Everybody's Looking for SSOmething: A large-scale evaluation on the privacy of OAuth authentication on the web

Yana Dimova
imec-DistriNet, KU Leuven
yana.dimova@kuleuven.be

Tom Van Goethem
Google / imec-DistriNet, KU Leuven
tov@google.com

Wouter Joosen
imec-DistriNet, KU Leuven
wouter.joosen@kuleuven.be

Abstract

The management of many different login credentials can be tricky for the average web user. OAuth eases this process by invoking identity providers (IdPs) as intermediaries, which identify the users and access their data on behalf of the website, without sharing their credentials. However, the information that IdPs share with websites is not always limited to basic data. Our work observes and documents that IdPs make a variety of resources (*scopes*) available to be requested by websites, most of which are not necessary for user identification (e.g., location, interests). By performing a large-scale analysis on OAuth-based login on the web, we show that 18.53% of websites using OAuth request at least one non-minimal scope. Additionally, our findings show that at least part of the requested information is redundant since websites provide alternative login methods that require less information from the user. Moreover, through a manual analysis we observe that revoking access to these scopes seems not to hinder the functionality of the website. Finally, when comparing OAuth-based login with registering a new account, we find that OAuth is often the more privacy-friendly option in terms of the amount of personal data being shared with the website.

Keywords

SSO, OAuth, privacy measurement

1 Introduction

Federated SSO allows users to log in with the same credentials on multiple (unrelated) domains and provides many benefits for both websites and users, such as ease of use, cost-effectiveness and efficiency. One of the most popular standards for federated SSO is OAuth 2.0 [19], which relies on an intermediary referred to as *the identity provider* (IdP) to authenticate the user on behalf of the website. Next to authentication, the OAuth protocol also makes it possible to share other types of user information that the IdP stores; these are grouped together according to IdP-defined *scopes*. Requesting additional user data, i.e., on top of the default information required for authentication, is trivial as websites can simply request more scopes in the OAuth flow. This raises some privacy concerns as companies and organizations might try to access more user data than they actually need to provide a functional service to the user.

In the EU, privacy and data protection legislation protects users from excessive collection of personal data by laying down principles for lawful processing. One of those principles outlined in the GDPR, is the principle of *data minimization* [34, art. 5.1(c)], which mandates that companies should stick to processing the minimum amount of information that is necessary to achieve their goal. Prior work has shown that some users are in fact concerned with the permissions that applications obtain from IdPs [1]. In addition, users may incorrectly estimate the privacy level of using SSO when they are insufficiently informed about the use of scopes requested from them [30].

We primarily focus on analyzing the privacy implications on users depending on how websites deploy their OAuth-based authentication. While IdPs can offer the possibility to access various types of data, ultimately the website (service provider) chooses which user data, i.e., scopes, to ask permission for. To better understand how websites engage with OAuth-based login and evaluate how they consider the privacy implications for their users, we perform a large-scale measurement on 100k websites, of which we find 7.23% to support OAuth-based login. We make the code that we used to automatically detect OAuth available in a public repository [1]. We find that 18.53% sites with OAuth request access to *non-minimal scopes*, i.e., for personal information that goes beyond the minimum amount that can be requested. This raises the question: *Do websites need this additional information to provide the service requested by the user?*

Through a series of experiments we aim to determine whether websites opt for the most privacy-preserving option for authentication and request the default required data. Our results clearly show that a significant share of websites are requesting more data than they actually need. For instance, we find that of the websites that allow users to login with multiple IdPs, 23.53% request strictly more user information from one IdP than from the other. As the same authentication is possible with either IdP, the website is requesting access to non-necessary data in (at least) one instance. In a second experiment we manually intervened in the OAuth authorization flow and reduced the requested scopes to the minimal set. Surprisingly, we found that in 65% of the examined websites, the login process was not disturbed. This means that non-minimal scopes requested by 65% of websites were in fact not essential for the provided functionality. On the bright side, this provides an opportunity for users to reduce the information they share with websites, as they can simply adjust the scope parameter before authorizing the website access to their data.

Furthermore, we explore how the OAuth-based authentication process relates to the typical registration with a username or email address in terms of data that the user needs to share. Surprisingly,

Proceedings on Privacy Enhancing Technologies 2023(4), 452–467

https://doi.org/10.56553/popets-2023-0119

[1] https://anonymous.4open.science/r/oauth_project_code-2003

despite the ease of requesting access to additional data from the IdP, we find that in most cases OAuth-based login is the more privacy-friendly option. Finally, we analyze how websites handle personal data in relation to other privacy aspects, such as cookie banners and online tracking. We find that there is no clear connection between the general privacy consciousness of website owners and the scopes they request access to through OAuth.

In general, our findings strongly indicate that a significant portion of websites are requesting data from users in an unjustified way and therefore raise privacy concerns. This is especially a problem given that only a small subset of IdPs provide a user-friendly authorization interface where users can select which scopes they want to grant access to, and some of them even employ manipulative designs to nudge the user into making less privacy-friendly choices. To solve these issues, we propose a number of improvements and guidelines for both IdPs and websites in order to improve control and transparency of information that users end up sharing with websites upon using OAuth-based authentication.

We summarize our main contributions as follows:

- We develop a fully automated method to detect OAuth-based login with support for 179 IdPs and perform a large-scale measurement on 100k websites.
- We report on the use of OAuth scopes by websites and find that 18.53% of websites require users to share information from non-minimal scopes with the website. Through a number of experiments we show that part of this information is oftentimes nonessential to the website.
- We compare the privacy implications of OAuth with the standard authentication procedure and assess whether the general privacy considerations of website owners affect how much data they collect from users.
- We suggest various improvements for both IdPs and websites that contribute towards a more privacy-friendly use of OAuth login for users.

2 The OAuth protocol

OAuth [22] is an authorization standard which includes three parties: a client (service provider), a resource owner (user) and an identity provider. Typically, OAuth is used in a scenario where the service provider requests access to resources that belong to the resource owner and are hosted by the identity provider (IdP). This is useful when a user wants to share their resources with a third-party application without putting their credentials at risk e.g., using Google to log into another website instead of having to create a new account with separate credentials. In such cases, the IdP verifies the identity of the user and accesses their resources on behalf of the website (upon permission from the user). The user benefits from this scenario in the sense that they will not need to go through a registration process on the website and do not need to keep track of an additional username and password.

The first step of implementing OAuth is for the application to register with the IdP, either by filling in a form or registering on the developer portal of the IdP. At that point, the authorization server assigns a unique identifier to the client that will be used when issuing authorization requests.

The most recent version, OAuth 2.0, includes support for a number of ways to obtain access tokens (tokens that are used to retrieve resources), called *flows*, with the goal of supporting multiple application types and therefore making OAuth more usable to developers. Generally, the authorization code grant and implicit grant are the most suitable for web applications.

Authorization code grant and implicit grant In the authorization code grant, the authorization server acts as an intermediary between the website and the resource owner. The credentials of the user are never shared with the client. The IdP's authorization server is a trusted party which authenticates the end-user on behalf of the web application.

The authorization code flow consists of three interactions with different entities:

- **Authorization** First, the website requests authorization from the end-user. This is typically by issuing an authorization request which redirects the user to a domain of the IdP, where the user is shown details about the third-party application and the set of requested permissions, called *scopes*. These scopes are IdP-specific and can be specified in the URL parameters of the authorization request, which makes it easy to passively extract them from all network traffic on a webpage. If the user has authorized the request, the IdP responds with an authorization token. Otherwise, the login flow is interrupted and the user is redirected back to the visited website.
- **Obtaining an access token** In the second step, the client exchanges the authorization token for an access token by issuing a request to the authorization server (typically belonging to the IdP).
- **Accessing resources** Finally, the client uses the access token to access protected resources of the end-user from the resource server of the IdP.

The implicit grant is a simplified version of the authorization code grant, where the first and second step are merged together, i.e., authorization is still requested from the user, but instead of issuing an authorization token, the authorization server responds directly with an access token.

OpenID Connect OpenID Connect [15] runs on top of the OAuth 2.0 framework and provides an additional authentication layer, allowing clients to obtain information about the identity of the end-user. It provides support for different types of clients. If an IdP supports OpenID, the client can include the scope *openid* in the URL parameters of the authorization request. The authorization server then returns an *ID token*, which contains information about the identity of the user. OAuth and OpenID have the same workflow, the difference lies in the fact that OpenID provides an id token to the web application instead of an access token. Some IdPs support both the use of OpenID Connect for identification of the user and the use of OAuth to access user resources, and provide websites the ability to combine both.

3 Methodology

In this section, we describe our crawler setup and the method used to automatically detect OAuth buttons.

3.1 Setup of experiment

To run our crawl, we used the latest Chromium version which we instrument headlessly via Chrome Devtools Protocol. We used 10 virtual machines running in parallel, each with 4 CPUs and 8GB of RAM. We used our local university network, which is based in the EU. Our crawl was run in October 2021 and took in total 20 days to complete. We used the top 100k websites from the latest list (July 2021) of the Chrome User Experience Report (CrUX) [6], which includes frequently visited websites by Chrome users. On 14104 websites (14.1%), the crawler failed mainly due to pages not being correctly loaded or websites being unreachable.

