Automating Website Registration for Studying GDPR Compliance

Karel Kubicek
ETH Zurich
Zurich, Switzerland
karel.kubicek@inf.ethz.ch

Jakob Merane
ETH Zurich
Zurich, Switzerland
jakob.merane@gess.ethz.ch

Ahmed Bouhoula
ETH Zurich
Zurich, Switzerland
ahmed.bouhoula@inf.ethz.ch

David Basin
ETH Zurich
Zurich, Switzerland
basin@inf.ethz.ch

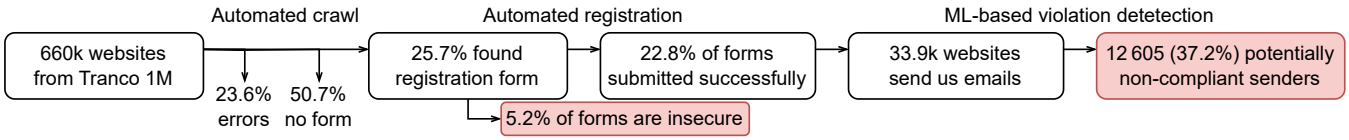


Figure 1: Overview of steps of our study and results.

ABSTRACT

Investigating how websites use sensitive user data is an active research area. However, research based on automated measurements has been limited to those websites that do not require user authentication. To overcome this limitation, we developed a crawler that automates website registrations and newsletter subscriptions and detects both security and privacy threats at scale.

We demonstrate our crawler's capabilities by running it on 660k websites. We use this to identify security and privacy threats and to contextualize them within EU laws, namely the General Data Protection Regulation and ePrivacy Directive. Our methods detect private data collection over insecure HTTP connections and websites sending emails with user-provided passwords. We are also the first to apply machine learning to web forms, assessing violations of marketing consent collection requirements. Overall, we find that 37.2% of websites send marketing emails without proper user consent. This is mostly caused by websites failing both to verify and store consent adequately. Additionally, 1.8% of websites share users' email addresses with third parties without a transparent disclosure.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; **Web application security**; • **Applied computing** → *Law*.

KEYWORDS

Crawling, Registration, Consent, GDPR, ePrivacy, Compliance

WWW '24, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3645709

ACM Reference Format:
Karel Kubicek, Jakob Merane, Ahmed Bouhoula, and David Basin. 2024. Automating Website Registration for Studying GDPR Compliance. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3589334.3645709

1 INTRODUCTION

Since the Internet's beginnings, users have been exposed to security and privacy abuses [11, 31]. Over the past decades, the advertising industry has also been in on the game, employing tracking technologies [39] to gather user data. Many of these abuses are financially motivated since users' behavioral data has economic value, for example, for targeted advertising.

To protect individuals, the European Union (*EU*) has enacted several laws regulating online data collection. The ePrivacy Directive mandates that the sending of marketing emails requires the recipient's prior consent. This consent is currently defined in the General Data Privacy Regulation (*GDPR*). Both of these laws have further requirements that pertain to the processing of personal data, such as following secure communication practices.

Even though data protection authorities can impose heavy fines for GDPR violations, the majority of studies analyzing EU websites' compliance find significant levels of non-compliance. For example, Libert [27] and Englehardt et al. [16] demonstrated that most websites track users through cookies or fingerprinting, respectively. Bouhoula et al. [6] showed that 65% of websites ignore users' cookie consent, and Linden et al. [28] found that almost half of websites' privacy policies violate GDPR requirements.

These studies have focused solely on websites' landing pages. Urban et al. [37] showed that browsing random pages beyond a site's landing page increased the incidence of privacy-invasive practices by 36%. However, their study was also limited to unauthenticated sections of websites, a limitation that has been addressed by only a few researchers. Englehardt et al. [15] and Mathur et al. [30] studied email privacy by signing up for US e-commerce and political

campaign newsletters, observing address sharing to third parties and email tracking. Jonker et al. [21] utilized a public credential database to log in to websites. However, their work was limited to websites with available credentials in that database. Only the study by Drakonakis et al. [14] addressed the registration process in general, but it was successful on just 1.6% of the Alexa top 1M websites, finding half of websites using insecure cookies.

Our work. We present a crawler that achieves a significantly higher registration and newsletter sign-up rate than previous work; in particular, it allows for the analysis of those parts of websites that require prior user authentication, which have been understudied. Utilizing this infrastructure, we examine the compliance of websites with security and privacy requirements for the registration process and analyze the emails received from these websites. The crawl process and associated statistics are depicted in Fig. 1.

To examine websites' compliance, we trained machine learning (*ML*) models on datasets from Kubicek et al. [25], predicting the legal properties of forms and the received emails. By processing the form's legal properties in decision trees, we can detect various kinds of potential violations of consent to marketing emails. We are able to identify instances where consent is likely not active, free, specific, or given at all, thereby violating GDPR requirements.

We evaluate both crawler and violation detection on a crawl of 660k websites, registering or signing-up for newsletter in 5.9% of them. Using an ML classification of email types, we evaluate the verification process of email address control, known as *double opt-in*, finding that 59.8% of websites fail to follow this process. Since we generate a unique email address for each registered website, we discovered that in 14.5% of the cases we received emails from domains other than the domain where we registered. We develop methods to evaluate the transparency of their disclosure practices, finding 1.8% of websites with undeclared or hidden senders.

Contributions. We make the following key contributions. (1) We develop a crawler that achieves more than double the rate of registration and newsletter sign-ups than prior work. Our crawler enables the automated analysis of those parts of websites that require prior user authentication, enabling privacy and security studies at scale that were previously not possible.[1] (2) We automate the detection of security and privacy violations using ML models that allow the fully self-contained processing of crawled registration forms and received emails. (3) We present new results on how tens of thousands of websites potentially violate GDPR consent requirements in the user registration process. Namely, 37.2%, which is 12 605, of websites send marketing emails despite insufficient consent.

2 LEGAL BACKGROUND

During the registration process, users provide personal information to websites, including their names, passwords, telephone numbers, and email addresses. Within the EU, the collection and processing of such information is regulated by the ePrivacy Directive and the General Data Protection Regulation (*GDPR*). The ePrivacy Directive regulates electronic communication, mandating prior consent (an *opt-in* regime, unlike in the US) for sending marketing emails.

The GDPR defines in Articles 4(11), 7, and Recital 32 the requirements for obtaining consent: it must be freely given, specific, informed, and unambiguous. For example, valid consent is considered to be given when users actively mark a checkbox that explicitly asks for consent to receive marketing emails. For forms exclusively dedicated to newsletter subscriptions, where the purpose of receiving marketing emails is implicit in the form's wording, the inclusion of a checkbox becomes redundant. Nevertheless, websites should first send an activation email to verify the user's possession of the registered address through a *double-opt-in* procedure and store the consent adequately [5].

Furthermore, Article 32 of the GDPR emphasizes the importance of implementing robust measures to ensure the secure and private processing of data. These requirements aim to prevent data breaches involving email addresses or passwords, which have led to significant fines [3, 20]. Collecting private data via insecure forms using HTTP or transmitting user-provided passwords in unencrypted emails may therefore violate Article 32(1)(a) of the GDPR [10].

3 CRAWLING INFRASTRUCTURE

We developed an infrastructure for crawling websites and automating user registration. For each website where the crawler registers, we provide a unique email address for a simulated user. Our infrastructure then analyzes the received emails to evaluate how the website uses the user's email address.

3.1 Crawler

Websites vary significantly in both their appearance and implementation, primarily due to the flexibility of JavaScript and CSS. Since all registration options must adhere to the same laws regardless of the technologies used, we focus on registration using email addresses. We therefore do not attempt to register using single sign-on, which was covered by other compliance studies [13].

Below we discuss the crawler's steps. First, the crawler navigates through websites to find pages containing a registration form, which it fills out and submits. Afterwards, it checks the registration state and finishes the double opt-in when this is requested by email.

3.1.1 Implementation. To simulate users' browsing patterns, our crawler utilizes a real browser orchestrated by Selenium. Since existing frameworks such as OpenWPM [16] or webXray [27] are not designed for the complex crawling that our task demands, we do not use them. To represent the majority of web users, we crawl websites using Chrome, but support Firefox as well.

To maximize the chances of successfully loading websites, we employ techniques to evade bot detection, which we describe in Appendix A.1. We have tested that our crawler is not flagged by any major Content Delivery Network (*CDN*), including Cloudflare, Fastly, Amazon CloudFront, and Akamai. Our crawler successfully loads 90.6% of websites, as opposed to 70% without bot evasion techniques.

