Empowering website visitors to manage their online privacy

Student Name: Jerzy Foss

Supervisor Name: Dr Craig Stewart

Submitted as part of the degree of [BSc Computer Science] to the

Board of Examiners in the Department of Computer Sciences, Durham University

Abstract —

Context/Background

Data is a fundamental element of the modern world. While data collection is often helpful to provide users with services, it is also highly lucrative and aggressive data collection can leave users feeling confused, concerned, and violated.

Aims

This paper investigates ways in which internet users can be given better control over data collected when they visit popular websites, thereby mitigating some of the risks to individuals.

Method

This paper suggests the use of a custom Google Chrome extension to automatically detect and control data collection techniques, particularly cookies and regular input fields with the intention of providing the user with an easier control method and auditing capabilities.

Results

Study participants found the cookie part of the extension to be highly useful and capable of applying preferences for all target websites, saving them time and effort. They found the risk rating system to be less useful but indicated the desire for a more complete system making use of the ratings. Participants strongly want browsers to implement elements of the extension.

Conclusions

This paper corroborates findings from previous works by others and shows that client-based tools can overall improve the web experience for users without over-reliance on website owners' goodwill. Furthermore, this paper proposes new standards for browsers to implement to aid users' ability to control cookies in a wider scope.

Keywords — data, privacy, cookies, audit, big data, privacy management

I INTRODUCTION

In the modern information age, data is often considered to be more valuable than oil. Unlike oil, data is easy to collect, store and instantly transfer around the world. While there are many subjects about which data may be collected, the most lucrative type of data collection for companies is personal data of individuals. The general public, understandably, demands a certain level of privacy with respect to their lives and privacy is regarded as important enough to feature in Article 12 of the UN's Universal Declaration of Human Rights (United Nations 1948). Despite the UDHR being created many years before the internet, Article 12 is still important in many ways. There are some obvious instances where people should be protected i.e. users don't expect a company shipping them a product to share their home address with the world. However, there are many instances where the data collection methods are much less obvious, the types of data collected are barely understood by the end user and the distribution methods are complex and global in ways that make controlling the flow of a data from a user to the wider world almost impossible.

Data privacy as a concept has been a multi-disciplinary topic of debate for many years. Politicians, corporations, academics, and consumers all have different aims when considering data privacy. While the concept of privacy has been defined in many different ways, one of the most common definitions is that privacy is the idea that individuals should be given the choice as to when, how and to what extent information about them is released (Westin 1970, p. 7). In the last couple of decades, there has been an explosion in the amount of data collected, both due to the rise of ubiquitous computing and the increased hunger of online services for personal data. Companies can collect and process personal data to provide a more personalised and relevant service to consumers and their use of such techniques can be immensely profitable. However, these systems often rely on collecting vast amounts of personal data from many people to accurately train models. Data must also be collected from an individual who is to be given a personalised service from this model. Even websites with primarily static content will often either collect data themselves, or act as a platform for major data collectors, using personalised advertising to monetise their content.

Data privacy concerns are well realised and ways to minimise these are incredibly desirable. Legislation alone is not enough; privacy literacy is low, and companies make controlling data difficult for consumers. Data privacy concepts have been studied across many papers. What consumers want from privacy protections is well-understood on a fundamental level, but companies are reluctant to change both because change is expensive and privacy-first approaches would destroy the golden goose that is data collection. Several papers propose solutions which require major changes to companies' approaches to data on a managerial and technical level for high privacy gains which keeps inertia high. Other papers have extensively studied the poor user experience related to data handling disclosure. This paper aims to find a medium between the poor corporate practices and privacy literacy.

This paper investigates how much users care about their personal data privacy and whether they feel that website hosts are taking sufficient steps to allow users to manage the data collected about them. This demonstrates how large the gap is between what users want to be able to do and what they are actually given the functionality to do. It then proposes a technical solution based on related research to attempt to improve the situation. The simple questions regarding the existing attitudes towards data privacy are used as a baseline to demonstrate exactly how well this technical solution improves the situation thereby answering the question of how website visitors can be empowered to manage their online privacy. Finding simple client-side wins that have an outsized effect on data privacy would allow users to regain control without the onus being on companies to act ethically.

One such simple win that has been chosen for investigation is that of cookie controls. Such warnings are legal requirements in many jurisdictions but there are not set standards for how they are implemented. As such, they are often designed to deliberately confuse website visitors in favour of maximising the data collection vector. Additionally, they are often in popup form which users find frustrating and will often click on any button that says 'Agree' (which may agree to all cookies without telling the user) to remove the popup from their screen and return to the task they visited a site for. This is obviously poor for both user experience and data privacy but can also be bad from a security standpoint as it trains users to click on attention-grabbing buttons to dismiss popups without much thought for what is happening in the background.

To combat this, this paper proposes a proof-of-concept Google Chrome browser extension will allow users to pre-define their desired cookie settings and these will then be automatically applied when the user visits a compatible website and then dismiss the popup. Of course, some users will want some websites to have less access to their cookies than other websites, so the extension has ways to provide such functionality. Moreover, where websites provide such functionality, the extension is able to configure settings for all cookie vendors on the website from a pre-defined list, particularly for websites with 100s of vendors but no way to deselect all of the in one go (a desirable operation for many users).

As a secondary objective, this paper proposes integrating, into the aforementioned Google Chrome browser extension, a way to measure the privacy risk of entering certain types of data into a website. While a fairly simple objective in of itself, it was originally considered as a steppingstone into creating a public database of website trust based on privacy risk. However, the trust database, while potentially good for the user, does not help in answering the research question and so has not been included in this paper. Instead, it was decided to only provide a way to measure risk on a given page.

Finally, this paper concludes by first summarising the efficacy of the implemented solution and briefly discussing it in the wider privacy context. It then finishes by proposing a set of extensions to the implemented system as well as recommendations for ways in which browser developers and regulators could jointly contribute to actions in helping to improve privacy controls.

