.

Google Chrome by default asks users if they would like to save passwords when they first enter them into a web page. If the user has signed in to the web-browser using their Google account, then the passwords are also automatically synchronised for use on multiple devices. These passwords can be viewed or deleted via a web portal (Google, 2016b).

Google Chrome offers two main methods of keeping the passwords secure: 1) using a master password to encrypt the files locally or 2) using a Google account to encrypt and synchronise the files online. It is worth noting however that a user only needs to sign into the web-browser once for this feature to function. Every time the browser is loaded after this the passwords are automatically synchronised meaning accounts would be accessible to an adversary that gained access to a computer without them having to authenticate.

The password manager built into the Firefox web-browser has two methods for storing user passwords. One method involves encrypting the username and password information using three-key Triple-DES encryption to a SQLite database. The URL for the account is stored in
plaintext. The DES keys are stored in a separate binary file. If an adversary gained access to both the binary file and the SQLite database, they would be able to decrypt the keys successfully on any computer.

To prevent this Firefox also gives the option of using a master password to encrypt the data by using the password and a hashing algorithm to generate a master key. This is then used to encrypt the Triple-DES keys before saving them to the binary file. However, the way that this master password mechanism is designed means that it is potentially vulnerable to brute force attacks. This is because the system rejects an incorrect master password in one millisecond without putting a limit on the amount of guesses that could be made (Zhao and Yue, 2014).

2.4.2. Local Password Managers

Keepass is an open source local password manager that is freely available for use, with several advance features to enhance the user experience. The program works by storing user passwords into a central password database which is secured with either a master password or a key file. This database can then be synchronised with multiple computers by storing it in a cloud storage location such as Dropbox or Microsoft’s OneDrive, but this is something a user would need to setup themselves as opposed to it being built in. Keepass is also designed to be portable, meaning that it can reside on a usb stick for access on multiple computers and it does not need to be installed (Reichl, 2003).

Another password manager that can be classified as local is 1Password, although it does offer some synchronisation features using cloud services such as Dropbox, or local network synchronisation. Applications are also available on Android and iOS that allow users to access their password information from mobile devices. Passwords are stored in a password vault database that is secured using a master password (AgileBits, 2016).

2.4.3. Cloud-Based Password Managers

LastPass is a popular cloud-based password manager available on multiple platforms to end users and enterprise clients (Henry, 2015). It works by synchronising password information amongst other features between multiple devices and the cloud storage server. User passwords are encrypted using a master password system. The service offers several advanced features including the ability to share passwords without revealing them to multiple users, the storage of form information such as addresses and credit card information and the ability to store secure notes that can be accessed from multiple devices.
The main feature of cloud-based password managers is the ability to seamlessly synchronise multiple account details between multiple devices. LastPass achieves this by providing downloadable applications for all major operating systems and mobile devices, ensuring that users can access their passwords and use their system without having to purchase additional equipment.

LastPass also features a password generator that can be used to make multiple types of passwords from different character sets, including passwords that can be pronounced to make them easier to be remembered by the user in the event that they cannot access the password manager.

Passwords can also be shared between different LastPass users, which allows the account details to be accessed by another user. Once permission has been granted the account details are copied over to the second user, but the password is not revealed. It should be noted that there is a warning on the LastPass user website stating that “Savvy end users could potentially access the password if they capture it using advanced techniques” (LastPass, 2016).

LastPass can also be used for remembering important, confidential information securely using two different features: Secure Notes and (Form filling). Secure Notes consists of a platform where users can input data into a text document that is then encrypted and synchronised across devices. (Form Filling) involves the software copying data from a website form such as credit card information or an address and storing it securely, allowing it to be automatically copied into similar forms when requested by the user.

A security analysis of the vulnerability of LastPass has been undertaken by Zhao, Yue, and Sun (2014). The analysis created a threat model for cloud-based password mangers, which included the types of credentials that would be targeted, who they would be targeted by and the types of attack that would be conducted to gain access to the password vault. The credentials used to secure the data are the username and master password. These are then used to create a local key and hash used for data encryption and decryption. If a user chooses to have LastPass remember their master password to make access even more seamless then the password is stored locally on the machine in a SQLite database, and encrypted depending on the software configuration. The study showed that if an adversary could gain computational access to the end-users machine then they would be able to decrypt the master password by calling the same functions used to encrypt the data. If the user had not decided to save the master password, then a brute force attack could be used to calculate the master password.
Hard-coded plaintext “lastpass rocks” is encrypted using the master password and username is stored locally as ciphertext, meaning that if an adversary decrypts the ciphertext to get “lastpass rocks” then they have used the correct master password.

Another popular cloud-based password manager is RoboForm which is available in several different packages, in the form of a password manager for individual devices or one package for multiple devices with synchronisation. RoboForm has several advanced features similar to LastPass, including the password manager which is secured using a master password that is not stored on the developer servers for additional security. Other features available include a form filler which can be used to store form information such as an address, a password generator with advanced options allowing for a user to specify the number and type of characters generated and finally a start page that allows users instant login to multiple websites without them having to navigate to the sites manually (Siber Systems, 2016).

2.4.4. Alternatives to traditional password managers

One of the security issues with using a password manager is that the passwords stored are being protected by another password, which as shown previously could be vulnerable to different attacks like brute force. The Passwordless Password Synchronization project (PALPAS) looked to address this by changing the way that passwords are stored in a way that if either the server or client application is compromised the data remains secure. It does this by not actually storing the passwords at all, but instead storing different parts of data needed on the server and client for the password to be re-computed whenever access is needed. This means that password generation is a fundamental part of this approach, and that a password cannot be recomputed without access to both the server data and the client data. Each password generated is unique and designed to conform to the intended service password policy (Horsch, Hülsing, and Buchmann, 2015).

Another type of authentication system proposed by Hayashi and Hong (2015) is the Unified Authentication system which is based on the concept of using a smart device to communicate with proxy software running on a computer or physical device. The prototype used was named Knock x Knock which had two main parts: a client running on the users’ iPhone which contained all of their passwords and a proxy application running on their Mac to input the authentication data into online services used by the user.
Knock x Knock was created to address two main issues: the all or nothing concept that most password managers are based around and the security issue of having all passwords stored locally or on a database. To solve the first problem Knock x Knock uses a multi-tier access system to allow different accounts to require a different level of security. These tiers in the prototype were: 1) ‘Quick’, which automatically unlocked when the iPhone is at a trusted location. 2) ‘Standard’, which activates when the master password has been entered on the phone and remains unlocked until the phone leaves the trusted location. 3) Secure which needed the master password to be entered with each access attempt. The client connects to the Mac using Bluetooth Low Energy (BLE) which has a maximum proximity range of 1-15 metres that could be adjusted by the user. The data sent between the clients is encrypted using an AES-128 connection that is authenticated using two pre-shared keys.

While this method of authentication helps to hinder common methods of attack on password managers – see section 2.3 - due to the way the information is stored on the user’s iPhone, Hayashi and Hong were aware of the possibility of theft of both the iPhone and Mac in which an adversary would be able to access passwords stored under the ‘Quick’ tier. The adversary would still need to know the users iPhone credentials and master passwords to access the other two tiers however, and during this time a user could either remotely wipe their iPhone or download the application onto another phone via iCloud and reset their passwords. However, what was not tested were whether these newly created passwords would be synced across to the stolen device if it was not wiped, but since the application has not been released to the public this is yet to be confirmed by a third party.

The general opinion for the usability of the prototype was positive, with users stating that it was convenient to use while staying secure and simple. Participants also reported that the prototype was enjoyable to use which is an additional benefit since most security systems are more regarded as a burden than something to be enjoyed (Hayashi and Hong, 2015).

McCarney et al. (2012) conducted a similar study using a smart-phone password wallet and a Firefox extension named TAPAS, except without using a master password. Instead a user would need to use their smartphone to authenticate login requests via notifications whenever the user attempts to login to a stored account using the extension. TAPAS was designed to be implemented in the industry today without the need for server side changes. The extension part of the implementation contained the decryption key while the wallet contained the cipher text, meaning that the system is secure from an adversary that only had access to either the
smart-phone or the computer, however the implementation was not designed to protect against the loss of both devices.

2.5. Conclusions

In this chapter the key issues surrounding password authentication were researched in detail. Password usability becoming a growing area of concern for the computing industry. There is a constant battle between keeping passwords useable and memorable for end users while also ensuring that they are secure and cannot be guessed easily by an adversary.

The security of password authentication was researched, with an in depth study conducted on several different types of attack that can compromise security. Methods to prevent these types of attack being successful were also researched in the form of two-factor authentication and single sign-on.

Finally the different types of password managers currently available were researched. Potential alternatives currently being researched were also studied but unfortunately none of these are currently available for testing with this project. The next chapter will look into creating the research methodology that will be chosen to conduct this project.
3. Research Methodology

3.1. Introduction

Chapter 2 has shown that while security analysis has been conducted on password managers, research has not been conducted to show what forensic artefacts are left behind by cloud-based password managers. In order to find these forensic artefacts, usage data for the software needed to be acquired and analysed using a forensic tool kit to determine what useful artefacts were found from this software. The password manager that will be tested and examined is LastPass, due to its popularity across the consumer and enterprise markets.

The formal method for the investigation was qualitative due to the nature of the project, allowing the results to be interpreted in their context as opposed to being straight quantitative figures.

3.2. Research Strategy

There are several different research strategies that may be suitable for use with obtaining the usage data required to conduct the forensic analysis of the password manager. A user case study could be used to obtain the data, in which users would operate with the password manager over a certain period of time and the data would then be acquired at the end of the study for analysis, however in doing so this would raise ethical issues since real, live data would be captured and viewed which could contain sensitive information including users actual passwords.

To avert this a closed user case study could be conducted in which users create new accounts and passwords on closed environment virtual machines, however this still raises potential privacy issues with the account data later being analysed. For this reason, data will be generated in a closed study without using any real users.

3.2.1. Software Testing

There are multiple standards for testing software in the computing industry. One of these standards is black box testing. This method of testing is used to examine the functionality of an application without the knowledge of the internal working. The tester is generally aware that an input results in an invariable output, but they are not aware of how the software
actually produces the output in the first place. This is ideal for investigating software that does not have well published information on it’s working such as commercial software, since this is not open source (British Computer Society, SIGIST, 2001).