3.1.2 Navigation. After loading each website with a fresh cache, our crawler determines the page's language using the `polyglot` Python package. If language detection fails, we rely on the <html>

[1] Our crawler is not publicly available as it can be misused for the Bomb attack [35]. However, interested researchers can request access via a form on the page with the study's supplementary materials https://karelkubicek.github.io/post/reg-www.

tag. If English is not the detected language, the crawler tries to switch to the English version, if one exists. We keep browsing the website regardless of the switch to English since we support the majority of European languages (see Appendix A.2).

Keyword matching. The detection of a link or button to change the language is based on matching keywords in the visible text, the 'alt' attribute of <img> tag, or the URL. We curated phrases for determining the purpose of page elements, such as a privacy policy link or marketing consent checkbox. Native speakers translated these phrases to all the supported languages. The curation was guided empirically by example websites. The matching procedure works as follows. First, we remove stop words from both the website and the keyword phrase. Then we lemmatize both texts, using the SpaCy [19] or lemmagen3 [22] lemmatizers, depending on the language support. Next, we map characters with accents or Cyrillic to lowercase ASCII counterparts. Finally, the processed keywords and phrases are matched. This keyword matching approach is also used for other navigation aspects, which are described below.

Navigating webpages. Our crawler uses a priority queue to determine the order of pages to visit of the site. The priority represents the likelihood that a given link leads to a registration or a newsletter form. We order the link categories starting with the highest priority as follows: the registration page, login page, privacy policy and terms and conditions, and others. Links within a category are ordered by their matching score. The 'other' links are selected randomly, preventing the crawler from getting stuck by, e.g., age walls on adult websites. The privacy policy and terms are collected after registration and are relevant for our legal evaluation.

The crawler is restricted to visiting at most twenty pages and the registration page is typically reachable within the first five pages. We allow the crawler to navigate beyond the original TLD+1 domain,[2] but only for a single step, i.e., links found on external domains are not considered for subsequent crawling. This allows registration on an affiliated website directly accessible from the original site. However, it restricts the crawler from navigating away from the original site and identifying unrelated registration forms. Moreover, the keyword-matching algorithm penalizes external domains.

Page content classification. When we load a page, we classify it according to the presence and type of a <form> tag. We apply the decision tree from Fig. 5 to classify the form as `registration`, `login`, `subscription`, `contact`, `search`, or `other`. We evaluated this procedure on a manually annotated dataset collected from 1000 randomly selected English websites from the Tranco 1M,[3] containing 426 forms. There were 12 `contact`, 32 `login`, 139 `subscription`, 163 `registration`, and 80 `other` forms. The procedure from Fig. 5 detected 74% of the `registration` and 94% of the `subscription` forms, with an overall accuracy of 82%.

3.1.3 Form interaction. Once we detect a `registration` form, or a `subscription` form when no `registration` form is found, we interact with it. We first extract the entire subtree of the <form> tag, which we process using the Beautiful Soup library. We use a similar keyword-matching method as in Section 3.1.2 to detect the type of input fields. We search for matches in the corresponding <label> tag and visible text, and in attributes such as autocomplete, type, label, placeholder, and value.

Once we determine the input type, we check which input fields must be filled, as indicated by the presence of the 'required' attribute, an '*,' or a bold label. Then we fill all the required inputs by simulating typing, ensuring that our fictitious credentials seem plausible. Most importantly, we generate a unique email address for every website.

Checkboxes and form submission. We interact with every required checkbox and <select> tag. Once the form is filled, we submit it using any detected submission button or by simulating pressing the Enter key. After submission, we look for a redirect or a change in the website content to detect the registration state. We compute the difference in the website's visible content and the form code to distinguish the following outcomes: the text differs and contains keywords indicating a 'successful' or 'failed' registration; the form is unchanged, usually indicating a 'failed' registration; the form changes after a redirect, indicating a multi-step registration; and none of the above applies, which we denote as an 'unknown' state.

If the registration failed but the same form is still present, we try filling in the credentials again, but this time we confirm all checkboxes. This increases the probability that a required checkbox like "I agree with the terms and conditions" is checked. However, it also increases the probability of consenting to sending marketing emails, which could be detrimental to the objective of our consent study.[4] Then the form is submitted again, possibly many times when the form changes and our heuristic detects a multi-step registration.

CAPTCHA solving. During any of the crawling steps, we might encounter a CAPTCHA. This usually happens during registration or when loading an index page is intercepted by CloudFlare or a similar DDoS-mitigation service. The crawler observes the type of CAPTCHA by the JavaScript that loads it. For reCAPTCHA or hCAPTCHA, we load a template substitute JavaScript that prevents crashes due to website changes of the CAPTCHA invocation. Image CAPTCHAs are detected by keywords directly in the forms. We use an external service that solves CAPTCHA using humans. A third of crawled websites use CAPTCHAs: 75% of them ReCaptcha v2, 20% ReCaptcha v3, 2% hCaptcha, and 3% image CAPTCHA.

Self-hosted mailserver. We self-host generated email addresses at `sybilmail.de`, configured to only receive emails using the Mail Delivery Agent implemented with the Python Maildir library.

3.2 Registration confirmation

Once the crawler determines that the registration state is either 'successful' or 'unknown,' it waits for a confirmation email. As shown in [25], only 85% of websites send emails to registered users and, of those, 59% send double-opt-in emails requiring activation. If we receive an activation email, we extract the activation link or code. The crawler visits the activation link or inserts the code into the open registration. For computational reasons, we wait for activation mail only for a limited period. We discuss this period and issues that we faced with confirmations in Appendix A.4.

[2] TLD+1 refers to the registered domain name preceding the top-level domain. For example, in both `bbc.co.uk` and `bbc.com`, the string 'bbc' represents the TLD+1.
[3] From an older crawl using https://tranco-list.eu/list/89WV/1000000.
[4] Checking all checkboxes hinders detecting the 'marketing email despite user did not consent' violations.

3.3 Deployment

We evaluated our crawler by visiting the June 2022 Tranco 1M list [26], available at https://tranco-list.eu/list/82Q3V. We selected the Tranco list to enable an accurate comparison with prior work that utilizes a similar crawling list. However, Ruth et al. [34] have observed that Tranco represents less accurately users' browsing patterns than the Chrome UX Report (*CrUX*) list. Hence we also evaluate the subset of Tranco that is present in the CrUX list. Unfortunately, due to a processing error, we crawled one million websites that were uniformly randomly sampled with replacement, rather than crawling all the websites. For this reason, our results are only based on 660 202 unique domains, corresponding to the first crawl.

The crawl was conducted from June to September 2022, averaging 10k websites per day on a server equipped with four Intel Xeon E7-8870 CPUs. We ran 60 Chrome browsers in parallel each within a separate docker container, using a freshly launched browser for every website. We used 12 IP addresses provided by the German Research Network, ensuring that the traffic originates in the EU. This, together with an EU address of our fictitious credentials, should indicate for the website that EU privacy laws may apply, which we further discuss in Section 7.

The crawler collected evidence in the form of HTML code from the index and registration pages, as well as extracted text from the privacy policy and terms and conditions. Additionally, we obtained screenshots of each step taken during registration and recorded all the observed cookies. Finally, the crawler collected information regarding the registration status, which we describe below.

3.4 Crawling results

From the 660 202 websites, 504 509 websites were successfully loaded in a supported language. Among the loaded websites, our crawler detected a `registration` or `subscription` form on 25.7% (169 765) of websites. Furthermore, our crawler estimated the success rate of form submissions defined in Section 3.1.3. The estimation indicates that 30.2% of form interactions were successful (51 290), 38.4% failed (65 220), and 31.4% resulted in an undefined state (53 255). The detection of the form submission's state is prone to false positives. See the manual investigation of the crawler registration state in [29, Sec. 6.4] or in the extended version of this study available from https://karelkubicek.github.io/post/reg-www. Further observations from the manual analysis are presented in Appendix B.

We also analyze the results based on whether the websites are in the CrUX list. Note that Tranco 1M and CrUX have only a 51.9% overlap. The crawl was significantly more successful for the CrUX websites. Specifically, 90.6% of the websites present in both lists were successfully loaded, in contrast with 65.3% for non-CrUX websites. Among the websites in the CrUX list, registration was detected as successful in 11.7% of cases (3.9% for non-CrUX websites). Our list choice supports a comparison with [14], relying on the DNS-based Alexa list with domains like `WindowsUpdate.com` without HTTP(S) endpoint. In the future, we recommend crawling the CrUX list to prevent unnecessary computations.

