João Luís Calha Tomás

Licenciado em Engenharia Informática

Análise automatizada do rastreamento de utilizadores na Web

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Orientador: José Legatheaux Martins, Professor,
NOVA University of Lisbon

Júri

Presidente: Nuno Manuel Robalo Correia
Vogais: Fernando Mira da Silva
José Augusto Legatheaux Martins

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Março , 2018

Análise automatizada do rastreamento de utilizadores na Web

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "unlthesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/unlthesis [2] http://www.di.fct.unl.pt

Agradecimentos

Agradeço à Universidade Nova de Lisboa e ao Capítulo Português da Internet Society. Aos meus familiares pelo apoio demonstrado, ao orientador pela ajuda e conhecimento partilhado, e aos meus colegas e amigos pelas críticas construtivas.

Resumo

A publicidade apresentada aos utilizadores da Internet de forma orientada aos interesses destes, permite questionar que tipo de informações é que os *websites* têm sobre os utilizadores, como é que são obtidas, se estas têm outro tipo de utilização e se são partilhadas com outros *websites*.

A utilização e partilha de informação dos utilizadores sem que estes estejam cientes, ou sem o seu consentimento, é um problema corrente que não tem sido possível ser controlado pelas entidades reguladoras, uma vez que, apesar de existir regulamentação, é bastante complicado penalizar os seus infratores.

Esta dissertação analisa o rastreamento dos utilizadores ao navegar na Internet, nomeadamente que tipo de mecanismos são utilizados para obter informação sobre os utilizadores e de como este rastreamento é feito na prática.

Existem hoje em dia ferramentas que permitem verificar os vários aspetos da interação entre um utilizador, o *website* que este está a visitar, e também, outros *websites* que estejam também a ser consultados. No entanto, estas ferramentas não apresentam uma análise comparativa desta interação com a política oficial apresentada pelos mesmos *websites*.

Assim, nesta dissertação foi proposta e implementada uma ferramenta com o objetivo de realizar esta mesma análise. A avaliação realizada demonstra que a criação de uma ferramenta para este propósito não é tarefa fácil, no entanto, esta ferramenta permite obter tanto a página com a política oficial dos *websites*, como também efetuar uma comparação da interação dos mesmo com os utilizadores. É ainda possível obter vários detalhes dos *websites*, bem como uma melhor perceção das informações guardadas e utilizadas por estes.

Palavras-chave: Rastreamento de utilizadores, cookies, privacidade, consentimento

Abstract

Internet advertising directed to a user's specific interests, raises questions about what kind of information websites have about their users, how it's obtained, if such information has any other use and if it's shared with other websites.

The usage and sharing of a user's information without their knowledge or consent is an ongoing problem which has proved to be very difficult for the regulatory authorities to control, since, even though such regulation exists, it's very hard to penalize those who offend it.

This dissertation reflects upon the aspects of user web tracking, in particular, the kinds of mechanisms which are used to obtain information about users and how the tracking process is carried out.

Nowadays, there are tools which allow the verification of the various aspects of the interaction between users, the website they're visiting, and also, other websites that may also be consulted. However, these tools don't present a comparative analysis between this interaction and the cookie policies of the websites.

Thus, in this dissertation, a local tool was proposed and implemented in order to carry out this same analysis. The evaluation performed shows that the creation of a tool for this purpose wouldn't be a simple task, however, this tool allows the gathering of both the page with the official policy of the websites, as well as allowing a comparison of the interaction between the same website and its users to be made. It is also possible to obtain various details pertaining to the websites, as well as a better perception of the information stored and used by them.

Keywords: User Tracking, cookies, privacy, consent

Índice

Capítulo 1

Introdução

1.1 Contexto e Motivação

Com o aumento que se verifica do número de fugas de informação com dados privados dos utilizadores de sites Web[1], existe uma maior preocupação relativa à privacidade destes ao navegarem na Internet. A privacidade é um conceito difícil de definir pois diz respeito às relações de um indivíduo em sociedade. É, no entanto, considerada um direito humano consagrado por diversas instâncias internacionais e este direito está consagrado, por exemplo, no Artigo 12º da Declaração das Nações Unidas sobre os Direitos Humanos (DUDH) de 1948, no Artigo 7º da Convenção Europeia dos Direitos Individuais e no Artigo 35º da Constituição da República Portuguesa[5].

A Informática, a Internet e a digitalização intensiva das interações entre pessoas e os serviços criou desafios novos à manutenção da privacidade individual. De alguma forma, esse direito procura ser acautelado de forma mais concreta em leis precisas, por exemplo, na Lei 67/98 de 26 de Outubro sobre a proteção de dados da República Portuguesa, e no Regulamento (UE) 2016/679 – relativo à proteção das pessoas singulares no que diz respeito ao tratamento de dados pessoais e à livre circulação desses dados (Regulamento Geral sobre a Proteção de Dados - RGPD). No entanto, a prática está a mostrar que a possibilidade desses normativos legislativos nacionais e europeus protegerem a privacidade dos cidadãos é relativamente diminuta.

Dada a potência expressiva da linguagem HTML e de outros objetos embebidos nas páginas Web, em particular de JavaScript executado pelos navegadores, dos mecanismos

[1] Ver exemplos de casos referentes à Sony, Yahoo, Cloudflare, e mais recentemente o caso da Cambridge Analytica.

para gestão da *cache* DOM do HTML 5.0, assim como da potência do conjunto dos mecanismos do protocolo HTTP, em particular do mecanismo dos cookies[2] e outros mecanismos dos seus cabeçalhos HTTP, é relativamente fácil realizar implementações de rastreamento dos utilizadores, que na prática ferem os direitos de privacidade dos mesmos.

Quando um utilizador acede a uma página Web, a entidade que gere essa página, daqui para a frente designada como primeira parte, não pode passar informação sobre esse acesso, feito por uma segunda parte, a uma terceira entidade ou terceira parte, sem que a segunda parte tenha conhecimento desse facto e o autorize.

Acontece que na Internet se generalizaram mecanismos económicos em que a grande maioria dos serviços são pagos via a publicidade, e não diretamente. Para o utilizador final os serviços são aparentemente gratuitos, mas na verdade eles são fornecidos em troca de publicidade ou, pelo menos, em troca de fornecimento de informações sobre si mesmo, que vão condicionar a forma como a sua navegação futura terá lugar.

Visto que o valor das primeiras partes é proporcional ao número de utilizadores (segundas partes) que clicam nas suas páginas, e o valor das terceiras partes é proporcional ao número de segundas partes que conseguem rastrear, caracterizar e identificar, existe assim um grande incentivo económico e político para que a informação sobre um utilizador seja fornecida a terceiras partes e esse incentivo leva a que sejam procurados todos os meios legais, não legais, ou simplesmente pouco éticos, para que essa violação da privacidade tenha lugar.

Desta forma, é fundamental criar uma ferramenta que permita apresentar aos utilizadores uma análise de *websites* sem que estes os tenham de aceder, protegendo assim o seu anonimato, bem como, de alguma forma, ajudar as entidades reguladoras a controlar este tipo de práticas.

1.2 Objetivos e resultados obtidos

O principal objetivo desta dissertação foi a recolha completa e robusta de informação dos *websites*, uma vez que para fazer a sua análise é necessário que estes dados estejam corretos. Para tal, foi utilizada uma ferramenta externa, o OpenWPM, que irá ser aprofundada em capítulos posteriores.

Uma vez que é irrealista fazer esta análise para todos os *websites* existentes, foi decidido que seriam utilizados os 50 *websites* mais populares de Portugal, sendo assim necessário fazer também um levantamento manual de quais as suas práticas.

A apresentação de uma análise comparativa entre o que os *websites* apresentam na sua política de privacidade e o que estes realmente utilizam, é também um dos principais focos da dissertação. Para tal, encontrar a página com a política de privacidade dos *websites*, tornou-se também noutro objetivo presente. Desta forma, tornou-se assim importante

[2] Poderia ser traduzido por "marca", no entanto, esta tradução dificultaria a leitura da presente dissertação.

classificar as diferentes páginas obtidas, e implicitamente os *websites*, de acordo com a informação guardada, de modo a ser possível separá-las em diferentes categorias.

Finalmente, pretende-se comparar a política apresentada pelo *website* com a que realmente utiliza. Estes dois últimos objetivos revelaram-se bastante complexos e foram apenas parcialmente atingidos, pois as soluções conseguidas não podem ser ainda consideradas definitivas.

Por fim, o desenvolvimento de uma ferramenta que permita a um utilizador obter toda esta análise e classificação para os *websites*, sem que este os tenha de aceder diretamente, foi também um dos objetivos presentes nesta dissertação. Infelizmente, este último objetivo não foi atingido.

1.3 Estrutura do documento

Esta dissertação está organizada em 6 capítulos. No segundo capítulo é apresentado o estado da arte, começando por introduzir o mecanismo dos *cookies*, rastreamento dos utilizadores recorrendo a este mesmo mecanismo ou a outros mecanismos, e ainda, ferramentas e estudos encontrados.

Em seguida, no capítulo 3, são apresentados os objetivos desta dissertação. No quarto capítulo encontram-se todos os aspetos de implementação tidos em conta, bem como os tempos de execução obtidos. O capítulo 5 apresenta todos os resultados obtidos.

Finalmente, esta dissertação termina no capítulo 6, apresentando algumas conclusões e trabalho futuro.

Capítulo 2

Estado da Arte

Este capítulo apresenta uma breve introdução ao que são, como funcionam e para que são utilizados cookies, nomeadamente como este mecanismo pode ser usado para rastreamento de utilizadores.

Os cookies são um mecanismo introduzido no protocolo HTTP para permitir ao servidor reconhecer que um dado pedido é feito por um cliente que já o contactou previamente.

O mecanismo utiliza os campos do cabeçalho HTTP (*header fields*) para registar referências entre o cliente, geralmente um navegador ou browser Web, e o servidor e vice-versa.

Como é bem conhecido, todos os objetos acessíveis pelo protocolo HTTP são especificados nos pedidos através de um URL, que tem uma sintaxe que permite especificar o nome remoto ou localização do objeto pretendido, a qual é constituída por:

protocol://server name/path name

Server name corresponde ao nome do servidor, ligado à primeira parte, *path name* ao caminho de acesso ao objeto no servidor, e *protocol*, o protocolo de acesso, geralmente HTTP. No que se segue, e também por conveniência do leitor, vamos utilizar a terminologia na linguagem inglesa, escrevendo os termos, *cookie*, *browser*, *website*, *cross-site*, *cross-site request forgery*, *cross-site scripting*, *user profiling*, *email*, *query strings*, *link*, *stateless*, *scripts*, *cache*, *canvas*, *fingerprinting*, *header*, *framework*, em itálico. Sempre que se justificar será introduzido o significado dum termo.

2.1 O mecanismo dos *cookies*

Cookies são identificadores gerados pelos servidores Web e memorizados pelos *browsers* dos utilizadores, de forma a que estes possam ser identificados cada vez que voltem a contactar o mesmo servidor. Como podem ser codificados diversos atributos no nome e valor do *cookie*, o mecanismo revela-se muito versátil e potente. Estes *cookies* servem assim

para guardar informação sobre os utilizadores, de modo a tornar mais fácil reconhecê-los quando estes voltam a contactar o mesmo servidor. Este mecanismo está definido em [3] e também descrito na Wikipédia[1].

Os *cookies* são enviados pelo servidor ao cliente através de respostas HTTP, as quais podem conter um ou mais cabeçalhos "Set-Cookie". Estes têm a forma geral:

Set-Cookie: <name>=<value>
[; Expires=<date>
; Max-Age=<digit>
; Domain=<domain>
; Path=<path>
; Secure
; HttpOnly
; SameSite=Strict|SameSite=Lax]

Quando o *browser* acede ao servidor e lhe transmite *cookies*, usa o *header field*: "Cookie:name".

Estes cabeçalhos são assim utilizados para enviar diferentes *cookies* do servidor para o utilizador e vice versa. Quando o *cookie* é enviado ao cliente pode conter um conjunto de atributos. Existem oito atributos num *cookie*, em que o único atributo obrigatório é um par nome-valor. Este nome pode ainda ter um de dois prefixos, que irão ser referidos posteriormente. Os atributos opcionais servem para definir o tipo de *cookie* que está a ser utilizado. Estes atributos opcionais são os seguintes:

- **Expires = <date>**

  Representa o tempo de vida máximo do *cookie*, dada uma data, o que permite definir a data em que este expira. Um *cookie* com este atributo tem o nome de "persistent cookie". No caso deste atributo não ser definido, o *cookie* irá apenas existir enquanto o cliente estiver operacional em execução e neste caso denomina-se "session cookie".

- **Max-Age = <digit>**

  Trata-se de uma forma alternativa de indicar o momento da expiração de um cookie, indicando o tempo de vida do *cookie* em segundos. Caso ambos os atributos "Expires" e "Max-Age" estejam definidos, o atributo "Max-Age" tem prioridade.

- **Domain = <domain name>**

  Especifica os servidores a que o *cookie* deve ser enviado pelo cliente. Nos casos em que este atributo não está especificado, o *cookie* só pode ser enviado para o servidor original do URL. Nos casos em que o atributo é especificado, então aplica-se ao servidor com aquele nome de domínio e a todos os servidores com nomes que sejam descendentes deste. Se o valor associado ao atributo não for um sufixo do *server name* do URL original, o *cookies* será rejeitado.