3.2 Detection of SSO buttons

To develop a fully automated method for detecting SSO buttons, we base ourselves on prior research. Just like several frameworks [9, 16, 44] and SSO measurement studies [30], we use a keyword-based approach, where we search for common terms related to SSO in all HTML elements of the page (a full list of the keywords we used can be found in Appendix A). Throughout this section, whenever we discuss login and OAuth buttons, we refer to their visual representation of a button on a website, and not to their underlying HTML implementations (which can be other than a HTML-button). Since SSO buttons are commonly found on login pages, or pages where the user can create a new account, we search for login and account registration buttons, prior to looking for SSO buttons. We assume that websites which allow users to create an account, will make that option prominent by including it on their homepage. Our approach consists of three steps: (1) visiting the homepage and searching for a login/registration page, (2) visiting the login/registration pages and searching for SSO buttons and (3) simulating a click on candidate SSO buttons to initiate the authorization process. We go into more detail for each of the three steps. The full code that we used to detect SSO buttons can be found in our public repository.

Detecting login and registration buttons In order to detect login and registration pages, we search for elements on the homepage that could lead us there and simulate a user click on them. To select candidate elements, we search all HTML elements of the page for phrases that are typically related to logging in or registering an account. We define both a set of primary login-related words and a set of helper keywords, which are often used in the same context as the login/registration terms (e.g., "new", "member"). Since we did not restrict our analysis to websites in English, we used the python library googletrans [18], based on Google Translate's API to translate all keywords in 28 languages. We manually verified whether the translation was correct for each language spoken by the authors to the best extent possible. While we acknowledge that some semantic information might get lost during the translation process, our method allows us to detect at least some basic keywords on websites that are not written in English. We assign a score to each element based on the occurrence of login-related keywords and helper keywords in the *outerHTML* of the element. We excluded elements which contain keywords related to cookie banners (e.g., "cookie", "accept") given that we did not want to include cookie banner elements. A similar approach is used in Jonker et al.'s Shepherd framework, which automates website login [23].

We consider four common URL paths that are typically used for login/registration pages (e.g., */login*). Next to those, we consider the six HTML-elements with the highest score (defined by the number of occurrences of keywords) to be potential login buttons, so that we get a total of 10 potential links. During a pilot crawl, we found that login and SSO buttons are not always implemented with `a`-elements. Therefore, we also considered other HTML-elements such as `buttons`, `divs` and `span`-elements with click-event listeners. In the next step, we either visit the link of the candidate login buttons, or simulate a user click on them, depending on the type of element.

Detecting SSO buttons on login pages Once we visit the potential login pages from the previous step, we again apply a keyword-based approach in order to find potential SSO buttons. For each login/registration link from the previous step, we search for HTML-elements with common OAuth phrases in combination with the name of one of the IdPs that we consider (e.g., *Login with LinkedIn*). Next, we assign a score to each element based on the occurrence of common SSO keywords. We exclude elements which contain social plugin-related words (e.g., "Share" or "Like"). In total, we consider the 10 HTML-elements with the highest score (again based on the number of occurrences of keywords) as potential SSO buttons.

Clicking on SSO candidate buttons In the final step, we click on each candidate SSO button and we monitor the web traffic to see whether any requests are issued to the OAuth authorization endpoint. All IdPs specify in their documentation that scopes are to be communicated via URL parameters of the authorization request, and do not provide a way for service providers to change the default scope. The only exception to this is Twitter, where the scope can be specified in the application on Twitter's developer platform. In this case we analyzed the DOM of the page to determine the requested scope.

3.3 Extracting OAuth scopes from IdPs

The `scope` parameter of the authorization request is used to determine the set of resources that the service provider can request from the user. These resources are IdP-specific i.e., the IdP chooses which resources a service provider can request via SSO. This information is typically located in the documentation of the specific IdP. In order to compile a list of the available scopes, we visited the website or developer portal for each IdP, and registered an account when required. We searched in the documentation for the authorization endpoint and which scopes are made available by the IdP, as well as other characteristics of the IdP, which we discuss in more depth in Section 4. However, finding information about the scopes of each IdP proved to be challenging given that some IdPs do not provide a full list of the possible scope values in the documentation i.e., they only provide example scopes or allow for custom permissions upon request from the service provider. Additionally, for 4 IdPs, we did not find any information regarding the use of scopes in the documentation. In such cases, we tried to register a test application on the platform of the IdP in order to find a list of available scopes e.g., for Twitter. Whenever it was not possible to register an application

with the IdP, we considered the set of observed scopes during our crawl for each of the IdPs as the set of available scopes.

3.4 Categorization of scopes

Since most OAuth scopes are specific to the IdP, we categorized each scope into seven categories in order to compare them among different platforms : 1) *minimal scope* i.e., the scope which shares the least possible amount of data with the service provider; typically this data includes attributes such as email address or basic profile information, (2) *personal* i.e., data about the user not included in the minimal scope, (3) *content* read i.e., read requests to provider-specific content such as access to user photos, (4) *content write* i.e., authorization to edit provider-specific content such as post on behalf of the user, (5) *behavioral* i.e., data that can be used to provide personalized content to the user (e.g., location, interests), (6) *sensitive* i.e., data relating to religion, political opinions, sexual orientation, health and financial data and (7) *functional* i.e., related to the management of access and authorization tokens, for instance Adobe provides the scope "offline_access" which indicates that a new refresh token should be generated.
The *minimal* scope is to be interpreted as the least amount of data that can be requested from a certain IDP, not as the strictest possible set of data that can be obtained via OAuth. We note that the *minimal* scope may differ among different IDPs. For instance, Twitter's most minimal scope includes not only the user's profile information, but also access to tweets and the timeline of the user. We elaborate further on the differences among IDPs in section 4.1. For the category of sensitive scopes, we base ourselves on the definition of special categories of personal data, as defined in the GDPR [34, art. 9]. This categorization method is based on previous work which compares scope attributes among different identity providers [30]. A full overview of the list of scopes available per IdP and their categories can be found in our public repository.

3.5 Page coverage

A limitation to our OAuth detection method is that we did not interact with websites after logging in, and thus might miss requests for additional scopes required for specific functionality on the site. We randomly selected a set of 100 websites that support both OAuth and manual registration. Then, we visited each website and spent 10 minutes interacting with various functions of the website. We started by visiting each link on the homepage, and moved on to following multiple chains of links to navigate further within the website. The pages we ended up visiting depend on the type of website, but in general we prioritized interaction with features that would reasonably request additional user data or specific *content read/write* scopes via the IDP (e.g., we tried to complete a purchase on webshops or tried to adjust and complement the information which we had already provided during login). While interacting with the website, we searched for signs of a new OAuth flow being automatically initiated and requesting additional scopes or an OAuth-related prompt or button popping up to ask for additional data from the user. However, we did not encounter this on any website.

3.6 Ethical considerations

Part of our experiments consist of passive observations of websites, and 2 of them required manual interaction with the website. For all experiments, we made sure not to impact real-world websites in a negative way. We only created one account for each IdP in order to log in with them on websites, and we created one account for each website visited during the manual experiments. Each time, we used the author's credentials and personal data to create the account.

4 Results

In this section, we discuss our findings on the prevalence of OAuth. We give an overview of the most popular Identity Providers and the use of scopes for OAuth-based authentication.

4.1 OAuth Identity Providers

During our crawl we searched for the use of 179 IdPs in total. We obtained this list from The ProgrammableWeb [41] - an independent website which collects data about the API usage of websites. We find actual implementations of OAuth-based login for only 33 IdPs (out of the 179) on at least one website from our dataset. In Table 1, we show the 15 most widely used IdPs and the number of websites that use them. Facebook and Google are at the top with a prevalence on respectively 5.52% and 3.96% of websites, while Twitter and Apple follow with a prevalence of respectively 0.76% and 0.7%. We note that these numbers are a lower bound for the actual number of websites using OAuth, since we miss a number of them during our multi-step detection process, as described in Section 4.5.
Available scopes The number and type of scopes that IdPs offer to developers differs. 9 of them only offer *minimal scopes*, the other 24 offer mostly a mix of *content write*, *minimal*, *content read* and *personal* scopes. Google is the only IdP that provides *sensitive* scopes due to the Google Health API. We do note that we are using a restrictive definition of "sensitive data" and that user photos or likes might also contain sensitive information about the user. A number of IdPs offer scopes related to analytics APIs, on average 1.22% of all scopes belong to the *behavioral* category.
Minimal scopes We define *minimal scopes* as the minimal set of data that the user needs to share with the service provider. Most IdPs provide a *user profile* or *email* scope that falls under this category. For some IdPs the *profile* or *user* scope includes a set of data that encompasses more information than with other IdPs. For instance, Google's *profile* scope includes the gender of the user, which we would categorize as being *personal* data.
We examined the documentation of the IdPs and found information about the content of the profile-summarizing scopes for 15 of them. For 5 out of these 15, at least one of the minimal scopes included information that we would categorize as non-minimal. For instance, Mailru includes the gender, location and birthday of the user in the *user info* scope. Github includes information about the repositories and gists of the user, the location, bio description and company and even the Twitter username of the user.
Authorization screen As part of the OAuth flow, websites need to obtain authorization from the user to share resources. For this reason, an *authorization screen* is shown to the user, which displays details about the client and the list of scopes for which permission is requested. Some providers allow the user to only give permission

for a subset of the chosen scopes, which is presented on the *authorization screen*. We find that only 5 IdPs from our dataset provide this option to users, among which Facebook, Google and GitHub. With Facebook, the user can edit the scopes only after performing an additional click, resulting in a window where all listed scopes are preselected. In 2018, Google announced that they are planning on implementing a more fine-grained choice for users [8] and we found several websites using this interface, in which the scopes are immediately listed and the choices are not preselected. While GitHub's consent screen does not present users with a choice regarding requested scopes, it is explicitly mentioned in the documentation that users might alter the *scope* parameter in the authorization URL to change the scopes.