3.5 Ethical considerations

We have identified the following three risks of our study. 1) *Legal risks arising from crawling*: we reviewed various legal regimes and concluded that our research activities do not violate laws related to fraud, trespass, or breach of contract. This is underpinned by the fact that our intentions are the pursuit of good-faith privacy research. 2) *Risks to website owners*: our single crawl negligibly impacts each individual website's capacity. Moreover, the registration rarely results in a manual action by website owners, as the vast majority of emails are automated. In Section 5, we present only aggregated results, preventing harm by wrongful accusation of individual websites for privacy violations. For that reason, we refrain from publicly disclosing our dataset of identified violations, except in cases where parties explicitly provide consent to adhere to the same ethical standards we uphold. 3) *Risks to CAPTCHA solvers*: we contracted with a third-party CAPTCHA solving service. Given the substantial prevalence of CAPTCHAs, accounting for one-third of our successful registrations, and their particular prevalence on higher-profit services, omitting CAPTCHA solving would introduce a significant bias. We carefully compared several providers, excluding those with evidently poor working conditions. Subsequently, we discussed the outsourcing with our university's legal department. Furthermore, we implemented multiple measures to avoid bot detection and, consequently, the need to solve CAPTCHAs. In follow-up research, we transitioned to CAPTCHA solving by research assistants employed at our university for email confirmations.

4 CLASSIFYING LEGAL PROPERTIES

Kubicek et al. [25] defined 21 legal properties relevant to consent compliance and annotated a dataset with them. In this section, we automate the prediction of these properties. Using the dataset from [25], we train two types of ML models: for emails and forms. For each type of model, we describe the feature engineering step, how models are trained, and the results.

4.1 Email features

The training dataset consists of 5725 mostly German and English emails. To reduce the complexity of dealing with multiple languages and to utilize all the training samples, we translate the subjects and bodies into English using LibreTranslate. From each translated email, we further process the headers, subject, and body.

4.1.1 Headers. Email headers constitute a set of key-value string pairs, such as 'Date,' 'Reply-To,' or 'List-Unsubscribe.' While several headers are standardized, there are many, often prefixed with 'X-,' that are custom to specific email servers. We define the *supported keys* as the set of all header keys in the training dataset. This resulted in 76 headers without the 'X-' prefix and 488 headers with it. For each email, we denote whether there is an entry for a given key, whether it contains an email, URL, other text, or whether it is empty.

4.1.2 Subject. The translated subjects are processed with a TF-IDF encoding [1] that we fit to the training dataset, as well as a universal sentence encoder [7]. This pretrained transformer model maps sentences to an embedding in $\mathbb{R}^{512}$.

4.1.3 Body. We extract both the TF-IDF encoding of the translated body and several manually-defined numeric features. These features include the number of characters or sentences of the email text, number of URLs, images, and links.

4.2 Training ML models for emails

Given that our features correspond to tabular data, we use the XGBoost model [9]. XGBoost is well-suited as it outperforms other training algorithms for datasets with few annotated samples but high dimensionality of the feature space.

We train the model using an established ML pipeline. We perform a stratified split of the dataset dedicating 75% for training, saving 25% of the unseen data for validation. We adjust for class-imbalance by sample-weighting. The models optimize the weighted 'multi:softmax' metric for multi-class and 'binary:logistic' for binary classification. All reported results are based on four-fold cross-validation. Given data scarcity, we skip hyperparameter tuning, which would require a further data split, and we use the default XGBoost hyperparameters.

We trained models that predict two distinct legal properties of emails. Our first model predicts whether an email is a marketing email (i.e., newsletters, notifications promoting service monetization, and surveys), a servicing double-opt-in email, or another kind of servicing email (confirmation emails or service updates). Our second model detects whether an email contains a method to unsubscribe, which we evaluate only on marketing emails.

In the last three rows of Table 1, we present the performance the mail-type model. This model achieves 97.7% balanced accuracy, while in the simplified task of deciding only whether email is marketing or servicing (aggregating double opt-ins with confirmations and legal updates), the balanced accuracy increases to 99.2%. The same balanced accuracy of 99.2% is achieved by the model predicting the presence of the unsubscribe options.

4.3 Form features

To transform forms of unlimited length to tabular features, we aggregate the form inputs by the crawler's keyword-based element classification. We group semantically similar inputs, such as the first and last name, full name, and username, see Appendix A.3 for details. We also reduce the complexity by excluding inputs irrelevant to legal classification, such as CAPTCHAs. From all inputs, we extract whether they are required or optional, and from checkboxes also their default values. We concatenate texts, such as corresponding labels, and translate them to English. Finally, we include the form type (`registration` or `subscription`) as a categorical feature.

We process the form texts similarly as emails. Note that checkbox labels often consist of complex, nuanced statements, such as "I don't want to receive special offers about [company name] products." To better capture the meaning of these statements, we extract both sentence embeddings and TF-IDF representations with a limit of 500 words. However, for other form inputs, which tend to have shorter labels like "Your email," we skip sentence embeddings and only use TF-IDF with a limit of 50 words.

The feature extraction produces 5839 tabular features: 69 numerical features about forms' input fields, 3154 TF-IDF columns, and five sequences of $\mathbb{R}^{512}$ sentence embeddings.

4.4 Training ML models for forms

Similarly, as with the email classification, we trained an XGBoost model for each of the 21 binary legal properties annotated by [25]. Note that the training dataset consists of only 668 annotated forms.

Table 1: Performance of legal properties models. 'Deterministic' model stands for the crawler's prediction.

Property	Model	F1	Precision	Recall	Support
Marketing consent	Deterministic	77.58%	80.43%	76.87%	41.92%
	XGBoost	82.33%	82.88%	82.08%	
Marketing purpose	Deterministic	68.06%	64.95%	74.65%	7.04%
	XGBoost	63.15%	61.71%	66.21%	
Marketing checkbox present	Deterministic	79.01%	83.23%	77.33%	35.18%
	XGBoost	81.67%	82.95%	81.04%	
Marketing checkbox pre-checked	Deterministic	71.74%	73.26%	70.44%	5.84%
	XGBoost	57.66%	57.58%	58.43%	
Marketing checkbox forced	Deterministic	55.67%	59.67%	54.22%	3.14%
	XGBoost	58.94%	59.84%	58.38%	
Tying policy and terms checkboxes	Deterministic	71.16%	71.48%	70.86%	16.77%
	XGBoost	77.71%	78.10%	77.92%	
Tying all checkboxes	Deterministic	51.84%	51.51%	64.49%	0.45%
	XGBoost	49.70%	49.70%	49.70%	
Forced policy	XGBoost	74.16%	74.34%	74.07%	26.95%
Forced terms	XGBoost	74.05%	80.28%	70.99%	5.24%
Forced policy and terms	XGBoost	72.55%	72.16%	73.25%	18.41%
Double-opt-in email	XGBoost	94.55%	94.16%	94.95%	12.45%
Single-opt-in email	XGBoost	89.65%	89.89%	89.41%	9.73%
Marketing email	XGBoost	99.11%	99.15%	99.08%	77.82%

To address this data scarcity, we also conducted experiments using the Tabnet model [2], a neural network model optimized for tabular data. One notable advantage of Tabnet over XGBoost is its ability to perform unsupervised pretraining on unlabeled data, allowing it to capture the distribution of classified data. For the pretraining phase, we provided the extracted features of 30k websites where the crawler detected `registration` or `subscription` forms.

Table 1 compares the results of XGBoost with predictions based solely on the crawler's keyword-based classification of form content. However, the crawler's prediction is unavailable for some legal properties, so for space reasons we skip such rows together with Tabnet as its performance is aligned with that of XGBoost. The table provides a summary of the F1 score, precision, and recall, while the last column indicates the percentage of positive samples in the training dataset. Note that the overall performance is highly dependent on the number of positive samples, making scarce properties insufficient for making legal judgments. To mitigate the risk of falsely predicting a privacy violation, we combine the ML predictions with the crawler's keyword-based deterministic prediction. When the presence of a legal property implies a violation, we combine predictions using AND and conversely when it implies compliance, we use OR. We further reduce false positives by conditioning predictions when possible, such as 'marketing checkbox forced' requires 'marketing checkbox present' in the first place.

5 POTENTIAL VIOLATION DETECTION

In this section, we describe our analysis of security threats and potential privacy violations concerning consent in forms and emails. For each method, we give context regarding related work and the EU privacy laws – the GDPR and the ePrivacy Directive.