II RELATED WORK

Data privacy scandals, such as Cambridge Analytica being given unauthorized access to data of Facebook users, have only increased the level of debate around personal privacy, especially in the digital realm. The data obtained allowed millions of adults in the US to be profiled and targeted for political advertising (Isaak & Hanna 2018). Global research indicates that 88% of people are concerned about how their data is used with 80% expecting legislators to intervene with data privacy protections (Spiekermann 2012). While the overall levels of concern are high, not everyone appears to be equally concerned, not are all concerns the same. Specific individuals may weight the importance of prior consent, lawful use/legitimate business interest and data access transparency in differing measures (Martin et al. 2016). As with human concerns about data privacy, the realistic value of different items of data can vary. The risk associated with revealing a given data item is context dependent for individual users. For example, revealing health data to a service will be considered more high risk than revealing shopping preferences and the reputation of the service will affect the perception of risk. In addition, the risk relating to revealing a particular piece of data of a given value depends on the other data being revealed as some similar data may provide little additional information (Yassine & Shirmohammadi 2009). These risks can be quantified which may be a useful starting point for analysis of websites' privacy within this paper.

Ultimately, as with many aspects of computer science, data privacy is a trade-off. A common framework for analysing data management is the Communication Privacy Management Theory (CPM Theory) which claims that sharing information and protecting your privacy are both aspects of privacy management with boundaries between private and public information (Petronio 2002). CPM Theory believes that the exchange of data on these boundaries revolves around rules and while it is commonly accepted that people own their own private data, it is also stated that people who gain access to other people's data become responsible for distributing it only according to further agreed rules. This idea of a trade-off was illustrated through meta-analysis where people with higher privacy concerns showed a lower intent to use, and lower actual use of online services but their privacy concerns had no impact on the use of online social networks (Baruh et al. 2017). The actual privacy concerns do not necessarily reflect actions, especially when the perceived benefits are great. Interestingly, the meta-analysis also revealed that people with greater privacy literacy were more likely to use online services and protected their privacy. This implies that increasing privacy literacy, for example by making privacy options and disclosure more accessible, may increase a user's ability to protect themselves in certain online scenarios and bring their actions more in line with their privacy concerns.

Over time, concepts such as Privacy by Design have emerged which involve combining the different aspects of data privacy design (engineering, management, and governance) within a company. A key obstacle with such an approach is challenging sentiments that the only way to keep advertising revenues high is to have lax privacy protections (Spiekermann 2012). However, (Yassine & Shirmohammadi 2009) showed that systems can be designed in such a way that an increased perception of privacy for consumers will have tangible financial benefits for a corporation. Furthermore, complex systems such as online social networks practically run on ingested data to make money. It is possible to make these much more privacy focused using a combination of symmetric and asymmetric encryption, giving revocable access to a user's profile to their friends. In addition, while not perfect, the system can be constructed using trapdoor functions which allow the social network to continue to make money through targeted advertising, give the user ad content they may be interested in and minimize the privacy leakage (Lin et al. 2014). As part of the process of ensuring that users give accurate trapdoors to the social network, the (Lin et al. 2014) paper states that users should be paid for their data. This is of course an additional benefit to the consumer and could help make the relationship between social network and user more balanced.

There have been some actions to help the privacy situation through governance over the years, but the issue is challenging and even before technical issues are considered, there is great variation in the attitude towards data protection from jurisdictions across the world. Severe deficiencies in the self-regulatory attitude to online consumer privacy in the US were identified (Culnan 2000). (Culnan 2000) acknowledges limitations in the study it is based on, but nevertheless identifies that only 14% of data usage disclosures properly covered areas that would be expected in a complete privacy policy. The core concerns laid out are that self-regulation is insufficient to protect consumers and that the Federal Trade Commission (FTC) does not have the authority to ensure that consumers are being treated fairly. These observations are useful as a baseline for comparing other privacy protection regimes. While on the federal level, there is a lack of comprehensive privacy protections, some regions such as the State of California have enacted more concrete laws. The California Consumer Privacy Act of 2018 (CCPA) (State of California 2018) affords much greater protections to citizens of California and can penalise any business that serves customers in the state should the company violate the rules. The EU has been ahead of the curve for some time in this respect. The General Data Protection Regulations (GDPR) were put into effect in the EU in 2018 (European Parliament and Council of European

Union 2016) and superseded the Directive on Data Protection of 1998. GDPR was widely seen as the toughest privacy protection regulation in the world and allows regulators to levy fines of up to €20 million or 4% of annual global turnover (whichever is greater). This greatly contrasts the seemingly lax approach to federal data protection in the US. While the CCPA and GDPR are different pieces of legislation, they follow reasonably similar principles around keeping the consumer safe and laying down key rights with respect to data collection, processing, and storage.

Having a comprehensive privacy policy is a core responsibility for data controllers and processors under legislation such as GDPR. It is widely known that privacy policies are usually long, drawn-out legal documents. However, some policies are even worse at clear communication, containing sentences of up to 103 words with multiple instances of the conjunctive 'or' (Pollach 2005). Most of the policies identified in this study were phrased in ways that raised uncertainty as to the true nature, extent, and frequency of data collection for the non-legal reader. Many corporations attempted to distance themselves from the act of data collection altogether in the minds of readers. However, (Pollach 2005) does acknowledge that it is unknown whether the linguistic choices are deliberate or a by-product of the legal nature of privacy policies. Whatever the case may be, this is problematic for consumers and may be wholly inadequate under more modern data protection legislation.

In fact, Article 12 of the GDPR has specific regulations for disclosure regarding data storage and processing in line with the rights of citizens. Paragraph 1 of Article 12 specifically uses words such as 'concise', 'accessible', 'clear' and 'plain'. While children are a more stringently protected category under GDPR, limited data about them may still be collected and Article 12 specifically makes the point of referencing that disclosure to children needs to be understandable by children. While (Pollach 2005) showed that their meaning may be unclear and it is often assumed that most people do not read privacy policies, from a survey of 1441 US adults (in a weighted sample) it seems that consumers categorically do not understand them (Strahilevitz & Kugler 2016). Their results conclude that the interpretation of a privacy policy by a non-lawyer will be different to that of a legal professional. Furthermore, it seems that consumers are predisposed to assuming the nature of data collection by some of the largest companies based on personal experience. Whether corporations deliberately intend to obfuscate their data practices out of maleficence or unintentionally conceal the true meaning of their disclosures, it appears that a more consumer-centred and transparent approach is required.