[1] https://en.wikipedia.org/wiki/HTTP_cookie

Este mecanismo é tão genérico que permite a utilização de "Supercookies", isto é, um *cookie* com uma origem no domínio de topo ou um sufixo público, como por exemplo, ".com". Este tipo de *cookie* é normalmente bloqueado pelos Web *browsers*.

- **Path = <path name>**

  Este atributo indica o *path name* do URL que tem de existir no recurso requisitado para que o *cookie* seja enviado. Se este atributo não for especificado, então todo o *path name* estará acessível. Se o atributo for especificado como, por exemplo, "path=/miei", quer dizer que o *cookie* só é enviado num pedido cujo URL tenha um *path* tal que "/miei" é um prefixo do mesmo.

- **Secure**

  Um *cookie* definido com este atributo só será enviado ao servidor quando o protocolo é HTTPS , isto é, HTTP sobre conexões seguras TLS. Este atributo não pode ser indicado quando o protocolo usado para o obter é HTTP.

- **HttpOnly**

  Os *cookies* HttpOnly não podem ser acedidos por *scripts* do lado do cliente e portanto ficam apenas controlados pelo *browser*. No caso do *browser* não suportar a utilização desta opção, esta irá ser ignorada e o cookie será tratado como um *cookie* tradicional e pode ser acedido e transmitido via *scripts*.

  Este atributo é uma das técnicas utilizadas para mitigar ataques de *cross-site scripting*, no entanto, quando não é utilizado em conjunto com outras técnicas de mitigação, não consegue eliminar completamente os perigos deste ataque.

- **SameSite=<Strict | Lax>**

  Com este atributo, os servidores podem definir que um *cookie* não deve ser enviado em conjunto com pedidos *cross-site*, o que permite uma proteção extra contra ataques de *cross-site request forgery*. Este tipo de ataque irá ser aprofundado mais abaixo.

Como referido anteriormente, o par nome-valor pode ter um de dois prefixos, prefixos estes que podem ser "__Secure-" ou "__Host-", onde ambos apenas serão aceites se o atributo "Secure" estiver presente no cookie. *Cookies* definidos com o prefixo "__Host-", têm também de apresentar um atributo "path" para o caminho "/" e não podem ter um domínio especificado.

Os *cookies* persistentes são memorizados pelo *browser* do utilizador no sistema de ficheiros do computador local. Quer isto dizer que, se o utilizador voltar a usar o mesmo computador antes do *cookie* expirar, este será de novo enviado para o servidor. Desta forma, este tipo de *cookie* aumenta o êxito potencial do rastreamento do utilizador se este utilizar sempre o mesmo computador, sobretudo se a sua duração for muito longa.

A principal motivação para a criação de *cookies* foi a facilidade com que permitem introduzir funcionalidades que proporcionem ao utilizador uma melhor experiência no acesso a *websites*.

- ***Cookie* de sessão**

  Permitem um melhor funcionamento da sessão pois desta forma ultrapassa o carácter *stateless* do protocolo HTTP. Estes *cookies* substituíram uma técnica em que a informação relativa à sessão de um utilizador era incorporada no URL, normalmente na parte especificada para consultas.

  Este método consiste no anexo de *query strings*, cada uma com identificadores únicos, a todos os *links* que existem no *website*. Sempre que um utilizador carrega num *link*, o *browser* envia estas *query strings* para o servidor, podendo assim identificar o utilizador. No entanto, guardar informação que identifica a sessão de um utilizador nestas *query strings*, facilita vários ataques de roubo de sessão, uma vez que o seu identificador pode ser facilmente obtido através do URL.

- ***Cookie* para autenticação**

  Inicialmente o protocolo HTTP permitia a autenticação de utilizadores num *website* após estes introduzirem o nome e palavra-passe corretos. Essa informação é transmitida para o servidor no *header field* Authorization[8]. Caso o servidor utilizasse este tipo de autenticação, o *browser* fazia o pedido das credenciais ao utilizador, e após estas terem sido obtidas, eram então guardadas pelo *browser* e utilizadas para todos os pedidos efetuados posteriormente.

  Atualmente, este tipo de autenticação é pouco utilizado, tendo sido substituída pela utilização de *cookies* de autenticação. Estes *cookies* servem para autenticar um utilizador que visite um *website* e só podem ser enviados por conexões seguras.

  Quando um utilizador tenta aceder a um *website* com autenticação, o cliente faz um pedido ao servidor, cuja resposta é um pedido de autenticação através do *header* WWW-Authenticate[8]. O cliente, após receber a resposta, apresenta um pedido de credenciais ao utilizador. Ao introduzir estas credenciais, o cliente envia ao servidor um pedido com o *header* de autorização correto. O servidor verifica as credenciais inseridas, autorizando ou não o cliente. A partir do momento em que o servidor autentica o cliente, é enviado um *cookie* de autenticação com uma chave, que será guardada pelo *browser* e este *cookie* é enviado em todos os acessos subsequentes de forma a autenticar automaticamente o cliente, mesmo em sessões posteriores.

  Naturalmente estes *cookies* devem ter um tempo de vida limitado o que obriga a que o utilizador se volte a autenticar periodicamente.

- ***Cookie* para personalização**

  Os *websites* utilizam este tipo de *cookies* para guardar informação relativa a preferências que o utilizador tenha escolhido, quer seja para personalizar o aspeto do *website*

de acordo com essas preferências, quer para mostrar conteúdos que possam ser relevantes para o utilizador. Quando um utilizador seleciona as suas preferências para aquele *website*, este associa-as a um *cookie* que será posteriormente utilizado sempre que aquele utilizador faça pedidos, e as respostas personalizadas de acordo com as preferências escolhidas.

Alternativamente o servidor pode tentar inferir as preferências do utilizador pela história das interações anteriores do mesmo utilizador.

2.2 Anonimização e re-identificação

Sempre que uma primeira parte exporta dados sobre acessos de um utilizador, deve fazê-lo de forma anónima.

Anonimização de dados[2] é o processo que transforma informação que permite identificar uma pessoa em concreto, num conjunto de dados anónimos não identificáveis.

A discussão da robustez da anonimização de dados é da mesma natureza da criptografia, ou seja, se não forem utilizados métodos adequados pode ser quebrada, levando assim à identificação dos utilizadores em concreto.

Se a metodologia utilizada para a anonimização dos dados não for a correta, esta pode levar, por exemplo, a que a dados diferentes sejam atribuídos identificadores idênticos, ou que esse identificador aleatório conserve uma relação com o original. Existe ainda o problema de que se este identificador não for regenerado com fatores de aleatoriedade num período muito longo, poderá eventualmente ser correlacionado com outros identificadores que utilizem o mesmo algoritmo de anonimização. A este processo que faz corresponder um perfil anónimo a um utilizador concreto dá-se o nome de re-identificação.

2.3 *Cookies* e rastreamento dos utilizadores

Qualquer acesso HTTP, transmite ao servidor (primeira parte) pelo menos a seguinte informação sobre o *browser* (segunda parte):

- o endereço IP de onde foi feito o acesso, o que permite conhecer a localização da segunda parte;
- a página Web solicitada;
- o sistema de operação;
- a sua língua;
- o tipo do browser que está a ser utilizado;
- muitas das preferências do *browser*;
- *cookies* gravados no *browser* de acessos anteriores;

- informação de controlo da *cache*.

Todas estas informações podem ser obscurecidas por utilizadores sofisticados mas tal é impossível a um utilizador comum. A informação sobre *cookies* é opcional e pode ser desativada, mas a grande maioria dos serviços não permite a sua utilização sem *cookies*.

Como foi visto anteriormente, os *cookies* são utilizados para guardar informação relativa aos utilizadores. De acordo com as normas de privacidade europeias, todos os *websites* que utilizem *cookies*, têm de obrigatoriamente perguntar ao utilizador se este aceita a utilização desses *cookies*. Sempre que um utilizador aceita que um *website* utilize *cookies*, este está a permitir que essa informação seja guardada por este mesmo *website*. Um dos principais problemas da utilização de *cookies* como meio de rastreamento de utilizadores consiste na utilização destes *cookies* por parte de outros *websites*, aos quais a mesma permissão não foi necessariamente dada, em que muitas vezes o utilizador não sabe que estas terceiras partes estão também a guardar a sua informação.

Este rastreamento de utilizadores feito por terceiras partes é maioritariamente utilizado para fins publicitários, em que empresas de publicidade utilizam a informação que obtêm através deste rastreamento para mostrar um conteúdo mais apropriado ao perfil do utilizador.

A utilização de *cookies* no rastreamento dos utilizadores começa quando um utilizador (segunda parte) faz um pedido sem um *cookie* a uma página de um *website*. O servidor (primeira parte) cria um identificador único que é enviado para o utilizador como um *cookie*, em conjunto com a página pedida. Nesta troca de pedidos, o utilizador fez um contrato implícito de fornecer as suas informações, referidas acima, à primeira parte, pressupondo que essa a utilize de uma forma adequada.

No entanto quando se acede a um *website*, não se está a aceder apenas a esse *website*, mas também a todos os *websites* que o mesmo indique no código HTML e nos *scripts* que envie. Esses outros *websites* podem ser parte integrante da primeira parte, mas também podem ser terceiras partes que têm contratos com a primeira parte, mas não com a segunda parte. Essa terceira parte pode também instalar *cookies*. Desta forma, as informações que a segunda parte forneceu à primeira parte são também enviadas para a terceira parte, em conjunto com o *cookie* em acessos posteriores.

Caso dois *websites* diferentes utilizem a mesma terceira parte, esta irá saber que se trata do mesmo utilizador através do identificador único. Duas terceiras partes que utilizem a mesma rede de publicidade ou rastreamento, podem partilhar os identificadores e fazer assim a ligação ao mesmo utilizador.

Esta correlação de utilizadores entre diferentes terceiras partes de modo a que ambos saibam que se trata do mesmo utilizador é principalmente, mas não só, feita através de "*cookie syncing*"[13], que tal como o nome indica, sincroniza os *cookies* entre os *websites*.

É ainda de notar que *scripts* pertencentes a redes sociais, por exemplo botões de partilha, exibidos num *website*, são também utilizados para rastrear o utilizador caso este

esteja conectado à rede social, uma vez que o *website* ao qual o *script* pertence guarda um *cookie* que permite ao *script* identificar que se trata do mesmo utilizador.

Todo este processo é denominado de "*user profiling*" e é este processo que as terceiras partes utilizam mais frequentemente. No entanto, este processo pode ser feito com ou sem a permissão ou conhecimento do utilizador. Muitas vezes, o *website* vende este perfil do utilizador, ou, intencionalmente, existe uma fuga de informação no *website*, permitindo que estas terceiras partes obtenham essas mesmas informações.

Existem ainda métodos que, apesar de utilizarem *cookies*, dependem principalmente de técnicas de código em Javascript e de deficiências de segurança. Servem como exemplo os seguintes métodos:

- **Zombie *Cookie***

  Zombie *cookie* é um *cookie* HTTP que, após ter sido apagado, se volta a criar automaticamente. Esta recriação é feita utilizando dados que foram guardados em várias localizações no lado do cliente, e por vezes, também no lado do servidor. Mesmo que um utilizador tenha optado por não utilizar *cookies*, estes *cookies* podem ser à mesma guardados e utilizados, uma vez que não dependem completamente de *cookies* tradicionais.

  Esta técnica baseia-se numa deficiência de segurança no mecanismo que utiliza, como por exemplo Flash[14]

- **Roubo de *Cookies***

  Existem algumas técnicas que permitem que um *cookie* seja roubado. Uma destas técnicas é chamada de "*cross-site scripting*", em que um atacante tira vantagem de um *website* que permita a utilização de HTML e JavaScript não filtrado. O atacante pode assim colocar código HTML e JavaScript de modo a que o utilizador envie o seu *cookie* para um outro *website* controlado pelo atacante.

- **Cross-site request forgery**

  É um ataque onde um utilizador executa comandos não autorizados por si num *website* onde esteja autenticado. Este ataque é conseguido quando um atacante introduz código malicioso. Este tipo de ataque tem o objetivo de forçar um utilizador a efetuar ações que o possam prejudicar, como por exemplo, transferências monetárias, alteração do seu correio eletrónico, entre outras.

Segundo os autores Steven Englehardt e Arvind Narayanan[7] e baseados em estudos anteriores, estes consideram que um *cookie* que tenha uma data de expiração acima de 90 dias, um parâmetro valor com tamanho de 8 a 100 bytes (um *cookie* pode ter até 4k bytes), em que o parâmetro não é modificado ao longo de diversas sessões da mesma medição, assim como outras características, e é um *cookie* enviado a terceiras partes, é classificado como *cookie* de identificação, pois o seu perfil indica que é sua intenção fazer

uma identificação dos utilizadores. Um *cookie* de terceiras partes deste tipo, tem todo o indício de que potencialmente pode quebrar o anonimato.

No entanto, é de salientar que a presença e utilização destes mecanismos por terceiras partes não revelam necessariamente a existência de qualquer tipo de problema, uma vez que depende de como a informação obtida está a ser utilizada pelas terceiras partes. Legalmente, se as primeiras partes publicarem quais as condições em que a informação está e pode ser utilizada, e se utilizarem medidas de anonimização corretas, não existem indícios de que existe qualquer tipo de problema.