Another standard is white-box testing which involves testing different parts of the internal structures and workings of a program, as opposed to testing the functionality experienced by the end user. This technique of testing can only be performed if access to the source code of a program is available (British Computer Society, SIGIST, 2001).

Black box testing will be used in this study to test LastPass because it is closed-source commercial software, meaning that the internal workings of the program cannot be obtained making black box testing ideal. Another reason for choosing this technique is that this study will be testing the functionality of the software from a user perspective as opposed to how it works internally.

3.2.2. Forensic Analysis Strategy

Once test data has been created using the software testing strategy it will need to be forensically acquired and investigated using a forensics tool kit. There are several different types of digital forensic frameworks that can be used for this task, however it is important to note that this study is not a forensic investigation in which evidence will need to be presented at the end, it is a study to see what forensic artefacts are left behind by “LastPass” meaning that not all elements of the forensic framework will need to be implemented.

Each digital forensic framework contains four main steps to conduct an investigation, which are: collection, examination, analysis and report. The collection step involves ensuring that all data collected is preserved forensically and correctly audited to ensure that the Association of Chief of Police Officers (ACPO) guidelines are met. This study will use the Collection, Examination and Analysis steps to preserve and acquire the artefacts and will then present them for finding in section 4 of the study.
Figure 1: Comparison of forensic process models (Casey, 2004)

Figure 1 shows a comparison of several different digital forensic models currently in use, each of them have slightly different steps to achieve the same goals of Collection, examination, analysis and reporting. For this study DFRWS 2001 (Palmer 2001) was chosen to be used since it provides the simplest form of a digital forensic investigation while retaining the key steps needed to analyse and present the data.

3.3. **Ethics**

Data used by this study will be solely generated for the purpose of acquiring and analysing the forensic artefacts left behind by the cloud-based password manager LastPass to avoid ethical issues of using real-world data of individual users.

3.4. **Limitations**

The data used through the software testing phase of study is purpose generated, meaning that there will be a lot less data available in comparison to using real-world testing and capturing data from a live machine.

Only select features from LastPass will be tested due to the scope of the study, meaning that potential forensic artefacts left behind by these features will not be looked at. The testing will also only be conducted on one operating system using one web-browser, meaning that results may be slightly different if the investigation looked at other operating systems or web-browsers that support the password manager.
3.5. Summary

This chapter discussed the methodology that will be used to conduct the study. The study will use a closed testing research strategy in which data will be generated using black-box testing of the password manager LastPass. This data will then be acquired, examined and analysed for forensic artefacts using the DFRWS 2001 digital forensic framework as shown in Figure 1.

![Figure 2: Project Phase Diagram](image)

Figure 2 shows the project phase planning diagram. The design phase (Chapter 4) will consist of experiments being designed to test the different features of the password manager. Step by step guides for creating the scenarios to allow reproducibility will be included as Appendix 9.3. Each experiment will be presented in the form of a table in Chapter 4. The implementation phase will be the actual running of the experiments which will be included in Chapter 5 and Appendix 9.3. The forensic analysis strategy will be shown in Chapter 4 and the actual analysis of the virtual machine images will be conducted in Chapter 5 and Appendix 9.4.

The next chapter will discuss the design of the testing plan that will be used to generate simulated usage data that can be analysed for forensic artefacts potentially useful in a forensic investigation.
4. Design Phase / Testing Plan

The password manager LastPass version 4.1.7.0 will be installed onto three clean Windows 10 virtual machines running on VMware Workstation Pro 11.1.3. The web-browser Google Chrome version 50.0.2661.94 will be used to conduct the tests. The data acquired from the tests will then be captured and examined using the Encase 7.11.01.05 and OS Forensics 3.3. Full details on the software used during this project including version and build number can be found in Appendix 9.1.

Windows 10 has been chosen since it is the newest version of the Microsoft operating system and is currently being pushed out to older windows installations as recommended update (Foley, 2016), meaning that many users will soon be using Windows 10 if they do not already have it. Google Chrome has been chosen as it is a popular web-browser that supports the LastPass password manager. Encase has been chosen to conduct the forensic investigation since it is an industry standard piece of software (SC Magazine, 2013).

4.1. Overall Testing Goals

The goals of the testing is to see what data is left behind by the password manager LastPass. Each experiment will be run and then data will be acquired for forensic analysis by creating forensic image files of each virtual machine hard disk using FTK Imager 3.4.2.6. The captured data will then be loaded into the Encase 7 and analysed for forensic artefacts. Some files, specifically SQLite databases like those used by LastPass cannot be viewed correctly in Encase 7. Therefore OS Forensics will be used to view these files.

4.2. Feature Testing

Several different features of the LastPass password manager will be tested with experiments to see what forensic artefacts left behind for analysis. Data will be generated from within a simulated environment using three virtual machines. Each experiment will be fully documented with screenshots in Appendix 9.3 to ensure the results are reproducible.

The password manager will be tested to see what password information is stored locally and how it can be accessed. The reason this feature is being tested is since it is the primary function of any password manager and is likely to present the most useful forensic artefacts during analysis.
Synchronisation will be tested by using two different virtual machines signed in with the same account to see what data is synchronised between the two machines. This feature is being tested since it is becoming more common for users to have multiple devices, if artefacts can be found relating to these different devices it could be useful in a forensic investigation.

Form filling will be tested to see where this information is stored since it can contain sensitive information including a credit card number. Address and credit card information could be very useful during a forensic investigation which is why this feature is being tested.

Shared folders will also be tested using two different virtual machines, each signed into different accounts to see where exactly this shared data is being stored. This could potentially be used to find associates of a user through forensic artefacts which is why this feature is being tested.

4.3. Experiment Design

Each experiment is made up of multiple tests to generate simulated data for forensic analysis at the next stage. As multiple virtual machines will be used during this investigation a baseline virtual machine will be created and copied meaning that all data should remain the same across all three machines except the usage data created through LastPass. This helps to ensure that the test results are in a closed environment that is not affected by outside variables.

The specifications for the baseline virtual machine is shown in Table 2.

<table>
<thead>
<tr>
<th>Name</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>AMD Phenom II x6 1045T Hex-Core (2.71Ghz) (Virtualised, one CPU Core assigned to VM)</td>
</tr>
<tr>
<td>RAM</td>
<td>2gb DDR3 (Virtualised)</td>
</tr>
<tr>
<td>Hard Disk Capacity</td>
<td>60gb</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 10 Education</td>
</tr>
</tbody>
</table>

Table 2: Specifications of Baseline virtual machine to be used during testing
4.3.1. Experiment 1: Password Manager testing

The password manager LastPass will be installed and logged into using a testing account with the username ‘100243363@unimail.derby.ac.uk’ and the password ‘testingpassword’. The reason for using the university email account as opposed to a generic testing account is that premium features of the password manager will be used throughout the testing process, and these features are freely available to students who register using their academic email account. Simulated data will be generated by creating three different testing accounts for the social media website Twitter, Reddit and an email account under Yahoo, then the remember master password feature will be ticked to see if the password is then stored locally. The individual tests are shown in Table 3.

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Description of Test</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Testing account with LastPass will be created</td>
<td>Account be present in the SQLite database under table “LastPassSavedLogins2” since remember email option is selected by default.</td>
</tr>
<tr>
<td>2</td>
<td>Testing account on social media website ‘Twitter’ will be created and added to LastPass</td>
<td>Account will be present in the SQLite database.</td>
</tr>
<tr>
<td>3</td>
<td>Testing account on social media website ‘Reddit’ will be created and added to LastPass</td>
<td>Account will be present in the SQLite database.</td>
</tr>
<tr>
<td>4</td>
<td>Testing account on the email website ‘Yahoo’ will be created and added to LastPass</td>
<td>Account will be present in the SQLite database.</td>
</tr>
<tr>
<td>5</td>
<td>Remember master password option will be selected in the Chrome extension</td>
<td>Master password should now be present in the SQLite database, and may be encrypted / hashed.</td>
</tr>
</tbody>
</table>

*Table 3: Experiment 1 tests and expected results*
4.3.2. Experiment 2: Synchronisation Testing

One of the features that makes LastPass popular is the fact that all data can be synchronised to several devices on the premium versions. With the standard free version data is only synchronised between the users device and the LastPass servers. To test to see what forensically useful artefacts are left behind simulated data will be created using several tests, including a test that involves deleting an account from one computer to see if it gets deleted on the second computer. It will be interesting to see if there is any evidence left behind showing that the account has been deleted. The individual tests are shown in Table 4.

<table>
<thead>
<tr>
<th>Test Number</th>
<th>Description of Test</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Account will be logged into LastPass on second virtual machine</td>
<td>Data will be transferred from source virtual machine to the target via the LastPass website, SQLite database should be created on the target virtual machine containing identical information to the source.</td>
</tr>
<tr>
<td>2</td>
<td>Social Media account ‘Twitter’ will be deleted from Virtual Machine 1.</td>
<td>The account should also be deleted from Virtual Machine 2.</td>
</tr>
</tbody>
</table>

Table 4: Experiment 2 tests and expected results

4.3.3. Experiment 3: Form filling testing

The ability to autofill in data for different web forms has been included with most popular web-browsers for several years. This functionality has also been included with popular third party password managers to provide additional security and convenience for end users. This experiment will generate commonly used form data in the form of an address and a credit card which will be added to the password manager. It will be interesting to see if there is extra security put in place for the credit card specifically, since this is very sensitive information. The individual tests and expected results are shown in Table 5.
<table>
<thead>
<tr>
<th>Test Number</th>
<th>Description of test</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Address information will be entered into LastPass for use with automatically filling in forms</td>
<td>Address information should be present in the SQLite database.</td>
</tr>
<tr>
<td>2</td>
<td>Credit card information will be entered into LastPass for use with automatically filling in credit card information</td>
<td>Credit card information should be present in the SQLite database.</td>
</tr>
</tbody>
</table>