Verification process 8 out of the 33 IdPs require the developer to go though a verification process if they want to use certain scopes. This is typically part of the registration process for applications, where the developer is required to fill in details about the OAuth application and configure parameters such as the callback URL and the scope. For instance, if an application owner wants to use Facebook Login with non-minimal scopes, they need to provide a detailed description about why they need the scopes, how they will be used and a screen recording of use scenarios. Facebook mentions specifically in their documentation that no review is required for an application that simply provides authentication. Along with Facebook, also Google, Instagram, YouTube and Line require an app review for the use of non-minimal scopes, while Discord only requires approval for certain scopes, and PayPal conducts a review regardless of the scope used. A number of providers mention that additional scopes not listed in the documentation can be accessed if necessary by contacting the IdP. This was the case for Patreon and Line. Only 10 IdPs clearly document that applications should only request scopes that are necessary for their correct functioning.

4.2 SSO on the web

We report on the prevalence of OAuth buttons on 6,211 (7.23%) websites. We found 10,304 OAuth buttons in total, with an average of 1.66 buttons per website and a maximum of 6 buttons, which we encountered on 3 websites. To assess the popularity of websites that use OAuth, we use the popularity ranking that has been introduced in the Chrome UX Report in 2021 [20]. We find that OAuth login is more prominent on popular websites, namely 10.4% of the top 1,000 most popular websites. The implementation rate is significantly lower on the top 10k most popular websites and the top 100k websites, where the prevalence is respectively 6.80% and 5.43%.

4.3 Scope categories

On websites that use SSO, we find 1,264 (12.27%) OAuth buttons that include non-minimal scopes. We found such buttons on 1,151 websites in our dataset (18.53%). About 380 (33.01%) websites with non-minimal scopes are using Facebook Login with a non-minimal scope, and 491 (42.66%) of them are using Twitter Login with a non-minimal scope. While Facebook Login is the most popular IdP in our dataset, OAuth with Twitter is less common as can be seen in Table 1, while still accounting for more than 40% of the websites that use a non-minimal OAuth scope. This is due to the fact that Twitter only allows developers to choose between 3 scopes in the Twitter Developer Platform: *Read*, *Read and Write* and *Read, Write and Access Direct Messages*. Since read is the most restrictive permission of the three, we consider it as *minimal scope*, while the other two belong to the *content write* category. On 75.08% of websites that use OAuth with Twitter, one of these *content write* scopes is requested. For Facebook, the percentage of websites that request a non-minimal scope is much lower, namely 8.01%.

The number of websites that use non-minimal scopes for OAuth with Google is almost three times lower than for Facebook (3.06% of websites using login with Google request a non-minimal scope, compared to 8.01% of websites with Facebook Login). One reason for this could be the fact that Facebook Login allows users to deselect part of the permissions. Therefore, service providers might be more likely to request more scopes assuming that users will only retain the ones they are comfortable with sharing.

Figure 1 shows the distribution of the use of non-minimal scopes on websites. We left out *sensitive* and *functional* scopes since we did not find any websites requesting *sensitive* scopes, and found that *functional* scopes are requested on only 10 (0.16%) websites. The figure shows the percentage of websites requesting *personal*, *behavioral*, *content read* and *content write* scopes either on their own, or in combination with another scope category. In general, 6.86% of OAuth buttons request a *content write* scope and 4.51% request a *content read* scope. We find that it is typical that the *content read* and *content write* scopes are requested simultaneously. More noteworthy, *personal* data scopes and *behavioral* scopes are both requested on 144 (2.32%) websites, supposedly in order to obtain additional user information and provide a more accurate personalized experience. We did not find any websites requesting the most privacy-invasive category of scopes, namely *sensitive* scopes. We do note however, that our definition of sensitive is strict in the sense that we only consider the data as such, not its potential privacy implications. For instance, Facebook likes and interests can reveal sensitive information about the user, but we consider them to be *behavioral* data.

4.4 Website categorization

Some non-minimal OAuth scopes might be requested for legitimate use-cases such as requesting the age of the user for adult content, while other websites will request OAuth scopes just for the sake of collecting more data or providing non-essential features such as personalized advertising. In order to get a deeper understanding of the motivation behind this data collection, we consider the type of service that the website provides. For this, we use McAfee's domain categorization API [28] for all websites in our dataset.

Figure 2 shows the 10 most prominent website categories for both websites using OAuth with minimal and non-minimal scopes.

Our results show that a number of websites request non-minimal scopes (18.53%). Out of those websites, some might use the additional user information for a legitimate purpose. For instance, users might share more personal data with websites belonging to the "Dating" category [7, 14, 17]. On the one hand, some non-minimal data might be required from the user in order to provide the dating service e.g., the user's age for verification purposes or the gender and even sensitive data such as sexual orientation to match with other users on the platform. On the other hand, users themselves have more incentive to provide details on their profile, given that

Table 1: Top 15 most prevalent IdPs in our dataset

Provider	Number of scopes	Number of websites	Percentage with non-minimal scope	Scopes customizable to user	Verification process by IdP	Sensitive scopes are available	IdP has guidelines on use scopes
Facebook	16	4743 (5.52%)	8.01%	●	●	○	●
Google	269	3400 (3.96%)	3.06%	●†	●	●	●
Twitter	3	654 (0.76%)	75.08%	○	○	○	●
Apple	3	604 (0.70%)	0.00%	○	○	○	○
Line	11	375 (0.44%)	46.93%	●	●‡	○	○
LinkedIn	4	103 (0.12%)	22.33%	○	○	○	●
Yandex	6	93 (0.11%)	5.38%	○	○	○	○
Mailru	7	75 (0.09%)	2.67%	○	●	○	○
GitHub	33	50 (0.06%)	14.0%	●*	○	○	○
Odnoklassniki	8	46 (0.05%)	17.39%	○	●	○	○
Microsoft	23	35 (0.04%)	82.86%	○	○	○	●
Live	23	29 (0.03%)	72.41%	○	○	○	●
Discord	24	27 (0.03%)	33.33%	○	●	○	○
Amazon	4	17 (0.02%)	5.88%	○	○	○	○
Patreon	6	8 (0.01%)	0%	○	●‡	○	○

†: in development stage
*: the documentation specifies that users can change the permissions in the URL, but there is no user-friendly options visible to the user
‡: verification is partially or wholly for requesting additional scopes that are not listed in the documentation

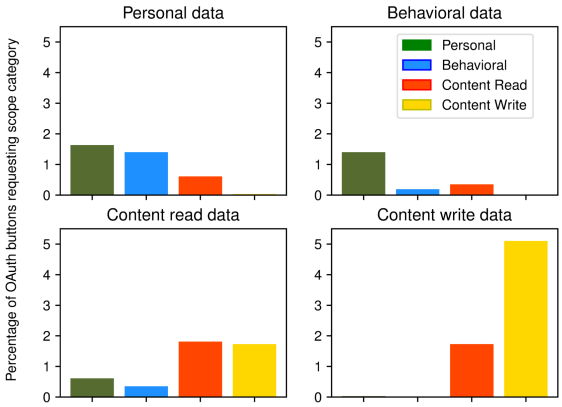


Figure 1: Distribution of categories of scopes used by websites. Each subfigure depicts a scope category and how often it is requested by websites, either on its own or in combination with other scope categories.

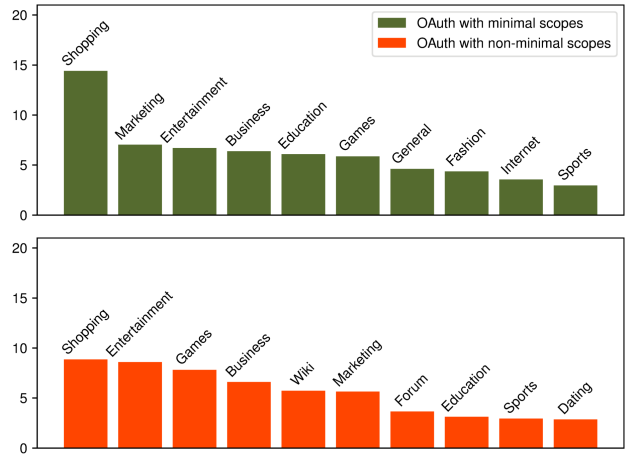


Figure 2: Top 10 website categories for websites with OAuth with minimal scopes and OAuth with non-minimal scopes

the more information they provide, the higher the chances are of finding a suitable match.

