Eötvös Loránd University
Faculty of Informatics

The quest of “dethroning“ passwords from (web) authentication schemes

Péter Burcsi
Assistant Professor at ELTE
Andreas Peter
Assistant Professor at University of Twente
Márk Szabó-Simon
Senior Software Engineer at LogMeIn

Antonios Papadopoulos
Computer Science

Budapest, 2019.

This thesis is dedicated to the memory of my late father.

Acknowledgement

This masters program has been a packed life adventure, where I got to re-discover myself through countless experiences and people. This thesis is the evidence that this journey is almost over and I would like to thank everyone that has contributed, in one way or another, for me to reach the finish line.

I would like to start by thanking Péter Burcsi for being my academic supervisor and professor during the last year of my studies. He is one of the most intelligent and good-hearted people I have met. Without his continuous guidance and support, I would not be where I am today. Also, a big thanks to my academic supervisor from the University of Twente, Andreas Peter, for supervising me and being my professor in one of the best courses I have taken.

I am also profoundly grateful to LogMeIn and LastPass for giving me the opportunity to be a part of their family while writing my thesis. Firstly, I would like to thank Márk Szabó-Simon for being my supervisor and helping me every step of the way. I would also like to thank Pedro, Bence, Robert, Feri, Boti, Andy and the many others I had the chance to work with while being a part of the most amazing engineering team.

Last but not least, heartfelt thanks go to all the people that believed and supported me the last two years. I would like to thank Zsolt, Leon and all my friends that helped me get back on my feet and were there when I needed them. I would also like to give a special thanks to Eszter Kiss for her immeasurable support and compassion, and EIT Digital Academy for letting me be a part of this ecosystem.

Contents

Chapter 1

Introduction

In today's society, passwords are an integral part of authentication. From logging in to your social media accounts to logging in your e-banking platform, a password is used somewhere through the process. However, selecting a secure password can be difficult as humans tend to use weak passwords, which they can easily recall from memory, and also have the tendency of using the same password for different websites. A solution to this problem has been the use of password managers, where a user needs to remember only one password to unlock the rest of their secure passwords. By using a password manager, the problem is narrowed down to the security of one master password. However, a master password is still a password at the end of the day, with the same fundamental flaws and attack surface.

A way of enhancing the security of passwords, is integrating them in an authentication setting where they are only used partly in the process. This kind of setting is called multi-factor authentication, where a combination of authentication factors can be used to verify a user's identity. The most commonly used type of multi-factor authentication is two-factor authentication where only two factors have to be provided by the user before they can be authenticated. Choosing the right two-factor authentication scheme for a specific purpose or product is not an easy task, considering the many options available nowadays.

This master thesis has two main objectives. The first is to provide a way of comparing and scoring two-factor authentication schemes in terms of their usability, deployability and security. To do so, an evaluation framework which has been adapted for the context of password management will be used. Using this framework, we will derive a model to score any two-factor authentication scheme, making comparisons between different schemes more rigorous and with less subjectivity. The second objective, is to provide a sample risk analysis for a password manager that is offering the option of two-factor authentication, using the FAIR risk framework. This analysis can be used as a baseline when conducting more realistic risk analyses on a password manager's security.

Throughout this thesis, we will be trying to answer the following research questions:

1. What is the trade-off between usability, deployability, and security when using extra software/hardware components in the password manager's authentication process and how can it be measured effectively?

2. How can a risk analysis framework be adapted and used in the context of password management and two-factor authentication?

The thesis is structured in the following way. In Chapter 2, we will explain the basic building blocks that will be used in the later chapters. We will explain basic authentication terminology and concepts, the usability-deployability-security(UDS) evaluation framework and the FAIR methodology. In Chapter 3, we will use the UDS framework and derive a model for comparing 2FA schemes. In Chapter 4, we will provide a sample risk analysis for a hypothetical scenario involving the LastPass password manager and a new 2FA scheme in their service offering. In the end, we will conclude with answers to our research questions and some general remarks.

Chapter 2

Preliminaries

A brief explanation of the main concepts will follow. By the end of this chapter, the reader should have a well-rounded understanding of multi-factor authentication, the evaluation framework used for ranking the schemes and the FAIR risk analysis methodology.

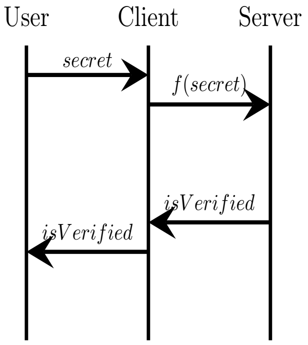
2.1 User authentication

In this thesis, the focus will be solely on human-by-machine authentication, which is the process of positively verifying the identity of a user in a computer system, as a prerequisite to allowing access to a system's resources[11]. Machine authentication will not be considered. From now on, every mention of authentication in the course of this paper, will refer to user authentication. Furthermore, we will borrow some terms from interactive proof systems when explaining some authentication protocols in order to describe the roles and exchanged messages.

Based on the artifacts used throughout the process, authentication can be divided into four distinct types:

Type I - Cognitive factors

In this authentication type, the user proves their identity by providing a piece of secret information from memory to the verifying party. Password-based authentication schemes are the most commonly used until today[12], where the user needs to recall a string of characters, numbers and/or symbols of arbitrary length every time they wish to authenticate. One of the major drawbacks of password-based authentication is the difficulty of remembering a lot of strong passwords[19], but this can be solved efficiently with the use of a password manager. Other common cognitive factors include Personal Identification Numbers(PINs), graphical passwords and lock patterns.



```
$\mathbf{0}$ : $User \rightarrow Client$: $secret$
$\mathbf{1}$ : $Client \rightarrow Server$: $f(secret)$

if $V(f(secret)) = 1$ then
    $isVerified = 1$
else
    $isVerified = 0$

$\mathbf{2.1}$ : $Server \rightarrow Client$: $isVerified$
$\mathbf{2.2}$ : $Client \rightarrow User$: $isVerified$
```

Figure 2.1: Interaction in Type I authentication schemes

Type II - Possession factors

In Type II authentication, the user proves ownership to a thing they own to the verifying party. This thing needs to be able to uniquely identify the prover and can be something tangible, like a document or security dongle, or something intangible, like a piece of software.

Official documents have been used for a long time to identify people, but have their downfalls. The main drawbacks of using a document, like a photo ID, for authentication is the time it takes to to verify its authenticity and association to the prover. This is why there has been a move towards more efficient solutions.

Smart cards have been used for a while due to their many benefits. They can automate the authentication process, decreasing the verification time and eliminating the need for a human verifier. However, they are bit lackluster when it comes to security[17], which has driven the development of more robust and secure solutions.

Security dongles are a step further down the road. They are specialized hardware devices, used solely for the purpose of authentication. Due to their specialized nature, they can be optimized for security while still maintaining their usability benefits. The most commonly used security dongles are one-time password(OTP) generators, and hardware authenticators.

Type III - Inherence factors

Type III authentication is all about being yourself. All the prover has to do is to either demonstrate average behaviour or provide temporary access to one of their physical unique

Figure 2.2: Type II authentication using a Yubikey 5 (Source: Yubico)

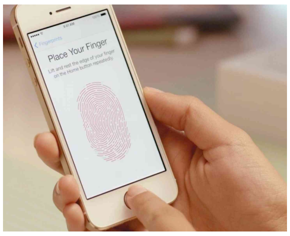


Figure 2.3: Type III authentication using the fingerprint scanner of an IPhone (Source: Business Insider)

characteristics to the verifier. Biometric authentication schemes fall in this authentication type, where specialized hardware is used to translate a prover's physical characteristic, like their iris or fingerprint, in a digital form, and then use that to verify their identity. Also, a slightly less accurate alternative to using physical characteristics, is using behavioural attributes like a person's voice or walking patterns.

Type IV - Location-based factors

Type IV authentication uses the location of the user to verify their identity. There have been proposals[9, 16] on how to achieve that for different scenarios and settings, and they are all relying on different implementations of a location signature. These are digital signatures generated by Location Signature Sensors(LSS) and can be used to verify a user's coordinates with a high accuracy.

2.2 FIDO Alliance

The FIDO Alliance is an industry consortiumn founded in 2013 by some of the biggest technology companies at the time with the goal to introduce new authentication standards, shifting away from the traditional password-based authentication. Since then, they have published a number of standards, with the most popular being WebAuthn, which was recently standardized by the World Wide Web Consortium[18], FIDO U2F and FIDO UAF. We will only focus on the latter two.

2.2.1 FIDO U2F

FIDO U2F is an authentication protocol which introduces a (cryptographic) hardware token as a second factor. There are only two operations described by the protocol, namely registration and authentication. The registration operation only happens once, when the user wants to register their token to a website that supports U2F, while the authentication operation is what happens every time the user needs to prove they possess their registered

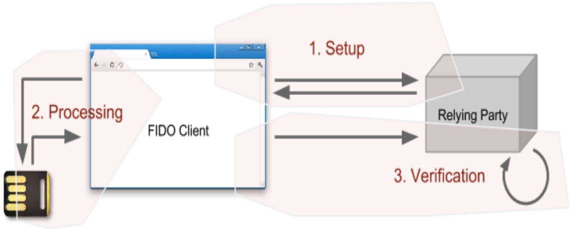


Figure 2.4: The FIDO U2F flow (Source: FIDO Alliance)

token. The protocol consists of three entities, the U2F device, the U2F client and the U2F server.

The U2F device is the hardware token itself. It needs to be able to generate origin-specific public/private key pairs and ECDSA signatures on the P-256 elliptic curve. The private keys are stored on the device, while the public keys are sent to the site during the registration operation alongside a unique Key Handle. Key Handles are strings of arbitrary length used for indexing the private keys on the device. During authentication, the user needs to authorize their request by performing a "test of user presence"(e.g. pressing a button on the Yubikey), before a signature using the private key indexed by the provided Key Handle is sent. An additional check of the origin of the request is performed, to defend against Man-In-The-Middle attacks. The U2F device can also be implemented in software, but that severely compromises the security of generating and storing the private keys.

The U2F server is queried by the site that requires authentication using the U2F protocol. Every time the user needs to authenticate using their registered device, the website sends its Key Handle to the U2F client and waits for the client to forward the U2F device's response. It then sends that response to the U2F server for validation. A website can run its own U2F server, or communicate with an existing one, through encrypted web requests. The website communicates with the U2F client for either registration or authentication through a JavaScript API.