Table 5: Experiment 3 tests and expected results

4.3.4. **Experiment 4: Shared Folder testing**

Folder sharing is a feature available for both enterprise and premium LastPass users, it allows the sharing of passwords and other data stored within LastPass between multiple accounts with different access permissions. The folder can be shared with read only access meaning that the recipient cannot edit any information stored in the account or address. The folder can also be shared with the password set to be hidden, meaning that it cannot be viewed at all by the intended recipient, but can be used to automatically log into web services. Individual tests and expected results are shown in Table 6.
<table>
<thead>
<tr>
<th>Test Number</th>
<th>Description of Test</th>
<th>Expected Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A second LastPass account will be created and signed into a third virtual machine. Social Media account information for ‘Reddit’ will be shared from the first LastPass account to the second. The hidden and read-only options will be selected for this test.</td>
<td>A new folder should appear on the second LastPass computer containing details of the account information for ‘Reddit’. This should be present in the LastPass SQLite database. The password for the account should not be able to be viewed in any way since it should be hidden to the recipient.</td>
</tr>
<tr>
<td>2</td>
<td>The sharing settings will be changed from hidden to visible for the ‘Reddit’ service.</td>
<td>The password should now be available to be viewed from the web portal.</td>
</tr>
</tbody>
</table>

Table 6: Experiment 4 tests and expected results

4.4. **Digital Investigation Plan**

Once the simulated data has been created and the forensic images of the virtual machines have been completed the data will then be verified, processed and analysed in Encase 7 and OS Forensics. Firstly the forensic images will be searched for internet artefacts in the form of the internet cache and history files to look for any usage data that could potentially be useful in a forensic investigation. Next the SQLite database that LastPass uses to store information locally will be examined to see if there is evidence of the usage data present. Finally a keyword search will be conducted across all of the forensic images to look for further traces of LastPass and the testing accounts created within the application. These results will then be compiled and analysed in Chapter 5.
4.5. **Summary**

This chapter discussed the design of experiments which will be used to create usage data for the password manager LastPass, and how this data will then be analysed using the forensic toolkit Encase 7. The results of the experiments and analysis will be reviewed in chapter 5.
5. Findings and Analysis

5.1. Introduction

This chapter will showcase the results of the simulated testing experiments and the forensic analysis that followed this. Screenshots of the simulated testing and forensic analysis have been included to show the findings of the results and to back up any arguments made. Extensive notes and screenshots were taken throughout the entire testing process and forensic investigation. These can be found in Appendices 9.3 and 9.4. Full version information for any software resource mentioned in this chapter can be found in Appendix 9.1.

5.2. Simulated testing results and analysis

5.2.1. Experiment 1

LastPass was installed onto the first virtual machine, which was setup using the baseline installation documented in Appendix 9.2. During setup there was an advanced options page which allowed for various default settings to be changed, including a setting labelled as “Automatically log me into LastPass when my browser starts” as shown in Figure 3. The fact that this is included by default is cause for concern since it means that any user who gains access to the computer and user account that LastPass is installed on will be able to access service details simply by loading the web-browser.

![Figure 3: Screenshot of the advanced Options screen visible during the LastPass setup on virtual machine 1.](image)
For testing purposes the password “testingpassword” was chosen to be the master password for the LastPass account. This was viewed by the LastPass password strength meter as being a secure password even though it would be vulnerable to a dictionary attack as discussed in section 2.2.4. Figure 4 shows the account creation screen and the password strength meters assessment of the chosen password.

![Figure 4: Screenshot of the account creation screen for LastPass on virtual machine 1.](image)

Once LastPass had been installed and logged in on virtual machine 1 several accounts were created to simulate real world usage. An account was created for Twitter, Reddit and yahoo email using the password generator built into the application. It is worth noting that when the Reddit account was first created two entries were present in the password vault as shown in Figure 5, however this could have been a onetime glitch since only one was present during the later experiments.

![Figure 5: Screenshot of the LastPass vault in Google Chrome from virtual machine 1 with the anomaly of the duplicate Reddit account.](image)
Finally the “Remember Password” option was selected on the LastPass account login screen within the Google Chrome extension, meaning that the master password is stored locally in the SQLite database located within the app data folder. The fact that the password manager can login automatically using the advanced options setting without this setting being enabled is strange.

5.2.2. Experiment 2

LastPass was installed on the second virtual machine, which was also setup by the baseline installation documented in Appendix 9.2. The previously created testing account was then logged into the LastPass browser extension, and the accounts were then synchronised and downloaded onto the machine as shown in Figure 6.

![LastPass password vault](image)

*Figure 6: Screenshot of the LastPass password vault on virtual machine two.*

Next the Twitter account was removed on virtual machine one to see if it is also removed from virtual machine two, and if any traces of the account existing are left behind on either virtual machine. The account remained visible on the second machine even after refreshing the page until Google Chrome was closed and relaunched. Virtual machine two was then restored to a previous snapshot taken before the account was deleted and the network connection was disconnected to prevent the computer from synchronising. With LastPass launching in offline mode due to lack of connectivity the Twitter account was visible in the password vault once more meaning that the service information is stored locally on the machine as shown in Figure 7.
5.2.3. Experiment 3

Using the Form Fills feature built into LastPass simulated address and credit card information was added to the password vault under the “Form Fills” tab. Once the information had been added the virtual machine was disconnected from the network to see if this information is stored locally or just available from the LastPass servers. The data was accessible with the virtual machine offline as shown in Figure 8 meaning that this information must be stored somewhere on the virtual machine.
5.2.4. Experiment 4

A second LastPass account was created and signed into a third virtual machine, which was setup using the baseline document in Appendix 9.2. Account information earlier created for ‘Reddit’ on the first LastPass account was shared with the second LastPass account to simulate an enterprise setting, in which account information would be shared across a department. On the first instance of sharing a folder the settings “Read-Only” and “Hide Passwords” were selected as shown in Figure 9, and the second account was unable to view or edit the password for the account.

![Figure 9: Screenshot of the shared folder within the LastPass vault on Virtual Machine 1.](image)

To further test that the password is not at all accessible by the second LastPass account while the “Hide Passwords” option was selected the “Copy Password” feature was attempted, which allows a user to copy a password for a service into their clipboard in the event that the account is not automatically logged into the service through LastPass. However this feature did not work, no password was copied into the clipboard and no error message was displayed warning a user that this feature is disabled since the password is hidden.

Next the “Hide Passwords” setting was disabled, and the second LastPass account could now view the password within the LastPass vault. The “Copy Password” feature also started working once this change had been synchronised across. The virtual machine was then disconnected from the network to see if shared accounts are stored locally or stored online and the data was still accessible and viewable as shown in Figure 10.
While the machine was disconnected from the internet it was also discovered that a history function exists in LastPass, showing detailed information on login data for the various service accounts stored in the password manager. This function can only be accessed online however. Figure 11 shows the detailed listing of history information from virtual machine 1.

**Figure 10:** Screenshot of the shared Reddit account in the LastPass vault on virtual machine 3 while the network was disconnected.

**Figure 11:** Screenshot of LastPass history within the LastPass vault on virtual machine 1.

### 5.3. Forensic Analysis of Simulated Testing

Forensic images were made from all three virtual machines used during the simulated testing using FTK Imager 3.4.2.6, and once these images were verified they were analysed using Encase 7.11.01.5. The full process for how each image was created and then verified within Encase is documented in Appendix 9.4 along with full notes and screenshots of the entire forensic investigation. This is the identification, preservation and collection steps in the DFRWS Digital Forensic Framework chosen for this investigation in section 3.2.2. Then the
evidence files were processed using the options shown in Figure 12 to look for potential forensic artefacts left behind by the simulated testing.

![EnCase Processor Options](image)

*Figure 12: Screenshot from EnCase 7 showing options used to process each forensic image.*

5.3.1. Internet Artefacts and System Information Analysis

Internet artefacts was the first area to be investigated as this lists the cache and history files from the various web-browsers used on the system, and LastPass is an extension to these browsers. The first virtual machine, used for all four experiments was examined first. Since this machine has been used the most it was expected to provide the largest quantity of internet artefacts. Two web-browsers were found to contain data in total, these being Internet Explorer and Google Chrome. Microsoft Edge was used originally to download Google Chrome, however the web-browser still uses a few of the same paths that Internet Explorer uses which is why the browser didn’t show up individually within the Internet Artefacts section. All of the WebCachev01.dat files located under Internet Explorer history contained reference to the LastPass installation. The URLs also contain reference to the account that was created during the setup process of LastPass in Experiment 1, along with the Windows account being used at the time. This means the information was transmitted back to the developer’s servers during install as shown in Figure 13. It is worth noting here that LastPass was actually downloaded using Google Chrome, not Microsoft Edge or Internet Explorer. This means that the setup process must use elements of Internet Explorer in order to download and install the application.
Evidence of the LastPass vault being accessed was first discovered within the Google Chrome HTML cache, in which the tutorial guide for how to use the password vault is clearly visible as an HTML document. At this stage “No Matching Results” was shown under “Remember Every Password” section since service accounts had not been made at this stage of the simulation, as shown in Figure 14.

Further evidence of LastPass use was visible in the code and image cache for Google Chrome in virtual machine 1, however little forensically useful information was found here besides
access times and URL addresses since it was all code or images used to make the application work as opposed to user data such as service accounts or form information. Evidence of cookies being used with LastPass were also found, however the files appeared to be empty except for the access times and URL information meaning they also have little forensic value. It’s possible that these cookies were used to authenticate with the application but without proof on how LastPass actually works internally this is just speculation. History was then checked for all three virtual machines, showing that the local password manager installed on the system was accessed 12 times on virtual machine 1, 2 times on virtual machine 2 and 7 times on virtual machine 3.

Next the system info parser records were reviewed for virtual machine 1. The system info parser is an option selected when the image files were processed in Encase that displays information on the operating system, hardware and software that had been installed on the computer. Evidence was found under “Installed Applications” of Google Chrome being installed but nothing was discovered for LastPass even though it had been installed a shown in Figure 15. This means that the tool cannot be exclusively relied in to collect information on all of the programmes installed within the forensic image. Identical information was found on virtual machine 2 and 3.