However, websites that request non-minimal scopes belong mostly to the same categories as websites that request only minimal scopes. It could be presumed that those websites have similar data processing activities and therefore would request similar OAuth scopes from the user. Given that this is not the case, our findings indicate that websites with non-minimal scopes request data that is not strictly essential for the proper functioning of the website and thus that websites might request more data than they actually need.

4.5 Evaluation of OAuth detection

We performed a manual analysis on 100 websites in order to evaluate our OAuth login detection method. The websites we examined were selected randomly from all reachable websites where our algorithm did not find any OAuth implementation. We visited each website and looked for OAuth buttons on both login and account registration pages. Whenever we found OAuth buttons, we clicked on them in order to initiate the OAuth flow and check whether it was implemented correctly. This was done because our automated detection method only considers actual requests to the authorization endpoint of an IdP as a valid OAuth login. We found that out of the 100 websites, our algorithm failed to detect OAuth buttons on 18 of them. The main cause for failing to detect an OAuth-based login on a website is twofold. On the one hand, our crawler failed

to detect the correct login or registration page on which the button was located and hence was unable to detect its presence. On the other hand, our method relies on the use of a set of expected keywords for the technical implementation of OAuth buttons, and any button that uses different keywords would be missed. Prior studies which have conducted a similar analysis find similar results - Ghasemisharif et al. [16] find 6.3% websites using SSO for the top 1M most popular sites and 10.8% for the top 10k sites; Zhou et al. [44] report a prevalence of Facebook Login on 9.3% of top 20k websites. We note that the reason for missing OAuth buttons on websites is unrelated to their stance on privacy. Hence, the set of websites that we evaluated can be considered a random selection of all websites that support OAuth-based login.

5 Multiple IdPs on the same website

In this experiment, we examine websites with multiple IdPs and compare the amount of information required from the user upon logging in in order to ascertain whether certain websites are requesting more scopes via one specific IdP than via another. We consider these IdPs to be used in a more privacy-intrusive way, in the sense that the user is required to share information which is not necessary, since the website provides an alternative login method which requires less permissions from the user.

Types of data We find that almost half of the websites using OAuth (3000 out of 6211 or 48.30%), include more than one identity provider. On 706 (23.53%) out of the 3000 websites, multiple IdPs are included where the user is required to share less permissions with one than the other. For this analysis, we exclude IdPs which only allow for minimal scopes to be requested.

The type of data that is requested by the more privacy-intrusive login in such cases belongs to the *content write* scope on 423 (59.91%) websites and the *content read* scope on 172 (24.36%) websites. The majority of those scopes are related to content and functionality specific to the IdP and are therefore less likely to be accessible via a different IdP. A large number of websites also requests personal and behavioral data. 200 (28.33%) websites request a *personal* scope and 107 (15.16%) websites request a *behavioral* scope. Some *personal* scopes, such as the user's birthday, are accessible via multiple identity providers. However, this is not the case for all *personal* and *behavioral* scopes, since some of them are exclusive to one IdP. We zoom in on the differences of available scopes between the two most popular IdPs (Facebook and Google) later in this section.

The impact of scope granularity Logging in with Twitter is more intrusive than logging with another IdP on 404 (57.22%) websites. This result is not surprising, since the scopes available via Twitter are limited in number (only three available scopes) and are all fairly privacy-intrusive i.e., each scope contains a large set of user information. For instance, the most limited scope (and the one we consider as being the *minimal scope*) allows the service provider to read tweets, lists and collections, profile information, account settings and accounts which the user follows, mutes and blocks. On the other hand, the second most intrusive scope facilitates both read and write access to the same resources as the minimal scope. The fact that none of these scopes are minimal (i.e., they do not include the bare minimum of information necessary to identify the user) gives no incentive to service providers to actually limit themselves to requesting the most restrictive scope out of the three, resulting in the majority of websites (75.08%) which use Twitter login, requesting non-minimal scopes.

The impact of provider-specific scopes On 207 (29.32%) websites, logging in with Facebook is more intrusive compared to a different IdP. That number is much lower for Google, namely on 52 (7.37%) websites. Therefore, we focus on the difference between scopes used when Google and Facebook both provide an SSO option.

In total, 487 (7.84%) websites in our dataset use both Facebook and Google SSO. On 263 (54%) of them, the same scope category is requested by both Google and Facebook, being almost in all cases (99.24%) the *minimal* scope. This leaves us with 224 (46%) websites where the requested scope categories differ. Logging in with Google results in a larger set of scopes being requested than logging in with Facebook on only 45 (20.09%) websites, while logging in with Facebook is more intrusive than logging in with Google on 175 (78.13%) websites.

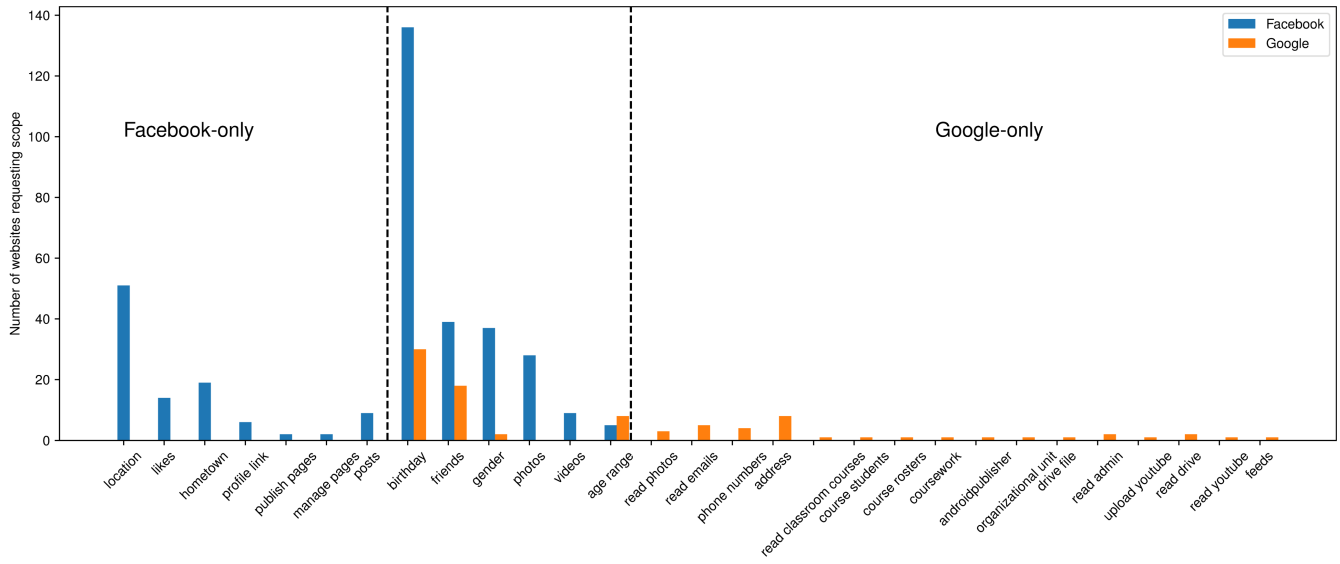
We go more in depth as to the reason why these is a difference. One hypotheses is that the scopes requested when using Facebook are not available via Google Login. In Figure 3, we show the scopes requested on all websites where the user can log in with both Google and Facebook. On the left side of the graph, we show scopes that are only available with Facebook Login and are not supported by Google. The middle part shows scopes that are available for both Google SSO and Facebook SSO, and the third part shows scopes that are only available for Google. While some of the frequently requested scopes are only part of Facebook's API (e.g., user location), the difference between the two platforms for the number of websites which request scopes that are both available for Google and Facebook is notable. The most-widely requested scope for Facebook Login is the birthday of the user, requested on 136 websites, is only requested on 30 websites when logging in with Google, indicating that this information is redundant. On the other hand, we note that not all of the scopes which are both available for Google and Facebook are comparable. For instance, a user might be more comfortable privacy-wise to share photos on Facebook, which might be public anyway, than to share an album of photos via Google. The same holds for sharing friends or contacts and videos. However, websites using Facebook Login request more often scopes that are available for both IdPs, as well as scopes that are specific to the IdP concerning personal data of the user such as the location of the user. Therefore, we conclude that websites are more privacy-intrusive in their use of Facebook SSO.

6 Adjusting scopes manually

In this experiment, we examine the effect of adjusting the requested scopes to a set of less privacy-intrusive scopes, both for IdPs which provide a user-friendly interface for changing them (e.g., Facebook) and for IdPs which do not provide this option. For the latter we resort to altering the authorization request parameters.

As described in the previous section, some identity providers allow users to choose which subset of scopes they actually want to share with the service provider. However, among the ones that do, some nudge users into completing the authorization process without adjusting the scopes.

Figure 3: OAuth scopes on websites that support both Google and Facebook login



Github's documentation specifies that users do have a choice about the requested scopes which they can communicate to the service provider by altering the scope parameter of the authorization request URL. Technically, this choice is applicable for every IdP (except for Twitter since it does not display the authorization URL to the user). This raises the question as to how websites actually handle the user adjusting the requested scopes.

6.1 Experiment

In our second experiment, we study the behavior of websites when the requested scopes are rejected by the user. Particularly, we look for signs that indicate whether the website is actually using the data or whether the website breaks when that data is missing.