2.4 Alternativas de rastreamento

Existem técnicas de rastreamento que se baseiam na utilização de outro tipo de mecanismos. Por exemplo, a utilização de "JSON Web Tokens", autenticação utilizando o protocolo HTTP, utilização de URLs e utilização de "Web storages".

- **JSON Web Tokens**

  Este mecanismo[10] é um pacote que pode ser utilizado para guardar informação sobre a identidade e autenticidade de um utilizador. Dada esta característica, estes pacotes podem ser utilizados assim para substituir *cookies* de sessão. Estes, ao contrário dos *cookies*, têm de ser explicitamente anexados, pelo *website*, a cada pedido HTTP.

- **Web storages**

  Similar aos *cookies*, mas com um limite de armazenamento muito maior e sem guardar qualquer informação no cabeçalho do pedido HTTP, este mecanismo[12] disponibiliza dois tipos de armazenamento de dados num *browser*. Ambos estes tipos, denominados de "local storage" e "session storage", atuam de uma forma parecida aos "persistent cookies" e "session cookies", respectivamente. Uma diferença notável entre a "session storage" e os "session cookies" é que esta não está ligada ao tempo de vida do *browser*, mas sim ao tempo de vida de uma página.

- **Canvas Fingerprinting**

  Este mecanismo[4] tira proveito do *canvas* HTML ao desenhar linhas ou gráficos não visíveis ao utilizador que são convertidos posteriormente em símbolos digitais. Uma vez que cada imagem neste *canvas* é processada de forma diferente em computadores diferentes, é assim possível que estes símbolos sejam utilizados para criar o perfil de um utilizador.

2.5 Ferramentas e estudos

Investigadores de diversas universidades têm dedicado bastante atenção ao problema do rastreamento, implementando diversas ferramentas de deteção do mesmo, e realizado recenseamentos sobre a utilização das diferentes técnicas de rastreamento.

Em seguida apresentam-se três exemplos de estudos.

1. Online Tracking: A 1-million-site Measurement and Analysis[7]

   Apresenta a maior e mais detalhada medição de rastreamento online até ao dia em que foi publicado. Este artigo é baseado numa análise feita a 1 milhão de *websites*. Este artigo descreve 15 tipos de medições em cada *website*, que incluem rastreamentos baseados em cookies e *fingerprinting*, os efeitos de ferramentas de privacidade e partilha de dados entre *websites* diferentes.

   Sucintamente, neste estudo foi concluido que o número de terceiras partes presentes em pelo menos 2 primeiras partes é maior que 81,000, mas apenas 123 destas estão presentes em mais de 1% dos *websites*. No entanto, 12 das 20 terceiras partes que estão no topo de utilização, pertencem à Google.

   É também concluído que a proteção contra o rastreamento funciona, quer isto dizer, que a utilização de ferramentas como o Ghostery, uBlock origin, ou até mesmo o bloqueio do Firefox a *cookies* de terceiras partes, é efetiva. O mesmo já não pode ser dito da proteção contra o *fingerprinting*, onde foi provado que apenas entre 10 a 25% dos *scripts* utilizados para este fim são bloqueados pela ferramenta Disconnect e uma combinação da EasyList e EasyPrivacy.

2. Third-Party Web Tracking: Policy and Technology[11]

   Examina implicações na privacidade dos utilizadores no que diz respeito ao rastreamento efectuado por terceiros. Este artigo apresenta ainda a regulamentação em vigor que foi imposta para ajudar a combater os riscos que o rastreamento apresenta para o utilizador, e ainda algumas das tecnologias que são utilizadas para correlacionar a atividade dos utilizadores entre vários *websites*, bem como algumas das opções que são disponibilizadas aos utilizadores para que estes não permitam o rastreamento.

   Neste estudo são apresentados vários problemas políticos e tecnológicos do rastreamento feito por terceiras partes, com o objetivo de levar a uma discussão inicial sobre este problema que, na altura da sua realização, se encontrava em rápido crescimento.

3. FPDetective: Dusting the Web for Fingerprinters[1]

Este artigo descreve o desenho, implementação e o desenvolvimento da ferramenta FPDetective, que irá ser referida numa próxima secção. Analisa também duas contra-medidas propostas como forma de defesa ao *fingerprinting*, encontrando fraquezas que podem ser exploradas para ignorar a proteção dessas contra-medidas.

Sucintamente, o estudo conclui que o rastreamento está a ficar cada vez mais penetrante na privacidade dos utilizadores, na medida em que os empresas de publicidade procuram melhorar a escolha dos seus alvos. Enquanto que grande parte do rastreamento de hoje em dia é feito através de *cookies* de terceiras partes, estudos anteriores mostram que os atributos do sistema e dos *browsers* podem ser também utilizados para identificar unicamente os utilizadores, através de *fingerprinting*, mesmo sendo esta forma menos precisa do que a utilização de *cookies*.

O estudo aqui apresentado, mostra que a utilização de *fingerprinting* é cada vez maior nos *websites* mais populares, e ainda que grandes entidades de comércio estão envolvidas na utilização deste método, apresentam um enorme desprezo em relação ao cabeçalho "Do not track", e estão também associadas com vários *malwares*.

Para o desenvolvimento destes estudos foram desenvolvidas diversas ferramentas. Servem como exemplo as seguintes:

1. FPDetective[1]

   Esta ferramenta é baseada numa *framework* que permite a deteção e análise de *fingerprinting* presente em *websites*. Focada na deteção de *fingerprinting* em si, esta ferramenta aplica a sua *framework* nas práticas de deteção de fontes.

2. OpenWPM[7]

   Embora esta ferramenta tenha sido criada com base na acima descrita, tem 4 aspetos que a diferenciam da primeira. O OpenWPM suporta medições tanto *stateful* como *stateless*, inclui instrumentação genérica para rastreamento. Nenhuma desta instrumentação requer código de navegação nativo, e utiliza uma arquitetura baseada em comandos de alto nível.

   Relativamente ao seu desenho, esta foi dividida nos três seguintes módulos:

   - ***browser managers***

     Atua como uma camada de abstração para automatizar cada instância dos navegadores, nomeadamente as ações que cada um deve realizar.

   - ***task manager***

     Monitoriza os *browser managers* e distribui comandos para estes. Serve ainda para simular a interação com o utilizador.

   - ***data aggregator***

     Atua como uma camada de abstração para a instrumentação do navegador, recebendo e pré-processando os dados.

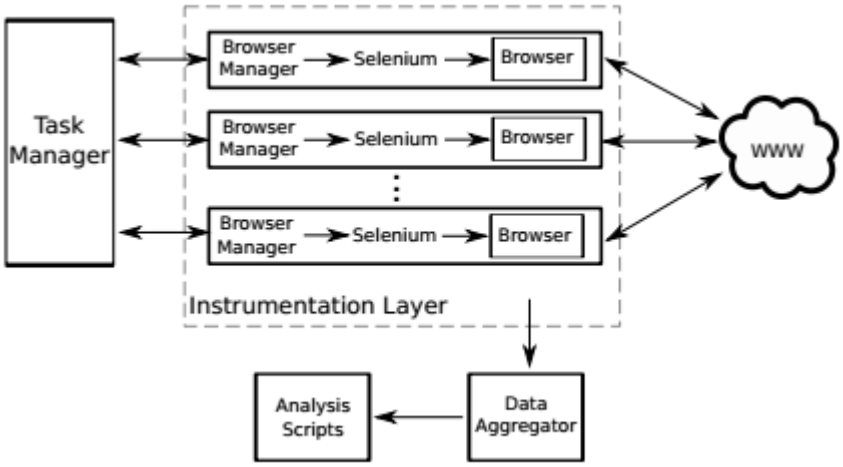

Figura 2.1: Visão geral do alto nível do OpenWPM. Retirado de [7]

Toda esta plataforma foi criada utilizando a linguagem Python e as suas bibliotecas, e é utilizada a *framework* "Selenium" para controlar a atividade dos *browsers*. Esta *framework* é utilizada apenas para o *browser* Firefox, no entanto também suporta vários outros, o que facilita a transição para qualquer um deles no futuro.

Uma grande desvantagem da *framework* é o fato desta ser bloqueante, o que leva a que fique bloqueada durante um tempo indeterminado caso se esteja a fazer um medição muito longa e ocorra algum problema que altere ou danifique os dados. De modo a precaver este problema, os *browser managers* isolam o Selenium do resto dos componentes ao utilizarem uma camada de abstração em volta deste.

Para isolar falhas, cada *browser manager* é inicializado como um processo em separado. Cada um cria uma instância do Selenium com configurações especificas, e é responsável por converter comandos de alto nível em sub-rotinas especificas dessa *framework*. Guarda ainda o estado de cada navegador, permitindo assim recuperar de falhas.

De forma a garantir escalabilidade e abstração, a plataforma, através do *task manager*, providencia uma interface na linha de comandos que controla múltiplos navegadores simultaneamente. Cada comando é lançado numa *thread* de comandos por navegador, podendo ser distribuído para esse navegador de uma forma sincronizada ou em que o primeiro a chegar é o primeiro a ser enviado.

Esta *thread* de comandos, caso exista algum problema no *browser manager* correspondente, entra numa rotina de recuperação de erros, em que o *task manager* arquiva o perfil do navegador corrente, para todos os processos correntes, e carrega um novo navegador com as mesmas configurações e dados.

Quanto à maneira como o utilizador pode consultar os dados, é dada a possibilidade destes serem acedidos de três formas diferentes que cobrem quase toda a interação

de um navegador com a Internet e o sistema. Os dados podem ser consultados de forma não tratada no disco do utilizador, ou guardados numa base de dados. Podem também ser acedidos a nível da rede, com a utilização de um *proxy* HTTP, ou então a nível de Javascript, através de uma extensão para o Firefox providenciada pelos mesmo autores.

Existem também diversas extensões (*plugins*) para *browsers* que tentam proteger o utilizador do rastreamento online, por exemplo:

1. Ghostery[9]

   Este tipo de ferramenta é destinada ao utilizador individual e só permite obter informação acedendo mesmo à página. Ao bloquear *websites* de acordo com uma lista criada pela própria extensão, esta garante uma navegação mais rápida e segura ao utilizador, permitindo ainda que este controle quais os *websites* onde esta extensão não deve atuar.

2. Disconnect[6]

   Esta ferramenta bloqueia serviços que foram identificados por si como rastreadores, criando também uma lista dos *websites* bloqueados. A proteção de rastreamento para um único *browser* é gratuita, no entanto o utilizador pode ainda comprar proteção para qualquer dispositivo, ou a utilização de uma Rede Privada Virtual.

Capítulo 3

Objetivos

Tal como apresentado na introdução, os objetivos desta dissertação foram os seguintes:

- Uma vez que a ferramenta externa OpenWPM é utilizada no desenvolvimento desta dissertação, um dos principais objetivos é entender o funcionamento desta.
- Fazer um levantamento das práticas usadas pelos *websites* mais populares em Portugal, utilizando a lista apresentada no *website*:

  http://www.alexa.com/topsites/countries/PT
- Utilizar então o OpenWPM de forma a obter informações de qualquer um destes *websites*.
- Encontrar a página com a política de *cookies* para cada um destes mesmos *websites*, caso esta exista.
- Analisar comparativamente as informações obtidas com o que é apresentado por cada *website* na página de política obtida.
- Classificar esta página, e implicitamente cada *website*, de acordo com a informação apresentada por estes e o que realmente é feito.
- Desenvolver uma ferramenta que permita a um utilizador obter esta análise e classificação dos *websites* apresentados, sem que este os tenha de aceder diretamente.

Capítulo 4

Implementação

A secção 4.1 apresenta as razões pelas quais se escolheu utilizar a ferramenta OpenWPM. Na secção 4.2 são apresentados os problemas a resolver, incluindo as soluções encontradas. Em seguida, a secção 4.3 apresenta a solução final adotada e as dificuldades encontradas. Finalmente, na secção 4.4 apresentam-se alguns dados sobre os tempos de execução.

4.1 OpenWPM

Como já referido anteriormente, ver a secção 2.5, esta ferramenta oferece uma diversidade de funcionalidades que permitem fazer *crawling* de qualquer *website*, como se se tratasse de um ser humano a utilizá-los. Com efeito, o OpenWPM lança e controla verdadeiros *browsers* (neste caso o Firefox, versão 58.0.2) e realiza os acessos especificados num *script*. Assim, o *website* é acedido por um verdadeiro *browser*, com a sua *cache*, dados e *cookies* memorizados, o seu *document store*, etc., o que não se passava com a generalidade das outras opções. Em particular, seria muito difícil, senão impossível, programar os acessos diretos sem que o site acedido não detetasse o acesso por um *bot* e de qualquer forma o comportamento real do *browser* dificilmente seria simulado.

Desta forma, foi decidido utilizar esta ferramenta para a obtenção de várias informações presentes nos *websites* escolhidos. Por esta razão é fundamental estudar como é que a ferramenta funciona, e mais importante, como pode ser aplicada às necessidades apresentadas nesta dissertação.

Por omissão, a ferramenta cria uma base de dados local, onde guarda toda a informação para cada *crawl* efetuado. Assim, esta base de dados tornou-se num dos principais instrumentos de suporte da implementação.