5.1 Security violations

Using our automated methods, we investigate websites' adherence to security best practices in private data protection as mandated by Article 32 of the GDPR. We focus on the personal information collection through user registration and newsletter sign-up processes.

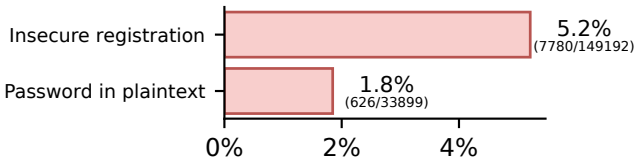


Figure 2: Security threats of registration to websites.

We present our findings in Fig. 2. We detect 5.2% of websites collecting sensitive information through forms from unsecured HTTP websites, failing to ensure the personal data confidentiality required by the GDPR. Utz et al. [38] found such violations on only 2.85% of websites. The difference might stem from our better selection of forms for inspection and the difference in the lists crawled. We also observed that 1.8% of websites, when sending us emails, included the user-provided password in plaintext. The data protection authority of Baden-Württemberg (Germany) [3] considers this a violation of Art. 32 of the GDPR. A similar incidence of 2.3% was observed in the manual study of Kubicek et al. [25].

5.2 Violations of marketing consent in forms

Our detection of potential violations of marketing consent in forms is based on the predicted legal properties used in the decision procedures defined by Kubicek et al. [25, Figs. 6 and 7]. Due to space constraints, in Fig. 3 we only report the aggregated results using these procedures. Note that the baseline of reported incidence is 33 899 of websites that send any email. According to Kubicek et al. [25], only 85% of registrations result in the website sending any email, and this factor should be taken into account when interpreting our results.

Over 43% of registrations resulted in websites that never sent us any marketing emails, potentially caused by issues with account activation (see Appendix A.4) and up to 44% of the marketing emails we received resulted from newsletter subscriptions, reflecting the crawler's higher success rate with `subscription` forms compared to `registration` forms. We found that at least 3.6% of senders violated the opt-in requirement of the ePrivacy Directive by sending marketing emails without any indication of marketing email consent. At least 4.3% of websites then violate the GDPR consent requirements by not including a marketing checkbox, pre-checking the checkbox by default, or tying the checkbox with privacy policy or terms. In 2.0% cases, we received a marketing email despite rejecting consent, where the checkbox was neither pre-checked nor checked by the crawler. The crawler checked all checkboxes on additional 450 (1.3%) of websites.

5.3 Email privacy violations

When users register, websites should verify the ownership of the registered email address through a double-opt-in process. Without this verification, our crawler could be used to subscribe arbitrarily selected email addresses to thousands of newsletters without the owners' consent, resulting in the Bomb attack [35]. The double-opt-in process also ensures that the website retains a clear record of consent. Using the ML model from Section 4.2, we classify the first email we receive from the website. The results presented in Fig. 4a show that 42.4% of websites adhere to the double-opt-in

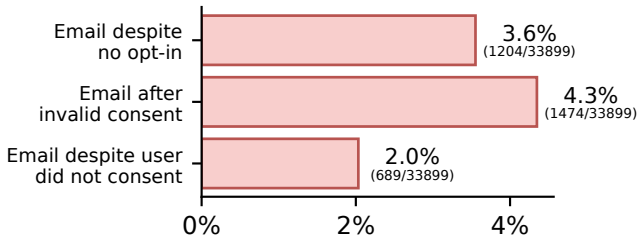


Figure 3: Portion of senders that violate at least one marketing consent requirement. This figure is based on the decision procedures from [25, Figs. 6 and 7].

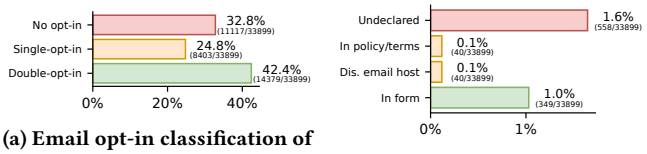
requirements and 24.8% of websites only send a confirmation email, not conforming to the double-opt-in practice. The remaining 32.8% of websites immediately send marketing emails to users.

5.3.1 Email sharing. To track how websites use email addresses, each registration was performed with a unique email address. Detecting when the website shares the email address to third parties, however, poses a challenge. For example, `facebook.com` sends emails from `facebookmail.com`. We developed the following heuristic to address this issue.

For a given registration, we extract a set of TLD+1 domains from which we receive emails. We then match these domains to other domains found in various sources documenting how the website declares this domain. We consider that domains match if the longest common subsequence between two domains is at least half of the shorter domain. This threshold of 0.5 was determined by empirical evaluation of a set of 200 domain matches, resulting in an accuracy of 91% with 2.5% of false negatives (wrongly predicting that domains are not similar) and 7.5% of false positives.

For each sender domain, we identify how the website discloses it. We take the first of the following outcomes, ordered from the most to the least disclosed. (1) The domain name where we registered and any domains that are similar. (2) The domain of the first received email. (3) Any common email host (e.g., `gmail.com`) if the name preceding the @ symbol is similar to the registration domain. (4) Any domain declared on the registration page is marked as 'In form.' (5) Any common host that was not matched previously as 'Dis. email host.' (6) Domains in the privacy policy and terms and conditions, are marked as 'In policy/terms.' (7) If all these checks fail, the domain is marked as 'Undeclared.' We list other methods we considered for third-party sharing detection in Appendix D.

If there are at least two senders and one of them is marked as 'dissimilar email host' or higher in the ordering above, we consider the website to be sharing the email address without a proper disclosure. As shown in Fig. 4b, 1.6% of our email addresses received emails from undeclared domains, including one website that shared our email address to 56 undeclared domains. Additionally, 0.1% of websites sent emails from domains that were only declared in the policy or terms, which are rarely read [4]. Finally, 1.0% of senders are correctly defined directly in the form, and the remaining websites sending emails do so from expected domains. The prevalence of this violation is comparable to results by Kubicek et al. [25].



(a) Email opt-in classification of the first received email.

(b) Email sharing classification.

Figure 4: Potential violations in emails.

6 MANUAL EVALUATION

To evaluate the trustworthiness of our automated methods in a real-world scenario, we manually analyzed a random sample of 100 websites that sent us at least one email.[5] We selected this sample for two reasons. First, it maximizes the number of websites for which our crawler has successfully filled out the form. Second, websites that sent us emails serve as a baseline for reporting violations. Our crawler submitted one `contact`, 54 `subscription`, and 45 `registration` forms. Our crawler misclassified six `subscription` forms as `registration` forms and one `registration` and `contact` form as `subscription` forms.

Out of the registrations or newsletter sign-ups, our crawler was unable to complete 25 double-opt-in procedures. Note that our evaluation of failed double opt-ins is conservative since we classified any lack of email confirmation as a failure, regardless of whether the website actually sends such an email. Nonetheless, considering that almost half of the websites use double opt-in, email confirmation should be improved in future work. Additionally, two registrations were incomplete, but the websites reminded us to finish the registration—a behavior that was studied by Senol et al. [36]. Finally, the crawler successfully submitted the remaining 73 forms.

We examined the email opt-in procedure and found that the first emails from 83 websites were correctly classified. The model misclassified that the first email was for marketing rather than single or double opt-in in nine and five cases, respectively. For a subsequent study described in Appendix C, we completed double opt-ins manually, which allowed us to inspect 110k labeled emails, which we summarize in Fig. 10b. The comparison with Fig. 10a suggests that we tend to classify emails more rarely to be marketing compared to the annotators of the dataset we used for training [25]. As future work, we will incorporate the larger annotated dataset for training to improve the mail-type model's robustness. For insecure registration and passwords sent via email, the sample had two violations each, and their prediction was accurate. We expect false positives to occur only if we misclassify a form.

We evaluated third-party sharing on a sample of 50 websites sending emails from multiple different domains. This sample contained 13 violations. Our method achieved a recall of 85% (two short sender domains were falsely detected on the registration page) and a precision of 79% (three senders used multiple domains belonging to the same company, which can be observed only from the email content).

We also evaluated 40 randomly selected instances for each type of form interface violations. For 'email despite no opt-in,' 17 out of the 40 cases were confirmed as violations. The 'email after invalid consent' type had 12 correctly identified violations. For 'email despite user did not consent,' 12 cases were correct, with false positives largely due to 16 pre-checked checkboxes, constituting other violation, and five instances where the crawler checked the checkbox. The false positives of all methods were mainly caused by misclassifying the form, namely `subscription` for `registration` in 49 cases or `contact` for `registration` in four cases. The findings suggest that particularly the form interface violations are susceptible to false positives, indicating areas for improvement already underway in our follow-up studies. The analysis focused on label accuracy over a comprehensive review, meaning some confirmed violations might not be regarded as such due to other factors.