Though the structure of privacy policies varies greatly across websites and industries, there have been some successful studies to automatically audit them (Libert 2018). The (Libert 2018) study uses a tool called policyxray which is part of webxray. This is intended to verify that 3rd party data collectors identified on a web page are disclosed in the policy. While the technique is solid in the realm of that study, the relatively primitive approach for privacy policy analysis is insufficient for investigating how to improve data processing disclosure. In addition, there is a significant requirement for manual intervention into the process of obtaining privacy policies. Creating an automated system with Natural Language Processing (NLP) capable of converting legal text into concise, plain points is out of scope for this paper and as such auditing privacy policies is out of scope. While the majority a lot of websites were found to use transport security via HTTPS, some were still lacking so this appears to still be a relevant basic data privacy metric which can be further analysed.

While cookie warnings have existed for a long time, their prevalence and complexity has increased since GDPR came into effect while companies try to adhere to the new regulations

(Utz et al. 2019). The technique used for analysis here involved taking screenshots and manually inspected them as opposed to the semi-automated techniques for privacy policy analysis in (Libert 2018). Cookie notices seem to suffer from the same sort of issues as privacy policies. If they give any choice at all, they seem designed to be convoluted and difficult to navigate with language designed to elicit certain actions from visitors. Moreover, it appears that the public does not have an accurate understanding of what cookies are and do with most notices not helping in this respect. This may again come into conflict with Article 12 and especially violates the spirit of informed consent which is a theme throughout the GDPR. There is a web mechanism for opting out of data collection "Do Not Track" (DNT), however, it is rarely respected (Libert 2018) so will likely have no impact on cookie consent. DNT requests should allow users to block advertising and tracking cookies while allowing functional cookies such as login (in contrast to most browser cookie controls). Using the same sorts of tools that were used by (Libert 2018) for privacy policies to analyse cookie controls and help distil them into a more accessible form is in scope for this paper, even with NLP considerations. Furthermore, similar techniques with help from a library like Readability.js should make it feasible to analyse data entry points on a particular website to audit those.

III SOLUTION

The solution is in the form of a Google Chrome extension, written in JS and HTML using Notepad++ as the development environment. The largest complexities in controlling cookies and detecting what data has been input onto a page are both fundamentally related to the issue of webpage scraping. It was originally considered to use a Machine Learning based system combined with Natural Language Processing to learn where to find cookie controls and what constitutes data entry points (fields) which would work on a significant percentage of websites irrespective of layout. Unfortunately, there were a number of issues identified with this: firstly, capturing images of a webpage can be a task in of itself. Secondly, finding out how to navigate through menus and popups visually (i.e. making the AI act as a user) and sandboxing these interactions to find the correct cookie menu for an unseen website would be quite hard. Thirdly, manually curating a large enough dataset of webpages to learn from would likely take more time than building the entire user control system several times over. For these reasons, it was decided that a less powerful but simpler approach of web scraping via the Document Object Model (DOM) had to be used. Since even in removing the visual aspects of a webpage, there an infinite number of ways to construct the HTML of a webpage, this presented unique challenges in balancing the generalisability of the solution with providing enough power on enough popular websites to actually provide some functionality. For example, a website may use the HTML class attribute with value ”cookie-control” on all inputs related to cookies, however other websites may use only ”cookie” or use something completely different like ”input-box-123”. It is therefore not possible to make assumptions about the layout. Being too specific by searching for ”cookie-control” won't generalise but on the flip side, the system risks detecting absolutely every input related to cookies or not which is undesirable. Due to the nature of Chrome extensions, there are 3 separate JS files that make up the solution. These are the Popup, the Options and Content (a 4th Background file is included but unused). Broadly, the Popup handles the display of the popup within the browser toolbar and pipes user interactions into Content for processing. Options is accessible via chrome://extensions and allows the user to manager their cookie vendor options. This was placed here since there are too many vendors to fit in a popup without clutter. Content does the majority of the processing regarding cookie controls and risk calculation.

A Using Content to detect cookie controls

The first stage requires identifying which controls on a page actually relate to cookies. The basis of this is a simple DOM scan to detect all HTML input tags which have type="checkbox". Some websites do not use checkboxes (even restyled as toggle switches) and will sometimes use buttons or standard divs styled to look clickable and with bound listeners. For this system, it has been assumed that input[type="checkbox"] is being used for every cookie control since detecting any element with an event listener isn't possible from a Chrome extension. The most important part of detecting inputs with the correct labels is a depth-first-search (DFS) for text nodes. Figuring out where to start this search is a little bit difficult. However, a reasonable assumption can be made that any text used to describe the input to the user will probably be nearby in the DOM. As such, a DFS root is chosen from any labels found within the parent node of the input and if this finds nothing, it is run again just from the parent node. It may seem like just detecting a label is enough but label tags are not a requirement so some sites may use regular divs. Additionally, sometimes developers will place tags within labels (such as spans to style part of the text differently) so running a DFS is necessary to dig down into this hierarchy. The DFS works by starting at the root and searching down the tree until it reaches the end before backtracking and searching other paths. In this case, each time some text node is found, the text is stored alongside the input that it is likely related to. An important point is that our implementation uses a JS array as an explicit stack rather than recursion since we cannot know in advance the depth of the DOM subtree to be searched and we do not wish to hit a stack overflow error due to exceeding the maximum recursion depth. While it would take a poorly designed website to hit these limits on modern browsers, a more robust implementation is preferred.