Porque foi adotado o OpenWPM

Numa fase inicial, poderia-se pensar que a simples utilização de uma biblioteca que permitisse extrair a informação e dados de *websites* seria suficiente para fazer *crawling* destes. Esta aproximação, apesar de ser fácil de utilizar, é um pouco ingénua, na medida em que é necessário reproduzir exatamente o comportamento do *browser*. Esta reprodução é cada vez mais fundamental, uma vez que os *websites* atuais utilizam sistemas de *cache*, deteção de acessos não-humanos, entre outros, devolvendo assim resultados inúteis para a nossa análise.

O OpenWPM oferece exatamente estas funcionalidades. Este permite o acesso a um *website* através de um *browser* real, simulando ainda a utilização deste como se fosse um humano, aumentando assim a probabilidade das informações devolvidas por este serem as corretas, semelhantes às obtidas por um utilizador humano.

No entanto, o grande defeito desta ferramenta está no acesso aos *websites*, que é bastante pesado. Ainda assim, o fato de, como já referido, tentar reproduzir uma verdadeira sessão de um utilizador, permitindo assim ter uma imagem mais realista do comportamento dos *websites*, é uma vantagem que torna imprescindível a utilização do OpenWPM.

Desta forma foi necessário, como será explicado numa próxima secção, encontrar um equilíbrio que permitisse atingir os objetivos pretendidos em tempo útil, ou seja, foi feita uma grande experimentação com os diferentes *websites* até encontrar um equilíbrio entre utilizar as funcionalidades do OpenWPM e funcionalidades implementáveis diretamente via bibliotecas de *Python* disponíveis.

Arquitetura do OpenWPM

O OpenWPM apresenta aspetos fundamentais na sua estrutura, podendo-se separar nos seguintes 4: lançamento do *browser*, *framework* de diálogo com o utilizador, funcionalidades de interação com o *browser* e base de dados.

Lançamento do *browser*

Ao lançar um *browser* este pode ser configurado para ser lançado em *background*, com uma interface gráfica emulada, permitindo assim um melhor desempenho na sua utilização, uma vez que a parte gráfica destes demora algum tempo a ser executada. Apesar de não ter sido utilizado, podem ainda ser lançados vários *browsers* concorrentemente, permitindo que cada um aceda a todos os *websites* pedidos, simulando distintos *browsers*.

Framework de diálogo com o utilizador

O módulo "TaskManager" é o responsável por começar todo o processo de *crawling*, receber as configurações dos *browsers*, e ainda, de lançar todas as funcionalidades pedidas pelo utilizador. A classe "CommandSequence" está encarregada de guardar, sequencialmente, todas as funcionalidades pedidas, bem como de aceder ao *website*, de forma a que

estas sejam enviadas para o mesmo *browser* como se se tratasse de uma única visita. O "TaskManager" executa então todas as funcionalidades guardadas nesta classe.

Interação com o *browser*

Toda a interação com o *browser* é feita recorrendo à *framework* "Selenium". Esta *framework* utiliza uma API especial do *browser*, especialmente desenhada para lhe injetar as ações do utilizador, como por exemplo: fechar janelas de *popup*, fazer *scroll* na página, ou esperar que página esteja completamente carregada.

Assim, é possível personificar o acesso aos *websites* obtendo resultados como se se tratasse de um utilizador real.

Base de dados

Sempre que um "TaskManager" é executado, este cria uma base de dados (caso não exista) para guardar todo o tipo de informação obtida durante a sua execução. Independentemente de algum *website* ser acedido ou não, são sempre criadas 3 tabelas na base de dados, com informações relativas aos *crawls* e tarefas realizadas. Estas tabelas são a tabela "crawl", onde são guardadas informações relativas a todos os *crawls* realizados, como o seu identificador, a sua data de criação e se terminou com êxito ou não, a tabela "sqlite_sequence", cuja função é guardar informações relativas ao número de entradas existentes noutras tabelas, e a tabela "task", onde são guardadas informações relativas às tarefas realizadas, nomeadamente a data de criação, e versões do *browser* e do *Selenium*.

Para além destas tabelas iniciais, com o decorrer do programa, o OpenWPM adiciona ainda as restantes tabelas, onde existem em quase todas os seguintes campos: identificadores do *crawl*, o número da visita, e a data de acesso.

A tabela "CrawlHistory" tem a função de guardar cada comando utilizado, o argumento recebido e se foi bem sucedido ou não. As tabelas "profile_cookies", "links_found" e "site_visits" são bastante intuitivas, na medida em que guardam toda a informação relativa aos *cookies*, todos os *links* encontrados na página principal de cada *website* acedido, todos os *websites* e *links* visitados, respetivamente.

As tabelas "http_requests" e "http_responses" guardam toda a informação relativa aos pedidos e respostas feitas pelo *website* visitado, como por exemplo: URL, método, *referrer*, estado da resposta, cabeçalhos, e o identificador do canal (identificador único do URL).

A tabela "http_redirects" guarda dois identificadores de canal, um antigo e um novo, podendo desta forma saber qual o URL que fez o pedido e qual o URL para onde foi redirecionado.

As seguintes tabelas foram acrescentadas ao modelo de dados de origem, uma vez que se sentiu a necessidade de guardar as outras informações para melhorar tanto o *crawling* como a análise. A tabela "website_status", que guarda o estado (completo ou incompleto) do *crawl* de cada *website* acedido. A tabela "cookie_description" tem a função de guardar,

para cada *cookie* encontrado, uma descrição e o seu propósito. Por fim, existem também as tabelas "html_pages" e "html_pages_cookies", cujo objetivo é guardar as páginas fonte de todos os URL's acedidos e *cookies* colocados, a contagem de ocorrências da palavra *cookie* nestas páginas, o número de domínios que colocam *cookies* nestas páginas, e o nome desses domínios.

Existem ainda outras tabelas que foram adicionadas mas não são relevantes para esta dissertação. Em seguida, apresentam-se exemplos concretos de funcionalidades disponibilizadas pelo OpenWPM. Nessa lista de métodos está implícito o *URL* da página a que os mesmos se aplicam.

- ***get(sleep, timeout)***

  Este comando permite aceder a uma página HTML, e obtém várias informações, como os pedidos e respostas HTTP, qual o site que foi acedido, se o comando foi bem sucedido, entre outras.

- ***extract_links(timeout)***

  Este comando permite extrair todos os links presentes na página visitada, guardando-os na tabela "links_found", presente na base de dados.

- ***dump_profile_cookies(timeout)***

  Este comando permite obter todos os *cookies* presentes no *website* acedido, sendo estes guardados na base de dados.

  Para além do nome, são também guardadas outras informações dos *cookies*, como o seu domínio, o valor, o seu prazo de expiração, entre outras.

- ***dump_page_source(suffix,timeout)***

  Este comando permite guardar a fonte da página acedida num ficheiro *html* local. O parâmetro opcional "suffix" pode ser utilizado para indicar um nome específico com que o ficheiro deve ser guardado, sendo que o nome dos ficheiros seguem sempre o seguinte formato:

  número da visita-hash em md5 do url-[parâmetro "suffix"].html

É ainda necessário referir que para utilizar qualquer comando, tem de ser lançado o comando "get" antes de qualquer outro.

Finalmente, é também de salientar que o comando "dump_profile_cookies" fecha a janela do *website* que estava a ser visitado, e que o comando "dump_page_source" faz com que nenhum *cookie* seja encontrado, o que requer que ambos sejam chamados separadamente.

Apesar da ferramenta conseguir recuperar de *crashes* e continuar a trabalhar, caso existam mais de 13 erros seguidos, esta é encerrada.

4.2 Problemas a resolver e soluções encontradas

Estes são os aspetos que foram identificados como os principais problemas a serem resolvidos de forma a atingir os objetivos pretendidos. Os dois primeiros problemas são mais simples de resolver relativamente aos últimos dois, que se mostraram mais complicados. São eles os seguintes:

- Aceder ao site e verificar que *cookies* são colocados após a visita à página principal;
- Aceder aos *links* encontrados na página principal, que os *browsers* normalmente expandem imediatamente, de forma a obter a lista suplementar de *cookies* colocados, uma vez que é aqui que vêm os *cookies* de terceiras partes;
- Obter a página que contém a política de *cookies* do *website*;
- Se se conseguir obter essa página, correlacionar a política publicada com os *cookies* de fato colocados.

Verificou-se que os dois problemas mais simples levavam muito tempo a executar com o OpenWPM.

Daí se ter, posteriormente, adotado a metodologia de primeiro fazer o *crawling* do *website*, e só depois ser feita a análise do mesmo, com base nos dados obtidos. Este aspeto será desenvolvido a seguir.

Os dois problemas seguintes revelaram-se bastante mais difíceis de resolver, pelo que as soluções adotadas foram evoluindo com a experiência da sua utilização.

A seguir começaremos por explicar como evolui a solução de encontrar a página com a política do *website*.

Inicialmente, o *script* efetuava o *crawling* do *website*, guardando todos os *links* presentes, bem como os *cookies* recebidos. Em seguida, eram procuradas palavras-chave[1] nestes *links*, e caso acabasse esta procura sem encontrar nenhuma palavra-chave, era devolvido o estado "Ausente". Esta procura de palavras-chave é feita com base na biblioteca "BeautifulSoup", especializada em obter dados presentes em páginas HTML e XML.

Caso a pesquisa da página selecionada como sendo a da política do *website* tivesse êxito, passava-se então à solução do último problema, que consiste em analisar essa política.

Caso fosse selecionado um *link*, este era então acedido, guardando a sua página fonte num ficheiro. Em seguida era procurado o nome de cada *cookie* presente no *website* acedido. Caso fossem encontrados dois ou mais nomes de *cookies* no ficheiro da página, era devolvido o estado "Bom", e o respetivo *link*. A ideia de ser necessário existirem dois ou mais nomes presentes no ficheiro foi tomada uma vez que alguns *cookies* apresentam nomes comuns, o que devolvia um estado incorreto.

[1] Nomeadamente as seguintes: cookie, privacy, policy, terms, rule, privacidade, regra, condicoes, termo, politica

No caso de não se verificar a ocorrência de dois ou mais nomes de *cookies*, era então procurado no texto da página onde a palavra "*cookie*" estivesse presente, um outro *link* em que a palavra "*cookie*" estivesse também presente. Se fosse encontrado um novo *link*, era devolvido o estado "Impercetível ao utilizador comum" e o seu novo *link*, caso contrário o estado "*Cookies* não especificados" era devolvido, acompanhado do *link* previamente determinado.

Posteriormente foram feitas melhorias na determinação do estado a devolver, nomeadamente a utilização da contagem do número de vezes que a palavra "*cookie*" aparece na página fonte, bem como quantos dos seus domínios estão também presentes no texto. Esta contagem era utilizada para separar os casos em que poderia existir um outro link onde a política de *cookies* fosse mencionada, de casos em que fosse dada pouca informação sobre esta.

Assim, foi determinado que caso a contagem da palavra "*cookie* devolvesse mais do que 4 ocorrências e que a contagem distinta do número de domínios fosse maior que 2, ou simplesmente que a contagem distinta do número de domínios fosse maior que 5, era devolvido o estado "*Cookies* não especificados" e uma mensagem de que poderia existir, ou não, outra hiperligação com mais informações.

As soluções que acabámos de descrever para os quatro problemas apresentados foram evoluindo com o tempo. Em particular, o problema de se obterem todos os *cookies* revelou que era necessário aceder aos *links* encontrados na página principal, de forma a obter os *cookies* de terceiras partes. De forma a resolver este problema, acedeu-se também a todos os *links* encontrados na página principal, guardando os *cookies* colocados por cada *link* visitado.

Como já referido, verificou-se que o desempenho do *script* ficou bastante lento, especialmente quando aplicado a vários *websites* seguidos (cada *website* demorava em média 1 hora e meia), bem como suscetível a problemas provenientes do OpenWPM, o que levou à utilização da metodologia de separar o *crawling* da análise. O leitor tem de ter em consideração que alguns dos *websites* mais populares têm várias centenas de *links* na página principal, ver tabela 5.1.

4.3 Solução adotada

Como já foi introduzido, a solução adotada desenvolve-se em duas fases:

1. Fazer *crawling* dos *websites* e colocar o resultado na base de dados
2. Usar a base de dados para fazer a análise do site

Crawling de *websites*

Sempre que um novo *crawl* de vários *websites* passados em parâmetro é lançado, é também criada uma nova tabela, caso não exista, na base de dados, que guarda o estado de cada

um deles.

Ao processar um *website* para obter todas as informações necessárias, é feita uma verificação do estado do mesmo na base de dados. Caso o *website* esteja completo e exista há menos de uma semana na base de dados, não é feito nada e termina o seu processamento. Caso contrário, é feito o acesso ao *website*, sendo que se este estiver completo e existir há mais de uma semana, toda a sua informação associada é primeiro apagada.

O acesso ao *website* em si é bastante simples, mas demorado. Com recurso ao OpenWPM, é então aberto o *browser* com o *website* pretendido, guardando todos os *links* presentes na página acedida e os *cookies* colocados. Como já referido, é necessário abrir de novo o *browser* e aceder ao mesmo *website* de modo a obter a página fonte deste, guardando-a num ficheiro cujo nome começa pelo número de identificação do *crawl* corrente, seguido do *website* convertido para a sua *hash*, e por fim o domínio do *website*.