We selected 100 websites randomly from all websites which request non-minimal scopes for at least one IdP (out of 1264 websites). For each website, we initiated the OAuth process by clicking on the OAuth button and then either 1) used the interface provided by the IdP on the authorization screen to change the scopes (e.g., for Facebook) or 2) changed the scope parameter of the authorization request. In each case, we changed the set of scopes requested by the service provider to a set of minimal scopes, and tried to complete the login process. We observed how websites reacted to the missing scopes by checking whether the login process was successful and whether any errors were displayed on the website.

6.2 Results

On 65 out of the 100 websites, logging in with minimal scopes instead of a non-minimal ones did not affect the login process or the user experience. On another 24 websites, the same information which we removed manually from the scope was requested by the website after the user had completed the login process via OAuth. For those 24 websites, we repeated the experiment but this time, we authorized the website to access all requested scopes, without adjusting them. Interestingly enough, for 8 out of the 24 websites, the website still requested the same information from the user after login as when no minimal scopes were granted. We observed this by checking whether any text fields for information requested from the user after login were already filled in or not. In this case, the text fields for the same data which is requested via OAuth remained empty after login. We conclude that these website do not use the data from non-minimal scopes which they specifically request from users given that whether the requested permissions are minimal or not, has no effect. For the other 16 websites, we saw that the information requested via OAuth is effectively used and the text fields for this data were correctly filled with the granted scopes after login. Finally, the login process was disturbed by the scope change on only 11 websites: on 9 of them logging in was not possible and on the other 2 the login succeeded but an error message was displayed to the user.

Our results indicate that most websites do not consider the requested non-minimal scopes to be vital to their operation. We do note that we only visited the page shown to the user after login, and did not test other functionality of the website. Additionally, the fact that the scope change had no impact on the majority of visited websites, provides the opportunity to assist privacy-conscious users with an automated method to change the scope parameter to exclusively minimal scopes (e.g., by implementing a browser extension) without causing a large number of websites to break.

While performing this experiment we noticed that 28.15% of websites with Facebook Login that we examined were requesting scopes to which they presumably had no access. In such cases, only the minimal scopes ended up being requested from the user. We suspect that these websites did not successfully complete the Facebook App Review process. However, we were not able to confirm this since Facebook Login's documentation does not specify how the application should act when the application review process has not been started yet or is still ongoing.

7 Alternative authentication methods

Most websites that provide SSO options also provide the option for the user to register a new account through a form, using their email address, username or both. We set out to explore whether there are any inconsistencies between data requested via SSO and user data required to be filled in during the manual registration process. If this is the case, it would suggest that the additional information requested during one of the authentication methods is actually non-essential for the service provider.

7.1 Experiment

In this experiment, we compare login via an IDP with regular user registration on the same website, in terms of the amount and type of information requested. We manually created accounts on 100 websites where OAuth login requests exclusively minimal scopes from the user and 100 websites where at least one IdP requests non-minimal scopes. We selected these websites randomly from our dataset of websites using OAuth, and manually verified that creating a new account was possible and that both the OAuth login and account registration process were functional (e.g., we discarded websites on which we were not able to complete the registration process). We ended up with 200 websites in total where we 1) logged in with one IdP and 2) registered a new account with the credentials and personal information of the authors.

For each website we examined which scope is requested when logging in with the IdP and which information the user is requested or required to fill in to create an account. We also noted whether any additional information was requested from the user after logging in or creating an account on the website.

Some scope categories such as *content read* and *functionality* are unlikely (or even impossible) to be requested from the user upon registration. Therefore, we focus only on data relating to the *personal*, *behavioral* and *sensitive* scope categories.

7.2 Results

Account registration vs. SSO We compare the amount of information requested from the user when logging in with SSO versus creating a new account on the same website. We evaluate this difference for websites requesting both minimal and non-minimal scopes.

Figure 4 shows the percentage of websites that request *personal*, *sensitive* or *behavioral* data from the user when registering a new account with respect to when logging in with OAuth. We make a distinction between 4 types of websites regarding the amount of information requested: 1) **strictly more** i.e., registering an account requests access to the same information as when logging in with OAuth and additional information on top of that, 2) **strictly less** i.e., registering an account requires the user to provide only a subset of the information requested via OAuth login, 3) **equal** i.e., registration and OAuth login request the same information and 4) **other** i.e., the sets of data requested during the two authentication methods differ and can therefore not be compared.

We see that for both websites with OAuth with minimal scopes and websites with OAuth with non-minimal scopes, registering an account requires the user to provide more personal, sensitive or behavioral information than logging in via OAuth. This is the case for 14 websites with OAuth with non-minimal scopes and 20 websites with OAuth with minimal scopes. Furthermore, we find that for 12 websites with non-minimal scopes and two websites with minimal scopes, strictly less information is requested for registration than for OAuth login. The latter is due to two websites that request minimal scopes during OAuth, but subsequently request additional personal, sensitive or behavioral information from the user after the OAuth process is completed.

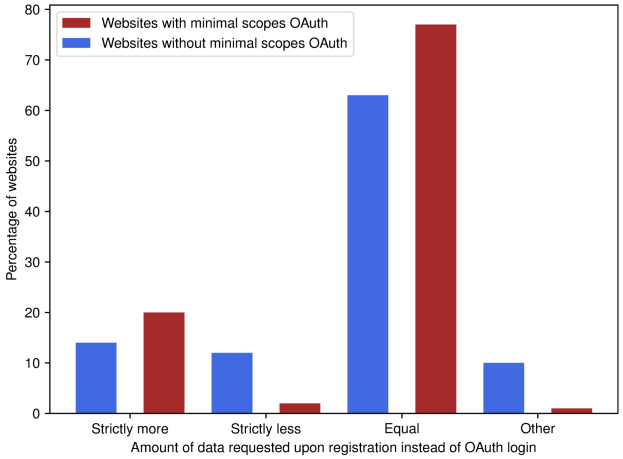


Figure 4: The amount of personal, sensitive and behavioral information requested when comparing registration on a website with OAuth login

Most websites request an equal amount of personal, behavioral and sensitive data through registration and OAuth login. This is more often the case for websites which request minimal scopes through OAuth: 63 websites with OAuth with non-minimal scopes and 77 websites with minimal scopes OAuth request equal amounts of data. Additionally, only one website with minimal scopes OAuth requests differing sets of scopes for both authentication methods, while the same occurs on ten websites with OAuth with non-minimal scopes. These results show that in some cases, logging in with OAuth requires users to share less information with the websites than when completing a registration process.

Dating websites While performing the login and account registration experiments, we noticed that one specific type of website was requesting excessive amounts of personal data from users when compared with other types of websites, namely the category of dating websites. In total, we encountered 10 dating websites during our analysis. On all of them, the amount of information requested from the user when logging in or registering a new account was much higher than for other websites. The requested data ranged from personal interests and habits of the user such as sexual orientation and alcohol consumption, to details about the user's appearance such as eye color and breast size. On average we found dating websites to be requesting six types of *personal*, three types of *sensitive* and one type of *behavioral* information. This number is the same for both login with OAuth and registration, because for 7 out of the 10 websites, the exact same information requested when creating an account, was required to complete after logging in with OAuth.

On the other three websites, logging in with OAuth resulted in the user having to share less data than registering an account. These results are not surprising since users who choose to register on dating websites want to be matched with other people who have similar preferences in terms of sexuality and sometimes specific appearance characteristics. Some of the data requested by these websites might be essential and necessary for them to provide their service. This example illustrates how whether data is necessary is dependent on the context of the application and how website categorization can help us understand this context and necessity.

8 Other privacy aspects

In this section we focus on evaluating how websites handle user data in general, with respect to other privacy aspects, in order to get a deeper understanding as to why websites request redundant information. In the previous sections we show that 18.53% of websites request non-minimal scopes via OAuth, and that most of this data is not essential, since some websites provide an alternative login method that requires less permissions, or the website seems to work without breaking when the data from non-minimal scopes is missing.

We evaluate three features that give an indication about how inclined websites are to respect user privacy and comply with data minimization principles. First, we consider the presence of cookie consent banners on websites, and if present, we assess specific characteristics of the banners such as the information displayed on them and whether any manipulative designs (interface choices that manipulate users into selecting less privacy-friendly options than intended) are visible. Second, we evaluate the occurrence of third-party tracking by means of cookies on websites. Finally, we examine whether websites nudge users into accepting marketing communication emails.

Our crawl is based in the EU and we visited each website with an IP-address from the EU during our experiments. Therefore, all websites must abide by European privacy and data protection laws, such as the GDPR [34][art. 3(a)]. We consider all websites which did not explicitly block EU-users from using them, to fall within the scope of this analysis.

8.1 Cookie banners

European privacy laws introduce a number of rights for data subjects and enforce a privacy-by-default approach for every company which processes personal data of EU citizens. The GDPR and the ePrivacy directive require most processing of personal data which is not essential to be consent-based. Additionally, conditions for valid consent are strict e.g., the user must perform a clear and affirmative action [34, rec. 32], prior to the data processing. Typically, a request for consent is presented to the user in the form of a cookie banner. Some cookie banners employ *manipulative designs*, methods which influence the behavior of users and potentially result in the user making an unintended choice. Nouwens et. al.[32], study the implementation of such manipulative designs on cookie banners of 5 widely-used CMPs, and find that only 11.8% of them, meet the requirements for obtaining valid consent. More recently, the European Data Protection Board has published a comprehensive list of manipulative designs and guidelines for publishers on how to avoid them [3]. In this experiment, we study the prevalence and characteristics of cookie banners on websites that use OAuth.