Well-designed websites will often make use of the aria-label HTML attribute. This is an accessibility feature often used by screen-readers but the advantage of it for our purposes is that it will usually be directly on the input element and accurately describe the element even if the surrounding regular labels do not. As such, aria labels for every element are also detected alongside the DFS resultant labels. At this stage, any field/label pairs that have the same detected string are grouped for both the aria labels and regular labels individually. It is obviously entirely possible that the same checkbox has been detected in both the aria label and regular label scan so it is necessary to somehow decide which label is correct if they differ. Due to the properties of aria labels, it was decided that an aria label for an element with some text will always be more important than another label and as such the regular label will just be ignored. The algorithm for performing this merge requires 4 levels of nested for loops: one to iterate over all regular label keys another to iterate all associated entries with this key, then the same for the aria labels for each regular entry. This may seem like a $O(n^4)$ algorithm. However, in practice for a well-designed website there will usually only be 1 or 2 entries for each label key for a regular label or aria label so the real-world upper bound is more like $O((2n)(2m))$ where n is the number of detected regular labels and m is the number of aria labels. This is made slightly worse since aria labels need to be iterated again to fill the gaps leading to an overall upper bound around $O((2n) * 2(2m))$ for the merge step.

The result of the merge can then be partitioned into cookie controls (i.e. cookie checkboxes), vendor controls (i.e. vendor checkboxes if they exist) and block cookie controls (i.e. select all). This is quite a simple algorithm which simply checks the id, class and name attributes as well as the text in the detected label to try to determine whether the input should be classified as a cookie, vendor or block control. At this stage, a taxonomy can be constructed. The one used for this system is very simple and detects keywords within the detected labels and then tries to match them to a set of predefined cookie types which appear on a large number of websites. The groups are core, advertising, functional, location, device, analytics, storage, personalised, social, research, products, offline, security. Multiple keywords can be associated with a single group which maximises the chance of a group being found for a given cookie. Additionally, anything that can't be sorted is added to an unsorted group and is controlled via other in the Popup.

B Using Content to detect cookie save buttons

This algorithm tries to find anything that may be similar to a ”Save and Close” or ”Save Preferences” button by looking at all input[type=”button”] and button HTML tags. To ensure that the system never tries to press random buttons on the page which could lead to submission of undesired forms, the search starts in any elements that have an id containing the term ”cookie”. This isn't ideal since it costs generalisability, but it prevents a lot of unintended behaviour so is a necessary evil.

C Managing cookies via the Popup

The Popup consists of some simple HTML using the Bootstrap 4 library for styling and is primarily comprised of toggle switches (checkboxes) corresponding to each group of cookies with an option to select or deselect all cookies. Each checkbox has a value change listener that automatically tells the group of cookies detected on the active tab to select or deselect based on the popup value. When the ”Apply Cookie Selection” button is pressed the Popup will attempt to trigger a save click on the page. This is done by making use of Google Chrome's messaging API to send changed data from the popup to the Content for the active tab which then applies it. The messaging API must be used to send desired states since the detected fields in the DOM of a tab cannot be accessed by popup code. The popup also contains a toggle for automatically attempting to apply cookies. This setting and the cookie settings themselves are all saved in Google Chrome synced storage when the ”Store Cookie Selection” button is pressed. This means they will be maintained across devices when logged in to a Google account. Of course, a privacy-conscious user may not wish to remain logged in so synced storage falls back to local storage by design in which case the system's options will just be stored in the browser so they can be used on multiple webpages.

D Automatically applying cookie settings

When the automatic application toggle is checked in the Popup, the extension will automatically attempt to apply the settings when a tab is opened. This consists of scanning the page for cookie options, setting them if they exist and then programmatically clicking the ”Save Preferences”-esque button. Since cookie warnings often appear later than the regular page content, this creates an issue whereby the Content detection cannot find any cookie elements on page load. This is especially the case when the warning is part of an external JS file which could be asynchronously loaded in any order. There is no easy way to overcome this since there are no listeners for this functionality and the system cannot know whether the warning is part of external

JS and if it is when it has loaded. To overcome this limitation, the automatic trigger is delayed by 1 second in the hope that the time will be sufficient for content to load. There are cases where this will not happen such as over slow networks. However, the Popup has a manual way to apply settings as mentioned so while this limits the user experience it isn't game-breaking.

E Managing vendors

The detection of vendors is done through the same mechanism as the detection of cookies but with filtering on the id, class, name and content of the detected checkbox as mentioned in A. In this case it is only possible to look for the word ”vendor” in any of these since vendors will generally be listed as company names and as such could have any value. On a lot of websites, the vendor checkboxes only become part of the DOM when beginning to navigate through the cookie popup so to prevent the extension breaking, the Popup contains an option to ”Rescan Vendors” which sends a message to Content to run the label detection code again. Once vendors are detected, the Popup can save all vendors on the page using the ”Store Vendors” button. This has a special mechanism compared to storing cookie options. Many pages will have 100s of vendors and the data are too large to store in one item Google Chrome synced storage which has a quota of 8192 bytes. Since storing all data in one item is preferable as it prevents the need to detect keys and try to build up complete JS objects, synced storage isn't suitable. As such, vendors are only stored in the browser local storage. Storing vendors will save any vendor that hasn't been seen before into the browser. Vendors that should be allowed can be chosen via the Options page which is only accessible via the Extension options feature within chrome://extensions. This isn't an ideal place to put these options. However, over many webpages there could be 1000s of vendors stored in the browser so displaying them in the Popup may make the Popup difficult to navigate or even unresponsive which is why it was deemed better to hide these choices slightly. All vendors have switches to allow or disallow them and of course there is an option to select or deselect all which some websites do not provide themselves which is obviously poor user experience especially when a page has a large number of vendors. Vendor choices are never automatically applied on a page. This is primarily due to the aforementioned issue of detecting vendors on page load. As such, the method to make use of the vendor feature is to ”Rescan Vendors” and then ”Apply Vendor Selection” when the correct page of the cookie options is open in the active tab. When displaying the list of available vendors in the Options page, it was crucial to implement a function to escape any valid HTML characters before including vendor names in the DOM to help protect against cross-site scripting attacks (as unlikely as they may be in the extension management page).