The U2F client(usually a web browser) acts as an intermediary between the device and the server. It communicates with the U2F device through an OS API, which implements communication methods through USB HID, Bluetooth Low Energy and NFC as of now. The U2F JavaScript API is currently supported by the latest versions of Google Chrome, Mozilla Firefox, Microsoft Edge and Opera[1].

For a more detailed description of the protocol, one can refer to the standard's specifications[3]. Additionally, a flow diagram of the U2F protocol can be seen on Figure 2.4.

[1] `https://caniuse.com/#feat=u2f`

2.2.2 FIDO UAF

FIDO UAF is an authentication protocol which lets existing local authentication mechanisms, like biometric scanners, to be used by online services and websites as additional authentication factors. As in the U2F protocol, there are two operations, registration and authentication, that operate in a similar fashion. One of the main differences between the U2F and UAF protocols is that the latter could eliminate the need of password-based authentication altogether, while U2F, at its current state, is just enhancing its security. The protocol consists of three entities, the FIDO authenticator, the FIDO client and the FIDO server.

A FIDO authenticator can be any secure element that is used for local authentication (e.g. fingerprint scanner), which satisfies certain criteria as described in the relevant standard. It needs to be able to respond to cryptographic challenges, during registration or authentication, and be able to attest its type and capabilities to the FIDO Client. This is similar to how some handshaking network protocols work, where the client sends the cipher suites it supports, and the server decides on which one to use for communication. In our case, the FIDO authenticator can be queried for its supported ciphers and/or version, and the FIDO client can then decide on whether it can use that authenticator and how. In the FIDO UAF protocol, there can be more than one FIDO authenticators and not all of them need to be used.

The FIDO client acts as the intermediary between the authenticators and the FIDO server. It is responsible for interacting with the FIDO authenticators on the device through the Authenticator API and communicating with applications or web browsers that want to use the UAF protocol. Using this architecture, the protocol can be interoperable through many devices and system types. An important detail is that the FIDO client runs on the same device as the FIDO authenticator(s), which can have some security implications[13] that will be touched in the last chapter. The FIDO server works similarly with the U2F server described in the previous subsection.

For a more detailed description of the protocol, one can refer to the standard's specifications[4]. Additionally, a flow diagram of the registration operation of the protocol can be seen for reference at Figure 2.5.

2.3 The UDS framework

The usability-deployability-security framework was introduced by researchers from the university of Cambridge[6], and aims to quantify the performance of a user authentication scheme. It allows us to rate a scheme using a set of benefits from three different

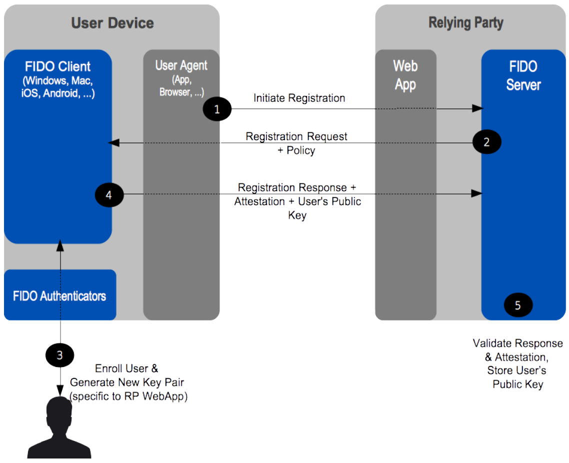

User Device
FIDO Client (Windows, Mac, iOS, Android, ...)
User Agent (App, Browser, ...)
Relying Party
Web App
FIDO Server
FIDO Authenticators
1
Initiate Registration
2
Registration Request + Policy
4
Registration Response + Attestation + User's Public Key
5
Validate Response & Attestation, Store User's Public Key
3
Enroll User & Generate New Key Pair (specific to RP WebApp)

Figure 2.5: The FIDO UAF registration flow (Source: FIDO Alliance)

dimensions(Usability-Deployability-Security). By doing do, we can rate and rank different schemes in a scientific and evidence-backed way.

There are three different distinct values that can be assigned to a benefit. If the benefit is fully satisfied by the scheme, then a value of 1 is assigned to it. Respectively, if the benefit is not present, at all, in a scheme, it is assigned a value of 0. In the case that the benefit is somehow satisfied, we will say that it is quasi-satisfied and give it a value of 0.5.

One of the main advantages of this framework, is the ability to fine-grain it for a specific context. This is done by assigning weights for each benefit, based on their perceived importance in that context. For example, the Memorywise-Effortless benefit could be assigned one of the highest values in the context of authentication schemes where the main users are elderly-aged people.

To calculate the score of a scheme, the following formula is used, where S_i is the score for scheme i, W_j is the weight of benefit j, and $b_{i,j}$ is the value of benefit j in scheme i (satisfied, quasi-satisfied, unsatisfied).

$$S_i = \sum_j W_j \cdot b_{i,j}$$

The absolute score of a scheme is not of much value on its own, but it provides a meaningful way of comparing and ranking schemes that share the same context and weight vector.

2.4 The Factor Analysis of Information Risk (FAIR) methodology

Factor Analysis of Information Risk (FAIR) is a risk management framework, originally published as a white paper by Jack Jones in 2005 and adopted by the Open Group as an

Usability	**Deployability**	**Security**
[U1]Memorywise-Effortless	[D1]Accessible	[S1]Resilient-to-Physical-Observation
[U2]Scalable-for-Users	[D2]Negligible-Cost-per-User	[S2]Resilient-to-Targeted-Impersonation
[U3]Nothing-to-Carry	[D3]Server-Compatible	[S3]Resilient-to-Throttled-Guessing
[U4]Physically-Effortless	[D4]Browser-Compatible	[S4]Resilient-to-Unthrottled-Guessing
[U5]Easy-to-Learn	[D5]Mature	[S5]Resilient-to-Internal-Observation
[U6]Efficient-to-Use	[D6]Non-Proprietary	[S6]Resilient-to-Leaks-from-Other-Verifiers
[U7]Infrequent-Errors		[S7]Resilient-to-Phising
[U8]Easy-Recovery-from-Loss		[S8]Resilient-to-Theft
		[S9]No-Trusted-Third-Party
		[S10]Requiring-Explicit-Consent
		[S11]Unlinkable

Table 2.1: Benefits of the UDS evaluation framework

open standard in 2009[2]. It consists of the following components:

- An **ontology** of factors and the relationships between them. This can be viewed as a tree structure, where the root node is the risk probability and we can derive its value by evaluating its children. The higher the level of a node, the more concise and easier it is to evaluate.
- **Methods** for measuring those factors.
- A **computational engine** which can derive risk by simulating the relationships in the ontology.
- A **scenario modeling construct** for formulating and analyzing risk scenarios.

Since 2005, the FAIR methodology has evolved significantly and a book about it has been written[10], which we have used as reference when conducting the risk analysis in Chapter 4. The most important concepts and factors that drive risk will now be explained.

2.4.1 Asset

“Assets are usually things that have intrinsic value, are fungible in some way, or create potential liability“[10]. Practically, it is anything that can have an effect on the primary stakeholders of the analysis when its state changes. In every risk analysis, the asset(s) need to be specified beforehand, as the measurements for our risk factors are highly correlated to the asset(s) under analysis. Reputation is usually not considered an asset in the context of risk management.

2.4.2 Threat

A threat can be anyone(or anything) that has a positive probability of causing harm(loss) to the asset under analysis, or to the risk analyst.

Threat communities(TCOMs)

Threats that share common characteristics can be analyzed together in groups labeled as threat communities. Individual members of those communities are called threat agents and can belong to one or many threat communities. The main reason of grouping threat agents together is because it is more efficient to forecast the behaviour of an average member of the group, than a specific member’s.

[2] `https://blog.opengroup.org/2017/01/24/what-is-open-fair/`

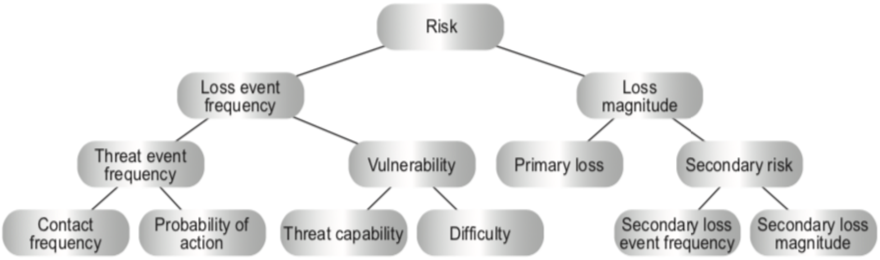


Figure 2.6: The FAIR ontology (Source: Elsevier[10])

Threat profiling

Threat profiling is the technique of identifying what are the common characteristics inside a threat community. Creating and maintaining threat profiles in an organization can help risk analysts to have a common ground when conducting their own analyses. An efficient technique to do so is described by Matthew Rosenquist in his whitepaper[14], which introduces the term of a Threat Agent Library(TAL).

2.4.3 Risk calculation

As described previously, the risk for a given scenario is calculated by evaluating its child nodes. In the FAIR ontology, these are the loss event frequency(LEF) and loss magnitude(LM). The complete ontology can be seen in Figure 2.6.

Loss event frequency (LEF)

It is “the probable frequency, within a given time-frame, that loss will materialize from a threat agent's action“[10]. In essence, it is how often is loss likely to happen, as a consequence of a threat agent's actions. In the cases where estimating loss event frequency directly is not feasible, it can be derived from its child nodes, threat event frequency and vulnerability.

Threat event frequency (TEF)

Threat event frequency is “the probable frequency, within a given time-frame, that threat agents will act in a manner that may result in a loss“[10]. The definitions of TEF and LEF seem interchangeable, but they have one fundamental difference. A loss event always results in a loss to the stakeholders, while a threat may(or may not). Threat event frequency can be estimated directly in most scenarios, but it can also be derived from its child nodes, contact frequency(CF) and probability of action(PoA). In the risk analysis at Chapter 4, we do not evaluate factors below threat event frequency.

Vulnerability

Vulnerability's definition in the context of the FAIR framework differs from the conventional definition used by many security professionals. In this case, it is “the probability that a threat agent's action will result in a loss“[10]. It can be estimated directly, or derived from its child nodes, threat capability(TCap) and difficulty(Diff).