Experiment In order to evaluate cookie consent banners, we crawl all websites from our dataset on which we found an OAuth button and take a screenshot of the homepage after all elements on the page are loaded.

After the crawl, we examined each screenshot manually in order to determine if the homepage of the website displays a cookie banner to visitors and if so, whether the banner includes any features that could have a negative effect on the user's privacy preferences. For the latter, we consider

- **Visual elements**: First we note whether the cookie banner is blocking the website. This is the case when at least 40% of the screen is taken up by the banner or when the background is darkened. Second, we inspect whether the cookie banner exhibits any manipulative designs.
- **Consent**: We analyze the mechanisms provided to the user to express their consent for cookie preferences such as whether there is an option to reject all cookies and whether there are any preselected categories of cookies which are non-essential.
- **Information**: We assess whether the user is sufficiently informed about the use of cookies.

If anything was unclear from the screenshot or the page had not loaded correctly, we manually visited the website in order to assign the characteristics to the cookie banner.

In order to assess whether the user is sufficiently informed about the use of cookies we consider 1) whether the purpose of the cookies is mentioned clearly (e.g., vague wording such as "We use cookies to improve your experience" is not precise enough) and 2) whether the cookie banner explicitly states how the user can adjust their settings or exercise their right to opt-out of cookies. We consider those two aspects the bare minimum necessary for users to assess the consequences of interacting with the cookie banner. We do note that this is a limited approach, since we do not read the full privacy policies of the websites.

Once we had performed the labeling of characteristics, we reduced them to a number of properties which do not seem to be in line with European privacy legislation, partially based on prior work by Matte et. al. [27] and Nouwens et. al [32]. We consider the following properties of banners to have a negative effect on user privacy: 1) *manipulative design*: the banner includes a manipulative design which makes it harder for the user to select one choice than another, i.e., by using a more prominent color for a button or smaller text, 2) *no choice*: no real choice is given to the user regarding cookies, i.e. the only option for the user is to accept all cookies, 3) *no reject all*: the user cannot reject all cookies as easily as accept them because the banner does not provide the option or because the option is present, but less obvious, e.g., the user has to click on "configure cookies" first, and can only then reject them all, 4) *preselected choices*: the banner displays pre-ticked boxes for some of the categories of non-essential cookies and 5) *not sufficiently informed*: the user did not receive sufficient information about the processing of their personal data, i.e., there is neither a link to a privacy policy, nor does the cookie banner include directly information about the data processing, as set out by the transparency requirements in the GDPR [36].

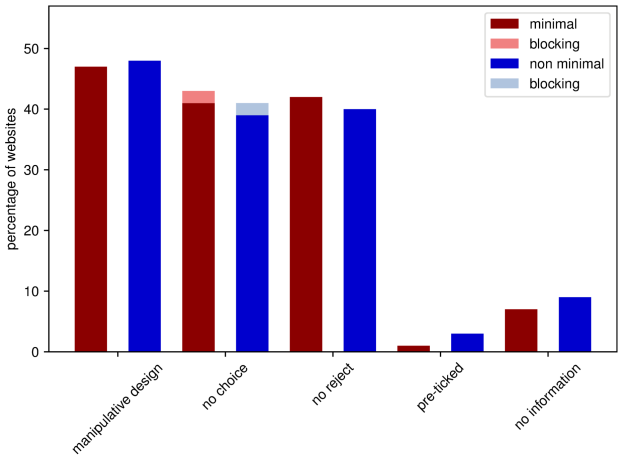


Figure 5: Properties of websites with cookie banners

Results In total, we found a cookie banner on 2220 (35.74%) websites that use OAuth. Websites that implement OAuth with minimal scopes, have a higher adoption rate for cookie banners, namely on 1851 (37.58%) websites, while only 369 (29.54%) websites that use non-minimal OAuth scopes have a cookie banner.
Figure 5 shows the prevalence of each property of cookie banners and how they relate to the use of minimal or non-minimal OAuth scopes on websites. "blocking" refers to the cookie banner being *blocking* on top of providing no real option for the user other than to accept all cookies. 49 websites do not include any of the mentioned properties, which is only 2.22% of websites with a cookie banner. This is consistent with prior work of Bollinger et al. [4], where the authors found at least one GDPR violation on 94.7% of the analyzed websites.
In general, the type of scope requested via OAuth on websites seems to have little to no influence on privacy properties of cookie banners.

8.2 Third-party tracking

One of the most popular ways to track users across multiple websites, is by using third-party cookies. Roesner et. al. show that in 2012, 445 out of the top 500 most popular Alexa websites, include at least one cross-site tracker [37].
According to European law, websites need to obtain consent for the purpose of online tracking [2, 39] [33, art. 5(3)]. This has resulted in a a drop of the number of third parties on popular websites by more than 10% [21].
Consent from the data subject needs to be obtained prior to the processing [27, 35], which entails that third-party tracking cannot be enabled by default on websites. In this section, we examine whether websites allow trackers to place third-party tracking cookies, before the user has interacted with the cookie banner.
Experiment To assess third-party tracking by default on websites, we visit each website in our dataset that uses OAuth and inspect all requests in the browser, in order to determine if third-party trackers are active on the page and whether they set cookies. We consider a third party as a tracker if at least one request is issued from the visited domain to the domain of the third party that would have been blocked by the EasyPrivacy blocklist [10] and the response sets a cookie which has a lifetime that is long enough to be used by trackers for cross-site tracking i.e., longer than 90 days [11, 12, 24, 43]. After the crawl, we validated manually that cookies with the previously-mentioned characteristics contained seemingly-random values or names and are therefore capable of uniquely identifying users.
Results We find that 1259 (68.0%) websites with minimal OAuth scopes include at least one third-party tracker and 222 (60.0%) websites with non-minimal scopes use third-party tracking. Thus websites with minimal scopes include more third-party tracking by default than websites with non-minimal scopes. This indicates that websites which limit the use of OAuth scopes are not purposefully including less third-party trackers. We conclude that there is no connection between the use of OAuth scopes and the use of online tracking by default.

8.3 Preselected marketing consent

A specific type of *manipulative design*, is when websites use preselected options, in order to nudge users into accepting a less favorable option with relation to their privacy. When it comes to pre-ticked boxes, the European Court of Justice has ruled in the Planet49 case [5] that they do not amount to valid consent.
We assess whether websites implement this practice both in relation with consent for the storage of cookies (in Section 8.1), and user consent for marketing communication emails by companies.
Experiment In Section 7, we created an account on 200 websites. During this process, we took note of whether we encountered any preselected choices that authorized the website to send a newsletter of marketing offers to the user by email.
Results During our registration experiment, we found 44 sites with the option to register for marketing communication on websites with OAuth with minimal scopes. 14 out of these 44 (31.82%) websites include a preselected choice. On websites that use OAuth with non-minimal scopes we found a much higher number of pre-ticked choices, namely on 30 (54.55%) out of the 55 websites. These results are surprising when comparing with preselected cookie preferences as an indication of choice, where only 1.26% employ this practice.

9 Discussion

In this section, we discuss the various privacy issues that we identified as part of our research and which may impact users in a negative way. Next, we propose possible solutions to solve these issues. Finally, we discuss the limitations of our methodology.

9.1 Privacy implications

There is a large number of IdPs and even though we only found OAuth implementations for 33 of them on the websites in our dataset, we noticed a lot of variation among the scopes made available by IdPs and the interface shown to users on the authorization screen. Only five IdPs allow the users to actually adjust the scopes according to their preferences. By not allowing the user to adjust the set of scopes and retain only the minimal scope, they are not presented with a fair, informed and freely given choice. The users

are not given a fair choice regarding their personal data since they can only accept all requested scopes, or abort the OAuth process, which raises privacy concerns regardless of the actual scopes that service providers end up requesting. A related problem is the use of *manipulative designs*, which manipulate the user into making unintended choices. We encountered this practice both when examining cookie banners on websites that use OAuth but also in the design choices of authorization screens by IdPs. For instance, a privacy-conscious user that only wants to accept the minimal amount of information sharing when logging in with Facebook, is confronted with an interface where an additional click is required to edit permissions and all scopes in the list are preselected.
Additionally, IdPs do not make a clear distinction between scopes that are minimal and necessary for the identification of the user and scopes that can provide additional functionality to the service provider. The most minimal scopes sometimes even include the sharing of non-essential information about the user such as the gender, birthday and location of the user.
We find that 18.53% of websites that use OAuth require the user to share data from non-minimal scopes. Some of the permissions requested might benefit the user by providing certain functionality on the website. However, we find that part of this data is redundant, since the website provides an alternative login method that does not require the same amount of data, as we found in Section 5. Alternatively, the data might not even be used, as we discovered during our experiment described in Section 6. Thus, our experiments indicate that some websites request more data than they need via OAuth, and therefore do not respect the basic privacy and data protection principle of data minimization. On top of that, some of data from our results might be more sensitive than currently described, due to our restrictive definition of *sensitive* scopes.
Another issue is that when users are faced with multiple options for logging in, i.e., some websites allow login with multiple IdPs and the creation of a separate account for their platform, they have no user-friendly way of determining the best possible option with relation to their privacy. This is because the website does not provide information about the data sharing policy of the different login options, but also because IdPs do not provide any information on how they will be treating the user data after login via OAuth. For instance, whether the fact that the user has linked their account with a specific application might be used to provide behavioral advertising is not communicated to the user, while prior work has shown that IdPs can track the user's browsing behavior upon using OAuth [26].
Finally, we see that the permissions requested via OAuth are mostly unrelated to the way that websites handle users' privacy and personal data in general. The use of manipulative designs, not giving users real choices with relation to their cookie preference and third-party tracking by default remain prevalent regardless of the OAuth implementation.