F Determining data entry risk levels

The risk analysis system first attempts to find all inputs on a page using the label/aria label method as in A. All the detected labels are then compared to a taxonomy which tries to determine what data each input is collecting based on a set of keywords. For example, ”first name”, ”forename”, ”surname” will all be considered types of ”name”. The keywords ”city”, ”county” will all be considered part of ”address” and ”passport” or ”driver's license” will be considered types of ”id”. While the special categories of personal data that are protected under GDPR (European Parliament and Council of European Union 2016) are in some cases unlikely to feature in fields (i.e. biometric probably won't have a field on any website), it has been included in the search for demonstration purposes. After detecting the broad groups of data item, the task of classifying risk levels can begin. This is a bit of an arbitrary process that is mostly based on knowledge of what each data type is and how it can be used both for legitimate or malicious reasons. For this paper, there are 6 levels of classification, but obviously multiple groups of data items can feature at any given level. Roughly speaking, the risk levels represent increasing severity of outcome should the data items classified at the given level be leaked or collected without lawful basis. At level 1 are name, appearance, country, gender, age. These are items that people will generally share on social media for the public at large. At level 2 are height, weight, job, education, marital status and relationship status. These are items that people may not always want to share due to risk of embarrassment or simply because they are trivia items they feel people do not need to know, but there is generally low risk of long-term damage for misuse. At level 3 are email address and phone number which are data items that people don't generally want shared due to spam and robocalls. In level 4 are address and date of birth. Address is quite obvious as most people do not wish to share their personal address for fear of harassment. Date of birth is a bit strange and may be better placed in a lower category. However, it was placed in level 4 for this paper since it is often used to verify someone without the use of official id. Also at level 4 is a category termed security. This includes fields such as favourite XYZ, pets or mother's maiden name. While these questions seem relatively harmless in of themselves, they are commonly used as security questions on websites which can be quite harmful for account security. While security conscious individuals will likely use random answers for security questions, the general public often won't and as such they receive a level 4 classification. At level 5 is information about a user's financial situation or their possessions as these may increase the chance of burglary or other external risks upon misuse. Finally, at level 6, are the legally protected data categories and banking details, including account numbers and CVV codes. While race and ethnicity may not seem like they should be regarded as high risk as bank details, it seems prudent to strongly protect data which are already legally protected especially where they may increase the risk of discrimination or harm to the user.

G Data entry risk calculation

Creating a final risk score for a page was quite difficult. There were a few considerations that largely influenced the calculation. Initially, trying to sum all the risk levels for every field was tried. This worked in terms of having a higher number represent a higher risk but the number was unconstrained so with many fields the number would become very large but for a user the difference between 2 very large risk numbers would essentially be meaningless. The second option was to use essentially a weighted mean of the different levels. This created an issue where a page with 100 inputs at risk level 1 and 1 at risk level 6 would average down to near 1 risk which doesn't accurately convey the risk of the level 6 field. Ultimately it was settled that the risk levels wouldn't be combined and the user would just be shown the number of fields at a given risk level in the Popup when using the "Scan Risk" feature and be free to interpret these as needed.

H Lifecycle

This project started out with a prompt relating to the use of intelligent agents to manage user data. Several different broad categories were considered at this point such as creating a machine learning system to help users monitor all the data leaving their device. The prompt and initial ideas relating to what could be created were used to choose relevant papers to read as research. The reading informed what had already been done and where there was room for further work. Both from the reading and the realisation that monitoring all data leaving a device is hard for security and ML reasons, it was decided that browsers, and in particular cookies would be the focus of the project. This led to the research question of ”how can website visitors be empowered to manage their online privacy” which then was further split down into basic, intermediate and advanced objectives with the basic objective of the cookie management part being a priority. At this stage, some research was done into creating extensions for different browsers and their features before Google Chrome being settled on since the API was well-documented and had a rich set of features which would work for scanning data on a webpage via a popup. A plan was made regarding how much time needed to be spent on each aspect and a GANTT chart was used for this. Generally, tasks included on the chart were more fine-grained than the simple objectives and included high-level programming tasks for the extension itself where this was feasible e.g. automate cookie controls. However, the plan based on the initial objectives was not static during the project execution since issues arose which needed further time or re-planning. An example of such an issue is that the detection of labels turned out to be much more difficult than originally anticipated and another is that navigating through cookie popups was infeasible, so a different approach had to be undertaken which led to some limitations as discussed in V. After the implementation had been completed, the resulting system was evaluated for correctness before being handed off to test users.

I Implementation process

The implementation was undertaken using iterative development with features from rapid prototyping. Iterative development involves planning out a component based on its requirements from the end user, designing a potential solution based on the plan, implementing it and then testing/analysing it for correctness against the objectives. However, in this case there was no end user giving a set of requirements, so the iterative cycles instead focused on building a prototype to achieve one of the high-level objectives laid out at the start of the project and based on related works. The act of breaking down the problem into chunks, designing a solution and then testing it was especially suitable for this project as the parameters and scope needed to be re-evaluated every time a component was deemed too complex, or the time required to implement it had been underestimated. Furthermore, in instances such as realising that detecting all clickable elements on a page isn’t currently possible, the design of the entire detection component had to be adjusted so the project benefited from the flexibility of iteration. Broadly speaking, as the software progressed, each prototype gained more features. For example, it went from being able to detect fields and output to the console, then detect and classify, then gave the user control over cookie fields, then allowed for option saving.

J Testing

A test plan was created that mostly focused on interactions with the extension Popup, a form of black box testing. Since the user was not able to enter arbitrary data through a set of switches and buttons, there was no erroneous or boundary data to consider so all tests were designed to test normal user interactions. Since the extension was not designed to handle websites designed in completely incompatible ways, these were not tested with the extension. Instead, tests focused on websites using a particular format of Popup using checkboxes that the extension was designed and expected to handle. Creating automated test cases wasn't really possible without replicating existing webpages and programming them with ways to output a ground truth, then comparing these truths to the detections and options in the extension. As such, the testing was all manual and mostly undertaken by visual inspection of the webpage and the underlying DOM compared to console output or the state of switches within the extension. For example, it was visually verified that when a switch was changed in the extension, the correct corresponding switch was changed within a website's cookie popup. This was performed for each cookie group on a compatible test website. If not manually hard coding in the states expected when each switch is changed for each website then an automated test approach would be at least as complex as the actual task of detecting and controlling the correct fields in the first place. A second example of tests regarded the save features. The synced and local storage were tested programmatically by saving some data and then attempting to restore it from storage, checking if the 2 values matched and also checking in the console to determine the format that storage returned data in.