In conclusion, while our results reasonably represent the landscape of violations, individual violations are sometimes incorrect. Therefore, individual violations should not be blindly trusted without inspecting the evidence we collected. Still, using our detection methods as a tool for privacy enforcement can considerably streamline the detection of violations, as it presents enforcement agencies with a set of potential violations alongside the evidence needed to manually check whether the violation actually took place.

7 LIMITATIONS

Bias. Our study is susceptible to a selection bias introduced by the crawler. As explained in Section 6, our crawler exhibits greater success when signing up for simple websites and forms such as newsletters compared to complex registrations. However, form complexity and website compliance may be correlated. Hence, our results may not be representative of the entire population of websites visited by users.

To mitigate this limitation, we propose involving real users in part of the process. For example, semi-automated techniques can be employed for email confirmation, ensuring that humans accurately handle the various double-opt-in processes used by websites. Additionally, violation detection can be similarly inspected.

Accuracy. All of our findings are prone to misclassification. Hence all violations should be regarded as potential violations. In particular, in cases where our methods exhibit low precision in identifying violations, caution should be exercised when using the results for enforcement purposes. We propose two complementary solutions to address this. First, one can carefully examine the evidence of the violation in the form of screenshots and website source code, similarly to our approach in Section 6. Moreover, a larger training dataset can be constructed by rectifying misclassified violations and adjusting the corresponding legal labels, thereby improving our models in the future. This is particularly crucial for properties with few positive samples, such as the pre-checked marketing checkbox.

Finally, our methods are not a complete audit as there may be additional unaddressed violations. Detecting email sharing might require a longer observation period to capture incriminating events.

Adversarial websites. Website operators could modify their forms, for example by including input fields or text labels invisible to users, to evade our violation detection methods, as was proposed by Zhao et al. [40, 41]. We assume that websites do not do this, since we have not published our violation detection models, making it

[5] A complementary sample of websites where our crawler attempted to register but received no emails is analyzed in Appendix B.1. The analysis suggests only a limited bias induced by our crawler and that the registration rate is higher than reported.

difficult for websites to exploit their weaknesses to evade detection. Moreover, the classification also depends on crawler's keyword-based prediction.

Territorial applicability of EU privacy laws. Although we access the websites from Germany and register a user located in the same country, note that websites with only a few EU visitors may not be obligated to comply with EU regulations. To ensure the enforcement of EU law, future studies can restrict their analysis to lists that are ranking websites by the origin of visitors, such as CrUX or Similarweb. In Section 3.4, we found that the registration rate is favorable when crawling such lists. By utilizing these lists and considering additional factors, like the website's language, one can estimate whether a website is targeting users located in the EU and, consequently, whether their privacy rights must be respected.

8 RELATED WORK

Drakonakis et al. [14] automated the registration process to detect insecurely configured cookies on over half of the websites. Their crawler registered successfully on 1.6% of Alexa top 1M websites, while our crawler achieved registrations on 5.9% of websites from the comparable Tranco list, although nearly half of our registrations can be attributed to newsletter sign-ups. In contrast, Drakonakis et al.'s method also relies on Single Sign-On (SSO) as part of their procedure, which is unsuitable for our mail violation detection requiring a unique email address for each registration. We attempted to re-evaluate their results, without success as their code is dependent on an outdated Google's SSO API. Zhou et al. [42] registers and inspect vulnerabilities specifically on websites with the Facebook SSO, making their work even less aligned with our study objectives.

A similar crawler was proposed by Chatzimpyrros et al. [8]. They claim that their crawler successfully registered on 26.4% of websites, which accounts for 80% of websites with any form. However, their claims are questionable. First, they regard `login` as `registration` forms. Second, they consider form submission as a successful registration. Finally, they do not report the number of senders, except for 0.03% of websites sending emails without crawler's form submission. Senol et al. [36] similarly investigated the detection of private data exfiltration prior to form submission. They found that nearly 3% of websites extract private inputs, such as email addresses.

Jonker et al. [21] developed a crawler that logs into websites using a legitimate crowd-sourced database of credentials called BugMeNot. They were able to login to 14.3% of approximately 50k websites present in this database, but they do not present any privacy or security results. While Jonker et al.'s approach is more effective in logging-in than our crawler, it is limited by the size of the BugMeNot database. Consequently, their approach is unsuitable for detecting violations during the registration process or in emails.

Englehardt et al. [15] automated newsletter subscription, which was successful on 5.7% of US e-commerce websites. They focused on identifying the presence of email tracking and email sharing, revealing third-party sharing by 30% of websites. Mathur et al. [30] studied the 2020 US political campaign with similar observations. In contrast, our research uncovered email address sharing by only 1.8% of the senders. This discrepancy suggests that privacy regulations such as the GDPR foster the protection of privacy, particularly in contrast to jurisdictions that lack similar regulations. Additionally, our crawler was more successful in subscribing to newsletters compared to these works.

Oh et al. [33] studied how website forms meet the GDPR consent requirements, specifying four conditions on consent with privacy policies and terms, including consent presence and tying of checkboxes. We focus on consent to marketing emails, and our methods involve observing the actual data use that violates the consent requirements. Hasan Mansur et al. [18] automated dark pattern detection across websites and apps, including the identification of pre-checked boxes as a default choice. Their findings also underscore the difficulty of detecting this type of violation. A comparable yet manual study was carried out by Gunawan et al. [17].

Consent compliance was thoroughly studied for subpages of websites that do not require prior user authentication. The focus of researchers lay mostly on cookie pop-ups and the consented privacy policies. We refer to a meta-study by Kretschmer et al. [24] that lists and compares publications with these two focal points.

9 CONCLUSIONS AND FUTURE WORK

We have developed a crawler capable of conducting large-scale studies on the security and privacy of website registration. Our crawler more than doubles successful registrations of prior work, signing up to 5.9% of 660k websites. This led to the collection of over 2 million emails. Using this crawler, we were able to detect a wide range of security and privacy threats, fully automating previous manual studies and scaling them by orders of magnitude. To do so, we automated the prediction of complex legal properties of forms and emails using ML. We observed 12 605 websites, which is 37.2% of the websites sending us emails, containing at least one potential violation, or sending a marketing email as the first email.

Our automation fosters various kinds of research. First, our crawler enables future work to analyze the security and privacy of authenticated sections, reflecting how real users browse websites. Second, the option to collect a large-scale dataset of forms and emails can foster research on communication practices. Examples include analyzing whether websites respect the unsubscribe action or studying whether tracking by third-parties is even more present in those parts of websites requiring authentication.

In future work, we will explore using our infrastructure for regulatory enforcement. Namely, by extending our training datasets, such as the annotation of emails of the subsequent crawl, we plan to enhance the predictive capabilities of our machine learning models in detecting violations. These enhanced methods can potentially help understaffed and under-resourced data protection authorities by pre-filtering non-compliant websites and collecting supporting evidence. This can foster efficient enforcement at scale and thereby improve security and privacy for users of the web.

ACKNOWLEDGMENTS

We thank Stefan Bechtold and Alexander Stremitzer for their guidance on the legal aspects of this work. Further, we thank David Bernhard, Patrice Kast [23], Luka Lodrant [29], and Vandit Sharma, for assistance developing the crawler and Joachim Posegga and the Deutsches Forschungsnetz for providing us with university VPN.

REFERENCES

[1] Stephen Akuma, Tyosar Lubem, and Isaac Terngu Adom. 2022. Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets. *International Journal of Information Technology* 14, 7 (2022), 3629–3635. https://doi.org/10.1007/s41870-022-01096-4

[2] Sercan Ö. Arik and Tomas Pfister. 2021. TabNet: Attentive Interpretable Tabular Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 8 (5 2021), 6679–6687. https://doi.org/10.1609/aaai.v35i8.16826

[3] Baden-Württemberg Data Protection Authority (LfDI Baden-Württemberg). 2018. LfDI - O 1018/115. https://gdprhub.eu/index.php?title=LfDI_-_O_1018/115.

[4] Yannis Bakos, Florencia Marotta-Wurgler, and David R Trossen. 2014. Does anyone read the fine print? Consumer attention to standard-form contracts. *The Journal of Legal Studies* 43, 1 (2014), 1–35.