Loss magnitude(LM)

Loss magnitude is “the probable magnitude of primary and secondary loss resulting from an event“[10]. Losses can either be primary or secondary, according to the type of stakeholder. Primary stakeholders are the individuals/organizations whose perspective is the focus of the risk analysis, while secondary stakeholders are everyone else who could be affected by a loss event and may react in a way that could result in a loss for the primary stakeholders. Loss can be of one the following forms:

- **Productivity:** Losses of this form are usually either losses from a company's reduced ability to execute their primary value proposition(s), or losses from employees being paid but being unable to perform their work duties.
- **Response:** Costs associated with managing the loss event. They can be mitigation/remediation costs, legal costs for responding to potential lawsuits, PR costs for trying to minimize the impact of the loss event to the media and more.
- **Replacement:** Costs associated with replacing a physical asset. The asset can be something easily replaceable, like a laptop computer, or even a person. The latter is usually the case when an insider is the threat agent.
- **Competitive Advantage:** Costs associated with reduced revenue streams due to the violation of the availability/confidentiality/integrity of a unique tangible/intangible asset. The asset is usually intellectual property of the company under analysis.
- **Fines and Judgments(F&J):** Costs associated with paying fines, or fees in general, as a consequence of the loss event.
- **Reputation:** Its effects are identical with the competitive advantage form of loss. The cause is usually reduced market shares, partners/suppliers stepping down and more.

Primary loss magnitude(PLM)

Primary loss magnitude is “primary stakeholder loss that materializes directly as a result of the loss event“[10]. Productivity, replacement and, sometimes, response costs fall in this category.

Secondary risk

“Primary stakeholder loss exposure(risk) that exists due to the potential for secondary stakeholder reactions to the primary event“[10]. Secondary risk can be treated as a separate risk analysis and has a similar tree structure as in figure 2.6. This risk must be derived from the secondary loss event frequency(SLEF) and secondary loss magnitude(SLM). Fines&Judgment, reputation and, sometimes, response losses are considered as part of the secondary loss magnitude.

Secondary loss event frequency(SLEF)

Secondary loss event frequency is “the percentage of primary events that have secondary effects“[10]. A loss event could or could not result into secondary losses and this frequency is trying to estimate that probability.

Measurement

Making estimates on the values going into each factor of the ontology can be a daunting task. This is usually the case when there is a lack of relevant data about what we are trying to estimate. To battle uncertainty, the creators of FAIR have introduced the measurement technique of “90% confidence“.

This technique works in rounds and should be used with a set of subject experts. It can help in reaching both accurate and precise estimates for frequencies and magnitudes. In the first round, the risk analyst introduces an absurd range as the initial estimate. In every subsequent round, the analyst decreases the range up to an arbitrary amount and the subject experts are asked how confident are they in that suggested range. If their confidence level is more than 90%, then the round is repeated for a stricter range. This keeps repeating until the confidence level of the subject expert is close to 90%. The range corresponding to the latest round is the output of this technique and can be used as an estimate for factors with high uncertainty.

2.5 The LastPass Password Manager

LastPass is a freemium password manager released in August 2008. Since its acquisition by LogMeIn in October 2015, it has been one of the main players in the password management market with an increasing number of active users. Each new user is assigned a digital vault, where they can store any sort of information, from plain passwords to addresses, credit card information and many more. The vault is encrypted with the user's master password, and is stored securely in the cloud, as well as in the user's authenticated devices.

LastPass does offer its users the possibility of enabling two-factor authentication for accessing their vaults, but the offered options are limited. The 2FA market is still on its infancy, but growing exponentially over the past years, which is why LastPass should embrace it and accommodate more 2FA options it in their product.

This thesis aims to provide a guideline on how LastPass, or any password manager, can integrate 2FA in their service offering, by providing a framework for evaluating 2FA solutions so executives can make calculated strategic calls.

Chapter 3

Ranking 2F authentication schemes for password managers

In this chapter, we will rank 2F authentication schemes using the UDS evaluation framework, in the context of password managers. We will start by choosing eight popular 2FA schemes that are currently in the market. For each scheme, we will check which benefits they satisfy and use that information as input to a deterministic scoring function defined by the framework. Then, we will choose an initial scheme ranking as our ground truth and use that for calibrating the framework for our context. After calibration, we will have a model for ranking other 2FA schemes, which we will use to pick the most effective 2FA scheme for password managers.

3.1 The 2FA schemes

For the purposes of this thesis, the 2FA schemes shown on Table 3.1 will be evaluated. The criteria for evaluating each benefit will only be extended to include quasi-satisfiability options for some specific benefits. The same criteria and interpretations will be used for every scheme in order to minimize the effect of a personal bias to a minimum when ranking the schemes. The reader can refer to the original paper for a thorough explanation on how rating the benefits works.

3.2 Benefit satisfiability

Due to our specific context, some benefits will be evaluated the same for every 2FA scheme. To avoid repetition, they will be mentioned here once and apply for all schemes. From the usability benefits, every scheme is Memorywise-Effortless since only one master password

Factor 1	Factor 2	Product
Password	SMS/Voice call	Discord
Password	Push notification	Microsoft Authenticator
Password	FIDO OTP	Yubikey 5
Password	FIDO UAF	NoPassword
Password	FIDO U2F HID	Yubikey 5
Password	FIDO U2F NFC/BLE	Yubikey 5
Password	TOTP(device)	OTP c100
Password	TOTP(app)	Google Authenticator

Table 3.1: A list of the 2F authentication schemes that will be evaluated and ranked

is needed for the prover to remember in order to "unlock" their personal vaults, Quasi-Infrequent-Errors as they might mistype their passwords and Scalable-for-Users as the password manager uses one password to "unlock" every other password, no matter how many websites the user is registered to. Furthermore, the password element makes each scheme not Physically-Effortless because the prover needs to type in their password, which is a physical task. Since the password manager acts as an intermediary between the user and other websites, every scheme is Server-Compatible. Furthermore, every scheme is Resilient-to-Leaks-from-Other-Verifiers because the verification between the user and the password manager has only one verifier, Resilient-to-Phishing since the password manager itself checks the website's authenticity before retrieving the user's credentials for it and Resilient-to-Theft because the master password can not be retrieved from the user's head without their consent. Also, they are all Requiring-Explicit-Consent because the prover needs to consciously type in their password, and Unlinkable as the authenticated website is not able to distinguish between a prover using a password manager and a prover who is not.

3.2.1 Password + SMS/Voice Call

In this scheme, the prover verifies their identity in two phases. In the first phase, they have to provide their credentials to the verifier, who cross-checks them with a database to validate them. If they are valid, then the verifier prompts the prover to either send him a text message or call him on his registered phone number. The text message contains a random code, which they need to send back to the verifier through a web form before a given time limit. If the code is correct, then the prover is authenticated. In the voice call scenario, the random code is communicated to the prover through text-to-speech instead.

Usability

The scheme is Quasi-Nothing-to-Carry since the prover only needs to carry their mobile phones with them, which is something they would probably carry anyway. It is Easy-to-Learn as the steps the user has to follow are straight-forward and would not cause issues for the average user. The scheme is not Efficient-to-Use as they have to wait for a text message or phone call which takes time, and not Easy-Recovery-from-Loss since in the case the user loses their phone, they will need to reset their 2FA option from their accounts, if such a flow exists.

Deployability

The scheme is Quasi-Negligible-Cost-Per-User since the more the users, the more the phone calls/text messages that are sent which increases the cost for the verifier. It is Browser-Compatible because the only extra component is an input field that the prover needs to fill with their received code. Also, it is Mature since it has been adopted and used for many years by different vendors. It is also Non-Proprietary since there is no active patent for it. The scheme is Accessible, as a blind or deaf prover can choose the method of receiving that random code.

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation since the random code sent by the verifier acts like a time-based one time password(TOTP) which is dynamic. It is Quasi-Resilient-to-Targeted-Impersonation since an attacker with enough knowledge about the user could perform a SIM swap attack[15], to bypass the second factor, and do a password recovery using the aforementioned information. The scheme is Resilient-to-Throttled-Guessing and Quasi-Resilient-to-Unthrottled-Guessing since the master password and the sent random code are limited length strings which could be enumerated by an unbounded adversary. It is also Quasi-Resilient-to-Internal-Observation since an adversary would need to have access on both the prover's phone and the machine they are performing the authentication on. The scheme is not No-Trusted-Third-Party since they need to trust the operator or service sending out the text messages and making the voice calls.

3.2.2 Password + Push Notification

In this scheme, the prover verifies their identity in two phases. In the first phase, a typical password-based authentication takes place where the user provides their credentials which need to be validated by the verifier. After validation, the verifier sends a push notification to the user's phone through their mobile app, which the prover needs to approve by clicking/pressing a confirmation button.

Usability

This scheme is Quasi-Nothing-to-Carry because the prover needs to carry their smartphone with them. It is Easy-to-Learn because it only takes one extra push of a button and Efficient-to-Use as that button press takes negligible time. The scheme is also Quasi-Easy-Recovery-From-Loss, as they would need to install the mobile app to a new device, disable the previous app instance and register the new one, which can take time.

Deployability

The scheme is Accessible since the prover only needs to provide their password and click on a button, Negligible-Cost-per-User since there is no cost for sending multiple push notifications requests over a network, Browser-Compatible since the push notification confirmation is happening on the backend, Mature because it's been used on iPhones since 2009[1] and Non-Proprietary as there is no active patent for it.

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation since the underlying information behind the confirmation button is usually encrypted and invisible to the user. It is also Quasi-Resilient-to-Targeted-Impersonation, since an attacker with the right information could register a new instance of the mobile app and have the push notifications sent there instead. The scheme is Resilient-to-Throttled-Guessing and Resilient-to-Unthrottled-Guessing since an adversary will need to crack the digital signature of the confirmation request in order to learn how to forge their own authentication approvals. It is Quasi-Resilient-to-Internal-Observation because an adversary would need to have access to both the user's phone that is running the authentication app and the machine they are performing the authentication on.

[1] https://pushcrew.com/blog/history-of-push-notifications/

3.2.3 Password + OTP USB

In this scheme, the prover goes through password-based authentication as a first step. If successful, they are then prompted to fill in a one time code, by plugging in their OTP USB device and pass a test of user presence on their device(usually tapping on it). The dongle then fills out the input field with the one time code by acting as a HID device(e.g. keyboard).

Usability

The scheme is Efficient-to-Use because a user can authenticate and register their OTP dongle in a reasonable time. It is not Nothing-to-Carry because they need to carry the OTP device with them. It is also not Easy-to-Learn because, without instructions, a user would have difficulty figuring out how the test of user presence operates, and not Easy-Recovery-from-Loss because, in the scenario where their OTP device gets stolen or lost, they would need to purchase a new one and register it to the password manager.