Em seguida, para cada *link* acedido, são guardados na base de dados, os *cookies* colocados, copiando também a sua página fonte para um ficheiro. Para todos os *cookies* guardados na base de dados, é obtida e guardada num ficheiro, uma breve descrição de qual o seu propósito. A mesma é obtida de forma a seguir descrita.

Todos os ficheiros obtidos, quer tenham páginas fonte, quer informações relativas aos *cookies*, são posteriormente guardados na base de dados. Ao fazer esta inserção, é também feita a contagem do número de vezes que a palavra "cookie" é mencionada em cada página do *website*, bem como o número distinto de domínios referidos nas páginas, sendo guardado o número correspondente de cada um para cada página.

Por fim, caso não ocorra nenhum erro, o estado do *website* é alterado e dado como completo na base de dados. Na eventualidade de ocorrer algum erro, este *website* é dado como incompleto, passando-se ao seguinte.

Em seguida explicam-se aspetos concretos da implementação que se acabou de descrever.

Mecanismos complementares ao OpenWPM:

De forma a obter a descrição e o propósito de cada *cookie*, foi utilizado o *website* "cookiepedia.co.uk", onde podem ser encontrados vários detalhes sobre *cookies* recenseados pelo *website*.

Apesar dos *cookies* serem obtidos com recurso ao OpenWPM, decidiu-se utilizar uma biblioteca de *Python* (biblioteca "Requests") para guardar as páginas fonte de cada *link* obtido do *website* principal. Esta biblioteca permite aceder a um *website* e guardar a sua página fonte de forma bastante mais rápida que o OpenWPM, melhorando assim o desempenho do programa. Esta decisão foi tomada uma vez que não é necessário que a obtenção da página fonte de *links* secundários seja feita de forma realista, uma vez que esta página devolvida raramente é alterada dependendo se o acesso é feito por um utilizador real ou não, e quando é alterada, apenas é necessário enviar o cabeçalho "User-Agent" de forma a obter a página fonte inalterada.

A utilização da biblioteca "hashlib" permitiu guardar o *url* do *link* visitado como o nome do ficheiro para a página fonte de cada um destes, uma vez que esta oferece uma diversidade de *hashes*. Cada *url* foi convertido utilizando o *hash* de md5. Foi ainda utilizada a biblioteca "BeautifulSoup", já referida anteriormente, para obter dados necessários das páginas HTML guardadas em ficheiros.

Dificuldades encontradas e como se ultrapassaram

Uma das principais dificuldades encontradas, foi o fato do OpenWPM utilizar um certo nível de concorrência, fazendo com que algumas das suas funcionalidades fossem executadas em *background*, continuando o programa. Desta forma, foi necessário alterar o acesso à base de dados de forma a que esta fizesse *commits* automáticos, uma vez que estava a provocar escritas concorrentes. Esta solução apenas atenuou o problema, pois se a máquina em que estiver a ser feito o *crawl* for lenta, o problema irá persistir.

Ainda de acordo com este problema, foi necessário criar esperas explicitas de forma a ter a certeza de que o programa apenas continuava depois do último ficheiro ter sido criado.

Outro problema encontrado durante esta parte, foi o fato de alguns *websites* devolverem páginas diferentes dependendo da localização (por exemplo, país) de onde é feito o acesso, o que faz com que as páginas retornadas sejam incorretas.

Foram encontrados vários problemas relativos à forma como o OpenWPM guarda os dados na base de dados ou problemas de acesso aos *websites*, sendo progressivamente resolvidos.

Vamos agora discutir como é realizada a análise de um *website*.

Como se classifica a política de *cookies* de um *website*

Para este efeito é necessário, como já referido, encontrar a página com a política de *cookies* de um *website* e depois classificar a mesma.

Encontrar a página com a política

Para classificar a política de *cookies* de um *website*, é primeiro necessário encontrar a página onde esta se encontra. De forma a encontrar esta página, é feito um pedido à base de dados para obter a página fonte do *website* principal.

Em seguida, para esta página fonte, são encontrados todos os *links* cujo texto visível ao utilizador contenha palavras-chave[1] relevantes para esta pesquisa. É então feita uma outra procura de *links*, mas desta vez procurando por palavras-chave no *url* em si (estas palavras são as mesmas já referenciadas na aproximação inicial). Ambos os *links* obtidos destas duas formas diferentes, são guardados numa lista, e posteriormente num conjunto, de modo a eliminar *links* iguais.

[1] Nomeadamente as seguintes: *cookie*, *privacy*, *learn more*, *policy*, *terms*, *click here*, privacidade, condicoes, politica, saiba mais, saber mais

É feita então a contagem do número de vezes que a palavra "*cookie*" é mencionada para todas as páginas fonte obtidas para cada *link* do *website* que está a ser analisado. Esta contagem é guardada como valor num dicionário, cuja chave é o nome do ficheiro com que cada página é guardada na base de dados, onde o nome do ficheiro contém a *hash* em md5 do respetivo *link*.

Cada *link* guardado no conjunto é então convertido para a sua *hash* em md5, e comparado com a chave do dicionário. É então calculado o valor máximo associado aos *links* onde foi encontrada uma correspondência de chaves.

Após todos os *links* do conjunto terem sido analisados, é então devolvido o *link* com o maior número de ocorrências da palavra *cookie*, bem como a sua página fonte.

Como classificar a política de *cookies* do *website*

A classificação da página da política de *cookies* é feita de acordo com os seguintes critérios:

1. Se mais de metade dos *cookies* colocados pelo acesso à página forem mencionados no texto da página da política, é atribuída a classe 1 e devolvido o *link* correspondente à página da política de *cookies*, pois considera-se que existe uma descrição razoavelmente completa da política de fato seguida.

2. Se o número dos *cookies* mencionados no texto da página for menor ou igual a metade do número de *cookies* colocados pelo *website*, é atribuída a classe 2 e devolvido o *link* correspondente à página da política de *cookies*, pois considera-se que existe algum esforço de apresentação da política seguida.

3. Se nenhum dos *cookies* colocados for mencionado no texto, mas a palavra "*cookie*" é mencionada mais de 5 vezes, é atribuída a classe 3 e devolvido o *link* correspondente à página da política de *cookies*, pois considera-se que é apresentada uma política genérica que no fundo qualquer site pode adotar.

4. Se nenhum dos critérios referidos acima for verificado, ou não existir uma página, é atribuída a classe 4 e devolvido o *link* correspondente à página da política de *cookies*, ou caso não exista, a página com o maior número de ocorrências da palavra "cookie" no seu texto, pois ou o site não revela a sua política ou não foi possível encontrar onde esta está.

Dados estes critérios, a sua utilização é bastante trivial. Relativamente ao número de ocorrências da palavra "cookie" no texto, o seu valor já foi encontrado quando se faz a procura da página da política de *cookies*. Quanto ao fato do nome dos *cookies* ser mencionado ou não no texto da página, uma vez que esta página é também encontrada pela mesma pesquisa mencionada anteriormente, apenas é necessário utilizar o pacote "BeautifulSoup" para procurar por estes nomes nessa página, guardando quantas vezes são mencionadas. Dependendo destes fatores, é então atribuído o critério correspondente ao *website*.

Apesar de não serem utilizadas para a classificação da política de *cookies* de um *website*, são também apresentadas a descrição e propósito de cada *cookie* que pertença ao *website* em causa.

Na análise do *website* são também contadas as ocorrências dos domínios que colocam *cookies* no *website*, no entanto, não foi possível, por falta de tempo, proceder a uma análise profunda e caso a caso de que informação suplementar seria possível extrair dessas contagens para melhorar o algoritmo de classificação. Este aspeto será de novo revisitado no capitulo com os resultados obtidos.

Descrevem-se em seguida as dificuldades encontradas e, caso seja possível, qual a sua resolução.

Dificuldades encontradas e como foram, se possível, ultrapassadas

Toda esta procura em páginas fonte não é certa, mesmo com recurso a uma biblioteca de *Python*. Podem ser retornadas como a página da política de *cookies* de um *website*, páginas que nada tenham a ver com este tema. Desde que a página mencione a palavra "cookie" mais vezes que todas as outras e tenha no seu *url* a palavra "cookie", será assim devolvida uma página incorreta.

Da mesma maneira, quando procurando pelo nome dos *cookies* no texto da página da política, palavras que constem na página fonte que sejam iguais a este nome devolverão também uma classificação errada. Uma atenuação feita a este problema foi procurar apenas por estas palavras no texto presente na página, ignorando assim *scripts* e outras *tags* que não sejam relevantes. No entanto, esta é a principal dificuldade de toda a parte de análise.

Existe também o problema da lista de palavras-chave que são procuradas para obter os *links* ser completamente estática, e conter apenas palavras comuns em *websites* de língua portuguesa ou inglesa. Voltaremos aqui no capítulo de conclusões.

Outro problema encontrado implica a biblioteca "BeautifulSoup". Ao procurar pelo nome de *cookies* numa página fonte, estes nem sempre são encontrados, mesmo quando o padrão inserido está correto (testado com a biblioteca "re" de *Python* e também no *website* "regex101.com").

Poderia-se utilizar esta biblioteca para efetuar a pesquisa, no entanto não seria possível filtrar palavras que aparecem em *tags* de HTML irrelevantes.

A utilização de *scripts* por parte dos *websites* para redirecionar o utilizador para a página com a política de *cookies* também apresenta um problema, uma vez que, apesar da página ser encontrada e classificada corretamente, é devolvido como *link*, o caminho do *script* (por exemplo, "javascript:document.forms['...'].submit();").

Outro problema encontrado, é o fato dos *websites* retornarem com o estado de *timeout*, não sendo possível fazer o seu *crawl*.

A utilização de uma página de entrada com botões que redirecionem para a página real do *website* torna também impossível obter um resultado correto.

Foram ainda encontrados outros problemas, nomeadamente o tempo que levava a fazer a contagem do número de ocorrências da palavra *cookie*, cuja resolução foi passá-la para a parte que acede ao *website* e guarda as informações deste na base de dados.

No capítulo 5 será apresentada uma melhor sistematização destes problemas.

4.4 Tempos de execução

Apresentam-se abaixo os tempos de execução tanto do *script* que faz *crawl* do *website*, como também o de análise do mesmo. Estes tempos foram obtidos para 3 *websites* representativos, um com poucos *links* presentes, outro com um número médio de *links*, e outro com muitos *links*.

Tempos de execução			
Website	Número de links	populate_db	site_analysis
live.com	9	110.3s	1.7s
youtube.com	95	1271.5s	5.1s
gearbest.com	904	8444.6s	27.7s

Figura 4.1: Tempos de execução para 3 *websites*.

Capítulo 5

Resultados

5.1 Resultados

A amostra e sua caracterização

Os *websites* escolhidos para a amostra são os que constam da tabela 5.1:

Estes *websites* apresentados são os 50 mais populares em Portugal de acordo com o *website* "Alexa", daí serem escolhidos como amostra para os resultados obtidos. Existem mais de 50 *websites* pois estes foram sendo alterados ao longo do tempo do projeto, variando o conjunto de *websites*.

Estes 55 *websites* podem ser separados nas seguintes categorias:

- motores de busca, como a google.com, yahoo.com, msn.com, etc.;
- sites de vídeos, como o youtube.com, instagram.com;
- sites de informação geral e técnica, como a wikipedia.com, github.com, stackoverflow.com;
- redes sociais, como o facebook.com, twitter.com, linkedin.com, etc.;
- sites de comércio, como o olx.pt, ebay, aliexpress.com;
- sites pornográficos, como o pornhub.com, xvideos.com, livejasmin.com, etc.;
- sites da banca, como a cgd.pt, santandertotta.pt;
- sites de informação, como a abola.pt, jn.pt, rtp.pt, etc.;
- sites de apostas, como o bet.pt, jogossantacasa.pt;
- sites do governo, como o portaldasfinancas.gov.pt, acesso.gov.pt.

Em seguida, será apresentada a tabela 5.1, que indica se foi possível encontrar a página com a política de *cookies*, apresentando ainda outra informações guardadas, que serão posteriormente utilizadas para fazer a análise a cada um dos *websites*.