K Verification and Validation

The final creation was verified by checking completed features against the original objectives since these were essentially the design specification. All of the features mentioned in the project plan barring the abandoned trust database were implemented and worked as per tests performed under J. The system was validated in a very specific way. Since in this case the system was designed based on objectives derived from work in other papers and analysing websites, there was no end user to initially lay out requirements. However, the passing of the system to users to test at the end and provide feedback on as in IV did act as a validation. In this case, the study participants were considered black box testers who tried the holistic system without any analysis of the code. Their responses demonstrate whether or not the correct software has been created to meet the objectives.

IV RESULTS

Returning to the original research question, this project's core creation is a client-side system to help website users manage their online privacy. Clearly, internally running tests on the system itself as in Solution part J analyses the correctness of the code relative to the stated objectives. However, it does not help to answer the research question. To do this, a study with real participants was undertaken, evaluating their attitudes to privacy without the system and whether they felt using the system improved their feeling of control, hence answering the research question. The format of the study was as follows: a group of users who browse the web on a regular basis were asked whether they wanted to participate in the study and given a consent form to complete if the answer was yes. Once completed, the user was provided with a copy of the Chrome extension and asked to install it in their browser. They were then provided with a list of 10 websites and asked to visit them and attempt to use the extension on them. They were also given information about how to clear only cookies that store cookie preferences for a given website to ensure that they had a cookie warning actually appear without clearing all browsing data and being signed out of any accounts. The websites chosen were mostly popular news sites with a band website and a story publishing website also in the selection. The full list of known-compatible websites

tested is as follows:
https://ico.org.uk/ - UK Information Commissioner's Office
https://time.com/ - News
https://www.wattpad.com/ - A story-writing/sharing website
https://nypost.com/ - News
https://edition.cnn.com/ - News
https://www.cnbc.com/ - News
https://www.cnet.com/ - News
https://www.hotjar.com/ - Behaviour analytics for websites
https://eu.usatoday.com/ - News
https://www.skillet.com/ - A band
There were a couple of reasons for the websites chosen. Firstly, these were all reasonably well-known types of website that a regular user will come across within their daily browsing and as such more representative than picking subject-specific websites such as stackoverflow.com. The second reason was that all the test websites used cookie warnings arranged in specific ways. The majority of websites are news-related since browsing articles is a common task and it is likely a user will encounter popups from them if they end up reading particular stories only available from one source. Furthermore, many news websites seem to follow the aforementioned cookie control format with only minor differences. When shortlisting websites that might be included, each one was tested with the extension to ensure that it was compatible. While this did limit the overall testing of the usefulness of the extension, this project only had a goal of creating a proof-of-concept that worked for a cross-section of websites and as such testing on websites that cannot be accurately detected on did not allow for the creation of a comparison to a baseline. However, it did still demonstrate that the extension was generalising to some extent as it was not hard coded for a specific website, nor were all the chosen websites' options completely identical. The websites provided were intended to test the cookie selection, the automatic setting of cookies and the vendor selection. The risk rating system could not be tested on the homepage of any of the provided sites, so the user was asked to use any site that has a form requiring personal information. Once the user had completed tests on the website list, they were asked to use the extension for some regular browsing. However, given the likelihood that a lot of websites visited either didn't have available cookie options or were incompatible for reasons mentioned before, this was an exercise to get a feel for using the extension more generally.

Once the user had completed their browsing, they were asked to complete a questionnaire with each question asking the user to provide a rating between 1 and 10 with 10 being the highest or best depending on context. After all users had been surveyed, the mean of responses to each question was taken and the results of this are mentioned below.

There were only 5 participants in the study at this stage so there are some overall limitations. See V for more details. On average, people rated their computer/internet literacy (a question asked since future research could be done with these results to see if the system is useful for all demographics) at 8.2 and their concern for personal data privacy at 9.4. Respondents rated their attempts to protect their privacy at 8.8. However, respondents also demonstrated that there is a significant divide between their desire for privacy and their ability to control it by rating websites' data collection transparency at 2.4 and their cookie controls at 4.2. This reinforces the research from (Baruh et al. 2017) stating that there is a divide between intent and reality when it comes to personal data privacy. Respondents rated their understanding of risks relating to sharing specific data items with websites at 8.8.

Compared to the baselines above, users rated the usefulness of the extension for automatically applying cookie preferences at 9.6 and strongly indicated they would like web browsers to automatically apply cookie preferences without a popup, giving such a theoretical capability a rating of 9.8. However, respondents only rated the risk rating system in its current form at 7.8 suggesting that while it has some use, it isn't that useful, especially compared to the cookie assignment system. Interestingly, the lowest rating given for the risk system was 5 which is above both the data collection transparency and cookie controls for websites. Despite this, respondents had a positive view of a more complete risk rating system; they gave a theoretical rating for a system that automatically analysed every page, storing cumulative results and creating a public database of intrusiveness ranking for popular websites of 9.2. Since respondents generally highly rated their personal understanding of risk, it is also possible that the risk rating system was only less useful to the particular group trialling the system but may still be useful with a broader sample. While simplistic, the results for cookies show a clear improvement over the baseline ratings when using the extension. Nevertheless, the high rating for a theoretical native cookie feature suggests that there is still a lot of further work to be done and room for improvement. Such a feature would, of course, solve the generalisation problem if a large enough number of websites and browsers successfully implemented it.

V EVALUATION

As seen in the results, the cookie settings part of the project is definitely a success despite the overall narrow scope. This means that this proof-of-concept system is valid and fulfils the requirements. Not only did testers quantitatively respond well to it compared to the baseline but some used the additional comments section of the questionnaire to reinstate how this is a great time-saving measure. One improvement that could be made to this component immediately is to tell the user when no valid cookie fields have been detected. This would be helpful for websites where the extension doesn't work to inform the user of the incompatibility rather than have them consider the extension as malfunctioning.