Deployability

The scheme is Accessible because the only actions required from the prover is providing their password and performing the test of user presence. It is also Browser-Compatible because the OTP device is filling in an input field by acting as a keyboard. It is also Mature because it has been out since 2008[2]. The scheme is not Negligible-Cost-Per-User because every user needs to have their own OTP USB, and Non-Proprietary because there is no active patent for it.

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation because of the random one-time code that the device fills in on its own, eliminating the possibility of a shoulder surfing attack. It is Resilient-to-Targeted-Impersonation because of the OTP device being unique and producing unique one-time codes. It is also Resilient-to-Throttled-Guessing and Resilient-to-Unthrottled-Guessing since an adversary will need to crack the digital signature of the one-time code in order to learn how to forge their own one-time codes for that unique device using the right parameters(e.g. incremental counter). The scheme is not Resilient-to-Internal-Observation because an adversary with a keylogger installed on the device the authentication takes place could retrieve both the user's password and their one-time code, creating the possibility of a man-in-the-middle attack. It is also not

[2] `https://www.yubico.com/about/about-us/yubico-fido-u2f-history/`

No-Trusted-Third-Party because the verification of the one-time codes is performed by a third-party server, usually owned by the device's vendor.

3.2.4 Password + UAF

In this scheme, the authentication is performed in two phases. In the first phase, the prover performs a typical password-based authentication. In the second phase, they get prompted to provide one of their registered biometric features for verification. The full process is described in Section 2.2.2.

Usability

The scheme is Quasi-Nothing-to-Carry because the user needs to carry their phone with them. It is Easy-to-Learn as most users already use biometric authentication to unlock their phones so, using it for authenticating to their password manager, is only a context switch. The scheme is also Efficient-to-Use because biometric authentication and password-based authentication are performed in a negligible time. The scheme is not Easy-Recovery-from-Loss because registering a new device is something which takes a lot of time and effort.

Deployability

The scheme is Accessible, because the user can choose their own biometric feature to use as their second factor. It is Negligible-Cost-Per-User because each user only needs to download an authenticator app on their phone. It is Browser-Compatible because the biometric authentication takes place through the user's authenticator app and not their browser. It is also Non-Proprietary because there is no active patent for it. The scheme is not Mature, because it has only been available since December 2014[3].

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation,Resilient-to-Targeted-Impersonation, Resilient-to-Throttled-Guessing and Resilient-to-Unthrottled-Guessing because of the biometric element involved in the authentication process. It is also Quasi-Resilient-To-Internal-Observation because an adversary, with control over both the device the authentication takes place and the user's phone which scans their biometric feature, could perform a man-in-the-middle attack and retrieve the user's vault, and No-Trusted-Third-Party because the verification of the biometric feature is being done on the user's phone.

[3] https://fidoalliance.org/overview/history/

3.2.5 Password + U2F HID

In this scheme, the authentication is performed in two phases. In the first phase, the prover performs a typical password-based authentication. In the second phase, they get prompted to plug in their U2F HID device to a USB port and perform the test of user presence to activate it. The U2F device then sends a HID interrupt, which gets forwarded to the server by the browser. The server then checks if the authentication request is valid, by verifying it through a third-party validation server, typically operated by the U2F device's vendor.

Usability

The scheme is Efficient-to-Use because a user can authenticate and register their U2F HID dongle in a reasonable time. It is not Nothing-to-Carry, because the user needs to carry the U2F HID dongle with them, and not Easy-to-Learn because of the test of user presence. The scheme is not Easy-Recovery-from-Loss because, if the device goes missing, the user needs to purchase a new one and register it to their account.

Deployability

The scheme is Accessible since the user only needs to type their password, plug in and activate their U2F dongle. It is also Non-Proprietary because anyone can implement a U2F HID compliant security dongle by sticking to the official specifications published by FIDO. The scheme is not Negligible-Cost-per-User since for every user group[4], a separate U2F dongle is required. It is not Browser-Compatible, because, at the time of writing, only Google Chrome and Opera browsers support the FIDO U2F API. For the same reason, it is also not Mature.

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation and Resilient-to-Targeted-Impersonation since an adversary needs the U2F device itself in order to log in as the prover. It is also Resilient-to-Throttled-Guessing and Resilient-to-Unthrottled-Guessing because an adversary would need to find the private key of the U2F device in order to forge authentication requests. The scheme is Resilient-to-Internal-Observation because of the U2F device, which is specialized hardware for the sole purpose of authentication, and expected to be secure against malware. The scheme is not No-Trusted-Third-Party due to the authentication requests being verified by a third-party server, usually operated by the hardware dongle's vendor.

[4] The same U2F dongle can be shared by multiple people, even for the same website

3.2.6 Password + U2F NFC/BLE

This scheme is almost identical with 3.2.5 with the only differences being in the way the U2F dongle performs the test of user presence and the way it communicates with the U2F client. In the NFC case, the user needs to place their hardware token close to their mobile phone's NFC reader for the authentication message to be sent. In the Bluetooth Low-Energy case, the user still needs to perform a test of user presence, but is not required to have their dongle that close to the device. Despite this difference, their benefits remain similar which is the reason we will be examining them together.

Usability

The scheme is Efficient-to-Use because a user can authenticate and register their U2F NFC/BLE dongle in a reasonable time. It is not Nothing-to-Carry because they need to carry the hardware token with them, not Easy-to-Learn because the way the U2F dongles operate is not straightforward for the average user, and not Easy-Recovery-from-Loss because, if the device is lost, a new one needs to be purchases and registered to the password manager.

Deployability

The scheme is Accessible, because the user only needs to type their password and activate their U2F dongle, and Non-Proprietary because the U2F NFC/BLE specifications are public information and anyone can implement a compliant token. The scheme is not Negligible-Cost-Per-User, because every user group needs to own their own U2F token, not Browser-Compatible, because at the time of writing, only Google Chrome and Opera browsers support the FIDO U2F API. The scheme is also not Mature, because, despite being out since 2014[5], it has been gaining attention only recently.

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation and Resilient-to-Targeted-Impersonation, since the U2F device itself is needed for an adversary to impersonate the user during authentication. It is also Resilient-to-Throttled-Guessing and Resilient-to-Unthrottled-Guessing, because an adversary would need to find the private key of the U2F device in order to forge authentication requests. The scheme is also Resilient-to-Internal-Observation, because the U2F dongle is specialized hardware used only for

[5] `https://www.yubico.com/about/about-us/yubico-fido-u2f-history/`

authentication, and not No-Trusted-Third-Party, because the authentication requests are validated by a third party server, usually operated by the hardware dongle's vendor.

3.2.7 Password + TOTP(device)

In this scheme, the authentication is performed in two phases. In the first phase, the prover performs a typical password-based authentication. After the verifier validates the prover's credentials, the prover is asked to provide a random code from their registered hardware TOTP device. The verifier then checks the code's validity and authenticates the prover. At a high abstraction, these TOTP devices are pseudo-random number generators(PRNGs), where the seed is known to the verifier. With this knowledge, the verifier can predict the sequence of generated random codes and check if the prover's code is a match.

Usability

The scheme is Easy-to-Learn, since the process is almost identical with 3.2.1, only differing in the device the user retrieves the code from. The scheme is not Nothing-to-Carry, because the user needs to carry the TOTP device with them, not Efficient-to-Use since they need to read the code from that device and type it to the client application, and not Easy-Recovery-from-Loss because the device needs to be replaced and re-registered in case of theft or loss.

Deployability

The scheme is Browser-Compatible, because the only addition on the client side is an input field. It is also Mature, because there have been standards for them since 1998[6] and Non-Proprietary, because there is no active patent for it. The scheme is not Accessible, because a blind user won't be able to read the codes from the device and not Negligible-Cost-per-User, because each user needs to own their own TOTP device.

Security

From a security standpoint, the scheme is Resilient-to-Physical-Observation because, even in the case of an adversary knowing the user's credentials and random code by filming their keyboard, they would still be unable to bypass the second factor. One could argue that, by observing enough codes, an adversary would be able to predict future codes, but that is an infeasible task. The scheme is also Resilient-to-Targeted-Impersonation, because each

[6] `https://tools.ietf.org/html/rfc2289`

TOTP device is unique, and Resilient-to-Throttled-Guessing. However, it is not Resilient-to-Unthrottled-Guessing, since the credentials and random code could be enumerated by an unbounded adversary. It is also not No-Trusted-Third-Party, because the vendor of the TOTP devices is usually the one to validate the codes.

3.2.8 Password + TOTP(phone app)

This scheme is similar to the previous one, with the main difference being the way of generating the TOTP codes. In this case, the user only needs to carry their smartphone, on which a TOTP application is installed and registered with the password manager.

Usability

The scheme is Easy-to-Learn, Quasi-Nothing-to-Carry, since the user needs their mobile phone with them, and Quasi-Easy-Recovery-from-Loss, since the user only needs to re-install and re-register the app on a new device, unlike in the previous 2FA scheme where a new TOTP device would need to be acquired. The scheme is not Efficient-to-Use.

Deployability

The scheme is Negligible-Cost-per-User, since additional instances of the TOTP app are essentially free, Browser-Compatible and Mature. It is also Non-Proprietary, because there is no active patent for it. The scheme is not Accessible, since, at the moment of writing, the code is only displayed on the user's screen.

Security

From a security standpoint, the scheme is Resilient-to-Physical observation and Quasi-Resilient-to-Targeted-Impersonation, since an attacker with enough knowledge about the user could register a new instance of the TOTP app. It is also Resilient-to-Throttled-Guessing and No-Trusted-Third-Party, since the seed of the app is set by the password manager. The scheme is not Resilient-to-Unthrottled-Guessing.