Resultados do *crawl*

Website	**Sucesso de encontrar a página de política**	**Nº de ocorrências da palavra *cookie* na página de política**	**Nº de *links* presentes na página de entrada inicial**	**Nº de *cookies* colocados pela página inicial e pelos seus *links***
google.pt	Sim	20	37	35
youtube.com	Sim	20	**109**	32
google.com	Sim	20	36	35
facebook.com	Sim	75	48	18
sapo.pt	Sim	37	**174**	**478**
live.com	Sim	**148**	9	22
wikipedia.org	Sim	10	**310**	31
olx.pt	Sim	15	**199**	**295**
reddit.com	Sim	14	**193**	**369**
instagram.com	Talvez	12	14	28
twitter.com	Sim	31	31	43
livejasmin.com	Sim	19	**197**	54
linkedin.com	Sim	14	65	53
abola.pt	Sim	23	**253**	63
bet.pt	Sim	20	**114**	0
yahoo.com	Sim	30	62	**180**
record.pt	Sim	21	**472**	**488**
cgd.pt	Não	2	**150**	46
iol.pt	Talvez	2	**131**	**360**
gearbest.com	Sim	28	***905***	**130**
imdb.com	Sim	20	**310**	***587***
publico.pt	Sim	10	**143**	**120**
twitch.tv	Sim	45	81	**129**
zipnoticias.com	Sim	6	41	2
bongacams.com	Sim	35	**223**	**135**
xvideos.com	Sim	14	***511***	39
ebay.com	Sim	99	**297**	**413**
onoticioso.com	Sim	0	68	50
jn.pt	Sim	5	**208**	**361**

vk.com	Sim	1	16	69
aliexpress.com	Sim	12	**154**	**195**
tugaflix.com	Não	-	37	12
observador.pt	Sim	16	**257**	69
msn.com	-	-	-	-
portaldasfinancas.gov.pt	Sim	-	23	16
cmjornal.pt	Sim	29	**352**	***539***
stackoverflow.com	Sim	8	***585***	1
dn.pt	Talvez	14	***550***	**320**
pornhub.com	Sim	12	**220**	**168**
microsoft.com	Sim	78	94	96
amazon.com	Sim	23	**196**	**396**
meusresultados.com	Sim	5	**371**	**151**
custojusto.pt	Sim	9	47	**128**
popads.net	Sim	1	17	11
clevernt.com	Sim	-	0	10
rtp.pt	Sim	11	**259**	**331**
jogossantacasa.pt	Talvez	0	76	45
wordpress.com	Sim	14	80	**150**
santandertotta.pt	Sim	3	**176**	45
office.com	Sim	78	86	**180**
wease.im	Sim	-	8	43
mrpiracy.xyz	Sim	-	8	26
acesso.gov.pt	-	-	-	-
ojogo.pt	Sim	-	**225**	5
github.com	Sim	13	53	0

Tabela 5.1: Tabela com os resultados do *crawl*.

Para as páginas apresentadas cujo número de ocorrências da palavra "cookie" é "-", significa que não existe ou não foi encontrada uma página com a política de *cookies* do *website*, uma vez não é possível fazer a contagem numa página que não existe.

É ainda de realçar que esta contagem do número de ocorrências da palavra *cookie* apresentada, é relativa ao número presente na página selecionada pelo *script*, e não na página que é indicada pelo *website* como a sua página de política de *cookies*.

Relativamente ao *website* "http://www.msn.com", encontra-se um erro ao extrair os *links* presentes na página principal através do OpenWPM. Este problema deve-se ao fato de um dos *links* encontrados não estar presente no *DOM* da página ou esta ter sido atualizada. A página "http://www.acesso.gov.pt" devolve sempre um estado de *timeout*, mesmo quando acedida de maneira comum. Desta forma, não é possível obter quaisquer

informações de ambos os *websites* referidos.

Existem ainda páginas de política de *cookies* com o estado de "talvez", que podem estar ou não corretas, dependendo do dia em que o *crawl* obteve a informação, ou da interpretação feita pelo leitor.

Classificar a página com a política

Como referido no capítulo 4, os *websites* para os quais foi feita a análise foram separados em 4 classes diferentes, em que a classe 1 é considerada a melhor, e a classe 4 a pior. Foi ainda introduzida uma nova classe (classe 0) que foi atribuída aos *websites* que não conseguiram ser guardados na base de dados.

Estas classes foram atribuídas de acordo com as heurísticas e experiências realizadas para todos os *websites* referidos. Assim, as classes definidas são as seguintes:

- **Classe 1**

  Classe em que grande parte dos *cookies* são mencionados na sua página de política, explicando qual a sua utilização, sendo também referidas terceiras partes que estejam presentes no *website*

- **Classe 2**

  A esta classe são atribuídos os *websites* que mencionem alguns *cookies* utilizados, bem como terceiras partes, podendo ou não referir qual o intuito da sua utilização.

- **Classe 3**

  Nesta classe estão presentes todos os *websites* que não mencionem *cookies* utilizados nem para que os utilizam. Estas páginas apresentam uma simples descrição do que são *cookies* no geral, ou muitas vezes, redirecionam o utilizador para outras páginas com mais informações.

- **Classe 4**

  Esta classe é atribuída a todos os *websites* que não apresentem uma página de política de *cookies*, ou que tenha apenas uma breve referência de que o *website* utiliza *cookies* (a contagem do número de ocorrências da palavra *cookie* no texto da página encontrada é inferior a 6).

- **Classe 0**

  Nesta classe encontram-se todos os *websites* onde foram encontrados problemas durante o seu *crawling* e que não conseguiram ser guardados no base de dados, daí não ser possível fazer a sua análise.

Em seguida, será apresentada a tabela 5.2, onde é indicada a palavra-chave utilizada para encontrar a página de política de *cookies*, e também para as quais o número de ocorrências da palavra *cookie* é o maior. Todos os *websites* apresentados nesta tabela

foram classificados corretamente, apresentando também a classe final atribuída a cada um destes.

Websites	Classe atribuída	Palavra encontrada
google, youtube, wikipedia, reddit, xvideos, amazon, wordpress, github	3	privacy
facebook, sapo, twitter, linkedin, bet, yahoo, bongacams, ebay, cmjornal	3	cookie
live, microsoft, office	2	learn more
gearbest	3	terms
publico	3	politica
zipnoticias, observador	3	privacidade
aliexpress	3	rule
custojusto	4	saber mais
rtp	3	condicoes
onoticioso, santandertotta	4	privacidade
jn	4	termo
vk	4	terms
meusresultados	4	cookie
popads	4	privacy

Tabela 5.2: Palavra-chave utilizada que identifica a página de política do *website*.

É assim possível observar que as palavras-chave escolhidas foram fundamentais para encontrar a página com a política de *cookies*, sendo também importante diferenciar *websites* que tentam apresentar uma informação detalhada sobre *cookies*, mesmo que não refiram algum deles explicitamente, de *websites* que apenas referenciam que utilizam *cookies*.

A tabela 5.3 apresenta as mesmas informações da tabela anterior (tabela 5.2), mas para os restantes *websites*, que necessitam de uma breve explicação detalhando o porque da classificação atribuída. Esta classificação pode estar, ou não, correta, sendo que os mesmos critérios descritos acima se mantêm.

Websites	Classe atribuída	Palavras encontradas	Razão do resultado obtido
olx, imdb, stackoverflow, pornhub	2	saiba mais, privacy	Um dos *cookies* colocados apresenta um nome comum, que é encontrado erradamente no texto.
instagram	3	privacy	Não são encontrados nomes de *cookies* no texto, e o número de ocorrências da palavra *cookie* é superior a 5, no entanto, a página encontrada está desatualizada.

record, twitch	3	cookie	A página referencia um ou vários *cookies* no seu texto. No entanto, esses *cookies* não são encontrados pelo *script*.
cgd	4	saber mais	O *website* não apresenta qualquer página com a política de *cookies*, no entanto são encontradas páginas que não se referem à política de *cookies*.
iol, dn	4 e 3	privacy	Se existir alguma notícia[1] presente no *website* que seja encontrada através de uma das palavras-chave, será então devolvida esta página. Caso não exista, é ser devolvida a página correta.
tugaflix	4	-	Não é devolvida nenhuma página, quando existe uma que referencia o uso de *cookies*.
portaldasfinancas, clevernt, wease, mrprivacy, ojogo	4	-	O *website* não apresenta qualquer página com a política de *cookies*, e nenhuma página é encontrada.
jogossantacasa	4	cookie	É devolvido o *script*[2] que redireciona o utlizador para a página com a política de *cookies*, não sendo assim possível inseri-lo na barra de endereços do *browser*. No entando, o *script* devolvido aponta para a página correta, estando a contagem da palavra "*cookie*" nessa página, incorreta.

Tabela 5.3: Palavra-chave utilizada na classificação de cada *website*, com uma breve razão da classificação obtida.

Como é possível observar, era necessário providenciar uma explicação mais detalhada do porquê da classificação atribuída aos restantes *websites*, uma vez que, independentemente da classificação estar correta ou não, a sua atribuição nem sempre é trivial. Toda esta informação pode ser encontrada, com mais detalhe, no anexo A.

A seguir, são apresentados dois gráficos (figura 5.1 e 5.2), onde é apresentada a contagem de quantos *websites* foram classificados corretamente, de acordo com os critérios estabelecidos, bem como a distribuição da classificação atribuída aos *websites*.

[1] Por exemplo: http://www.tvi24.iol.pt/politica/manuela-ferreira-leite/manuela-ferreira-leite-espera-para-ver-quem-sao-os-herdeiros-do-passismo

[2] Por exemplo: javascript:document.forms['frmPrivacyContent'].submit();

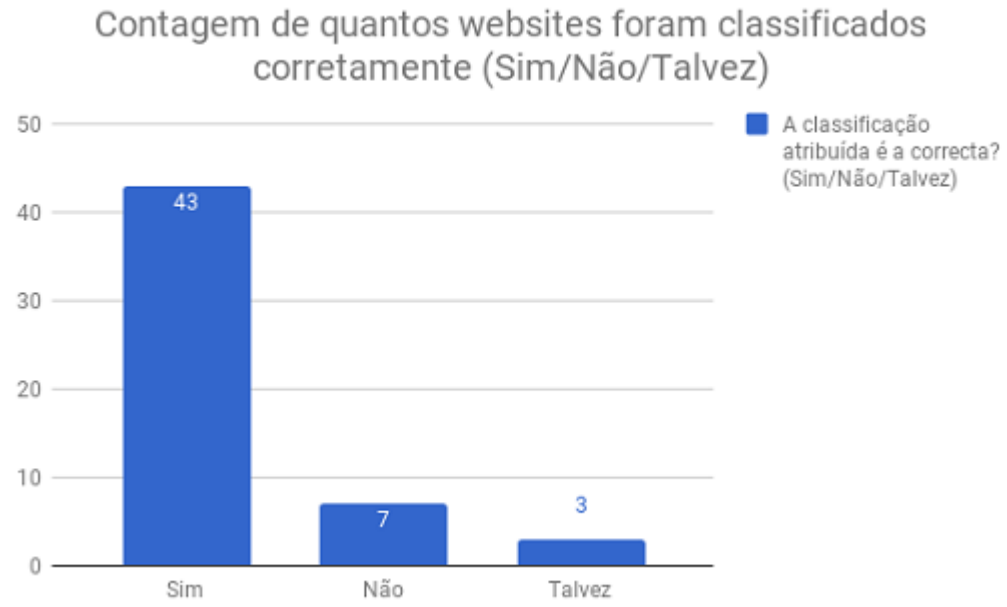

Figura 5.1: Contagem de quantos *websites* foram classificados corretamente.

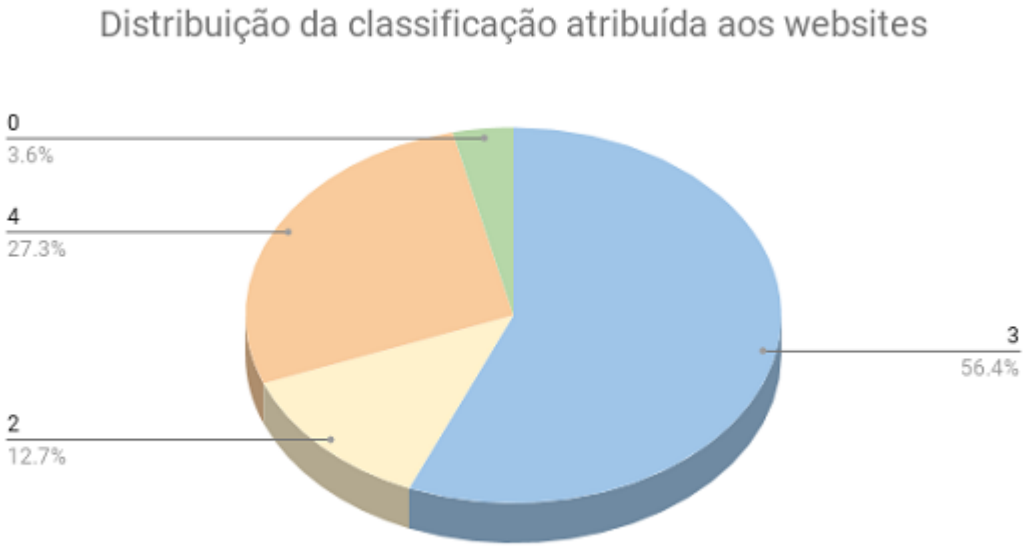

Figura 5.2: Distribuição da classificação atribuída aos *websites*.

É possível observar que nenhum dos *websites* se encontra na classe 1, sendo a mais popular a classe 3. Quer isto dizer que a grande maioria dos *websites* apenas apresenta uma página genérica onde confirma a utilização de *cookies* e outros meios de rastreio, mas não especifica qual a utilização destes.

Resultados sobre a política

Foi também utilizada a extensão "Ghostery", apresentada no capítulo 2.5, para fazer uma análise comparativa da classificação do tipo de rastreadores encontrados por esta, com a classificação do tipo de *cookies* obtida, com recurso ao *website* "cookiepedia.co.uk".

A classificação do tipo de *cookie* de origem desconhecida (*unknown*), é, na grande maioria dos casos, devida ao fato desta ser uma *cookie* colocada pelo próprio *website*, não sendo assim uma *cookie* de terceiras partes e menos propícia a rastreio agressivos e partilhas de informação entre *websites*.