<table>
<thead>
<tr>
<th>Name</th>
<th>File Offset</th>
<th>Last Written</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>chrome.exe</td>
<td>19/04/16</td>
<td>12:36:37</td>
<td>1</td>
<td>Program Files (x86)</td>
</tr>
<tr>
<td>lmp32.exe</td>
<td>10/07/13</td>
<td>14:19:39</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>oshim.dll</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>IEDAG.EXE</td>
<td>10/07/13</td>
<td>14:19:49</td>
<td>1</td>
<td>Program Files\Internet Explorer\IEDAGCMD.EXE</td>
</tr>
<tr>
<td>IEDAGCMD.EXE</td>
<td>10/07/13</td>
<td>14:19:49</td>
<td>1</td>
<td>Program Files\Internet Explorer\IEDAGCMD.EXE</td>
</tr>
<tr>
<td>IEPLS.EXE</td>
<td>10/07/13</td>
<td>14:19:49</td>
<td>1</td>
<td>Program Files\Internet Explorer\IEPLS.EXE</td>
</tr>
<tr>
<td>install.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Journal.exe</td>
<td>10/07/13</td>
<td>17:43:56</td>
<td>2</td>
<td>ProgramFiles\Windows Journal\Journal.exe</td>
</tr>
<tr>
<td>Kensmmanagers...</td>
<td>10/07/13</td>
<td>14:19:49</td>
<td>2</td>
<td>\Windows\System32\kensmmanagers.exe</td>
</tr>
<tr>
<td>mi.exe</td>
<td>10/07/13</td>
<td>17:43:39</td>
<td>2</td>
<td>CommonProgramFiles\Microsoft Shared\WindowsPowerShell.exe</td>
</tr>
<tr>
<td>mplayer2.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>ProgramFiles\Windows Media Player\mplayer.exe</td>
</tr>
<tr>
<td>notepad.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>SystemRoot\System32\notepad.exe</td>
</tr>
<tr>
<td>PowerShell.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>SystemRoot\System32\WindowsPowerShell\v1\PowerShell.exe</td>
</tr>
<tr>
<td>setup.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>ShoppingTool.exe</td>
<td>10/07/13</td>
<td>17:43:56</td>
<td>2</td>
<td>SystemRoot\System32\ShoppingTool.exe</td>
</tr>
<tr>
<td>table32.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>TabTip.exe</td>
<td>10/07/13</td>
<td>17:43:56</td>
<td>2</td>
<td>CommonProgramFiles\Microsoft shared\ink\TabTip.exe</td>
</tr>
<tr>
<td>web.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>ProgramFiles\Windows\Media\web.exe</td>
</tr>
<tr>
<td>wdi.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>ProgramFiles\Windows\Multimedia\wdx.dll</td>
</tr>
<tr>
<td>xmpplayer.exe</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>ProgramFiles\Windows Media Player\xmpplayer.exe</td>
</tr>
<tr>
<td>WORDPAD.EXE</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>ProgramFiles\Windows\Accessories\WORDPAD.EXE</td>
</tr>
<tr>
<td>WRITE.EXE</td>
<td>10/07/13</td>
<td>12:03:46</td>
<td>2</td>
<td>ProgramFiles\Windows\Accessories\WRITE.EXE</td>
</tr>
</tbody>
</table>

Figure 15: Screenshot of the installed applications on virtual machine 1, as found by the System Info Parser option in Encase.
5.3.2. SQLite database analysis

The SQLite database is used to save data locally by LastPass was examined next. This database is located within the windows user accounts appdata folder, full information on the path can be found in Appendix 9.4 in the SQLite database analysis section. While the contents of the file could be partially viewed within Encase 7 using the transcript window as shown in Figure 16, the data was not very readable.

![SQLite database analysis](image)

Figure 16: Screenshot of the SQLite database for the LastPass Google Chrome extension on virtual machine 1 viewed using the transcript window in Encase 7.

The majority of the data stored here appears to be encrypted using the method stated in section 2.4.3, however this data can be decrypted since the master password is known. If the master password was not known then the virtual machine would need to be loaded into a live environment in order to decrypt the password by also using the method shown in section 2.4.3. This security vulnerability was researched in section 2.4.3 (Zhao, Yue, and Sun, 2014) but it appears that it still has not been fixed. The username can be clearly seen on line 19 of the transcript window in Figure 16.

To make the data more readable the SQLite database for each virtual machine was exported from Encase 7 and loaded into OS Forensics 3.3 since it has a built in SQLite database viewer. The database was found to contain 6 different tables used for storing user data. The tables found to contain forensically useful data were LastPassData, LastPassPreferences,
LastPassSavedLogins2 and sqlite_sequence. Figure 17 shows what appears to be the username hash for the first LastPass account since it is identical in each entry. If this value was different then it would be possible that it is a username hash for the different service accounts added to the password manager, but this is not the case. One of the entries appears to be for an RSA key which could be assumed to be used to encrypt the various data used by the LastPass extension. The username hash was also identical on virtual machine 2 since both machines were signed in with the same account. The values for otp and the rsakey were different however, meaning different encryption values were used to encrypt this data. The table on virtual machine 2 is shown in Figure 18.

![Figure 17: Screenshot of the LastPassData table within the SQLite database found on virtual machine 1, in OS Forensics 3.3](image1)

![Figure 18: Screenshot of the LastPassData table within the SQLite database found on virtual machine 2, in OS Forensics 3.3](image2)

The next table examined was the LastPassPreferences table that appeared to contain data for different options used by the password vault as shown in Figure 19. Some of these options have the username_hash field which is assumed to belong to the first LastPass test account, while others have no username at all. It can therefore be assumed that the preferences are split into account based options and default options for all users. “Remember Email” and “Remember Password” are shown with the value ‘1’ since these options were ticked during the simulated testing in experiment 1. From a forensics standpoint this means that if the virtual hard disk was loaded and run on a virtual machine, and access to the windows user account that had LastPass installed was available a forensic investigator would be able to access all the services and data stored on the account simply by launching the Google Chrome web-browser and navigating to the password vault. This information was also shown to be available while the computer is disconnected from the internet during testing, which is important since it means that the data would remain intact and accessible even if the end user managed to delete the information directly from the LastPass website. In the event that the virtual machine was connected and the data was removed through synchronisation it could be
recovered by copying the original evidence file again and booting this with the network connection once more disconnected.

<table>
<thead>
<tr>
<th>id</th>
<th>username_hash</th>
<th>prename</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>652807b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>tom@ljou62</td>
<td>Name [a-d]</td>
</tr>
<tr>
<td>36</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>sharedfolder6order</td>
<td>Name [a-d]</td>
</tr>
<tr>
<td>39</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>emergencytrustedfolderorder</td>
<td>Folder [a-d]</td>
</tr>
<tr>
<td>46</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>espassword</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>leekemorphine</td>
<td>$6$6565565$105.0</td>
</tr>
<tr>
<td>45</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_length</td>
<td>10.0</td>
</tr>
<tr>
<td>46</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_upper</td>
<td>1.0</td>
</tr>
<tr>
<td>47</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_lower</td>
<td>1.0</td>
</tr>
<tr>
<td>48</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_digits</td>
<td>1.0</td>
</tr>
<tr>
<td>49</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_special</td>
<td>0.0</td>
</tr>
<tr>
<td>50</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_start_with</td>
<td>0.0</td>
</tr>
<tr>
<td>51</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_ends_with</td>
<td>0.0</td>
</tr>
<tr>
<td>52</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_hex</td>
<td>10.0</td>
</tr>
<tr>
<td>53</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_username</td>
<td>0.0</td>
</tr>
<tr>
<td>54</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_password</td>
<td>1.0</td>
</tr>
<tr>
<td>55</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_auto</td>
<td>0.0</td>
</tr>
<tr>
<td>56</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_pattern</td>
<td>10.0</td>
</tr>
<tr>
<td>57</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>generate_strength</td>
<td>4.0</td>
</tr>
<tr>
<td>58</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>esrandom</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Figure 19: Screenshot of the LastPassPreferences table within the SQLite database found on virtual machine 1, in OS Forensics 3.3

The LastPassSavedLogins table should have contained all of the service information for the accounts created during the simulated testing according to (Reference), however it appeared to be empty when viewed through OS Forensics 3.3 as shown in Figure 20. This means that the data can currently only be viewed while the virtual machine is running by viewing it directly through the LastPass vault which by default logs in automatically when the browser is launched.

<table>
<thead>
<tr>
<th>id</th>
<th>username_hash</th>
<th>encryption</th>
<th>password</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>AES-256</td>
<td>0.0</td>
</tr>
<tr>
<td>36</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>AES-256</td>
<td>1.0</td>
</tr>
<tr>
<td>39</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>AES-256</td>
<td>true</td>
</tr>
<tr>
<td>46</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>AES-256</td>
<td>Name [a-d]</td>
</tr>
<tr>
<td>42</td>
<td>652387b168e12c0c08d149072e1d5c232f21cc869b7c2b48f2023000209</td>
<td>AES-256</td>
<td>Name [a-d]</td>
</tr>
</tbody>
</table>

Figure 20: Screenshot of the LastPassSavedLogins table within the SQLite database found on virtual machine 1, in OS Forensics 3.3

The next table to be examined was LastPassSavedLogins2 which contained information for the LastPass user account as shown in Figure 21. The username is visible along with an encrypted copy of the master password and the last login time, although it is unknown what format the time is stored in; attempted conversion to Unix time didn’t supply the correct result.
When compared against the same table from virtual machine 2, the master password field has a different value which means that it was encrypted using a different seed or salt. This means it can be assumed that either the salt or seed is randomly generated when an account is logged into LastPass since both virtual machines were logged in using the same LastPass account. The table from virtual machine 2 is shown in Figure 22.

Virtual machine 3 was signed in using the second LastPass account, meaning that the database values were different to that of virtual machine 1 and 2. A different username hash was found in the LastPassData table, and different values were present for the username and password in the LastPassSaveLogins2 table as shown in Figures 23 and 24.

No trace of the first LastPass account, “100243363@unimail.derby.ac.uk” was found within the database even though the account had shared a folder with the second account in experiment 4.

5.3.3. **Keyword search**

A keyword search was conducted on all three virtual machine images to look for further artefacts relating to LastPass usage. Firstly LastPass was searched and a total of 149,147 hits
were found across the three virtual machines. These results were then sorted by the date of creation, date of access and which the machine located on to make them easier to analyse. References to the LastPass chrome extension were found but no usage data was located here besides the database that had already been discovered. The remaining results were then examined but no further usage data was found besides LastPass web page caches that had already been previously analysed in section 5.3.1.

A new search was conducted with the keyword ‘100243363’ in hope to find further usage data since this is contained within the username for the first LastPass account. 147 results were found across all three virtual machines. The username and a private encryption key for a server were found within a windows event log file named “Microsoft-Windows-WinInet-Capture%4Analytic.evtx” as shown in Figure 25, however this likely is only the encryption key used for SSL communication with the LastPass web server as opposed to the key used to encrypt data locally on the machine.