Scheme	U1	U2	U3	U4	U5	U6	U7	U8	D1	D2	D3	D4	D5	D6	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
Password + SMS/VoiceCall	★	✓	★	✗	✓	✗	★	✓	✗	✓	✓	✓	✓	✓	✓	★	✓	★	★	✓	✓	✓	✗	✓	✓
Password + Push notification	★	✓	★	✗	✓	✓	★	★	✓	✓	✓	✓	✓	✓	✓	★	✓	✓	★	✓	✓	✓	★	✓	✓
Password + OTB USB	★	✓	✗	✗	✗	✓	★	✗	✓	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✓
Password + UAF	★	✓	★	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	★	✓	✓	✓	✓	✓	✓
Password + U2F HID	★	✓	✗	✗	✗	✓	✓	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Password + U2F NFC/BLE	★	✓	✗	✗	✗	✓	✓	✗	✓	✗	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓
Password + TOTP(device)	★	✓	✗	✗	✓	✗	★	✗	✗	✗	✓	✓	✓	✓	✓	✓	✓	★	✗	✓	✓	✓	✗	✓	✓
Password + TOTP(app)	★	✓	★	✗	✓	✗	★	★	✗	✓	✓	✓	✓	✓	✓	★	✓	★	✗	✓	✓	✓	✓	✓	✓
✓ : Satisfied | ✗ : Not satisfied | ★ : Quasi-satisfied																									

Table 3.2: A list of the 2F authentication schemes that will be evaluated and ranked

3.3 Calibrating the framework for 2FA

3.3.1 Initial ranking

There cannot be an absolute ground truth for ranking the 2FA schemes, because there is always a percentage of subjectivity involved when choosing a scheme over another. For that reason, we have tried to integrate this subjectivity in our search for that truth, by gathering empirical evidence from recent studies and reports. We have used a weighted average method, using arbitrary weights based on the perceived accuracy of the source, to derive our final ranking.

Our first source of information was the annual report from Javelin[2], sponsored by FIDO Alliance, which talks about the state of strong authentication. The report tries emphasizing the difference between strong MFA and traditional MFA, with the latter lacking standardized cryptographic handshakes during authentication. Unfortunately, some of our examined 2FA schemes, like Password + SMS/Voice Call, and Password + OTP USB and Password + TOTP(device), do not include a strong cryptographic handshake.

From the report, we have used a set of figures to help us derive the initial ranking. Specifically, we have used the adoption rates of online and web authentication methods(figures 4 and 5), as well as the most desired authentication features among consumers(figure 8). Since all of our examined schemes share password-based authentication as their first factor, direct comparison of the second factors was possible. For the 2FA schemes that appear in multiple figures, we have assigned uniform weights to combine them.

A statistical report[1] from 2017 about the share of internet users who use 2FA in the United States was also used. Specifically, we only used the percentages of adoption rates for 2017, combining them uniformly with the data from the Javelin report. Lastly, the distribution of motivation across 2F technologies(figure 3) from another research study[8] in 2013 was also used. Using the aforementioned sources, we have derived the ranking in Table 3.3 as our ground truth.

3.3.2 Calibration

In order to rank the schemes, we have borrowed a few concepts from the field of machine learning and information retrieval. Our modeling problem can be reduced into a Learning-to-Rank problem, where we are trying to minimize a loss function. We have used the pointwise loss function, as described in [7], to get an estimation on how accurate or wrong a given ranking is. The formula for this function is the following:

Rank	**Scheme**
1	Password + UAF
2	Password + Push Notification
3	Password + SMS/VoiceCall
4	Password + TOTP(phone app)
5	Password + U2F NFC/BLE
6	Password + U2F HID
7	Password + TOTP(device)
8	Password + OTP USB

Table 3.3: The "ground truth" ranking of the 2FA schemes

$$L(ranking, targetRanking) = \sum_{i=1}^{8}(f(targetRanking[i]) - l(ranking[i]))^2$$

We will also use an algorithm called Simulated Annealing [5]. Simulated Annealing is a probabilistic technique used for finding the global minimum of a function. In our case, the objective is finding the global minimum of the aforementioned loss function and its corresponding weight vector. Specifically, we will set the initial ranking described in the previous section as our target with a loss value of 0. We will then initiate the Simulated Annealing process, by assigning a random weight vector and adjusting the individual benefit weight values in a probabilistic number of steps. The weight vector which corresponds to the resulting state of the Annealer will be regarded as the calibrated model. By using that model, anyone will be able score and rank any 2FA scheme, as long as the context and benefit satisfiability remain unchanged.

W(U1)	**W(U2)**	**W(U3)**	**W(U4)**	**W(U5)**
0.30377763945236528620	0.11105707472884073471	0.32104815702247711459	0.41694971043000658185	0.41429295613281406051
W(U6)	**W(U7)**	**W(U8)**	**W(D1)**	**W(D2)**
0.22690084166939402527	0.44802681476130597898	0.44802681476130597898	0.66829076454136108511	0.71557445387241568578
W(D3)	**W(D4)**	**W(D5)**	**W(D6)**	**W(S1)**
0.32447723099758106306	0.39430884941865092935	0.094588679924534903656	0.84797428574721375912	0.78959711867017402277
W(S2)	**W(S3)**	**W(S4)**	**W(S5)**	**W(S6)**
0.33070879920402865976	0.6842464124362598928	0.0083861917219962070	0.091887699540458124201	0.64590773303722560407
W(S7)	**W(S8)**	**W(S9)**	**W(S10)**	**W(S11)**
0.16249323547906019058	0.45654092988532698517	0.23454290038113475431	0.21966853759744010312	0.44024045172518340430

Table 3.4: The calibrated weight vector model

Algorithm 1 Weight calibration algorithm

```
Data: schemeNames, targetRanking, steps, Tmax, Tmin, step
Result: modelWeights
initialState = getRandomWeights()
rwc = Annealer(initialState, targetRanking, schemeNames, Tmax, Tmin, step)
loss = -1
while loss != 0 do
    state = rwc.moveState()
    loss = rwc.calculateLoss()
    if loss == 0 then
        modelWeights = state
    end
end
```

The above algorithm describes the process at an abstract level[7]. Using our designated ground truth, we derived the weight vector model, as shown in Table 3.4.

3.3.3 Results

Having the weight vector model, a company involved in password management can assess 2FA schemes directly and see how these compare to each other. To do so, they would need to assess a 2FA scheme's benefits and then plug in the benefit satisfiability vector alongside the calibrated weight vector to get score values with which they can rank schemes.

[7] For a more detailed python implementation, check `https://github.com/Uyatashi/2fa_ranking`

The assumptions on which the initial ranking was created will always have an element of subjectivity. Hence, the calibrated model should also be company and people-specific. By using the algorithm described in the previous subsection and the associated code, one can calibrate a weight vector using their own assumptions for the initial ranking.

Chapter 4

Risk analysis for the Password + UAF scheme

(Removed by request of LogMeIn for containing confidential information)

In this chapter, we will perform a quantitative risk analysis for the Password + UAF 2FA scheme, using the FAIR risk analysis methodology. We will start by introducing the scenario on which we will base our assumptions. Then, we will identify the asset at risk, treat communities, threat types and effects in order to scope our analysis. After scoping, we will make estimations on the frequencies and magnitudes of potential primary and secondary loss events. Finally, we will present and interpret the calculated risk probabilities, as well as provide some advice on how to reduce them.

4.1 Background

For the purposes of this analysis, we have created the following hypothetical scenario:

During a recent meeting, upper management of LastPass has decided to introduce an additional 2FA option into their password manager. After consulting the results of the 2FA ranking framework, they have decided to include the Password + UAF scheme in their service offering. However, before proceeding, they would like to know how this new component would affect their existing infrastructure and the confidentiality/availability/integrity of the user vaults. For that reason, they have assigned a team of risk analysts the task of assessing that future scenario and its implications for the company.

We will assume that there are one million user vaults stored in a data center, away from the company's premises. The company has decided that only users with vaults from that specific data center will be able to enable 2FA for their vaults. This would be a trial run and there would be a full roll-out after six months.

4.1.1 Asset at risk

Taking into consideration the size and impact of the organizations that are using LastPass(e.g. GitHub, Mozilla), the most important asset is the user vault and will be the main block in our analysis. Passwords for accessing important systems are stored within, which should remain both confidential and available for all users. Informational integrity of the vault is also quite important, as it could result in a long-term loss of availability and loss of internal money[1].

4.1.2 Threat Communities(TCOMs)

For the purposes of this thesis we will examine four major threat communities, cyber criminals, nation states, privileged insiders and non-privileged insiders. For each threat community, we have created a threat profile by building on existing templates from the FAIR book[10], as seen in tables 4.1 and 4.2. Only malicious(intentional) threat events will be considered for this analysis.

[1] Funds that are raised within the firm, and are used for its daily operations.

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Threat community	**Motive**	**Primary intent**	**Sponsorship**	**Preferred targets**	**Capability**	**Personal risk tolerance**	**Concern for collateral damage**
Cyber criminals	Financial	Maximize their profits by abusing users' sensitive information stored in their encrypted vaults	Non-state sponsored or recognized organizations(illegal organizations or gangs).	Individuals with high access privileges within their organization and wealthy individuals.	Professional hackers. Well-funded, trained and skilled. May employ relatively desperate actors with or without native skillsets.	Relatively high; however, willing to abandon efforts that might expose them.	Not interested in activities that expose themselves or others from their organization. Prefer to keep their identities hidden.
Nation states	Nationalism	Data gathering or disruption of critical infrastructure using information/credentials from LastPass vaults.	State sponsored, yet often clandestine.	High profile users with access privileges on important information systems	Highly funded, trained and skilled. Can bring a nearly unlimited arsenal of resources to bear in pursuit of their goals.	Very high; up to and including death.	Some, if it interferes with the clandestine nature of the attack.

Table 4.1: Threat profiles (1 out of 2)

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Threat community	Motive	Primary intent	Sponsorship	Preferred targets	Capability	Personal risk tolerance	Concern for collateral damage
Privileged insiders	Vindictive or personal gain	Gain retribution for perceived wrongs or to acquire money for alleviating a personal stressor by interfering with the availability/integrity of user vaults.	None. In rare cases, there is collusion between various bad actors, however, most are lone wolves	Users' vaults stored on servers where the attacker already has access.	Skillset varies. Tends to be very well versed in the systems to which they have access. Could have very high general computer science skills, yet may not be well-skilled in hacking.	Very low. Attacker typically pressured into scenario that compels them to act. This could be work related pressure (e.g. layoff), or personal pressure(e.g. personal financial stress).	In highly cohesive groups, there is very little tolerance for collateral damage, except in cases where the attacker feels wronged by the group.
Non-privileged insiders	Spite or personal gain	Gain retribution for perceived wrongs or to acquire money for alleviating a personal stressor by interfering with the availability/integrity of user vaults.	None. In rare cases, there is collusion between various bad actors, however, most are lone wolves	Users' vaults stored on servers where the attacker has or can somehow gain access.	Skillset varies. Likely to have limited access to systems. Most are likely to have limited skills required for pulling of a hacking attack, yet some may be studying and practicing on their own as a hobby or in pursuit of career progression.	Varies with personal circumstances.	Varies with personal circumstances.