9.2 Improvements

We provide a number of suggestions for improvements concerning the privacy issues mentioned above, both for websites and IdPs.

9.2.1 Explicit consent Since IdPs make information available that is not necessary for identification of the user, they must rely on consent from the user, as is laid down in European Privacy and Data Protection legislation. This explicit consent should be implemented in authorization screens of each IdP and should provide a way for service providers to clearly indicate which scopes are necessary for authentication and which are optional. Additionally, the interface should not include any designs that manipulate users into making certain choices.

9.2.2 Fine-grained permissions Requesting explicit consent for optional scopes can only be effective if scopes made available by IdPs are sufficiently fine-grained. Some IdPs are already doing this, for instance by separating read and write permissions and providing one scope per piece of user information. Nonetheless, Twitter only provides three different scopes, all of them including write permissions, resulting in login with non-minimal scopes on 75% of websites. Twitter is planning on adjusting the scopes to allow more fine-grained permissions in their API v2 [38], but we did not encounter any website using the updated version during our crawl. IdPs should avoid bundling multiple pieces of user information into a single scope altogether, such as Google's *profile* scope, which includes the user's gender.

9.2.3 Use of scopes by service providers Service providers should always limit themselves to the bare minimum necessary for the correct working of their application and should be encouraged to do so by IdPs by specifying clear guidelines on the use of scopes in their documentation and the enforcement of review processes for non-minimal scopes. Some IdPs already implement an application review process, which encourages websites to think twice before requesting certain scopes solely because they might be useful in the future. Facebook, the most widely used IdP in our dataset, requires a review process for permissions, except for those that are only used for authenticating the user. The stricter the review process, the more likely it is that websites will limit the number of non-minimal scopes they intend to use, e.g., when the website developer is required to submit a screen recording and description of the use for each requested scope. Websites should especially refrain from requesting user data that they are not using. When users decide to decline optional scopes, the service providers should respect the choice of the user and not cause the website to show an error or break.

9.2.4 Transparency towards users Websites should be more transparent towards users about the data sharing involved, especially when they provide multiple login options. Currently, the consequences for the user upon selecting a certain login option are not clear. Morkonda et al. [30] suggest improvements for transparency towards users. They argue that websites should provide a clear overview of the benefits for the user when sharing a certain permission with the website (e.g., a description of the additional functionality that is provided by granting access to the given
permission), and that information needs to be presented prior to initiating the login process, especially when the user can choose between multiple SSO options. We build on these suggestions by arguing that this should be the case not only for multiple SSO options, but for all authentication options available on the website i.e., also for the registration of a new account on the same website.

9.3 Limitations

Our method for automatic detection of OAuth buttons is limited in a number of ways. First, since we search for OAuth buttons by using a set of predefined OAuth authorization endpoints, we cannot detect IdPs other than the ones from our list. A simple alteration to our crawler could consist of searching for HTML elements that have a high similarity with OAuth buttons that we already found on the same page, since websites tend to use the same layout for all SSO options. Second, we were unable to gather data from some websites because the cookie banner would block any clicks or other interaction with the website until the banner has been filled in. This could be solved by making use of addons that automatically close cookie banners such as Consent-O-Matic [31].

Different service providers might request non-minimal scopes for legitimate purposes, depending on the application. For instance, a website with adult content might request the birth date of the user to verify their age. In Section 4.4 we use domain categorization to determine the context of websites and what the reasonable expectation of data collection might be and keep this in mind when performing experiments and making claims about the data collection. However, domain categorization services might not be sufficiently precise to capture the entire contextual information of a website. For example, the category "Business" is fairly broad and includes "Web pages that provide business-related information, such as corporate overviews or business planning and strategies" [29]. In such cases, it is hard to make assumption about whether certain data collection by the website is justified.

10 Related work

Previous work covers various security and privacy issues related to the use of federated SSO, OAuth and OpenID connect.

In 2021, Morkonda et al. [30] conducted a measurement study of SSO on the web that focuses on privacy consequences for users. They built OAuthScope, a tool to extract request parameters (among others the scope of the request) with support for four major IdPs (Facebook, Google, Apple and LinkedIn). The authors use this tool to analyze how the top 500 most popular websites in five countries implement OAuth. They find a prevalence of 10.3% websites using Facebook Login and 9.8% websites using Google's OAuth, both in Germany. Their findings show that there is some variation between requested scopes when multiple IdPs are on the same page. In our work, we perform a large-scale measurement study of OAuth scopes on *100k* websites. Moreover, we perform an in-depth analysis of the scopes that IdPs make available and provide additional experiments where we compare different login options for users in terms of privacy and assess how websites handle personal data of users.

Zhou et al. [44] develop *SSOScan*, an automatic vulnerability checker for applications using Facebook Login. With their tool, they perform a large-scale analysis of 20,000 US websites where they find OAuth implementation vulnerabilities on 20% of websites that use Facebook Login. In total, they find Facebook Login on 9.3% of websites.

Ghasamisharif et al. [16] implement an automatic detection tool for SSO implementations on websites with support for 65 IdPs. With their tool, they search for SSO on 1 million websites and assess authentication security with multiple novel attack scenarios. They find a prevalence of OAuth login on 6.3% websites, which is in line with our findings (7.23% of websites of the top 100k enable OAuth login).

In 2020, Drakonakis et al. [9] built a black-box auditing framework for attacks on authentication cookies. Among other, they deploy their tool on websites that support Facebook and Google SSO.

Balash et al. [1] explore how users perceive connecting Google with third-party applications via SSO. 46% of users indicated that they are concerned about the use of at least one specific Google scope.

Wanpeng and Mitchell [26] focus on privacy risks for users with regards to the information that IdPs receive when the user logs in with SSO. They show that IdPs are capable of tracking the user across different websites and learn about their browsing behavior, and propose several ways to mitigate this privacy issue.

Jonker et al. [23] develop a framework called Shepherd for login automation on websites. While the main focus is not SSO, they do provide support for login automation through Facebook login. They perform a large-scale evaluation of their tool and find adoption of SSO for Facebook on 20% of the top 10k most popular websites according to Alexa rankings.

Several other studies perform an in-depth evaluation of the security threats arising with the use of OAuth [13, 25, 40, 42].

11 Conclusion

Our work provides an analysis of the privacy implications of OAuth-based authentication for users on the web. We discuss issues that impact the user, introduced by both SSO identity providers and service providers. We conclude from our results that the range and granularity of scopes that IdPs make available differ. Most of them provide both minimal scopes in the sense that they are the least amount of information that the IdP can share about the user, and non-minimal scopes which can be used by websites to provide additional functionality or personalization of content. IdPs also design the authorization interface that is shown to users. We find that in practice, these interfaces seldom allow users to customize the scopes that they want to share with the service provider, and when they do, they sometimes employ manipulative designs to nudge users into accepting the sharing of all requested scopes. Our findings show that the non-minimal information that websites request via OAuth is often redundant, since an alternative authentication option is available requiring less information from the user, or because not granting access to that information has no effect on the website and in some cases the requested information is not even used. This practice appears to not be consistent with privacy and data protection legislation such as the GDPR, since it violates the data minimization principle. We perform a large-scale measurement of the use of OAuth scopes on 100k websites and analyze the use of scopes by websites, the privacy implications for users when multiple authentication methods are available, and how websites that use OAuth handle personal data. We hope that our research will contribute towards more privacy-friendly login options for users and more awareness about information shared via OAuth scopes.

Acknowledgments

This research is partially funded by the Research Fund KU Leuven, and by the Flemish Research Programme Cybersecurity. We would like to thank the reviewers for their constructive comments.

References

[1] David G Balash, Xiaoyuan Wu, Miles Grant, Irwin Reyes, and Adam J Aviv. 2022. Security and Privacy Perceptions of Third-Party Application Access for Google Accounts. In *31st USENIX Security Symposium (USENIX Security 22)*. 3397–3414.

[2] European Data Protection Board. 2021. Guidelines 8/2020 on the targeting of social media users. https://edpb.europa.eu/system/files/2021-04/edpb_guidelines_082020_on_the_targeting_of_social_media_users_en.pdf.

[3] European Data Protection Board. 2022. Guidelines 3/2022 on Dark patterns in social media platform interfaces: How to recognise and avoid them. https://edpb.europa.eu/system/files/2022-03/edpb_03-2022_guidelines_on_dark_patterns_in_social_media_platform_interfaces_en.pdf.