[5] Bayerisches Landesamt für Datenschutzaufsicht. 2022. Guidance from supervisory authorities for the processing of personal data for direct advertising purposes under the General Data Protection Regulation (GDPR) (Orientierungshilfe der Aufsichtsbehörden zur Verarbeitung von personenbezogenen Daten für Zwecke der Direktwerbung unter Geltung der Datenschutz-Grundverordnung (DS-GVO)). https://www.datenschutzkonferenz-online.de/media/oh/OH-Werbung_Februar%202022_final.pdf

[6] Ahmed Bouhoula, Karel Kubicek, Amit Zac, Carlos Cotrini, and David Basin. 2024. Automated Large-Scale Analysis of Cookie Notice Compliance. In *33st USENIX Security Symposium (USENIX Security 24)*. USENIX Association, Philadelphia, PA, 17 pages. https://ahmedbouhoula.github.io/post/automated

[7] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. arXiv:1803.11175 [cs.CL]

[8] Manolis Chatzimpyrros, Konstantinos Solomos, and Sotiris Ioannidis. 2019. You Shall Not Register! Detecting Privacy Leaks Across Registration Forms. In *Computer Security*. Springer, Cham, 91–104.

[9] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (San Francisco, California, USA) *(KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785–794. https://doi.org/10.1145/2939672.2939785

[10] CNIL. 2020. GDPR Developer Guide – Secure your websites, applications and servers. https://lincnil.github.io/GDPR-Developer-Guide/#Sheet_n%C2%B06:_Secure_your_websites,_applications_and_servers

[11] CNN. 2004. AOL worker arrested in spam scheme. https://money.cnn.com/2004/06/23/technology/aol_spam/.

[12] Nurullah Demir, Matteo Große-Kampmann, Tobias Urban, Christian Wressnegger, Thorsten Holz, and Norbert Pohlmann. 2022. Reproducibility and replicability of web measurement studies. In *Proceedings of the ACM Web Conference 2022* (Virtual Event, Lyon, France) *(WWW '22)*. Association for Computing Machinery, New York, NY, USA, 533–544. https://doi.org/10.1145/3485447.3512214

[13] Yana Dimova, Tom Van Goethem, and Wouter Joosen. 2023. Everybody's Looking for SSOmething: A large-scale evaluation on the privacy of OAuth authentication on the web. *Proceedings on Privacy Enhancing Technologies* 2023 (2023), 452–467. Issue 4. https://doi.org/10.56553/popets-2023-0119

[14] Kostas Drakonakis, Sotiris Ioannidis, and Jason Polakis. 2020. The Cookie Hunter: Automated Black-box Auditing for Web Authentication and Authorization Flaws. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 1953–1970. https://doi.org/10.1145/3372297.3417869

[15] Steven Englehardt, Jeffrey Han, and Arvind Narayanan. 2018. I never signed up for this! Privacy implications of email tracking. *Proceedings on Privacy Enhancing Technologies* 2018 (2018), 109–126. Issue 1.

[16] Steven Englehardt and Arvind Narayanan. 2016. Online Tracking: A 1-Million-Site Measurement and Analysis. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1388–1401. https://doi.org/10.1145/2976749.2978313

[17] Johanna Gunawan, Amogh Pradeep, David Choffnes, Woodrow Hartzog, and Christo Wilson. 2021. A Comparative Study of Dark Patterns Across Web and Mobile Modalities. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW2 (Oct. 2021), 377:1–377:29. https://doi.org/10.1145/3479521

[18] S M Hasan Mansur, Sabiha Salma, Damilola Awofisayo, and Kevin Moran. 2023. AidUI: Toward Automated Recognition of Dark Patterns in User Interfaces. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, Melbourne, Australia, 1958–1970. https://doi.org/10.1109/ICSE48619.2023.00166

[19] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2023. spaCy: Industrial-strength Natural Language Processing in Python. https://doi.org/10.5281/zenodo.7970450

[20] Information Commissioner's Officer (ICO). 2020. Monetary Penalty on Marriott International Inc. COM0804337. https://gdprhub.eu/index.php?title=ICO_-_Monetary_Penalty_on_Marriott_International_Inc..

[21] Hugo Jonker, Stefan Karsch, Benjamin Krumnow, and Marc Sleegers. 2020. Shepherd: a Generic Approach to Automating Website Login. In *Proceedings MADWeb 2020: Workshop on Measurements, Attacks, and Defenses for the Web*. Internet Society, Reston, VA, 1–10. https://doi.org/10.14722/madweb.2020.23008

[22] Matjaz Juršic, Igor Mozetic, Tomaz Erjavec, and Nada Lavrac. 2010. Lemmagen: Multilingual lemmatisation with induced ripple-down rules. *Journal of Universal Computer Science* 16, 9 (2010), 1190–1214.

[23] Patrice Kast. 2021. *Automating website registration for GDPR compliance analysis*. Bachelor's thesis. ETH Zurich.

[24] Michael Kretschmer, Jan Pennekamp, and Klaus Wehrle. 2021. Cookie banners and privacy policies: Measuring the impact of the GDPR on the web. *ACM Transactions on the Web (TWEB)* 15, 4 (2021), 1–42.

[25] Karel Kubicek, Jakob Merane, Carlos Cotrini, Alexander Stremitzer, Stefan Bechtold, and David Basin. 2022. Checking Websites' GDPR Consent Compliance for Marketing Emails. *Proceedings on Privacy Enhancing Technologies* 2022 (2022), 282–303. Issue 2. https://doi.org/10.2478/popets-2022-0046

[26] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. 2019. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS 2019)*. The Internet Society, San Diego, CA, USA, 1–15. https://doi.org/10.14722/ndss.2019.23386

[27] Timothy Libert. 2015. Exposing the Hidden Web: An Analysis of Third-Party HTTP Requests on One Million Websites. *International Journal of Communication* 9 (2015), 3544–3561.

[28] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. 2020. The privacy policy landscape after the GDPR. *Proceedings on Privacy Enhancing Technologies* 2020 (2020), 47–64. Issue 1. https://doi.org/10.2478/popets-2020-0004

[29] Luka Lodrant. 2022. *Designing a generic web forms crawler to enable legal compliance analysis of authentication sections*. Master's thesis. ETH Zurich. https://doi.org/10.3929/ethz-b-000534764

[30] Arunesh Mathur, Angelina Wang, Carsten Schwemmer, Maia Hamin, Brandon M. Stewart, and Arvind Narayanan. 2020. Manipulative tactics are the norm in political emails: Evidence from 100K emails from the 2020 U.S. election cycle. https://electionemails2020.org.

[31] David McCandless and Tom Evans. 2022. World's Biggest Data Breaches & Hacks. https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/.

[32] Abraham Mhaidli, Selin Fidan, An Doan, Gina Herakovic, Mukund Srinath, Lee Matheson, Shomir Wilson, and Florian Schaub. 2023. Researchers' Experiences in Analyzing Privacy Policies: Challenges and Opportunities. *Proceedings on Privacy Enhancing Technologies* 2023 (2023), 287–305. Issue 4. https://doi.org/10.56553/popets-2023-0111

[33] Junhyoung Oh, Jinhyoung Hong, Changsoo Lee, Jemin Justin Lee, Simon S Woo, and Kyungho Lee. 2021. Will EU's GDPR Act as an Effective Enforcer to Gain Consent? *IEEE Access* 9 (2021), 79477–79490. https://doi.org/10.1109/ACCESS.2021.3083897

[34] Kimberly Ruth, Deepak Kumar, Brandon Wang, Luke Valenta, and Zakir Durumeric. 2022. Toppling Top Lists: Evaluating the Accuracy of Popular Website Lists. In *Proceedings of the 22nd ACM Internet Measurement Conference* (Nice, France) *(IMC '22)*. Association for Computing Machinery, New York, NY, USA, 374–387. https://doi.org/10.1145/3517745.3561444

[35] Markus Schneider, Haya Shulman, Adi Sidis, Ravid Sidis, and Michael Waidner. 2020. Diving into Email Bomb Attack. In *2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, Valencia, Spain, 286–293. https://doi.org/10.1109/DSN48063.2020.00045

[36] Asuman Senol, Gunes Acar, Mathias Humbert, and Frederik Zuiderveen Borgesius. 2022. Leaky Forms: A Study of Email and Password Exfiltration Before Form Submission. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, USA, 1813–1830. https://www.usenix.org/conference/usenixsecurity22/presentation/senol

[37] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. 2020. Beyond the front page: Measuring third party dynamics in the field. In *Proceedings of The Web Conference 2020* (Taipei, Taiwan) *(WWW '20)*. Association for Computing Machinery, New York, NY, USA, 1275–1286. https://doi.org/10.1145/3366423.3380203

[38] Christine Utz, Matthias Michels, Martin Degeling, Ninja Marnau, and Ben Stock. 2023. Comparing large-scale privacy and security notifications. *Proceedings on Privacy Enhancing Technologies* 2023 (2023), 173–193. Issue 3. https://doi.org/10.56553/popets-2023-0076

[39] Tim Wambach and Katharina Bräunlich. 2017. The Evolution of Third-Party Web Tracking. In *Information Systems Security and Privacy*, Olivier Camp, Steven Furnell, and Paolo Mori (Eds.). Springer International Publishing, Cham, 130–147.