Table 4.2: Threat profiles (2 out of 2)

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Asset at risk	**Threat community**	**Threat type**	**Effect**
Vault(s)	Cyber criminals	Malicious	Confidentiality
Vault(s)	Privileged Insiders	Malicious	Confidentiality
Vault(s)	Non-privileged insiders	Malicious	Confidentiality

Table 4.3: Scoping table for relevant risk scenarios

4.2 Scope

After identifying our primary asset at risk and the threat communities, we can create a scoping table with the potential risk scenarios. We have narrowed the scoping table down based on anticipated frequencies and magnitudes for each scenario.

Scenarios where confidentiality is breached by privileged or non-privileged insiders were excluded from the analysis, since the attack surface for bypassing/cracking a user's vault encryption has no differences between internal and external attackers, due to the zero-knowledge security model of LastPass[2]. We will not analyze the scenarios where the treat community is a nation state due to their lower threat event frequency. We expect that a nation state attacker could retrieve a user's credentials with less effort in a sophisticated spear phising attack than by having to compromise their user vaults. In the case where the user is using randomly generated secure passwords which are stored in their vault, the cost for an attack is still comparable considering that the attacker still needs to compromise the user's authenticating device.

We have also excluded the scenarios where availability or integrity is affected, as they are unlikely to be the objective of an attack. In the cases where a cyber criminal is the attacker, their main objective is to retrieve a user's credentials(break confidentiality). Making the credentials unavailable or corrupt for the original user would probably not be providing any net value for the attacker. The same assumption applies for insiders. The final scope can be seen in Table 4.3.

4.3 Scenario 1: Privileged insider - Confidentiality

A privileged insider in this scenario is anyone with access privileges on the data center storing the user vaults. Since we are dealing with confidentiality, the only privilege we will

[2] `https://www.lastpass.com/enterprise/security`

TEF Minimum	TEF Most Likely	TEF Maximum	Confidence
0.02	0.025	0.2	Low

Table 4.4: TEF estimates for scenario 1

be considering is read access.

4.3.1 Loss event frequency(LEF)

Given our hypothetical background, no loss events have occurred from privileged insiders at LogMeIn, so we will have to derive the loss event frequency from threat event frequency and vulnerability.

Threat event frequency(TEF)

To make an estimation for TEF, we have consulted with subject experts within LogMeIn to pick their minds on what the range of frequencies might be. To do so, we have started with absurd estimates and narrowed them down using the 90% confidence technique, described in 2.4.3. Our initial range was:

- Minimum: once every 100 years
- Maximum: five times a year

Taking in account the number of people having read access on the data center, their years at the company, their financial status and the experts' inputs, we have narrowed down the range to:

- Minimum: once every 50 years
- Maximum: once every 5 years

For the most likely frequency, we have chosen an estimate near the minimum for a couple of reasons. First and foremost, the people with access privileges on the data center are a select few DevOps engineers, that have been through thorough background checks and have worked for the company for at least five years(on average). Also, the company is paying them above market value, so financial distress is highly unlikely to be the cause of a threat event.

Due to the lack of data for the specific threat community, we have assigned a confidence of low for our estimates, as seen in Table 4.4.

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Vuln Minimum	Vuln Most Likely	Vuln Maximum	Confidence
1%	5%	10%	High

Table 4.5: Vuln estimates for scenario 1

Vulnerability

Vulnerability was trickier to estimate. In order for a threat event to escalate into a loss event, the attacker would need to compromise the target user's phone. Having access to the encrypted vault means they would also have access to the associated metadata and, potentially, could leverage that to launch a spear-phising attack. By compromising the target's phone, the attacker can indirectly derive the key used to encrypt/decrypt[3] their vault.

Nevertheless, in order for such an attack to be successful, there are many conditions which need to be satisfied. First, the privileged insider needs to be technically capable of launching such an attack which, according to our threat profile, is highly unlikely. Secondly, the user's phone needs to be compromised. There are many ways for this to happen but, the most likely one, is when the user has rooted their smartphone device and have authorized a malicious app with inappropriate privileges, or when the user hasn't upgraded their phone's operating system and the attacker is able to elevate their privileges through a kernel exploit.

Taking the above into account and after consulting with people from the LastPass security team, the risk analysts have made the estimates seen in Table 4.5.

4.3.2 Loss Magnitude

In order to derive the loss magnitude for this scenario, we have separated the costs associated with primary or secondary losses and have made estimations about their values.

Primary loss

To make estimates on the primary loss, we have looked into each form of loss separately.

Productivity

No operational disruption would occur if a loss event took place, due to the fact that the data center is isolated from the rest of the infrastructure. Therefore, we will not consider any productivity loss for this scenario.

[3] The vaults are encrypted using symmetric encryption.

Loss Type	Minimum	Most Likely	Maximum	Confidence
Primary replacement	€30,000	€40,000	€70,000	High

Table 4.6: Primary LEF replacement estimates for scenario 1

Replacement

In the case of a loss(or even a threat) event, the perpetrator would be terminated and, probably, replaced by LogMeIn. The associated cost in that case would be classified as a replacement cost. For estimating the costs for a terminated employee of this seniority, we have consulted with people from the HR department of the company. We came up with the esimates in Table 4.6.

Response

The primary response costs associated with such a loss event are forensic/investigation costs and person-hours spent in meetings regarding the incident.

Regarding the forensic/investigation costs, there are two groups of employees that are involved. The first group is people from the security team of LogMeIn with experience in forensic analyses, who will be investigating the incident during work hours. The second group consists of people in middle and upper management, as well as some executives and engineers.

For a team of five forensic experts, we have made the following estimations regarding the time spent on investigating the incident:

- Minimum: 16 hours
- Most likely: 40 hours
- Maximum: 100 hours

Assuming that the average hourly rate for the first group is €50, we were able to calculate the primary response costs for this scenario. All the primary loss costs can be seen at Table 4.7.

Secondary loss

To make estimates on the secondary loss, we have first identified the relevant secondary stakeholders for this scenario. The main ones are customers with premium or enterprise

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Loss Type	Minimum	Most Likely	Maximum	Confidence
Primary replacement	€30,000	€40,000	€70,000	High
Primary response	€4,000	€10,000	€25,000	Medium
Sum	€34,000	€50,000	€95,000	Medium

Table 4.7: Primary loss estimates for scenario 1

accounts and regulators. We have excluded users with free and team accounts from our analysis. Having identified the stakeholders, we went through each form of loss separately, as in the primary loss section.

Response

We have included the following secondary response costs in our analysis:

- Meeting costs
- Customer notification costs
- Customer support costs
- Legal costs
- PR costs

When estimating the primary response costs, we mentioned a second group of employees who are involved. The costs associated with that group are classified as secondary response costs, as they are not directly related to the loss event itself, but to its repercussions. For a group of 150 employees, we have made the following estimations regarding person-hours spent in meetings and can be found below:

- Minimum: 10 hours
- Most likely: 40 hours
- Maximum: 120 hours

Assuming that the average hourly rate for this group is €70[4], we have calculated the secondary response costs for meetings to be the following:

[4] A separation of the costs for the different subgroups would not make a significant difference in this analysis.

- Meeting costs minimum: €70 × 10 × 150 = €105,000
- Meeting costs most likely: €70 × 40 × 150 = €420,000
- Meeting costs maximum: €70 × 120 × 150 = €1,260,000

The number of compromised user vaults is also of very important for this risk analysis. Since the difficulty of compromising a user vault remains the same per vault attacked[5], we have made the following estimates about the number of compromised vaults per attack:

- Compromised vaults minimum: 0
- Compromised vaults most likely: 1
- Compromised vaults maximum: 1,000,000

For estimating the notification costs, we have consulted subject experts within LogMeIn and our own intuition. It is important to separate the customers into groups for this scenario. Let's assume that 500,000 of the user vaults stored in the data center belong to individual customers with premium subscriptions and the remaining 500,000 belong to enterprise users. The average enterprise user is typically not involved directly with the subscription. We will be considering as enterprise customers, the people in charge of the enterprise subscription for their company. For the 500000 enterprise user vaults, we are assuming that there are 20 enterprise customers. We have estimated that the cost per notification for customers with a premium subscription is €200 and for customers with an enterprise subscription is €50.

- Customer notification costs minimum: €0
- Customer notification costs most likely: €2 × 1 = €2
- Customer notification costs maximum: €2 × 50,0000 + €200 × 20 = €1,004,000

We have also included customer support as part of our secondary costs for our analysis. To calculate these costs, we have consulted with the LastPass support team to get an estimate on the cost per phone call and the percentage of affected customers that would make that call. As it turned out, the cost per call was estimated to be around €15(taking account of the hourly rate and the time spent per call) and the percentage of affected customers that would engage in such a call is around 70%. The costs associated with other

[5] For every vault, the owner's smartphone needs to be compromised.

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

means of communication(e.g. email) can be calculated similarly, but won't be part of the analysis.

Having the aforementioned assumptions in mind, we have made the following cost estimates:

- Customer support costs minimum: €0
- Customer support costs most likely: €15 × 1 × 0.7 = €10.5
- Customer support costs maximum: (€15 × 500,000 + €15 × 20) × 0.7 = €5,250,210

Taking into account the estimates on the number of compromised vaults, we have engaged the legal team of LogMeIn into giving us estimates on filing fees costs, court costs, person-hours for legal representation on trials and other relevant expenses. In the most likely scenario of 1 compromised vault, we have assumed that there would probably be no legal costs since the user will have been partly at fault for having their device compromised. Having access to the encrypted version of the vault on its own wouldn't be enough for a privileged insider. Our estimations were the following:

- Legal costs minimum: €0
- Legal costs most likely: €10,000
- Legal costs maximum: €500,000

We have also included PR costs in our analysis. To get these estimates, we have consulted with representatives from the marketing and legal departments and were able to come up with the following numbers:

- PR costs minimum: €0
- PR costs most likely: €0
- PR costs maximum: €1,000,000

We have added up and rounded all the secondary response costs, as seen in Table 4.8.

Fines & Judgments

Fines and judgment loss estimates were more straightforward. After consulting with people from the legal department of the company, we were able to define the range but had some difficulty estimating the most likely value. For that, we have made a similar assumption as in the PR costs, that a compromise of 1 vault, would most likely incur no fines.