![Figure 25: Screenshot of keyword search result for “100243363” from virtual machine 1 in Encase 7.](image)

Next the actual master password was searched to see if there were any references to it in the system, however no results were returned as shown in Figure 26.
Finally the keyword “l.acott1” was searched since this is contained within the username for the second LastPass account. Only data was found on virtual machine 3 with 4 hits in total. This is surprising because in experiment 4 the username was entered on virtual machine 1 so some trace of the username is to be expected within the browser cache but none was found.

5.4. **Summary**

This chapter discussed all the results from the four simulated testing experiments and the forensic analysis undertaken. The results taken from each experiment and the digital forensic investigation were analysed in preparation for full contextual analysis in Chapter 6.
6. Discussion

6.1. Introduction

This chapter will discuss all of the results from the experiments and digital forensic investigation that were undertaken in chapter 5. These results will be analysed and put into context in regards to digital forensic value.

6.2. Forensic value of the data acquired

All data captured from the experiments could have potential value in a forensic investigation. This includes the username of LastPass account and metadata showing when the password manager was accessed. Unfortunately due to the way that LastPass encrypts and saves the data meant that the service account information created in experiment 1 and 3 could not be accessed directly from the forensic image. It was however discovered that if the password manager had been installed using the default settings then a live acquisition of data could be made. By default, the password manager automatically logs in to the LastPass vault when the web-browser is launched, allowing an adversary or a forensic investigator to access the data and copy the files. This however goes against standard forensic practice because by doing so the integrity of the data is technically compromised since it would be altered during the booting of the operating system. This could be circumvented by using a copy of the evidence in an isolated environment to capture usage and password information and recording it manually however this practice would not be seen as being forensically sound due to issues with data integrity.

During Experiment 4 it was discovered that a full history of all account login information could be found through the LastPass vault portal, including information on access times, accounts users and IP address information. This information could be potentially used to find out which locations a suspect is using the password manager if acquired, however this data could only be recovered while the virtual machine was running and connected to the internet. Therefore the data would either have to be recovered during the acquisition phase while the device was still powered on, or a copy of the evidence file would need to be loaded into a virtual machine with access to the internet which is not a forensically sound practice.
6.3. Other Findings

During the investigation a few problems were encountered with Encase 7. Firstly there were several issues initially processing the data, this was discovered to be due to lack of space on the University’s network drives so once the cache was completely moved to an external drive it started working correctly as expected. Secondly it was expected that SQLite files were able to be viewed natively in the form of a database table, not a text file. Unfortunately this was not the case which was why OS Forensics 3.3 had to be used for this part of the investigation.

6.4. Conclusion

This chapter looked at the results of the experiments and forensic analysis undertaken to see what potentially useful forensic artefacts were uncovered and put these into context with research found in the literature review regarding sound forensic practice. The next chapter will provide an overview of the project as a whole against the aims and objectives set out in section 1.2 and provide recommendations for improvements and future work.
7. Conclusions

7.1. Conclusions

7.1.1. Aim and Objectives

In this chapter the original aims and objectives set out at the beginning of the project will be assessed individually, with an explanation to how each one was met.

“The aim for the project is to perform forensic analysis on the cloud-based password manager LastPass and evaluate the forensic value of artefacts left on the local machine when using selected features.”

To evaluate the forensic value of artefacts left by LastPass a forensic analysis was conducted on simulated usage data created through four experiments, in which different features of the LastPass password manager were tested over several virtual machines. It was discovered that user data was being stored in a SQLite database named ‘1’ within the Google Chrome app data folder, including the email address used to register the account. Further information including the master password was found to be encrypted when certain settings were selected during the setup process of the password manager account. This data could be recovered through live usage of the virtual machine but could not be viewed through forensic analysis of the file system itself. Several other artefacts were found within the internet browsing history and cache relating to the use of the password manager, however actual account data was not found to be recoverable this way but again could be accessed through live usage of the virtual machine.

“Determine the current state of research being performed on password managers from both a forensics and a security standpoint.”

Very little information was found during the literature review on forensic analysis of password managers. However an abundance of data was found on the current security research of password managers, including that of LastPass which showed several security vulnerabilities that were looked into throughout the investigation. Extensive research was also conducted on the usability of passwords for authentication to express why password managers and similar solutions are becoming more popular in the field of computing and security. This research combined with the security research into password managers and password...
authentication lead to an understanding as to why people are struggling with the burden of password authentication over multiple accounts. Areas in which organisations can adjust their password policies or adopt tools to assist with this burden were also highlighted, specifically that the trend that password expiration policies provide a security benefit is not necessarily true.

“Design experiments to create usage data for the forensic analysis of cloud-based password managers”

Four experiments were designed to test specific features of the LastPass password manager with the aim to create usage data for later forensic analysis. These experiments involved adding accounts and other data to LastPass as well as testing synchronisation and sharing features of the password manager to look for the potential of useful forensic evidence.

“Capture data using an image of the output from the experiments in the form of forensic image files.”

Three virtual machines were used in total for the testing of the LastPass features, and three forensic image files of their virtual hard disks were created using FTK Imager 3.4.2.6. These image files were then verified in Encase 7.11.01.05 to ensure that data integrity of the evidence was consistent with digital forensic practices.

Examine the output using the selected forensic analysis methodology to see what artefacts are left behind.

The data collected had already been identified, preserved and collected during the forensic imaging process described above. The data was then examined and analysed in accordance with the DFRWS 2001 framework. Since the evidence is not going to be presented in court and the purpose of the project was to see what artefacts of forensic value were left behind from using LastPass the presentation phase was not completed.

Analyse the final results obtained in terms of potential use in Forensic Investigations.

It was then determined in section 6 that while the data collected had potential forensic value, a lot more value could be found via live acquisition directly from the virtual machines while they were running, especially if certain default settings were enabled which allows for access to the LastPass vault simply by launching the web-browser. Information including passwords
for different services, credit cards and usage history could be found this way, however doing so is not necessarily forensically sound since the data integrity of the hard disk would be compromised. Therefore it is recommended that this is done on a copy of the original data with the internet connection disconnected if just access to the accounts is needed. If access to history or other advanced features is required then it was discovered that a connection to the internet is needed.

7.1.2. Improvements and future work

If this project were to be carried out again there are a number of different recommendations that could be made to improve the outcome of the project.

Firstly real-world usage could be investigated in which a much vaster amount of data would be able to be captured and analysed since more artefacts may have been available if more in depth usage was captured, but this was unfortunately out of scope for the investigation. There are however potential ethical implications from conducting such a study since real-world usernames and accounts would be captured, meaning that all data would need to be anonymised or created purposely for the project.

Secondly LastPass could be reverse engineered to identify how exactly the program encrypts and stores the data used by the password manager, since this data is stored locally. The expected location of this data was within a table of the previously discussed SQLite database, however when investigated it was found that this table was empty even though the information could be accessed while the virtual machine was disconnected from the internet.

Live acquisition in the form of capturing the ram while the password manager was in use could also lead to different forensic artefacts being found, in the form of account data being stored in the ram during general usage. This is an area that could be looked at in depth with further study with a future project.

Finally another cloud-based password manager could have been tested and directly compared to LastPass in terms of artefacts left behind from usage data which is another area that could be looked at in depth through a future project.
8. Bibliography


9. Appendices

9.1. Resources used during testing and forensic analysis

Table of resources used during testing

<table>
<thead>
<tr>
<th>Software name</th>
<th>Developer</th>
<th>Version</th>
<th>Location</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware Workstation</td>
<td>VMware</td>
<td>11.1.3 build-3206955</td>
<td>Preinstalled on the University’s lab machines</td>
<td>Used to create and run virtual machines for testing</td>
</tr>
<tr>
<td>Windows 10 Education</td>
<td>Microsoft</td>
<td>Windows 10 Education</td>
<td>Downloaded from the University’s shared storage</td>
<td>Operating system which LastPass is installed on for testing</td>
</tr>
<tr>
<td>Google Chrome</td>
<td>Google</td>
<td>50.0.2661.94</td>
<td>Downloaded from <a href="https://support.google.com/chrome/answer/95346?hl=en-GB">https://support.google.com/chrome/answer/95346?hl=en-GB</a></td>
<td>Web browser used with LastPass extension</td>
</tr>
<tr>
<td>LastPass Google Chrome Extension</td>
<td>LastPass</td>
<td>4.1.7</td>
<td>Installed through the LastPass Universal Windows Installer</td>
<td>Chrome Extension used for testing and analysed for forensic artefacts</td>
</tr>
</tbody>
</table>
Table of resources used for Forensic Analysis

<table>
<thead>
<tr>
<th>Software name</th>
<th>Developer</th>
<th>Version</th>
<th>Location</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS Forensics</td>
<td>Passmark Software</td>
<td>3.3 Build 1004 (64 Bit) Free Edition</td>
<td><a href="http://www.osforensics.com/osforensics.html">http://www.osforensics.com/osforensics.html</a></td>
<td>Used to view SQLite database files, and access chrome cache?</td>
</tr>
<tr>
<td>Encase Forensic</td>
<td>Guidance Software</td>
<td>7.11.01.05</td>
<td>Preinstalled on the University’s lab machines</td>
<td>To analyse forensic images captured from the virtual machines for artefacts left from testing</td>
</tr>
</tbody>
</table>
9.2. **Virtual machine baseline setup notes**

VMware Workstation 11.1.3 launched to create virtual machine for baseline. VMware version information

![VMware Workstation 11.1.3 version information](image)

Virtual Machine Setup Steps

1. Clicked “Create a New Virtual Machine”
2. Typical (recommended) selected, clicked “Next”

3. Windows 10 education ISO selected for installation, clicked “Next”
4. “Windows 10 x64” selected under “Version”, clicked “Next”

5. VM named “Windows 10 x64 (Baseline VM)”, storage location entered
6. Default option “Split virtual disk into multiple files” kept, clicked “Next”

Baseline Virtual machine has now been setup. Next step is installing Windows 10 Education
1. Virtual Machine is powered on by clicking “Power on this virtual machine”

2. Machine booted and the setup menu loaded, Clicked “Next”
3. “Install now” clicked

4. Clicked “Skip” on product key page
5. “I accept the licence terms” clicked

6. Clicked “Custom: Install Windows Only”
7. Clicked “Next”

![Windows installation screen]

8. Waited for Windows to be installed

![Windows installation progress]

- Copying Windows files
- Getting files ready for installation (3%)
- Installing features
- Installing updates
- Getting finished
9. “Do this later” selected

10. Clicked “Customise settings”, since many Windows 10 features are not needed for this project.
11. All options switched to “Off” position, clicked “Next”

Customise settings

Personalisation
Personalise your speech, typing and inking input by sending contacts and calendar details, along with other associated input data to Microsoft.