[40] Junbo Zhao, Yoon Kim, Kelly Zhang, Alexander Rush, and Yann LeCun. 2018. Adversarially regularized autoencoders. In *International Conference on Machine Learning*, Vol. 80. PMLR, Stockholm, Sweden, 5902–5911. https://proceedings.mlr.press/v80/zhao18b.html

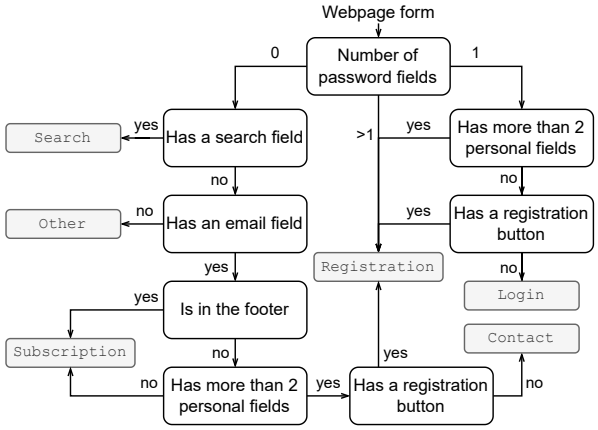


Figure 5: Crawler's form classification procedure.

[41] Zhengli Zhao, Dheeru Dua, and Sameer Singh. 2018. Generating Natural Adversarial Examples. In *International Conference on Learning Representations ICLR*. OpenReview.net, Vancouver, BC, Canada, 16 pages.

[42] 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)*. USENIX Association, San Diego, CA, USA, 495–510.

A CRAWLER

A.1 Bot-evasion techniques

We implemented the following methods to further decrease the chance of our crawling being detected as a bot activity.

Browser. We use Undetected Chromedriver,[6] which extends the usual Chromedriver with numerous bot evasion techniques, such as removing fingerprints unique to Selenium. Unfortunately, there is no equivalent driver available for Firefox.

Fingerprinting evasion. For each page load, the crawler checks the load status. This functionality is not directly implemented by Selenium, so we use Chrome DevTools Protocol for Chrome and Selenium Wire for Firefox. The use of Selenium Wire is however prone to TLS fingerprinting. The proxy and browser differ in the ciphersuite, which is inspected by modern bot detection systems like Cloudflare. While the Firefox-based crawler is prone to this detection, the Chrome implementation does not use any proxy. Additionally we must run Chrome with a non-root user. Chrome disables sandboxing protections when run as root, making it flagged as a bot by Cloudflare.

Interaction speeds. Interactions with the website cannot occur instantaneously, as humans have limited reading and writing speeds. Our crawler introduces random time delays before each click and during typing to mimic human behavior.

IP address. As we study the impact of the EU's privacy regulations, we focused our data collection on traffic originating from within the EU. We considered using commercial VPNs, datacenter or residential proxies, or a university VPN located in the EU. According to a study by Demir et al. [12], residential proxies are the least likely to be detected as bot traffic, closely followed by university VPNs, while datacenters and commercial VPNs are blocked more frequently. Since purchasing a large number of residential IP addresses from services like Bright Data is expensive (≥$10k for our crawl), we used a VPN provided by a university in Germany, which gave us access to a block of 12 IP addresses.

[6] https://github.com/ultrafunkamsterdam/undetected-chromedriver

A.2 Supported languages

Our crawler supports 37 languages, with most of the keywords being translated by native or proficient speakers of the language, whom we instructed in observing multiple registration or newsletter-subscription websites prior to the translation. These languages are: Bulgarian, Bosnian, Catalan, **Czech**, Welsh, **Danish**, **German**, **Greek**, **English**, **Spanish**, Estonian, Basque, **Finnish**, **French**, Galician, Croatian, **Hungarian**, Icelandic, **Italian**, Luxembourgish, Lithuanian, Latvian, Macedonian, Maltese, **Dutch**, Norwegian, **Polish**, **Portuguese**, Romanian, **Russian**, **Slovak**, Slovenian, Albanian, Serbian, **Swedish**, **Turkish**, and **Ukrainian**. From these languages, only 18 of them are supported by LibreTranslate and therefore are suitable for detection of all the violations. We highlighted these languages in bold. Note that the LibreTranslate support is constantly improving, both in the terms of translation quality and the number of supported languages, which rose to 28 by the camera-ready version of this publication.

We are aware of the following limitation of the machine translation. First, nuances in the form or email text might be lost. Second, as the training data is in German and English, the models should reflect well the websites in these languages, yet their performance can drop on websites in a language absent in the training data. Finally, the ePrivacy Directive implementation is not absolutely consistent among EU countries, the impact of such inconsistencies remains a limitation. We believe that the generalization of our methods to so many languages, even if constrained by the machine translation quality, is an important contribution of our work in the context of understudied non-English websites [32, Sec. 4.4.2].

A.3 Form features

For form classification, we use aggregated form features, features for specific input types that we order to provide stable tabular features' ordering, and features extracted from specific parts of text in the form. The aggregated features include the number of inputs in total, the number of the <input>, <textarea>, <select>, and <button> tags. Our crawler distinguishes various form inputs by their semantics, which we aggregate into the following groups.

- Each of: mail, password, phone, username
- names: first, middle, last or full name
- name-other: organization, title, honorific prefix, other text fields
- address: street, house number, city, ZIP, country, full address
- Both age and sex
- checkbox: terms of service
- checkbox: privacy policy
- checkbox: privacy policy and terms of service
- checkbox: marketing, privacy policy and terms of service
- checkbox: marketing
- checkbox: SMS

- checkbox: age
- checkbox: other
- birthday: day, month, year, full birth, other <select>
- submit buttons: registration, subscribe
- other buttons: login, contact, other

For each of these groups, features correspond to the number of inputs in the group, whether any of these inputs is required, the default values, i.e., text for text input or Boolean for checkboxes and radio buttons, and the text of the closes label. The texts are then processed by the TD-IDF model, with a vocabulary size of 50. In the case of checkboxes, the submission button, and the entire aggregated text of the form, the text is processed by the TF-IDF model with a 500 words vocabulary and embeddings are extracted using the universal sentence encoder [7].

A.4 Email confirmation

Since letting the crawler wait for an activation email is computationally expensive, our crawler only waits for up to 30 seconds. If an activation email is received after this period, we activate the registration using a standalone script that processes the incoming emails from all the crawlers running in parallel. However, this script lacks the registration page session, such as cookies, which reduces its success rate compared to the stateful crawler within the 30-second period. We analyzed the distribution of confirmation emails over time in our crawl and observed that less than half of the activation emails arrived within this 30-second period. To achieve a higher success rate for account activation, we recommend waiting for five minutes in future work, since 97.7% of websites that send activation emails do so within this period. Further increasing the waiting period to, say, fifteen minutes would only marginally improve this rate to 99.0%. The longer waiting time, however, comes at the expense of crawling time.

Unfortunately, due to technical issues the independent confirmation script was malfunctioning for about half of the crawl. The combination of a shorter period of waiting by the crawler and the faulty script resulted in lower confirmation rates. This caused the presented results in Section 5 to be more conservative. Namely, websites that violated the consent in the form but then complied with the double-opt-in requirement and never sent us a marketing email are falsely considered compliant.

B MANUAL ANALYSIS OF THE CRAWLER

We conducted a manual investigation of 200 crawled websites to evaluate form detection. Out of the 200 pages, 19 failed to load, the analysis presented below pertains to the remaining 181 websites.