Response cost	**Minimum**	**Most Likely**	**Maximum**	**Confidence**
Meetings	€105,000	€420,000	€1,260,000	Medium
Customer notification	€0	€2	€1,004,000	Medium
Customer support	€0	€10.5	€5,250,210	Medium
Legal	€0	€10,000	€500,000	Medium
PR	€0	€0	€1,000,000	Medium
Sum	€105,000	€430,011	€9,014,210	Medium

Table 4.8: Secondary response cost estimates for scenario 1

- F&J costs minimum: €0
- F&J costs most likely: €0
- F&J costs maximum: €3,000,000

Reputation

We have only included market share costs as the secondary reputation costs for this analysis. We have excluded capital costs and costs of acquiring or retaining employees for simplicity. The results of our analysis would not have differed much if were to include them.

Market share costs are the costs that would incur from losing customers in the password management market after a loss event. To calculate them, we have consulted with subject experts from the financial team to get some figures on how much is a customer worth for the company and what percentage of affected customers would decide to cancel their subscriptions. A premium customer would be worth €24 and an average enterprise customer[6] would be worth €1,800,000[7], for the period of one year. For the second figure, the percentage we got was close to 90%.

- Market share costs minimum: €0
- Market share costs most likely: 1 × €24 × 0.9 = €21.6
- Market share costs maximum: (500,000 × €24 + 20 × €1,800,000) × 0.9 = €43,200,000

All secondary loss estimates have been added up and rounded in Table 4.16.

[6] An average enterprise customer with a company of a size of 25,000 employees.

[7] 25,000 vaults × €72(annual cost per vault user for enterprise subscriptions)

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

SLEF Minimum	SLEF Most Likely	SLEF Maximum	Confidence
98%	99%	100%	Medium

Table 4.9: SLEF estimates for scenario 1

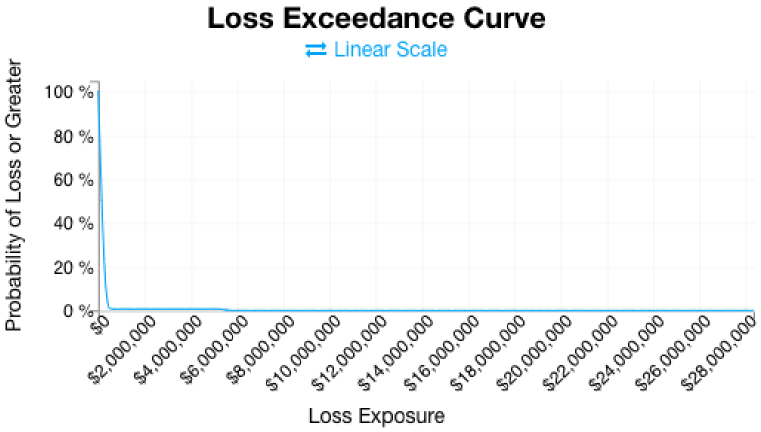


Figure 4.1: The loss exceedance curve for scenario 1

Secondary loss event frequency(SLEF)

Having estimated the secondary losses for this scenario, we would also need to calculate how often they occur in the case of a loss event. Due to the sensitive nature of the contents of a user vault, secondary losses would be triggered almost every time. After careful consideration, we have come with the SLEF estimates in Table 4.9.

4.3.3 Annual loss exposure

Having derived all the necessary risk factors of the FAIR ontology, we have used FAIR-U to calculate the annual risk for our scenario. FAIR-U is a simple computational engine for the FAIR framework, provided by RiskLens[8]. The loss exceedance curve seen in Figure 4.1 is a visualization on how likely a loss of a given magnitude is likely to occur. The average annual loss for this scenario is estimated to be €39,800, with the minimum being €0 and the maximum being €28,100,000.

In Tables 4.10 and 4.11 are the simulation results for primary and secondary loss events respectively, for the period of 1 year. Finally, the vulnerability for the scenario has been estimated to be 5.36%.

[8] https://www.risklens.com/blog/introducing-fair-u-the-free-way-to-try-fair/

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

	Minimum	Average	Maximum
Loss Events / Year	0	0	1
Loss Magnitude	€35,200	€53,500	€79,100

Table 4.10: Simulation result for primary loss events (scenario 1)

	Minimum	Average	Maximum
Loss Events / Year	0	0	1
Loss Magnitude	€250,800	€9,400,000	€39,700,000

Table 4.11: Simulation result for secondary loss events (scenario 1)

4.4 Scenario 2: Cyber criminal - Confidentiality

In this scenario, member(s) of the cyber criminal threat community are trying to gain illicit access to one or more user vaults in order to obtain their secrets(passwords, bank card information, addresses and more).

4.4.1 Loss event frequency(LEF)

We weren't able to estimate the loss event frequency directly, due to the lack of relevant data on past attacks. LastPass has experienced loss events from this threat community in the past(2011[9] and 2015[10]), but this data could not add much value in our analysis for two reasons. First and foremost, both attacks took place before the acquisition of LastPass by LogMeIn, so loss magnitude estimates are not relevant anymore. Also, technical details regarding the attack are outdated.

Threat event frequency(TEF)

In our hypothetical scenario, we have consulted with people from the security assurance team of LogMeIn to get estimates on the threat event frequency range. There was a plethora of data available, so there was no need to employ the 90% confidence technique. The range we ended up with was:

- Minimum: once a month
- Most likely: once a day
- Maximum: 2 times a day

[9] https://www.pcworld.com/article/227268/lastpass_ceo_exclusive_interview.html

[10] https://blog.lastpass.com/2015/06/lastpass-security-notice.html/

TEF Minimum	TEF Most Likely	TEF Maximum	Confidence
12	365	730	High

Table 4.12: TEF estimates for scenario 2

Vuln Minimum	Vuln Most Likely	Vuln Maximum	Confidence
0%	0.1%	5%	High

Table 4.13: Vuln estimates for scenario 2

These estimates correspond to threat events on the 1,000,000 vaults of our analysis, and do not represent threat events to any vault from the entire cyber criminal threat community. They have been translated into frequencies at Table 4.12.

Vulnerability

To estimate vulnerability, we have extended some of the assumptions from the previous scenario. An attacker from this threat community would have access to neither the encrypted user vaults nor their associated metadata. After careful consideration, we have come up with the folllowing attack scenario as the most probable to happen.

A blind attack[11] towards vulnerable smartphone devices would first take place. If one of the victims of that initial attack has a user vault at the data center under analysis, then the attacker would be able to compromise the victim's vault through their smartphone, assuming that its their registered UAF device. After compromising the device and, if necessary, the device's secure storage to get the UAF authenticator's private key for LastPass, they could retrieve the victim's master password by installing spyware on their smartphone. Having both the master password and the private key, the attacker could derive the decryption key for the victim's vault.

Based on these assumptions and the lack of successful attacks against LastPass since the acquisition, we have assigned the vulnerability range estimates seen in Table 4.13.

4.4.2 Loss Magnitude

As in the privileged insider scenario, we went through the primary and secondary losses and made estimations about their values. We were able to reuse estimates from the previous scenario, as most of them were also applicable here.

[11] An attack aimed to a set of targets and not to a specific member of that set.

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Loss Type	Minimum	Most Likely	Maximum	Confidence
Primary loss	€4,000	€10,000	€25,000	Medium

Table 4.14: Primary loss estimates for scenario 2

Primary loss

We have looked into each form of loss separately to estimate the primary loss for this scenario.

Productivity

The same assumptions as in the first scenario apply, so we haven't considered any productivity costs.

Replacement

Since we were dealing with external attackers, the replacement costs associated with recruitment processes were not applicable. We have assumed that there are no replacement costs.

Response

We have used the same estimates for the primary response costs, as in the first scenario. The primary loss can be seen at Table 4.14. One can notice that it is lower than in the privileged insider scenario, because of the absence of replacement costs.

Secondary loss

The secondary stakeholders are, again, customers with premium or enterprise accounts and regulators. Due to the nature of the threat community, the loss estimates which involved regulators were larger and more severe than in the privileged insider scenario. To make a precise estimation, we went over each relevant form of loss.

Response

The secondary response costs we have included in our analysis were the same as in the first scenario. Minimum and maximum values remained the same for most of them, but the most likely values differed due to the motives of this threat community and its higher risk tolerance(compared to privileged insiders).

We have raised the most likely estimate for meeting hours to 2 weeks, for the same employee count(150 employees), and kept the same minimum and maximum values from the previous scenario. The estimated costs for meetings were the following:

- Meeting costs minimum: €70 × 10 × 150 = €105,000
- Meeting costs most likely: €70 × 80 × 150 = €840,000
- Meeting costs maximum: €70 × 120 × 150 = €1,260,000

We have increased our most likely estimate for the the compromised vaults count, as cyber criminals have a higher technical capability than privileged insiders. A threat agent from that community could utilize device or software specific flaws/bugs, to infiltrate the phones of one or more users with a vault in the data center under scope. Here are our estimates:

- Compromised vaults minimum: 0
- Compromised vaults most likely: 5
- Compromised vaults maximum: 1,000,000

For estimating the customer notification and support costs, we have used the same set of assumptions as in the privileged insider scenario and the same customer distribution. Given our new estimates on compromised vaults, we have calculated the most likely costs to be the following:

- Customer notification costs most likely: €2 × 5 = €10
- Customer support costs most likely: €15 × 5 × 0.7 = €52.5

For the legal and PR costs, we have consulted people from the marketing and legal departments at LogMeIn, to get their estimates on this new scenario. As it turned out, minimum and maximum cost estimates didn't change, but the most likely ones did significantly. In retrospect, it makes sense for a company to be held more liable against loss events originating from cyber criminals rather than privileged insiders as, for the latter, there is not much a company can do to defend against. Our most likely estimates for these costs were:

- Legal costs most likely: €100,000
- PR costs most likely: €200,000

All secondary response costs were added up and rounded, as seen in Table 4.15.

DISCLAIMER: The risk analysis in this chapter is **fictional**. All figures and numbers in the analysis are **not real** data and shoult not be associated with LogMeIn's operations.