[4] Dino Bollinger, Karel Kubicek, Carlos Cotrini, and David Basin. 2022. Automating cookie consent and GDPR violation detection. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association. https://doi.org/10.3929/ethz-b-000525815

[5] JUDGMENT OF THE COURT (Grand Chamber). 2019. Bundesverband der Verbraucherzentralen und Verbraucherverbände - Verbraucherzentrale Bundesverband e.V. v Planet49 GmbH. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:62017CJ0673&from=en.

[6] Google Chrome. 2022. Chrome UX Report.

[7] Camille Cobb and Tadayoshi Kohno. 2017. How public is my private life? Privacy in online dating. In *Proceedings of the 26th International Conference on World Wide Web*. 1231–1240.

[8] Adam Dawes. 2018. More granular Google Account permissions with Google OAuth and APIs. https://developers.googleblog.com/2018/10/more-granular-google-account.html.

[9] Kostas Drakonakis, Sotiris Ioannidis, and Jason Polakis. 2020. The Cookie Hunter: Automated Black-box Auditing for Web Authentication and Authorization Flaws. *Proceedings of the ACM Conference on Computer and Communications Security* (oct 2020), 1953–1970. https://doi.org/10.1145/3372297.3417869

[10] EasyList. 2022. EasyPrivacy Filter List. https://easylist.to/#easyprivacy.

[11] Steven Englehardt and Arvind Narayanan. 2016. Online Tracking: A 1-Million-Site Measurement and Analysis. In *2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)*. 1388–1401. https://doi.org/10.1145/2976749.2978313

[12] Steven Englehardt, Dillon Reisman, Christian Eubank, Peter Zimmerman, Jonathan Mayer, Arvind Narayanan, and Edward W. Felten. 2015. Cookies That Give You Away: The Surveillance Implications of Web Tracking. In *24th International Conference on World Wide Web (WWW '15)*. 289–299. https://doi.org/10.1145/2736277.2741679

[13] Daniel Fett, Ralf Küsters, and Guido Schmitz. 2016. A Comprehensive Formal Security Analysis of OAuth 2.0. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. Association for Computing Machinery, 1204–1215.

[14] Colin Fitzpatrick, Jeremy Birnholtz, and Jed R Brubaker. 2015. Social and personal disclosure in a location-based real time dating app. In *2015 48th Hawaii International Conference on System Sciences*. IEEE, 1983–1992.

[15] OpenID Foundation. 2014. OpenID Connect. https://openid.net/connect/.

[16] Mohammad Ghasemisharif, Amrutha Ramesh, Stephen Checkoway, Chris Kanich, and Jason Polakis. 2018. O Single Sign-Off, Where Art Thou? An Empirical Analysis of Single {Sign-On} Account Hijacking and Session Management on the Web. In *27th USENIX Security Symposium (USENIX Security 18)*. 1475–1492.

[17] Jennifer L Gibbs, Nicole B Ellison, and Chih-Hui Lai. 2011. First comes love, then comes Google: An investigation of uncertainty reduction strategies and self-disclosure in online dating. *Communication Research* 38, 1 (2011), 70–100.

[18] Suhun Han. 2023. googletrans 3.0.0. https://github.com/ssut/py-googletrans.

[19] D. Hardt. 2012. The OAuth 2.0 Authorization Framework. https://www.rfc-editor.org/rfc/rfc6749.

[20] Johannes Henkel. 2021. Adding Rank Magnitude to the CrUX Report in BigQuery. https://developer.chrome.com/blog/crux-rank-magnitude/.

[21] Xuehui Hu and Nishanth Sastry. 2019. Characterising Third Party Cookie Usage in the EU after GDPR. In *Proceedings of the 10th ACM Conference on Web Science* (Boston, Massachusetts, USA) *(WebSci '19)*. Association for Computing Machinery, New York, NY, USA, 137–141. https://doi.org/10.1145/3292522.3326039

[22] IETF. 2012. The OAuth 2.0 Authorization Framework. https://datatracker.ietf.org/doc/html/rfc6749.

[23] Hugo Jonker, Stefan Karsch, Benjamin Krumnow, and Marc Sleegers. 2020. Shepherd: a generic approach to automating website login. (2020).

[24] Martin Koop, Erik Tews, and Stefan Katzenbeisser. 2020. In-Depth Evaluation of Redirect Tracking and Link Usage. *Proceedings on Privacy Enhancing Technologies* 2020, 4 (2020), 394–413. https://doi.org/10.2478/popets-2020-0079

[25] Wanpeng Li and Chris J Mitchell. 2014. Security issues in OAuth 2.0 SSO implementations. In *International Conference on Information Security*. Springer, 529–541.

[26] Wanpeng Li and Chris J Mitchell. 2020. User Access Privacy in OAuth 2.0 and OpenID Connect. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 664–6732. https://doi.org/10.1109/EuroSPW51379.2020.00095

[27] Célestin Matte, Nataliia Bielova, and Cristiana Santos. 2020. Do Cookie Banners Respect my Choice? Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework. In *2020 IEEE Symposium on Security and Privacy (SP)*. 791–809. https://doi.org/10.1109/SP40000.2020.00076

[28] McAfee. 2023. Customer URL Ticketing System. https://sitelookup.mcafee.com/.

[29] McAfee. 2023. Reference Guide: McAfee TrustedSource Web Database. https://sitelookup.mcafee.com/download/ts_wd_reference_guide.pdf.

[30] Srivathsan G Morkonda, Paul C van Oorschot, and Sonia Chiasson. 2021. Exploring privacy implications in OAuth deployments. *arXiv preprint arXiv:2103.02579* (2021).

[31] Midas Nouwens, Rolf Bagge, Janus Bager Kristensen, and Clemens Nylandsted Klokmose. 2022. Consent-O-Matic: Automatically Answering Consent Pop-Ups Using Adversarial Interoperability. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts* (New Orleans, LA, USA) *(CHI EA '22)*. Association for Computing Machinery, New York, NY, USA, Article 238, 7 pages. https://doi.org/10.1145/3491101.3519683

[32] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. (2020). https://doi.org/10.1145/3313831.3376321 arXiv:2001.02479v1

[33] Official Journal of the European Union. 2002. DIRECTIVE 2002/58/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector (Directive on privacy and electronic communications). https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32002L0058&from=EN.

[34] Official Journal of the European Union. 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data. https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN.

[35] Article 29 Working Party. 2016. Article 29 Working Party Guidelines on consent under Regulation 2016/679. https://ec.europa.eu/newsroom/article29/items/623051.

[36] Article 29 Working Party. 2016. Article 29 Working Party Guidelines on transparency under Regulation 2016/679. https://ec.europa.eu/newsroom/article29/items/622227/en.

[37] Franziska Roesner, Tadayoshi Kohno, and David Wetherall. 2012. Detecting and Defending Against Third-Party Tracking on the Web. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*. USENIX Association, San Jose, CA, 155–168. https://www.usenix.org/conference/nsdi12/technical-sessions/presentation/roesner

[38] Twitter. 2022. Twitter API v2 authentication mapping. https://developer.twitter.com/en/docs/authentication/guides/v2-authentication-mapping.

[39] M. Veale and F. Zuiderveen Borgesius. 2022. Adtech and Real-Time Bidding under European Data Protection Law. In *German Law Journal 23(2)*.

[40] Rui Wang, Shuo Chen, and XiaoFeng Wang. 2012. Signing Me onto Your Accounts through Facebook and Google: A Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services. In *2012 IEEE Symposium on Security and Privacy*. 365–379.

[41] The Programmable Web. 2023. ProgrammableWeb. https://www.programmableweb.com/.

[42] Feng Yang and Sathiamoorthy Manoharan. 2013. A security analysis of the OAuth protocol. In *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. 271–276.

[43] Zhiju Yang and Chuan Yue. 2020. A Comparative Measurement Study of Web Tracking on Mobile and Desktop Environments. *Proceedings on Privacy Enhancing Technologies* 2020, 2 (2020), 24–44. https://doi.org/10.2478/popets-2020-0016

[44] Yuchen Zhou and David Evans. 2014. SSOScan: Automated Testing of Web Applications for Single Sign-On Vulnerabilities. In *23rd USENIX Security Symposium (USENIX Security 14)*. 495–510.

A Keywords used for detection

In table 2, we include the keywords that we used to automatically detect login and OAuth buttons, as described in section 3.2. We translated each keyword in 28 languages and used the translations to search for login and OAuth buttons. Helper words indicate keywords which are often used in the same context as login or OAuth buttons. We assign a score for each HTML element when searching for login and OAuth buttons, and the helper words weigh less in the total score than the actual keywords in the first column of the table. We excluded elements which contained keywords referring to cookie banners and social plugins. The full code for our crawler can be found in our public repository.

Login and account registration			SSO		
Keywords	**Helper words**	**Cookie banner words**	**Keywords**	**Helper words**	**Social plugin words**
auth	account	cookie	log in with	oauth	share
log in	new	banner	login with	auth	download
login	member	accept	sign in with	log in	follow
sign in	user	accept	signin with	login	
signin		gdpr	sign up with	sign in	
sign up		compliance	signup with	signin	
signup		compliant	connect with	connect	
register		popup	continue with	sign up	
registration		privacy		signin	
join		policy			
create					

Table 2: Keywords used to automatically discover login/registration and SSO buttons