In Fig. 6, we present the evaluation of `registration` form detection. Among the sampled websites, 55 had a `registration` form, of which our crawler successfully detected two-thirds. Additionally, the crawler identified a wrong form (e.g., a `contact` form or password reset form) in 10.5% of the evaluated websites. Furthermore, in 4.7% of the websites, the crawler misclassified a `subscription` form as a `registration` form.

Fig. 7 illustrates the evaluation of discovered `subscription` forms. Our findings reveal that 73.0% of websites do not have a `subscription` form (although note that many websites contain both a `subscription` form and a `registration` form). The crawler

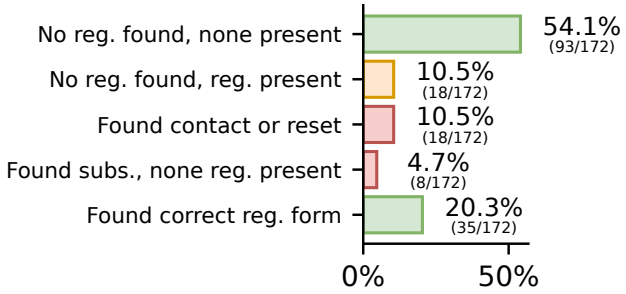


Figure 6: Evaluation of crawler-detected `registration` forms.

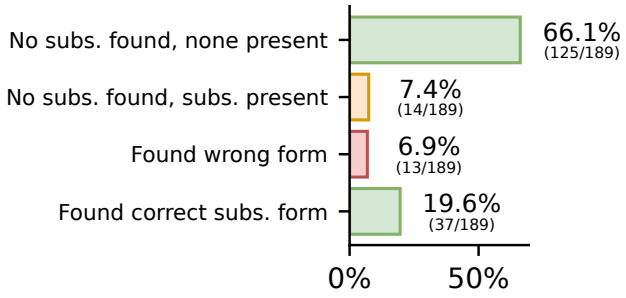


Figure 7: Evaluation of crawler-detected `subscription` forms.

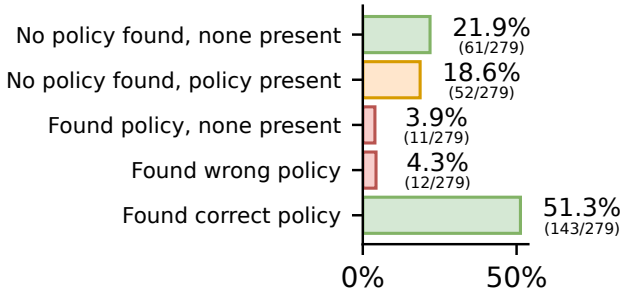


Figure 8: Evaluation of found privacy policies.

accurately determined the absence of this form on two-thirds of the websites, and on 19.6% of the websites, it correctly identified the existing form. However, the crawler failed to detect the `subscription` form on 7.4% of the analyzed websites, and in 6.9% of websites, it found an incorrect form.

Figs. 8 and 9 illustrate the evaluation of the detected policies and terms and conditions on a list of 300 websites. Our manual evaluation showed that almost 75% and 65% of websites contain privacy policies and terms and conditions, respectively. Our crawler can then detect the correct privacy policy on 51% of websites and correctly conclude that there is no policy on 21% of websites. On 19% of websites, it fails to find the policy and in the remaining 9% of cases, it finds a wrong document. The crawler is correct in finding the terms and detects the absence of terms on 37% and 21% of websites, respectively. It failed to detect terms on 13% of websites and in the remaining 29% of cases, it detects a wrong document.

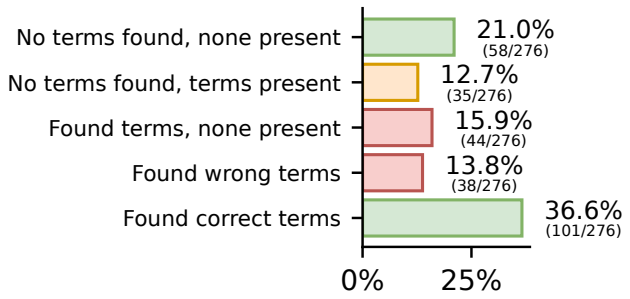
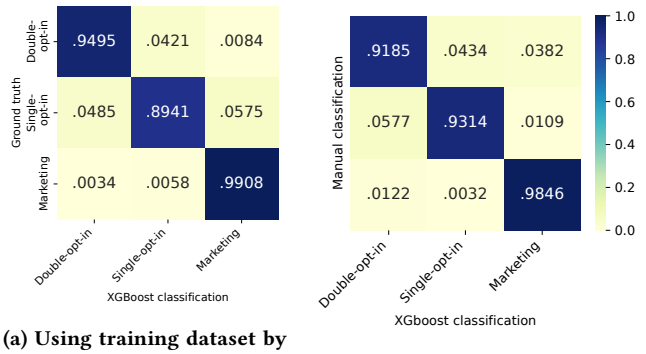


Figure 9: Evaluation of found terms and conditions.



(a) Using training dataset by Kubicek et al. [25].

(b) Using our labeled 110k emails.

Figure 10: Confusion matrices of mail type classification.

B.1 Failed registrations

We inspected 100 websites where the crawler detected registration or newsletter form, but we have not received any email. First, we noticed that forms of 22 of these websites were successfully submitted, the websites only do not send any emails. Second, 33 forms required either telephone or some other information not available to the crawler (library card number, bank account, etc.), although on five of these websites, a random input like "Ignore this automated message" was accepted. Third, other reasons for not being able to register were wrong form detection, namely our crawler submitted registration data to eleven `comment`, eight `login`, seven `contact`, and four `other` forms. Some of the websites were having a proper `registration` of `subscription` form, so our manual submission succeeded in 52 cases. On these websites, we observed slightly higher violation rates of no opt-in, invalid consent, or no consent given, but these can be explained by conservative interpretation of our models. However, the email classification seems problematic, we received a marketing email first only in 9% of cases, while 60% of websites followed the double-opt-in procedure.

C SUBSEQUENT CRAWL AND EMAIL ANNOTATION

For a subsequent study, we crawled the May 2023 CrUX list of the top 100k German websites, resulting in 8230 websites sending us emails. The initial 10 596 emails were annotated by two researchers, one of them among the authors. Since the remaining emails were almost entirely marketing, we marked them as marketing if they were classified so by the previous model. The dataset contains 110 151 emails, of which are 7% double opt-ins, 5% single opt-ins, and 88% marketing.

D EMAIL SHARING

In addition to the described methods in Section 5.3.1, we explored the following methodologies to minimize false positives and negatives in our violation detection for third-party sharing.

TLS certificates. We considered the extraction of company information from TLS certificates. However, note that only a minority, less than 30% of websites, include company names within their TLS certificates. This practice is predominantly observed among highly popular websites, whereas our automated crawling and classification methods perform the best on websites of medium popularity. Furthermore, our observations revealed that websites associated with the same parent companies commonly employ different company names in their certificates, calling into question the usefulness of this approach.

Co-occurrences. We investigated the co-occurrence of senders who send emails to multiple addresses registered by our crawler. This analysis uncovered two distinct scenarios. First, email hosting providers such as Gmail were observed to send emails to multiple accounts, suggesting that co-occurrence could be indicative of websites that are compliant with privacy regulations. Conversely, we identified clusters of websites that shared email addresses among themselves without belonging to the same corporate group and without obtaining proper user consent, which strongly suggests privacy violations.

Company databases. We explored the use of databases such as Whois, Crunchbase, and Orbis to discover connections between domains owned by the same companies. However, Whois data has become increasingly sparse due to privacy concerns. Moreover, both Crunchbase and Orbis feature inconsistent company name records, leading to false positive violation reports and occasionally attributing incorrect company names, resulting in false negative violation reports. We also considered the webXray dataset curated by Libert [27],[7] but it primarily targets third parties within the tracking industry, which seldom overlap with email senders.

Besides the classification of the 3rd-party sharing in Section 5.3.1, we also detect whether the third party is a well-known newsletter sender (e.g., Mailchimp or Sendgrid; we keep a list of almost 100 such newsletter senders). However, we detect these domains rarely, namely in 27 cases do the senders belong to the undeclared sharing category. The newsletter companies recommend configuring the sending domain to the first party (using CNAME). For future work, we plan to inspect the IP address in the SMTP connection sending us the email, allowing us to inspect email address processing, since the newsletter companies are, according to the GDPR, 'processors' of personal data, and therefore must also be declared.

Received 6 October 2023; revised 18 February 2024

[7] https://github.com/agilemobiledev/webXray/blob/master/webxray/resources/org_domains/org_domains.json