Off

Send typing and inking data to Microsoft to improve the recognition and suggestion platform.

Off

Let apps use your advertising ID for experiences across apps.

Off

Location
Let Windows and apps request your location, including location history, and send Microsoft and trusted partners some location data to improve location services.

Off

12. All options switched to “Off” position, clicked “Next”

Customise settings

Browser and protection
Use SmartScreen online services to help protect against malicious content and downloads in sites loaded by Windows browsers and Store apps.

Off

Use page prediction to improve reading, speed up browsing, and make your overall experience better in Windows browsers. Your browsing data will be sent to Microsoft.

Off

Connectivity and error reporting
Automatically connect to suggested open hotspots. Not all networks are secure.

Off

Automatically connect to networks shared by your contacts.

Off

Send error and diagnostic information to Microsoft.
13. Clicked “Join a domain”, machine will not actually be joining a domain, this is just to complete the setup process.

14. Account setup with name “Test Account” and password “Testingpassword”. Password hint is “Test Account” since this is required. Note: Microsoft state that the password should be something that is easy for you to remember but hard to guess, doesn’t specify any advice on how to make a password fit this criteria.
15. Clicked “Yes”. Baseline Windows 10 Setup now complete.

16. Clicked “I Finished Installing” on VMware window

17. Clicked “VM” then “Install VMware Tools” on VMware window
18. Double clicked on “VMware Tools” in the virtual dvd drive on the Virtual Machine Window

19. Clicked “Yes” to the UAC Prompt
20. Clicked “Next”

21. Clicked “Next”
22. Clicked “Install”

Application install: Google Chrome needs to be installed ready for the installation of LastPass.

1. Microsoft Edge launched from the start menu, “Google Chrome” searched in the edge search bar
2. “Chrome Browser” selected from the Bing search engine page
3. Clicked “Download Chrome”

4. Left default option “Set Google Chrome as my default browser” selected. Clicked “Accept and Install”
5. Clicked “Run”

![ChromeSetup.exe finished downloading.][1]

6. Clicked “Yes” on UAC prompt

![User Account Control][2]

7. Google Chrome version 50.0.2661.94 is now installed

![Google Chrome installed][3]
Finally a snapshot of the baseline setup is created in case the virtual machine needs to be restored to the baseline. This was done from the VMware window by selecting “VM” > “Snapshots”.

![Windows 10 x64 (Baseline VM) - Take Snapshot](image)

Taking a snapshot lets you preserve the state of the virtual machine so that you can return to the same state later.

Name: Baseline Installation (Prior to LastPass Install)
Description: Windows 10, VMware Tools and Google Chrome have been installed in preparation for testing.
9.3.  Simulated usage data creation notes: Experiments 1-4

Experiment 1
LastPass was setup on Virtual Machine 1 and testing data was created for forensic analysis later using the forensic toolkit.

Test 1
LastPass cannot be installed without a LastPass account, because of this the installation took place after the virtual machine baseline and is documented below. During the installation a testing account was created with the username “100243363@unimail.derby.ac.uk” and the password “testingpassword”.

1. The web browser Google Chrome was loaded, and “LastPass” was entered into the search box. The first link was then clicked

2. On the LastPass landing page the “Get LastPass Free” button was clicked. After account creation the premium version of LastPass will be activated using the free offer
of LastPass premium for students.

3. LastPass was then downloaded onto the virtual machine. “lastpass_x64.exe” was then clicked from the download bar.
4. Clicked “Run”

Open File - Security Warning

Do you want to run this file?

- Name: C:\Users\Test Account\Downloads\lastpass_x64.exe
- Publisher: LastPass (Marvasol Inc)
- Type: Application
- From: C:\Users\Test Account\Downloads\lastpass_x64.exe

Run  Cancel

Always ask before opening this file

While files from the Internet can be useful, this file type can potentially harm your computer. Only run software from publishers you trust.

What’s the risk?

5. Clicked “Yes” on the UAC prompt

User Account Control

Do you want to allow this app to make changes to your PC?

- Program name: LastPass Installer
- Verified publisher: LastPass (Marvasol Inc)
- File origin: Hard drive on this computer

Show details  Yes  No

Change when these notifications appear
6. Clicked Advanced Options
7. **Advanced Options Screen.** Clicked “Install LastPass”. Note that by default LastPass automatically logs a user into LastPass when the browser starts which is a significant potential security risk, since it means anyone with access to the user’s computer and windows account could gain access to the Password Vault. This is very useful from a Forensics standpoint since it means that if the a copy image of a suspects computer was loaded into a virtual machine (A copy being used to ensure that the original data is preserved) then access could be granted to a user’s password vault. For the purpose of testing these options are left as default since this is likely to be what a user would have done on install.

![LastPass](image)

**Advanced Options**

- **Browser Add-ons**
  - Internet Explorer
  - Google Chrome
  - Firefox
  - Opera

- **Install Options**
  - Directory: `C:\Program Files (x86)\LastPass`
  - Add icon to desktop

- **Privacy Options**
  - Automatically log me into LastPass when my browser starts
  - Keep a history of my logins and form fills
  - Send anonymous error reporting to help improve LastPass

Reset to default recommended options.
8. Clicked “Close Programs”

9. Clicked “Create Account”
10. Entered account details: Email “100243363@unimail.derby.ac.uk”, Master Password: “testingpassword”. This password was apparently a fairly secure choice according to the password meter. Clicked “Create Account”

11. LastPass installed, and Google Chrome was launched, prompting for the extension to be added to the web browser. Clicked “Add Extension”
12. Clicked the LastPass logo to show login screen

13. Entered account information into login boxes and clicked “Log In”
14. LastPass is now logged in, and the LastPass account has been setup

15. A snapshot was created in the event that the testing machine needs to be reset to this stage
Experiment 1: Test 2

A social media account for twitter was created using the testing credentials, a password for this account was generated using LastPass.

1. Google Chrome was launched and “Twitter” was searched. The first link provided by the search engine was clicked

2. Clicked “Sign Up”
3. The following credentials were entered: Name “100243363”, Email “100243363@unimail.derby.ac.uk”. The LastPass logo was clicked in the password field, then “Generate a new password” was selected.

4. Clicked “Use Password”
5. Clicked “Save Site”

6. Clicked “Sign up”
7. The twitter account is now present in the LastPass vault

8. Snapshot of the experiment was created for later review
Experiment 1: Test 3

Testing account on social media website ‘Reddit’ was created next.

1. Google Chrome was launched and ‘Reddit’ is searched. The first link provided by the search engine was clicked.

   ![Google search for Reddit](image1)

   **A privacy reminder from Google**

   ![Reddit login page](image2)

   ![Reddit homepage](image3)

2. Clicked “Login or sign up”
3. Username “100243363” entered, password automatically generated using LastPass as done with Experiment 2. Clicked “Use Password”.

4. Clicked “Save Site”
5. Clicked “Sign up” to complete sign up process

6. Reddit was now visible in the password vault. Interestingly there are two accounts here for reddit, although the account in the group “none” only contains the generated password, while the account in the group “Social” contains all the login information. It is odd that the same thing did not happen with twitter in test 2.
7. Snapshot created of the experiment in VMware for later review.

Experiment 1: Test 4

An email account on Yahoo will be created using the testing credentials, a password for this account will be generated using LastPass

1. Google Chrome was launched and “Yahoo email” is searched. The first link provided by the search engine was clicked
2. Clicked “Sign up”

![Yahoo sign up screen]

3. First name “Testing”, second name “account”, email address “T100243363” entered and the generate password button on LastPass was clicked, followed by “User Password”. T100243363 needed to be used this time since Yahoo states that all email addresses must start with a letter.

![Generated password on LastPass]

![Yahoo sign up screen with email and password entered]
4. Clicked “Save Site”

![Image of Save Site dialog box]

5. Interestingly the password wasn’t automatically filled in to the “Password field”. Entered the password manually into the field. Testing phone number and Date of Birth entered, then clicked “Continue”
6. Yahoo account was now present in the LastPass vault. Previous occurrence of Reddit being in the vault twice seems to have fixed itself.

7. Snapshot created of the experiment in VMware for later review
Experiment 1: Test 5

Remember master password option was now selected, meaning the password should have been encrypted and saved to the local SQLite database.

1. Google Chrome was loaded and “Remember Password” box is ticked, “Yes” was clicked after the warning prompt then the password is entered and “Log in” was clicked. LastPass is now logged in, with the master password remembered and saved to the database.

2. Snapshot created of the experiment in VMware for later review.
Experiment 2: Synchronisation testing

Test 1

LastPass was downloaded and installed onto the second virtual machine, which was a clone of the baseline with identical specifications. Once LastPass was downloaded the previously made account was logged in, and the database should have downloaded onto the system, identical to the database on the first system.

1. The steps from Experiment 1 were followed on the second virtual machine up to step 9, where “Login to Existing Account” was clicked

2. Email and Master Password account details entered, clicked “Log In”
3. Browser extension enabled in Google Chrome

4. Clicked LastPass logo

5. Logged into account using same setup as VM1
6. Accounts have now been downloaded onto the second virtual machine.

7. Snapshot created to save data for later review

**Experiment 2: Test 2**

Social media account “Twitter” was removed from Virtual Machine 1, it should also have been removed from the Virtual Machine 2 database.

1. LastPass vault was loaded up on Virtual Machine 1, delete icon clicked on the “twitter.com” account. Clicked “Yes”
2. Password vault was loaded on Virtual Machine 2, Twitter is still present. Tried refreshing the password vault, Twitter was still present.

3. Closed the Google Chrome web browser and relaunched it, opening the password vault again and the Twitter account is no longer present.

4. Restored virtual machine to previous snapshot taken at the end of Experiment 6 and disconnected the virtual machine from the internet. Twitter is now back in the virtual machine’s password vault, meaning that the data is kept offline on the machine as well as being kept on the LastPass servers.
Experiment 3: Form Filling

Test 1
An address has been added to the LastPass account using the ‘Form Fills’ feature to make it easier to enter address information on web pages. This information should then have been stored in the SQLite database in appdata, along with being accessible from the browser interface.