While the vendor settings are useful, they are weak compared to the cookie settings due to the lack of automation derived from the fields not being added to the DOM in many webpages until the correct part of the cookie warning is opened. The risk rating system is somewhat weak in that it could do with some information about how to interpret the different levels. For example, having a way to click the different levels and giving the user some information about the types of data associated with the level and why they are considered a risk would likely improve the ability to meet the risk analysis objective, particularly for users who are not intricately familiar with data protection. While the 2 columns are labelled with the risk level and the number of data items at each level, some testers did understandably use the additional comments section of the questionnaire to point out that the risk UI in its current form is a little bit difficult to parse. Furthermore, some changes to the UI such as colour coding the levels and better styling them would improve overall usability.

As previously alluded to, there are several limitations which are why the scope had to remain narrow. Unfortunately, it was determined that there is no easy way to detect which elements are actually within a cookie popup (barring an implementation of an advanced ML imaging system), or even to detect which element is focused so there is a significant chance of all checkboxes on any page being detected. The button detection mechanism to automatically find and click a ”Save Cookies” style button on the webpage suffers from this issue in an even larger way and may find and click any button that is similar to ”Save Preferences”. This is an issue for websites that have many forms or may contain user accounts where there are certainly contexts of preference. In practice, a lot of popular websites that have detectable cookie controls in the first place will fall into similar categories to the 10 test websites and won't have user accounts or won't have preferences on the same page that a user without any selected cookie preferences is likely to see a popup on. The issue with reasonably detecting buttons is also why the extension is unable to handle websites that style their cookie controls as switches with buttons in the DOM - there would be too many potential candidates and changing the state of this pseudo-switch would involve emulating a click which could trigger all sorts of unintended actions on the website if the detection isn't fully precise. Furthermore, websites that use buttons in this manner often also have short vendor lists that use AJAX (send asynchronous requests for more data) to load more vendors when the user scrolls to the bottom. To implement vendor controls for these would require emulating scrolls until no more data is found and then comparing the full list. Depending on the speed of network requests, this could end up particularly slow. A second class of website that the extension my not be able to correctly handle is any website where the cookie popup displays in an iframe. Depending on the configuration between the main website and the embedded content, the extension may not be able to correctly scan the iframe content if it has access at all. The final class of website that the extension won't work correctly on is any that don't use checkboxes or even buttons to control cookies, but instead style arbitrary HTML tags to look like clickable elements. There is no way in Chrome extensions to detect all the click handlers on elements and therefore determine what can be clicked and what can't. Even with better detection systems, this 3rd type cannot be implemented and instead would take a highly advanced ML system that can understand the semantic meaning of different types of element and attempt to click correctly.

A key lesson learned, from the standpoint of the system implementation itself, relates to the limitations and is how complex and technically challenging the implementation of even low-level intelligent systems that generalise can be. It seems likely that no matter how mature an implementation of a primitive system becomes and how many cases are considered, there will probably always be a sufficient number of slightly different websites that the exceptions end up swallowing the rule. Moreover, minor rearrangements to the underlying DOM structure of a compatible website could render the solution no longer compatible. As such, and while simplicity is preferred, complexity of a future system implementation is almost inevitable. In continuation, a solution seeking to work in 99%+ of cases would probably require a system that is capable of powerful image recognition and near human-level reasoning to accurately infer semantic meaning of website structure.

The results are also quite limited since the test group of users was necessarily very small. In this case, the sampling was purposive with a small chosen target group as opposed to a large random sample of the population. For this reason, the study is more qualitative. Ideally, to make the study quantitative, the project could be given to at least 1000 people of varying demographics to test. However, finding such a group in the first place would be difficult. Furthermore, not all participants would know how to set up extensions within their browser, particularly the manual way (outside the Chrome Web Store) which is required since for this project the extension is unpublished. Even with screenshots, guiding them through and answering questions would have been infeasible given time constraints. In non-pandemic times, setting up a large number of pre-configured devices, choosing assistants and bringing participants in to complete the study would be the preferred solution for including a large number of testers. The test data is also qualitative and relies on users giving what boil down to their ”feelings” regarding the situation with an without the extension. For future work, a more quantitative approach could be taken. For example, it may be possible to measure the time taken for each user to start doing a task on a website that they originally intended both with and without the extension. The difference between these would show time savings but wouldn't necessarily measure the difference in ease of use since an experienced user who understands privacy and is used to cookie popups will likely not find a popup challenging in the first place so for them only time would be relevant.

While the organisation of the project was generally solid to the extent that it was completed with all final objectives, there were a few issues. Firstly, the original time estimates for completing certain tasks such as the label detection turned out to be very far out. While on the face of it, the concept of web scraping, even in this application seems quite simple, it turned out to be far from it. The intricacies required to generalise a primitive solution enough to even handle the test websites are quite severe. As such, there should have been 2 to 3 times more time allocated for this process which essentially involved testing small tweaks on the different websites until things worked, without giving in to requiring a specific DOM structure. Secondly, the original project plan mentioned Natural Language Processing to try to classify the input labels. Any form of ML for this application was significantly too ambitious and shouldn't have been included in any goals. However, despite this, early research did demonstrate that it wasn't suitable so there was very little time lost in considering this path. The approach of only planning broad tasks and creating small prototypes of features before integrating them into the complete system iteratively was definitely the correct methodology. This is primarily because creating an overall top-down design at the start wasn't possible given that many of the minutia of implementation were only discovered at development time. As such, the chosen strategy allowed for significant flexibility and as long as individual pieces of code were designed with integration in mind i.e. functions returning sensible data structures, there was no risk of creating confusing, unmaintainable code. This contrasts with deeply designing the entire system before attempting any implementation and then all tests failing since the system had never been exposed to live websites. One part that wasn't designed by iterative trial and improvement was the Popup UI. This wasn't started on until a significant portion of the scanning code had been completed and as such there was some idea of how the user would interact with the system. These ideas were used to create some quick sketches which were very close to the final version for the cookie settings component at least. The vendor rescanning controls were not planned and had to be added later after limitations were found. As such, they are not as user-friendly as they could be. In terms of organising projects, a lesson was learned to under-promise with the initial objectives and then hope to be able to over-deliver if all objectives end up completed. If this is not done, the initial goals can be unmanageable and potentially unfocused on the task at hand. This was the case for the abandoned idea of a public trust database.