Response cost	**Minimum**	**Most Likely**	**Maximum**	**Confidence**
Meetings	€105,000	€840,000	€1,260,000	Medium
Customer notification	€0	€10	€1,004,000	Medium
Customer support	€0	€52.5	€5,250,210	Medium
Legal	€0	€100,000	€500,000	Medium
PR	€0	€200,000	€1,000,000	Medium
Sum	€105,000	€1,140,063	€9,014,210	Medium

Table 4.15: Secondary response cost estimates for scenario 2

Secondary Loss Type	**Minimum**	**Most Likely**	**Maximum**	**Confidence**
Response	€105,000	€1,140,063	€9,014,210	Medium
F&J	€0	€1,000,000	€3,000,000	Medium
Reputation	€0	€108	€43,200,000	Medium
Sum	€105,000	€2,140,171	€55,214,210	Medium

Table 4.16: Secondary loss estimates for scenario 2

Fines & Judgements

F&J most likely loss was the only estimate that was changed from the previous scenario. Given the nature of the attacker(external), even for a single compromised vault(let alone 5), there could be hefty fines, even if the right course of remediation actions is followed. With the help of people from the legal team, we have arrived to an estimate of €1,000,000 as the most likely cost.

Reputation

We have re-used the same set of assumptions when estimating the secondary reputation loss. For the market share costs, we have adjusted our most likely figure to its corresponding compromised vaults estimate, as seen below:

- Market share costs most likely: 5 × €24 × 0.9 = €108

All secondary loss estimates for this scenario have been added up and rounded in Table 4.16.

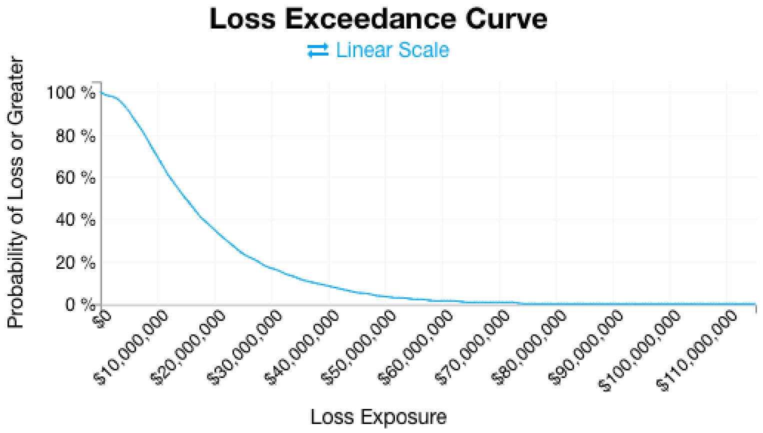


Figure 4.2: The loss exceedance curve for scenario 2

	Minimum	Average	Maximum
Loss Events / Year	0	1.76	4
Loss Magnitude	€4,100	€11,500	€24,000

Table 4.17: Simulation result for primary loss events (scenario 2)

Secondary loss event frequency(SLEF)

We have kept the same SLEF estimates as in the previous scenario, as a loss event would have secondary losses almost every time. This is mainly due to the fact that, if there is a successful compromise of a user vault and is detected by the company, then there is a legal obligation to disclose this incident to secondary stakeholders.

4.4.3 Annual loss exposure

Having provided our estimates for this scenario to the FAIR-U simulator, we got the loss exceedance curve seen in Figure 4.2. The average annual loss for this scenario is estimated to be €18,600,000, with the minimum being €0 and the maximum being €114,000,000.

In Tables 4.17 and 4.18 are the simulation results for primary and secondary loss events respectively, for the period of 1 year. Finally, the vulnerability for the scenario has been estimated to be 0.48%.

	Minimum	Average	Maximum
Loss Events / Year	0	1.74	4
Loss Magnitude	€615,600	€10,600,000	€42,600,000

Table 4.18: Simulation result for secondary loss events (scenario 2)

The third scenario would not have contributed a greater risk than scenario 2, so it has been skipped from this analysis. The reasoning behind our decision is that the non-privileged insider threat community has the same attack surface as cyber criminals, but with a lower threat capability.

4.5 Risk interpretation

Having analyzed both scenarios, we can make some conclusions about the risk, after integrating the password + UAF scheme as a a second factor, in the LastPass password manager. Despite the estimated most likely primary loss being lower in the privileged insider scenario, we are more concerned about the scenario involving cyber criminals. The loss exposure in the latter is about 467% higher than in the first scenario, because of the significantly higher secondary loss. Considering that the main value proposition of LastPass is keeping people's sensitive data safe, a higher secondary loss makes perfect sense.

We have not discussed any controls in this analysis. Better controls could help LastPass reduce its vulnerability percentage and, subsequently, its loss events. They could also, indirectly, reduce the secondary loss from legal and F&J costs, as the liability would be shifted more towards the end user. Some controls that could be integrated are:

- Checking if the device is rooted, or not, before sending any authentication requests to the LastPass servers. The parts of the vault decryption key could be retrieved easily by a 3rd party malicious app, if the device has been rooted.
- Allowing users to register UAF authenticators from only a specific set of vendors. Practically, it reduces the uncertainty behind using insecure authenticators that may not follow the proposed standard to the letter(e.g. storing the private keys in shared memory). A good way to achieve that is allowing users to only register authenticators that have been certified by FIDO[12].

The threat event frequency is something that can be reduced indirectly in our case. Attackers decide to attack based on the perceived value of the target and the effort required to be successful. By introducing additional controls, we are increasing the required effort for an attacker and, therefore, making attacks possible to a smaller subset of the cyber criminal threat community that has a higher threat capability. The perceived value of the vault is not something that LastPass can directly affect, as its correlated to the target user and the amount of information an attacker can get about them through reconnaissance. For example, the perceived value of the vault of a user who has their government job listed on

[12] `https://fidoalliance.org/certification/fido-certified-products/`

their social media profile and has made a post about their experience with using LastPass, would be quite high no matter what LogMeIn did to prevent that.

Chapter 5

Conclusion

Password-based authentication is slowly shifting out due to its many fundamental flaws and we should be as much prepared and well-informed as possible while this happens. In this thesis, we have contributed two major components that can help in in this transition: a way to compare 2FA schemes and a way to assess their impact.

First, we have introduced an extension to the UDS framework for multi-factor authentication settings. Using the extended framework, we have calibrated a model, which can be used directly by companies and individuals in the password management domain, to rank and score 2FA schemes for their own agendas. By checking on the benefit satisfiability of the 2FA schemes, it was apparent that schemes with higher security scores, did poorly in the usability and deployability domains. Striking the right balance between the three is essential and should be tailored to the target user/customer group.

In the second part of the thesis, we have conducted a FAIR risk analysis for a future scenario involving a company in the password management market, that is interested in including 2FA in their service offering. Risk analysts can use it as a guideline when conducting their own analyses and want to communicate risk to management and executives. Due to the abundance of cyber risk frameworks with no universally agreed terminology and concepts, having a framework adapted and used for password management, can be a stepping stone to more baseline analyses from the academic world.

Essentially, what we are offering is a set of tools to organizations in the password management domain, for conducting more rigorous and objective strategy planning. These tools can help them in prioritizing engineering tasks and allocating resources to where is necessary within the organization.

Bibliography

[1] Use of two-factor authentication among u.s. online users 2017. https://www.statista.com/statistics/789942/us-use-of-two-factor-authentication, 2017. [Accessed May 2nd, 2019].

[2] The state of strong authentication. https://fidoalliance.org/2019-strong-authentication-report/, 2019.

[3] F. Alliance. Fido u2f complete specifications. https://fidoalliance.org/specs/fido-u2f-v1.2-ps-20170411/FIDO-U2F-COMPLETE-v1.2-ps-20170411.pdf, April 2017. [Accessed May 15th, 2019].

[4] F. Alliance. Fido uaf complete specifications. https://fidoalliance.org/specs/fido-uaf-v1.1-ps-20170202/FIDO-UAF-COMPLETE-v1.1-ps-20170202.pdf, February 2017. [Accessed May 21st, 2019].

[5] D. Bertsimas and J. Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1):10–15, 1993. ISSN 08834237.

[6] J. Bonneau, C. Herley, P. C. v. Oorschot, and F. Stajano. The quest to replace passwords: A framework for comparative evaluation of web authentication schemes. In *2012 IEEE Symposium on Security and Privacy*, pages 553–567, May 2012. doi: 10.1109/SP.2012.44.

[7] W. Chen, T. yan Liu, Y. Lan, Z. ming Ma, and H. Li. Ranking measures and loss functions in learning to rank. In Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 315–323. Curran Associates, Inc., 2009.

[8] E. D. Cristofaro, H. Du, J. Freudiger, and G. Norcie. Two-factor or not two-factor? A comparative usability study of two-factor authentication. *CoRR*, abs/1309.5344, 2013.

[9] D. E. Denning and P. F. MacDoran. Location-based authentication: Grounding cyberspace for better security. *Computer Fraud Security*, 1996(2):12 – 16, 1996. ISSN 1361-3723. doi: 10.1016/S1361-3723(97)82613-9.

[10] J. Freund and J. Jones. *Measuring and Managing Information Risk: A FAIR Approach.* Butterworth-Heinemann, Newton, MA, USA, 2015. ISBN 9780127999326.

[11] L. O'Gorman. Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, 91(12):2019–2020, Dec 2003. ISSN 0018-9219. doi: 10.1109/JPROC.2003.819605.

[12] S. Palfy. Five reasons why passwords aren't going away any time soon. `https://www.forbes.com/sites/forbestechcouncil/2018/09/04/five-reasons-why-passwords-arent-going-away-anytime-soon`, note = "[Accessed March 8th, 2019]",, September 2018.

[13] C. Panos, S. Malliaros, C. Ntantogian, A. Panou, and C. Xenakis. A security evaluation of fido's uaf protocol in mobile and embedded devices. In *TIWDC*, 2017.

[14] M. Rosenquist. Prioritizing information security risks with threat agent risk assessment. 2009.

[15] M. Rouse. Sim swap attack (sim intercept attack). `https://whatis.techtarget.com/definition/SIM-swap-attack-SIM-intercept-attack`, September 2018. [Accessed April 3rd, 2019].

[16] N. S. and B. S. Location-based protocol for the pairwise authentication in the networks without infrastructure. 2018-May:190–197, 2018. doi: 10.23919/FRUCT.2018.8468300.

[17] M. Tunstall. *Smart Card Security*, pages 217–251. Springer International Publishing, Cham, 2017. ISBN 978-3-319-50500-8. doi: 10.1007/978-3-319-50500-8_9.

[18] W3C. W3c and fido alliance finalize web standard for secure, passwordless logins. `https://www.w3.org/2019/03/pressrelease-webauthn-rec.html`, March 2010. [Accessed April 30th, 2019].

[19] J. Yan, A. Blackwell, R. Anderson, and A. Grant. Password memorability and security: empirical results. *IEEE Security Privacy*, 2(5):25–31, Sep. 2004. ISSN 1540-7993. doi: 10.1109/MSP.2004.81.