1. LastPass loaded in Google chrome on Virtual Machine 1, Clicked “Form Fills” then clicked “Add Form Fill Profile”

2. Entered University of Derby information and clicked “Save”
3. University of Derby information is now available from the password vault.

4.Disconnected the virtual machine from the internet to see if the form information is accessible offline, or if it’s just accessible via the cloud. Information was present and editable while the virtual machine was offline.
Experiment 3: Test 2

Credit card information has been added to LastPass also using the Form Fill feature, again this information should be accessible through the web panel and via the SQLite database after decryption.

1. LastPass loaded in Google chrome on Virtual Machine 1, Clicked “Form Fills” then clicked “Add Form Fill Profile”

2. Entered testing credit card information and clicked “Save”
3. Test credit card information now available from the password vault

4.Disconnected the virtual machine from the internet to see if the credit card information is accessible offline, or if it’s just accessible via the cloud. Information was present and editable.

5. Snapshot created of the experiment for later review in a forensic toolkit
Experiment 4: Shared folder testing

Test 1
A second LastPass account has been created and signed into a third virtual machine, copied from the baseline. Account information for Reddit was shared with this second account to simulate an enterprise setting, where account information is shared across a department. The account details should appear in the second accounts “Shared” page, it is unknown if this information will be available locally meaning it had been saved and encrypted to the SQLite database, or just available online.

1. LastPass needs to be installed in order to make the second testing account. Following Experiment 1 to step 10 to install LastPass and create a new account using the Email “l.acott1@unimail.derby.ac.uk” and the master password of “testingpassword”. Clicked “Create Account”

2. Logged into the LastPass extension in Google Chrome using same settings as used previously
3. LastPass is now logged into on the third virtual machine using the new account

4. LastPass vault loaded on first Virtual Machine, clicked “Sharing Centre” then “Add Shared Folder”

5. Named folder “Test shared folder” and clicked “Create”

6. Clicked “Manage”
7. Sent an invite to the other text account, with permissions set to “Read Only” and “Hide Passwords” for now

8. Other account is now visible in the shared folder. Clicked “Save”

9. Dragged and dropped reddit testing account into “Shared-Test” folder, meaning that the folder should now be accessible from the second LastPass account on Virtual Machine 3
10. Account is now visible on the third virtual machine, and the password cannot currently be viewed.

11. Snapshot created for later forensic investigation to see if the password is visible via the SQLite database.
Experiment 4: Test 2

Next the “Hide Passwords” box will be unticked to see what happens on the third virtual machine. The expected result is that the password should now be available to be viewed from the web portal.

1. Unticked “Hide Passwords” option on Virtual Machine 1 for the second account, meaning the password should now be visible on Virtual Machine 3 using the second account.

2. Password now visible on Virtual Machine 3

3. To see if the shared account information is stored locally or online the internet connection was disabled on Virtual Machine 3, then the password vault was loaded and the account details were able to be viewed.
Other findings

Access history for the different logins and forms can be found by navigating to the password vault, then going to “More Options” then “History”. This can only be accessed online, but provides detailed, potentially forensically useful information on account logins including IP address information. Also while the option is available to copy a password to clipboard through the LastPass menu, while the “Hidden” option is enabled for an account this feature does not work but is still shown as available; when the “Copy password” button is clicked nothing happens.
9.4. Forensic analysis testing notes

Forensic Analysis of testing data images

Firstly the final images from each virtual machine were analysed for forensic artefacts, looking for metadata on the activity conducted within LastPass including an examination of the SQLite database used by LastPass to see what data is on the machines themselves. The images were first examined in Encase 7. Full version information for the resources used in this Appendix can be found in Appendix 9.1.

Virtual Machine Forensic Images

Firstly before any analysis could take place a forensic image must be created from the virtual machines hard disk, specifically from the latest snapshot. To do this FTK was used to mount the hard disk locally with write block enabled to prevent the data from being changed during this process. This is a critical part of acquisition for a forensic investigation to ensure that the integrity of the evidence is upheld. Once the drive had been mounted it was then forensically imaged as an evidence file, with the file extension E01. During this process the image was also verified and given an MD5 hash which can be compared against throughout the investigation to ensure again that the integrity is not compromised. **Note: The following steps show the entire process used to create the forensic image for Virtual Machine 1. The exact same steps were used to create forensic images for virtual machines 2 and 3, except selecting different virtual hard disk files to be mounted at step 2. Screenshots of the images for virtual machines 2 and 3 after creation and verification are shown at step 14 and 15.**

1. FTK Imager was loaded on the host system and the imaging mounting button was clicked.
2. The latest snapshot from virtual machine 1 was loaded into the system, meaning that when the forensic image was created, the latest data recorded onto the virtual machines hard disk was captured. The mount type was set to “Physical & Logical” with the mount method set to “Block device / Read Only” to ensure that the entire disk is imaged, and that it could not be written to while the data was being copied. The “Mount” button was then clicked to mount the disk.

3. Next the “Close” button was clicked since the drives are now mounted.
4. The next step is to add the mounted drive to FTK so that a forensic image could be created. To do this the “Add Evidence Item” button was clicked.

5. Physical drive was selected then “Next” was clicked

6. The virtually mounted drive was selected then “Finish” was clicked to add the drive to FTK
7. The virtual drive had now been added to FTK with both its partitions. Partition one contained system files used by Windows 10, Partition 2 contained the main operating system files along with all user files, including those used during testing to generate the LastPass data. Both partitions were added to the Forensic Image since this is what would happen if this hard disk was seized during a digital forensic investigation. The “Export Disk Image” button was clicked to do this.

8. The “Add” button was then clicked to select the image destination.
9. “E01” was selected as the destination image type since this evidence file type is fully compatible with Encase 7. “Next” was then clicked.

10. On this screen evidence information is normally entered for use in a forensic investigation. Since this project is to analyse the artefacts left behind and not to actually conduct a forensic investigation for court this form was filled out using basic information. Next was then clicked.
11. The image destination folder was then selected, and the file named “Forensic Image VM1, Snapshot 7”. “Finish” was then clicked.

12. “Start” was then clicked. Note that “Verify images after they have been created” has been selected to ensure that data integrity is kept throughout this process.
13. The forensic image has now been created and verified. This process was then repeated for virtual machines 2 and 3.

14. Screenshot of the forensic image for virtual machine 2 after creation and verification

15. Screenshot of the forensic image for virtual machine 3 after creation and verification
Adding Forensic Image Files to Encase for analysis and Evidence Processing

Each image file was added to the Encase for analysis, to look for forensic artefacts left behind by the simulated testing. The following steps show the process of adding the evidence to Encase, validating the evidence and processing the evidence for analysis.

1. Encase 7 was loaded and a new case was created by clicking “New Case”. Case name, path and information details were filled in then the “OK” button was pressed.

2. Next the “Add Evidence” button was clicked to add the forensic images created using FTK Imager to the case.
3. Since the forensic images have been saved as evidence files, the “add evidence file” button was clicked from the menu.

4. The first evidence file was selected and “Open” was clicked.

5. The evidence file was then verified as shown under the “File Integrity” field.
6. Next the evidence was processed by clicking “Process Evidence” then “Process” using the options listed in the screenshot below.

![EnCase Processor Options](image1)

7. Processing has now completed for VM1. Next added the evidence file for VM2 and ran the exact same processing options.

![EnCase Forensic](image2)
8. Processing is now completed for VM2. Next added the evidence file for VM3 and ran the exact same processing options.

Internet Examination

1. Firstly Virtual Machine 1 is opened and the records from the processing are viewed. Since Lastpass works through a web browser the first place looked at will be the internet tab. Clicked “View” > “Records” > “Internet” to get to the screen below. Next clicked “Internet” in the main window.
2. Microsoft Edge was only used during testing to download Google Chrome, however there is evidence that it was used during the LastPass installation as well which can be seen in the screenshot below. All the WebCacheV01 files contain references to the LastPass website, it also looks like the account information for both the Windows User and for the LastPass account was also recorded and submitted to LastPass in the form of the URL along with the date of installation – in this case being 02/05/2016. The WebCacheV01 file was located under Internet > Internet Explorer (Windows) > History > Daily. These files were last accessed on 02/05/16 at 12:04:13.

3. Google Chrome can be seen to have been downloaded from Microsoft Edge within the “Downloads” folder, WebCachev01.dat. This file was last accessed on 29/04/16 when the baseline virtual machine was created.
4. The first evidence of the LastPass vault being accessed can be found in the Google Chrome HTML Cache, which shows the tutorial / setup program and a basic HTML output of the document. “No Matching Results” is likely shown under “Remember every Password” since no passwords had been added to the account at this stage, and this appears to be a tutorial for how to use LastPass, which was shown the first time the vault was loaded on the virtual machine. This file was flagged for later review in Encase.

5. Further evidence of LastPass being used can be found in the Code section of the cache, but this reveals no useful forensic information besides access and request times. Data in the actual documents is code used at the time to display the task to the user. Since this LastPass code appears after and around Twitter code, it is safe to assume that this was used in the process of generating the new password, and then viewing the account in the password vault afterwards in Experiment 1, test 2.
6. No other useful information was found in this section besides more references to LastPass in terms of images cached, and several references to cookies for LastPass which could be used during the login procedure for the password manager, but these files were showing up as empty in Encase and therefore have little forensic value.

7. Next Virtual Machine 2 was analysed for similar artefacts, this time skipping the LastPass download since this would be identical to that of Virtual Machine 1. History for the extension use was discovered, again showing traces of the account name along with traces of the password vault being loaded locally with visit counts.