VI CONCLUSION

This paper has reinforced the findings in prior works that end users care about their privacy but that there is a conflict between the desire for privacy and the practical ability to have privacy while still actively using the internet. This imbalance is due to the ways that website owners choose to implement their data controls. While prior work focused on privacy policies, this focuses on cookie controls. While websites may implement poor controls due to a lack of understanding of data privacy or because it benefits their business to obfuscate their data processing, this work has shown that there are feasible ways for users to regain some control over their own data without the reliance on a high-level of goodwill from data collectors. Users have also indicated that they understand the risks of sharing data so don't necessarily need assistance with this but would like to have an easy way of knowing who to trust. This project's solution works purely on the client-side and is a Google Chrome extension for easily allowing users to control their cookies without reading cookie warnings and often being able to apply preferences automatically. While the study results were positive, the test group was very small and further work perhaps should be undertaken on a broader group with control for different demographics to ensure that the results still hold. Nevertheless, the created extension has been shown to resonate with users and they strongly would like further work to be undertaken in extending the scope of the solution. As mentioned in IV, users would like for browsers to implement cookie controls that give such fine-grained control rather than just the regular all-or-nothing option of blocking all cookies from a site that can break login systems. In this vein, this paper now proposes a general solution for extending the proof-of-concept system by implementing cookie controls natively in the browser. A potential form for this would be the updating of web standards such that browser JavaScript has an API for website developers to register all of their cookies and their associated groups with the client i.e. cookie named X is a location cookie, cookie Y is a social media cookie, cookie Z is a functional cookie, Z is required for the website to work properly. The same could be done for registering vendors. The browser should then be able to read this registry provided on the website and automatically apply the user's predefined cookie options upon a visit, hence banishing obtrusive, ambiguous cookie popups to an archive of a darker time for the web. Unfortunately, this does require a certain amount of goodwill from website owners and developers, but a simple series of hooks would likely be much easier to publicly pressure companies to implement than attempting to force them to abandon mass data collection practices altogether. Once some popular websites start to make use of the browser API, it could eventually become a demanded standard that the public comes to expect. Moreover, if regulators were to regularly use their enforcement powers to ensure that companies are following the spirit of laws such as GDPR, particularly with respect to transparency of data collection, this would further increase pressure to adopt the new standard. With respect to the repeatedly mentioned trust rating database for websites, a proposal for how this might work is as follows. The risk analysis system could be extended to automatically detect filled fields on pages though this would be a challenge with detecting actual form submissions. These risk levels for analysed websites would then be sent to a centralised server along with a user-controlled rating for the website if they choose to complete one. A rating in this case might be a yes/no response to "do you feel this website adequately protects your data?" Continuing, data would be combined with publicly acknowledged data breaches as well as the same data from any websites owned by parent or child companies owing to the fact that management would likely have similar priorities in these instances. Furthermore, if an advanced method to analyse the legal text of privacy policies and cookie policies were created, this data could also be fed to the central server. The resulting combination could result in a highly intelligent system that is able to rate websites for the risk of the overall data they collect about a user, the companies' attitude to data protection, the users' understanding of the companies' data collection and whether the companies' disclosure accurately and simply conveys their real-world data practices. This paper has created a baseline for controlling and analysing data but there is still much research to do in this field and while there would be some significant technological and social hurdles in the proposed future works, they are potentially very powerful.

References

Baruh, L., Secinti, E. & Cemalcilar, Z. (2017), 'Online privacy concerns and privacy management: A meta-analytical review: Privacy concerns meta-analysis', *Journal of Communication* **67**.

Culnan, M. (2000), 'Protecting privacy online: Is self-regulation working?', *Journal of Public Policy & Marketing - J PUBLIC POLICY MARKETING* **19**, 20–26.

European Parliament and Council of European Union (2016), 'European parliament and council of european union (2016) regulation (eu) 2016/679'. Accessed 28th October 2020.
URL: *https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679*

Isaak, J. & Hanna, M. J. (2018), 'User data privacy: Facebook, cambridge analytica, and privacy protection', *Computer* **51**(8), 56–59.

Libert, T. (2018), An automated approach to auditing disclosure of third-party data collection in website privacy policies, pp. 207–216.

Lin, Y., Wang, C. & Chen, W. (2014), A content privacy-preserving protocol for energy-efficient access to commercial online social networks, *in* '2014 IEEE International Conference on Communications (ICC)', pp. 688–694.

Martin, G., Gupta, H., Wingreen, S. C. & Mills, A. (2016), 'An analysis of personal information privacy concerns using q-methodology', *CoRR* **abs/1606.03547**.
URL: *http://arxiv.org/abs/1606.03547*

Petronio, S. (2002), *Boundaries of privacy: Dialectics of disclosure*, Suny Press.

Pollach, I. (2005), 'A typology of communicative strategies in online privacy policies: Ethics, power and informed consent', *Journal of Business Ethics* **62**, 221–235.

Spiekermann, S. (2012), 'The challenges of privacy by design', *Communications of The ACM - CACM* **55**, 38–40.

State of California (2018), 'California consumer privacy act, 2018 cal. legis. serv. ch. 55 (a.b. 375) (west)'. Accessed 28th October 2020.
URL: *https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375*

Strahilevitz, L. & Kugler, M. (2016), 'Is privacy policy language irrelevant to consumers?', *The Journal of Legal Studies* **45**, S69–S95.

United Nations (1948), 'Universal declaration of human rights'. Accessed 28th October 2020.
URL: *https://www.un.org/en/about-us/universal-declaration-of-human-rights*

Utz, C., Degeling, M., Fahl, S., Schaub, F. & Holz, T. (2019), '(un)informed consent: Studying GDPR consent notices in the field', *CoRR* **abs/1909.02638**.
URL: *http://arxiv.org/abs/1909.02638*

Westin, A. (1970), *Privacy and Freedom*, Bodley Head.
URL: *https://books.google.co.uk/books?id=rapOSAAACAAJ*

Yassine, A. & Shirmohammadi, S. (2009), Measuring users' privacy payoff using intelligent agents, pp. 169 – 174.