A classificação do tipo de *cookie* que seja estritamente necessária (Stricly Necessary), é, normalmente, associado a *cookies* que guardam se o utilizador já aceitou ou não a política de *cookies* apresentada no *website*. No entanto, pode ainda servir para proteger o *website* de ataques especificos[3], ou ainda para correr *scripts* do *webstite*.

Por fim, é necessário mencionar que muitas das atribuições da classificação do *Ghostery* são iguais às atribuídas pelo *script*, apenas muda o nome que lhe é dado (por exemplo, o *Ghostery* atribui "Essential", enquanto que o *script* apresenta "Functionality").

Esta comparação é então demonstrada pelos seguintes gráficos (figuras 5.3 e 5.4), onde é indicada a percentagem de *websites* onde foram encontradas as classificações atribuídas aos seus meios de rastreio.

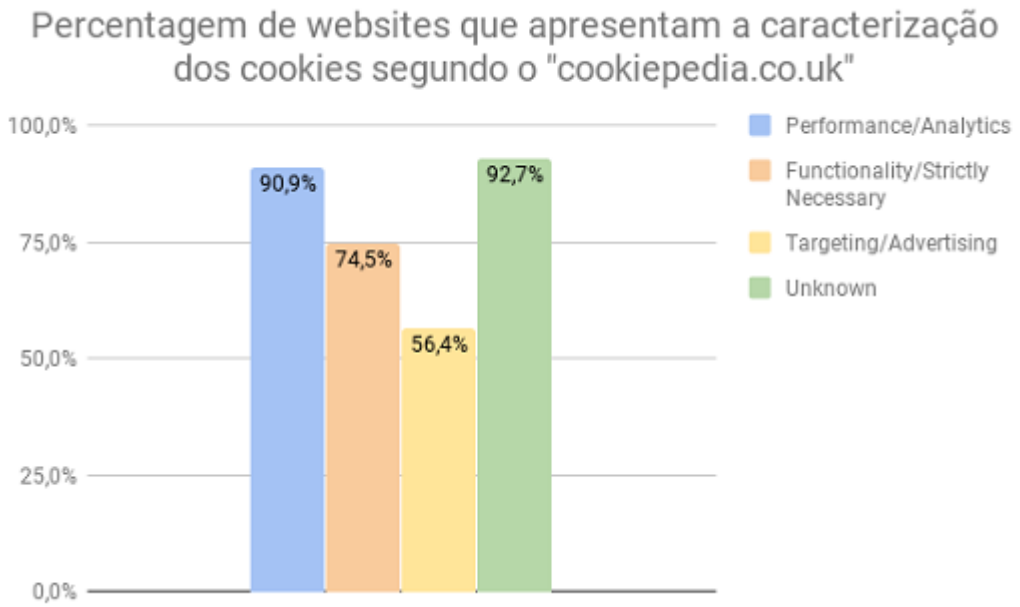

Figura 5.3: Percentagem de *websites* que utilizam os meios de rastreio apresentados.

[3]Por exemplo, o *cookie* "csrftoken" ajuda a proteger *websites* de ataques de *Cross-site request forgery*.

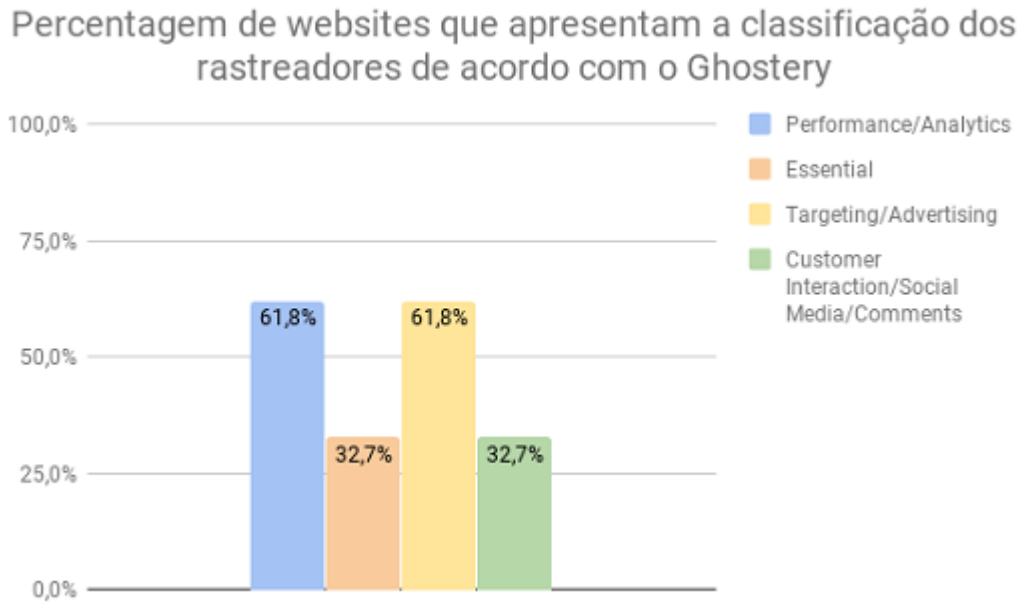

Figura 5.4: Percentagem de *websites* que utilizam os meios de rastreio apresentados.

Uma informação mais detalhada desta caracterização de *cookies* e classificação de rastreadores pode ser encontrada no anexo B.

Em relação à análise feita pelo *script* aos *cookies* colocados pelos *websites* acedidos, é possível categorizar os *websites* que não utilizam *cookies* de terceiras partes para fins publicitários da seguinte maneira:

- Os monopólios, como a Google e o Facebook, entre outros, uma vez que, para além de utilizarem outros tipos de rastreadores para este fim, utilizam os seus próprios *cookies*, não precisando assim de utilizar terceiras partes.
- Os *websites* do governo e da banca, onde não é relevante qualquer tipo de publicidade, uma vez que são mantidos pelo Estado, e os bancos dos respetivos *websites* da banca.
- *Websites* que contam com contribuições voluntárias dos utilizadores, nomeadamente a Wikipedia.
- Os que faturam diretamente, como os *websites* de apostas, alguns pornográficos e o Github.
- *Websites* de conteúdo pirateado.

Capítulo 6

Conclusões e trabalho futuro

6.1 Conclusões

Tal como tem sido posto em evidência, o rastreio dos utilizadores é uma realidade generalizada. Mesmo restringido à análise da utilização de *cookies*, uma forma relativamente fácil e "clássica" de fazer rastreio, os mesmos são utilizados a fundo, de forma a obter todas as informações possíveis dos utilizadores.

De todos os aspetos presentes nesta dissertação, um dos mais importantes a referir é o fato da grande maioria dos *websites* aqui referidos devolver a página da política de *cookies* correta. Uma vez que toda a análise é feita a esta página, torna-se assim imperativo devolvê-la corretamente. Os casos em que a estratégia seguida falhou são "cgd.pt", "instagram.com", "iol.pt", "tugaflix.com", "dn.pt" e "jogossantacasa.pt", pois são encontradas páginas incorretas através de uma palavra-chave, a política está referenciada noutro *website*, ou é devolvido um *script javascript* que redireciona para a página correta.

Outro aspeto positivo a retirar é a realização do *crawl* que, apesar de lento, é bastante completo e robusto. Falhando apenas em dois casos, "msn.com" e "acesso.gov.pt", devido aos problemas já referidos anteriormente.

É ainda de referir que os problemas relativos a esta parte do *crawl* são provenientes da maneira de como os *websites* estão construídos, o que dificulta a obtenção da informação através de ferramentas e bibliotecas externas. Isto sugere que competiria aos reguladores definir formas normalizadas dos *websites* apresentarem a sua política, o que facilitaria a compreensão dos utilizadores e permitiria mais facilmente implementar sistemas automáticos de verificação.

Um dos aspetos menos positivos é a classificação dos *websites* não estar completamente correta para uma grande parte destes. Mesmo que os domínios dos *cookies* colocados fossem utilizados na atribuição da classificação, esta não iria melhorar de tal maneira

que os resultados fossem precisos. Isto é, a utilização dos domínios na classificação das páginas de política iria apenas reforçar ou alterar a classificação para um número bastante reduzido de *websites* aqui presentes, e não para todos os que estão a ser classificados erradamente, podendo ainda introduzir classificações erradas a vários outros.

O fato de se utilizar listas com palavras-chave é também um problema, uma vez que estas palavras apenas funcionam para as línguas Portuguesa e Inglesa, sendo assim impossível encontrar páginas para *websites* que estejam noutra língua.

Relativamente à classificação dada aos *websites*, penso que o resultado é também positivo, uma vez que em 78% dos *websites* receberam uma classificação correta, com mais 6% a poderem ser interpretados como também corretos. Ou seja, se, no dia em que o *crawl* é efetuado, os *websites* de notícias não contenham uma palavra das que estão a ser procuradas, irá ser devolvida a página correta; ou se considerarmos que a página apresentada poderá estar desatualizada.

Quanto à análise comparativa da classificação obtida e a classificação dada pela ferramenta "Ghostery", é necessário ter em conta que esta ferramenta utiliza outros meios para classificar os *trackers* presentes nos *websites*, e não só os *cookies*. Ainda assim, os resultados obtidos estão equiparáveis, e por vezes, até melhores. Servem como exemplos os *websites* "livejasmin.com", "vk.com", "jogossantacasa.pt", "office.com", e "wease.im".

Foi ainda criado um servidor com uma interface "REST", recorrendo à biblioteca de *Python* "Flask". Este servidor está funcional, na medida em que apresenta uma página em que é possível introduzir um *website*, devolvendo posteriormente a análise e classificação obtida para este *website*. No entanto, é necessário que este *website* esteja presente na base de dados, com toda a sua informação. Não só é este aspeto um problema, como também o fato de apenas ser possível acedê-lo de forma local, não estando assim disponível para qualquer utilizador.

Dito isto, penso que este estudo é útil, uma vez que são discutidos vários aspetos presentes na *Web* e deste tema específico, apresentando diversos problemas encontrados e resoluções possíveis. Ainda assim, para fazer uma ferramenta interativa para utilizadores que permita classificar qualquer *website* introduzido, é algo que ainda não está na sua forma final.

6.2 Trabalho futuro

A criação de uma ferramenta *online* que permita obter a classificação de um *website* a pedido de um utilizador é um dos mais importantes pontos para o trabalho futuro. Apesar desta ferramenta existir e estar funcional, não é possível que seja acedida de forma não local, muito menos por vários utilizadores.

Em seguida, também de acordo com o ponto anterior, uma melhoria a ser considerada é a alteração do tipo de base de dados que está a ser utilizada, uma vez que a melhoria do desempenho sempre que é feito o acesso à base de dados seria necessário para tratar vários utilizadores ao mesmo tempo. Por outro lado, a base de dados utilizada é a SQLite,

que executa em memória, o que a torna mais lenta conforme for crescendo caso o servidor não disponha de memória suficiente.

Existe ainda o problema do fato das palavras-chave utilizadas para encontrar a página com a *política* de *cookies*, apenas funcionarem para palavras portuguesas e inglesas, dificultando assim acessos a *websites* escritos noutra língua.

A introdução de uma forma inteligente para classificar a qualidade das páginas com a *política*, por exemplo, através de aprendizagem automática, é também uma melhoria a ser considerada.

Por fim, dada a importância do tema, providenciar algumas recomendações para a regulação dos *websites* por parte das entidades capazes, é também para se levar em consideração.

Bibliografia

[1] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens e B. Preneel. "FP-Detective: dusting the web for fingerprinters". Em: *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM. 2013, pp. 1129–1140.

[2] *Anonimização*. https://en.wikipedia.org/wiki/Data_anonymization. Accessed: 2017-07-04.

[3] A. Barth. *HTTP State Management Mechanism*. Rel. téc. 6265. Abr. de 2011. 37 pp. DOI: 10.17487/RFC6265. URL: https://rfc-editor.org/rfc/rfc6265.txt.

[4] *Canvas Fingerprinting*. https://browserleaks.com/canvas. Accessed: 2017-06-21.

[5] V. R. Constitucional. *Constituição da República portuguesa*. Lisboa, 1976.

[6] *Disconnect*. https://disconnect.me/. Accessed: 2016-07-04.

[7] S. Englehardt e A. Narayanan. "Online tracking: A 1-million-site measurement and analysis". Em: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM. 2016, pp. 1388–1401.

[8] R. T. Fielding e J. Reschke. *Hypertext Transfer Protocol (HTTP/1.1): Authentication*. Rel. téc. 7235. Jun. de 2014. 19 pp. DOI: 10.17487/RFC7235. URL: https://rfc-editor.org/rfc/rfc7235.txt.

[9] *Ghostery*. https://extension.ghostery.com/intro#start. Accessed: 2016-07-04.

[10] *JSON Web Token*. https://en.wikipedia.org/wiki/JSON_Web_Token. Accessed: 2017-04-04.

[11] J. R. Mayer e J. C. Mitchell. "Third-party web tracking: Policy and technology". Em: *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE. 2012, pp. 413–427.

[12] *Web Storages*. https://en.wikipedia.org/wiki/Web_storage. Accessed: 2017-04-04.