8. Virtual Machine 1 was compared for the history results, showing that the password vault had been visited 12 times.

9. Next Virtual Machine 3’s internet artefacts were analysed in a similar way, showing that the local vault had been accessed 7 times.
10. Next the system info parser records were reviewed for virtual machine 1, looking for evidence that the applications LastPass and Google Chrome had been installed. Data was present for Google Chrome however none was found for LastPass even though there is proof of this within the web browsers history, meaning that this module of Encase 7 processing cannot be relied on exclusively for finding all programmes installed on a forensic image. This was accessed by going to Records > Evidence Processor Module Results > System Info Parser – Records then expanding the tree view and selecting “Software” then “Installed Applications”. It’s worth noting that LastPass was present under “Installed Microsoft Applications” or “Uninstalled Applications” either. Same information was found for the other two virtual machines.
SQLite Database Analysis

1. Next the SQLite database used to save data locally by LastPass was examined. The database location is listed on the LastPass website under (Reference), it’s worth noting that the physical location of the data on an end-users computer will depend on what web browser they are using the password manager on, in this case it was Google Chrome. The location for this database file is: D:\Users\Test Account\AppData\Local\Google\Chrome\User Data\Default\databases\chrome-extension_hdokiejnpimakedhajhdlecepplioahd_0\1. The contents of the file can be viewed using the transcript window, however the data is not very readable, and a majority of the data is encrypted using the method stated in the Literature Review of this project. This data can be decrypted since the master password is known, if the master password was not known then the virtual machine would need to be run in a live environment in order to decrypt the password. This security vulnerability was shown in (reference), but it appears that it still has not been fixed. The username can clearly be seen from the transcript view.
2. To attempt to make the data more readable the SQLite database was exported from the virtual machine image, and opened using the SQLite viewer built in to OSForensics. EnCase does not appear to have a similar tool available, besides the ability to view the transcript, text and hex data of the file. The SQLite database files from the other two virtual machines were also exported for comparison. The database has 6 different tables for storing information named LastPassData, LastPassPreferences, LastPassSavedLogins, LastPassSavedLogins2, _WebKitDatabaseInfoTable_ and sqlite_sequence. On VM1 LastPassData contains 7 items. Without knowing information on how the software is coded it is difficult to determine what this data is actually used for, username_hash can be assumed to be the hash of the username for LastPass since it is identical in each entry. If it was different then it is possible it would be a hash of the various accounts stored on LastPass.
3. The next table, LastPassPreferences appears to contain data for the different options used by the passwordvault, some of these have the username_hash assumed to belong to the LastPass test account while others have no username at all. It can therefore be assumed that the preferences are split into account based options and default options for all users. “Remember Email” and “Remember Password” are shown with the value 1, which makes sense since these options were ticked on the LastPass interface during testing. What this means from a forensics standpoint is that if these options are ticked within the database, the full account information should be completely accessible if the virtual hard disk was to be loaded into a virtual machine and the forensic investigator had access to the Windows user account. To comply with best forensic practices this would need to take place on a copy of the virtual hard disk, with internet disabled to prevent the local database from being updated from the cloud. While further information can be found when the virtual machine is connected to the internet including history information on when the accounts were last signed in, the local accounts could be deleted remotely. However, re-copying the source data or rolling back the virtual machine to a snapshot before the internet connection was made should allow access to the original data as captured.

<table>
<thead>
<tr>
<th>id</th>
<th>username_hash</th>
<th>preference</th>
<th>prefvalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>37</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>loginBoxOrder</td>
<td>Name [red]</td>
</tr>
<tr>
<td>38</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>sharedFolderOrder</td>
<td>Name [red]</td>
</tr>
<tr>
<td>40</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>rememberOrder</td>
<td>Folder [red]</td>
</tr>
<tr>
<td>42</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>rememberEncryption</td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_len</td>
<td>12.0</td>
</tr>
<tr>
<td>47</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_hash</td>
<td>1.0</td>
</tr>
<tr>
<td>48</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_file</td>
<td>1.0</td>
</tr>
<tr>
<td>49</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_special</td>
<td>0.0</td>
</tr>
<tr>
<td>50</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_rgb</td>
<td>1.0</td>
</tr>
<tr>
<td>51</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_safety</td>
<td>0.0</td>
</tr>
<tr>
<td>52</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_password</td>
<td>1.0</td>
</tr>
<tr>
<td>53</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_pronounce</td>
<td>0.0</td>
</tr>
<tr>
<td>55</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>generate_word</td>
<td>0.0</td>
</tr>
<tr>
<td>59</td>
<td>0263b7a3136b8c23c000044701a07d072452c2914b76a645321932908</td>
<td>icons</td>
<td>1.0</td>
</tr>
</tbody>
</table>

4. The LastPassSavedLogins table appears to be empty which is surprising since this is where the account information for the services added to LastPass should be located according to (Reference). This means this data may only be able to be found through accessing the “Live” virtual machine. Searches will be conducted after analysis of each SQLite database has been completed.
5. The LastPassSavedLogins2 table contains the account information for the LastPass application. The username (Email) is clearly visible, however the password has been encrypted, as stated in (Reference). last_login field is populated, however it doesn’t appear to use Unix time stamp so the meaning of the data is unknown. The purpose of the “protected” field is also unknown.

<table>
<thead>
<tr>
<th>username</th>
<th>password</th>
<th>last_login</th>
<th>protected</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:100245383@inmail.derby.ac.uk">100245383@inmail.derby.ac.uk</a></td>
<td>W/9/B/MJ.c52W9hgtH8h3uape=</td>
<td>1462203171361</td>
<td>2</td>
</tr>
</tbody>
</table>

6. _WebKitDatabaseInfo did not contain any useful information so it was skipped.

database_sequence contained two entries, LastPassPreferences and LastPassData but the meaning of these figures is unknown.

<table>
<thead>
<tr>
<th>name</th>
<th>seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastPassPreferences</td>
<td>75</td>
</tr>
<tr>
<td>LastPassData</td>
<td>36</td>
</tr>
</tbody>
</table>

7. Next Virtual Machine 2’s SQLite database was loaded to be analysed. For the most part the data in the LastPassData table is the same as Virtual Machine 1’s, however the rsaeky and otp entries have swapped positions and the data for otp, rsaeky and key are different.
8. On the LastPassPreferences table a lot less entries are made under the username_hash for the user account, however this virtual machine was only used for one experiment so it is likely with more frequent use that this would change. Entries for rememberpassword and rememberemail are marked as 1 as with Virtual Machine 1, meaning this information could also be accessed from the interface on a live virtual machine without the investigator knowing the master password.

<table>
<thead>
<tr>
<th>id</th>
<th>username_hash</th>
<th>prefername</th>
<th>prefervalue</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>openpopoverHKeyCode</td>
<td>220.0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>openpopoverHKeyCode</td>
<td>off</td>
<td></td>
</tr>
</tbody>
</table>

9. LastPassSavedLogins was empty, identical to Virtual Machine 1. LastPassSavedLogins2 however contained the username and encrypted master password for the LastPass testing account. The encrypted password on this machine is different to Virtual Machine 1, meaning that the salt or seed value changes between installs. This is likely specified either in LastPassPreferences or in sqlite_sequence which can be used to decrypt the password when known as stated in (Reference).

<table>
<thead>
<tr>
<th>username</th>
<th>password</th>
<th>last_login</th>
<th>protected</th>
</tr>
</thead>
<tbody>
<tr>
<td>100243653@unились.яркие.ук</td>
<td>1a64f14f1b5b24b9b913df5d7d2a45c2</td>
<td>1462188010071</td>
<td>2</td>
</tr>
</tbody>
</table>

10. _WebKitDatabaseInoTable_ contained identical information to Virtual Machine 1. sqlite_sequence however contained different values, which supports the theory that these values could be used during the encryption process which would explain the fact that the encrypted master password is different on this virtual machine and that some of the values in LastPassData are also different.

<table>
<thead>
<tr>
<th>name</th>
<th>seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>LastPassPreferences</td>
<td>36</td>
</tr>
<tr>
<td>LastPassData</td>
<td>9</td>
</tr>
</tbody>
</table>
11. Finally the SQLite database for virtual machine 3 was loaded into OSForensics to be analysed. Completely different results are expected since LastPass was being used with a different account to the other two virtual machines. LastPassData was looked at first, showing a different username_hash as expected since this is testing account 2.

12. Next LastPassPreferences was loaded, rememberemail and rememberpass have values of 1.0 meaning the data can be accessed via live acquisition as explained previously.

13. LastPassSavedLogins is empty as with the other two virtual machines. LastPassSavedLogins2 contains the second accounts username and encrypted password.

14. As with the other two virtual machines _WebKitDatabaseInfo_ contain no useful information and was skipped. sqlite_sequence contained different values to the other two virtual machines, further supporting the idea that these are used as part of the encryption process, and are likely randomly generated at install or login.
Encase Keyword Search

1. Going back to Encase, to look for further artefacts several searches were conducted across all three evidence files. Firstly “lastpass” was searched and the results were saved under “Search – “lastpass”. 8712 items were found, with a total of 149,147 hits across the three virtual machines.

2. Files were then sorted by “File Created”, “Last Accessed” and “Primary Device” to get a timeline of events when scrolling through the data. Note: not all files had file creation or access dates, these files were shown first. Entered the MFT artefact and discovered an entry named vault.html. From this found a file address for the LastPass extension within appdata, in a similar location to where the database file was found. Note: This entry shows the extension local in virtual machine 2, the extension can be found in the same place on virtual machine 1 and 3.
3. Navigated to the location and found the installation files for the lastpass extension, no useful forensic data was found here besides version number 4.1.7_0.

4. Going back to the search results and looking at the MFT artefact records shows “1-journal” being created and written to several times during the testing period. The journal file is likely used to update the password manager’s sqlite database located previously while the database is in use. Again, these mft artefacts are present on all 3 virtual machines, the screenshot below shows them from virtual machine 1.
5. After skimming through the files nothing new was really found, besides generic lastpass web page caches that didn’t actually contain any user data on them. A new search was run for “1000243363” since this is the start of the username used for the testing account. 147 hits were found on 52 items across all 3 virtual machines. It will be interesting to see if there is any reference to that account on the third virtual machine since LastPass was logged into that computer using the second testing account, “l.acott1@unimail.derby.ac.uk” but data was shared between the accounts. The data was then sorted using the same variables: “File Created”, “Last Accessed” and “Primary Device”.

6. Microsoft-Windows-WinNet-Capture%4Analytic.evtx shows evidence of user account sign in using encryption and a private key. The master password isn’t actually listed but private key and a decryption key is listed, however this is likely the https SSL encryption for the connection to the server, not the encryption of the password itself.
7. After going through the rest of the search data nothing else new was found. Next trying a search of the master password “testingpassword” to see if there are any traces of that on the system. Index search revealed no results for this.

8. Next “l.acott1” was searched since this is in the username for the second testing account. Only records for this were found on virtual machine 3, despite the fact that the account was invited to the shared folder through virtual machine 1.

Finished forensically analysing the virtual machines in Encase, found all important artefacts left behind by the simulated use of LastPass.