[13] M. Zawadziński. *Cookie Syncing*. http://clearcode.cc/2015/12/cookie-syncing/. Accessed: 2017-05-16.

[14] *Zombie Cookie*. https://www.techopedia.com/definition/25736/zombie-cookie. Accessed: 2017-04-04.

APÊNDICE A

Tabela com o estado da classificação dada a cada *website* e a sua razão.

Website	Estado da classificação atribuída	Classe	Razão do resultado obtido
google.pt	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
youtube.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
google.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
facebook.com	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
sapo.pt	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
live.com	correto	2	Encontrada através da palavra "learn more", apresentando o maior número de ocorrências da palavra "cookie".

wikipedia.org	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
olx.pt	incorreto	2	Apesar de encontrar a página correta, a palavra "um" aparece no texto, e é também utilizada como nome de um *cookie*, devolvendo assim uma classificação incorreta.
reddit.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
instagram.com	talvez	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie". No entanto, esta mesma página não está atualizada, e a que é apresentada pelo *website* como atual, é a mesma que a do *website* "facebook.com".
twitter.com	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
livejasmin.com	correto	3	Encontrada através da palavra "policy", apresentando o maior número de ocorrências da palavra "cookie".
linkedin.com	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
abola.pt	correto	3	Encontrada através da palavra "saiba mais", apresentando o maior número de ocorrências da palavra "cookie".
bet.pt	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
yahoo.com	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
record.pt	incorreto	3	Apesar de encontrar a página correta, esta página referencia 7 *cookies*, dos quais apenas 1 é realmente utilizado. No entanto, este nome não é encontrado pelo *script*.

cgd.pt	incorreto	4	O *website* não apresenta qualquer página com a política de *cookies*, no entanto são encontradas páginas através da palavra "saber mais".
iol.pt	talvez	4	Se existir alguma notícia presente no *website* que seja encontrada através de uma das palavras-chave, será então devolvida esta página. Caso não exista, deverá ser devolvida a página correta, no entanto com um número de ocorrências da palavra *cookie* incorreto, pois é devolvido uma página em PDF, onde não é feita a contagem corretamente.
gearbest.com	correto	3	Encontrada através da palavra "terms", apresentando o maior número de ocorrências da palavra "cookie".
imdb.com	incorreto	2	Apesar de encontrar a página correta, as palavras "as" e "data" aparecem no texto, e são também utilizadas como nome *cookies*, devolvendo assim uma classificação incorreta.
publico.pt	correto	3	Encontrada através da palavra "politica", apresentando o maior número de ocorrências da palavra "cookie".
twitch.tv	incorreto	3	Apesar de encontrar a página correta, esta página referencia 1 *cookie*, mas este não é encontrado pelo *script*.
zipnoticias.com	correto	3	Encontrada através da palavra "privacidade", apresentando o maior número de ocorrências da palavra "cookie".
bongacams.com	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
xvideos.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
ebay.com	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".

onoticioso.com	correto	4	Encontrada através da palavra "privacidade", apresentando o maior número de ocorrências da palavra "cookie".
jn.pt	correto	4	Encontrada através da palavra "termo", apresentando o maior número de ocorrências da palavra "cookie".
vk.com	correto	4	Encontrada através da palavra "terms", apresentando o maior número de ocorrências da palavra "cookie".
aliexpress.com	correto	3	Encontrada através da palavra "rule", apresentando o maior número de ocorrências da palavra "cookie".
tugaflix.com	talvez	4	Não é devolvida nenhuma página, no entanto há uma que pode ser encontrada adicionando a palavra "ajuda" à lista de palavras-chave.
observador.pt	correto	3	Encontrada através da palavra "privacidade", apresentando o maior número de ocorrências da palavra "cookie".
portaldasfinancas.gov.pt	correto	4	O *website* não apresenta qualquer página com a política de *cookies*, sendo que nenhuma página é devolvida.
cmjornal.pt	correto	3	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
stackoverflow.com	incorreto	2	Apesar de encontrar a página correta, a palavra "visit" aparece no texto e é também utilizada como nome de um *cookie*, devolvendo assim uma classificação incorreta.
dn.pt	talvez	3	Se existir alguma notícia presente no *website* que seja encontrada através de uma das palavras-chave, será então devolvida esta página. Caso não exista, deverá ser devolvida a página correta.
pornhub.com	incorreto	2	Apesar de encontrar a página correta, as palavras "platform" e "orientation" aparecem no texto, e são também utilizadas como nome *cookies*, devolvendo assim uma classificação incorreta.

microsoft.com	correto	2	Encontrada através da palavra "learn more", apresentando o maior número de ocorrências da palavra "cookie".
amazon.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
meusresultados.com	correto	4	Encontrada através da palavra "cookie", apresentando o maior número de ocorrências da palavra "cookie".
custojusto.pt	correto	3	Encontrada através da palavra "saber mais", apresentando o maior número de ocorrências da palavra "cookie".
popads.net	correto	4	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
clevernt.com	correto	4	O *website* não apresenta nenhuma página com a política de *cookies*, sendo que o *script* não encontra nenhuma página.
rtp.pt	correto	3	Encontrada através da palavra "condicoes", apresentando o maior número de ocorrências da palavra "cookie".
jogossantacasa.pt	talvez	4	É devolvido o *script* que redireciona o utilizador para a página com a política de *cookies*, não sendo assim possível inseri-lo na barra de endereços do *browser*. No entanto, o *script* devolvido aponta para a página correta.
wordpress.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".
santandertotta.pt	correto	4	Encontrada através da palavra "privacidade", apresentando o maior número de ocorrências da palavra "cookie".
office.com	correto	2	Encontrada através da palavra "learn more", apresentando o maior número de ocorrências da palavra "cookie".
wease.im	correto	4	O *website* não apresenta nenhuma página com a política de *cookies*, sendo que o *script* não encontra nenhuma página.

mrpiracy.xyz	correto	4	O *website* não apresenta nenhuma página com a política de *cookies*, sendo que o *script* não encontra nenhuma página.
ojogo.pt	correto	4	O *website* não apresenta nenhuma página com a política de *cookies*, sendo que o *script* não encontra nenhuma página.
github.com	correto	3	Encontrada através da palavra "privacy", apresentando o maior número de ocorrências da palavra "cookie".

Tabela A.1: Tabela com os resultados obtidos da análise aos *websites*.

Apêndice B

Tabela com a comparação entre os resultados obtidos pelo *script* e a classificação dada pelo Ghostery

Website	**Caracterização dos *cookies* colocados**	**Classificação do *tracking* pelo Ghostery**
google.pt	Performance, Analytics, Unknown	-
youtube.com	Performance, Analytics, Unknown	Targeting, Advertising
google.com	Performance, Analytics, Unknown	-
facebook.com	Unknown, Strictly Necessary, Performance, Analytics	Social Media
sapo.pt	Performance, Analytics, Unknown, Strictly Necessary, Functionality, Targeting, Advertising	Targeting, Advertising, Performance, Analytics, Essential
live.com	Unknown, Performance, Analytics	-
wikipedia.org	Unknown, Funcionality, Performance, Analytics, Strictly Necessary	-
olx.pt	Performance, Analytics, Unknown, Targeting, Advertising, Strictly Necessary	Performance, Analytics, Targeting, Advertising, Essential, Social Media

reddit.com	Performance, Analytics, Unknown, Targeting, Advertising, Strictly Necessary	Targeting, Advertising, Essential, Performance, Analytics
instagram.com	Performance, Analytics, Unknown, Strictly Necessary	Targeting, Advertising,Social Media
twitter.com	Performance, Analytics, Unknown	Performance, Analytics
livejasmin.com	Strictly Necessary, Performance, Analytics, Funcionality, Unknown, Targeting, Advertising	Essential, Performance, Analytics
linkedin.com	Funcionality, Performance, Analytics, Targeting, Advertising, Unknown	Performance, Analytics, Targeting, Advertising, Social Media
abola.pt	Performance, Analytics, Targeting, Advertising, Strictly Necessary, Unknown	Targeting, Advertising
bet.pt	-	Essential, Customer Interaction
yahoo.com	Functionality, Performance, Analytics, Unknown, Targeting, Advertising, Strictly Necessary	Targeting, Advertising
record.pt	Performance, Analytics, Functionality, Targeting, Advertising, Strictly Necessary, Unknown	Targeting, Advertising, Performance, Analytics, Customer Interaction, Social Media
cgd.pt	Perfomance, Analitycs, Strictly Necessary, Unknown	-
iol.pt	Performance, Analytics, Funcionality, Targeting, Advertising, Strictly Necessary, Unknown	Targeting, Advertising, Essential, Performance, Analytics, Customer Interaction, Social Media
gearbest.com	Functionality, Performance, Analytics, Strictly Necessary, Unknown, Targeting, Advertising	Targeting, Advertising, Essential, Social Media
imdb.com	Unknown, Performance, Analytics, Strictly Necessary, Targeting, Advertising, Functionality	Targeting, Advertising, Customer Interaction, Social Media
publico.pt	Functionality, Performance, Analytics, Strictly Necessary, Targeting, Advertising, Unknown	Performance, Analytics, Targeting, Advertising, Essential

twitch.tv	Strictly Necessary, Performance, Analytics, Functionality, Targeting, Advertising, Unknown	Targeting, Advertising, Performance, Analytics, Essential
zipnoticias.com	Strictly Necessary, Unknown	Social Media, Targeting, Advertising
bongacams.com	Performance, Analytics, Unknown, Strictly Necessary, Functionality	Performance, Analytics
xvideos.com	Performance, Analytics, Unknown, Strictly Necessary	-
ebay.com	Targeting, Advertising, Performance, Analytics, Strictly Necessary, Functionality, Unknown	Targeting, Advertising
onoticioso.com	Performance, Analytics, Targeting, Advertising, Unknown	Targeting, Advertising, Performance, Analytics, Social Media
jn.pt	Strictly Necessary, Performance, Analytics, Functionality, Targeting, Advertising, Unknown	Targeting, Advertising, Performance, Analytics, Essential, Social Media
vk.com	Unknown, Performance, Analytics, Strictly Necessary, Targeting, Advertising, Functionality	Social Media, Performance, Analytics, Comments
aliexpress.com	Performance, Analytics, Unknown, Targeting, Advertising, Strictly Necessary	Targeting, Advertising, Performance, Analytics, Customer Interaction
tugaflix.com	Performance, Analytics, Strictly Necessary, Unknown	Performance, Analytics
observador.pt	Performance, Analytics, Targeting, Advertising, Unknown, Functionality, Strictly Necessary	Performance, Analytics, Targeting, Advertising, Essential
msn.com	-	Essential
portaldasfinancas.gov.pt	Unknown, Performance, Analytics	-
cmjornal.pt	Strictly Necessary, Performance, Analytics, Targeting, Advertising, Functionality, Unknown	Targeting, Advertising, Performance, Analytics, Customer Interaction, Social Media

stackoverflow.com	Targeting, Advertising, Performance, Analytics, Functionality, Strictly Necessary, Unknown	Targeting, Advertising, Performance, Analytics
dn.pt	Strictly Necessary, Performance, Analytics, Targeting, Advertising, Functionality, Unknown	Targeting, Advertising, Performance, Analytics, Essential, Social Media
pornhub.com	Performance, Analytics, Strictly Necessary, Unknown, Targeting, Advertising	Targeting, Advertising, Performance, Analytics
microsoft.com	Performance, Analytics, Targeting, Advertising, Unknown, Functionality	Targeting, Advertising, Performance, Analytics, Essential, Social Media
amazon.com	Performance, Analytics, Unknown, Targeting, Advertising, Strictly Necessary, Functionality	Targeting, Advertising, Performance, Analytics
meusresultados.com	Strictly Necessary, Performance, Analytics, Unknown	Targeting, Advertising, Essential
custojusto.pt	Targeting, Advertising, Unknown, Performance, Analytics, Strictly Necessary	Targeting, Advertising, Performance, Analytics
popads.net	Performance, Analytics, Strictly Necessary, Unknown	Targeting, Advertising, Performance, Analytics
clevernt.com	Unknown, Performance, Analytics	Performance, Analytics
rtp.pt	Strictly Necessary, Performance, Analytics, Functionality, Targeting, Advertising, Unknown	Performance, Analytics, Targeting, Advertising, Essential
jogossantacasa.pt	Performance, Analytics, Unknown, Targeting, Advertising, Strictly Necessary, Functionality	Performance, Analytics
wordpress.com	Targeting, Advertising, Performance, Analytics, Unknown, Strictly Necessary, Functionality	Targeting, Advertising, Performance, Analytics
santandertotta.pt	Performance, Analytics, Unknown, Strictly Necessary	Performance, Analytics, Targeting, Advertising, Essential
office.com	Targeting, Advertising, Performance, Analytics, Functionality, Unknown, Strictly Necessary	-

wease.im	Targeting, Advertising, Performance, Analytics, Unknown	Performance, Analytics
mrpiracy.xyz	Performance, Analytics, Strictly Necessary, Unknown	Performance, Analytics, Targeting, Advertising
acesso.gov.pt	-	-
ojogo.pt	Performance, Analytics, Unknown	Performance, Analytics, Targeting, Advertising, Social Media, Customer Interaction, Essential
github.com	-	Performance, Analytics

Tabela B.1: Tabela com a comparação da classificação atribuída pelo *script* e pelo *Ghostery*.