

PrivacyScore: Improving Privacy and Security via Crowd-Sourced Benchmarks of Websites*

Max Maass¹, Pascal Wichmann², Henning Pridöhl², Dominik Herrmann²

¹ Technische Universität Darmstadt, Secure Mobile Networking Lab, Germany
mmaass@seemoo.tu-darmstadt.de

² Universität Hamburg, Security in Distributed Systems Group, Germany
{wichmann|pridoehl|herrmann}@informatik.uni-hamburg.de

Abstract. Website owners make conscious and unconscious decisions that affect their users, potentially exposing them to privacy and security risks in the process. In this paper we introduce PrivacyScore, an automated website scanning portal that allows anyone to benchmark security and privacy features of multiple websites. In contrast to existing projects, the checks implemented in PrivacyScore cover a wider range of potential privacy and security issues. Furthermore, users can control the ranking and analysis methodology. Therefore, PrivacyScore can also be used by data protection authorities to perform regularly scheduled compliance checks. In the long term we hope that the transparency resulting from the published assessments creates an incentive for website owners to improve their sites. The public availability of a first version of PrivacyScore was announced at the ENISA Annual Privacy Forum in June 2017.

Keywords: Scanner, Tracking, Compliance, Security, Privacy, Data protection

1 Introduction

Setting up and running a website requires expert knowledge and a substantial amount of resources. Software systems have to be configured correctly, kept up-to-date and secured against attacks that are discovered during the lifetime of a site. The continuing flow of reports about security incidents on major websites indicates that many website operators are incapable of maintaining a sufficient level of security.

Vulnerabilities resulting from mistakes and negligence of website operators constitute a privacy risk for users. For instance, insecurely configured transport encryption may be broken by eavesdroppers, and sensitive data on web servers may be stolen by criminals.

However, the privacy of users may also be under attack by the website operators themselves. Modern web design relies on third-party services: analytics

* A German version of this paper with a more detailed discussion of the legal considerations is available at [21]. This is the authors' version. The final publication is available at link.springer.com via dx.doi.org/10.1007/978-3-319-67280-9_10

services provide insights about visitors and ad networks generate revenue. Site owners commonly choose privacy-infringing third-party services, even though many typical requirements can be met with privacy-friendly alternatives, e. g., by running a local analytics tool such as Piwik [27].

The security and privacy risks resulting from (un)conscious decisions of site operators are complex and elusive. There is no straightforward way for end users to determine whether a site takes security seriously and whether it respects their privacy.

Existing website scanning services do not give a comprehensive impression of security and privacy features of a site. First, these services are mostly geared towards skilled administrators to assist with self-assessment. Secondly, most scanners focus on a very specific area: the properties of the encrypted connections from the browser to the web server. Thirdly, the interface of existing scanners makes it difficult to compare the results of different sites side-by-side. As a result, site operators have little incentive to improve security and privacy on their website beyond the status quo.

Our project PrivacyScore aims to fill this gap. Building on existing work, and in cooperation with data protection authorities (DPAs) and data protection non-governmental organizations (NGOs), we are currently designing and implementing the **PrivacyScore Benchmarking Portal**³ to assess both security and privacy measures of websites. The public availability of a first version of PrivacyScore was announced at the ENISA Annual Privacy Forum in June 2017.

The target audience of PrivacyScore are *end users* who want to know how a particular website ranks in its peer group, *researchers* who want to perform studies about security features of websites, and *NGOs and DPAs* that want to check the compliance of websites with data protection laws.

Our platform offers a scanning infrastructure and is built on the crowd-sourcing paradigm. It offers the following distinctive features:

- Users can set up crowd-sourced lists of websites that belong to a certain category (e. g., all public schools in France). These site lists and their corresponding rankings are publicly visible and updated automatically on a regular basis to create an incentive for operators to improve their security and privacy compared to their peers or competitors. The results contain actionable advice for operators who want to improve their rating.
- Both list creators and ordinary users can adapt the rating and ranking of sites via customized ranking schemes. This allows list creators to create benchmarks that highlight particular checks, while giving users the opportunity to tailor the ranking to their personal privacy preferences.
- PrivacyScore is released under an open-source license (GPLv3 or later) and the data collected on the platform is made available to the public in human-readable and machine-readable form.

³ Available online at <https://privacyscore.org/>.

The rest of the paper is structured as follows: After having reviewed related work in Sect. 2, we present the main features of PrivacyScore in Sect. 3. Section 4 provides details on our security and privacy checks, while Sect. 5 outlines the implementation. Finally, we discuss legal and ethical considerations in Sect. 6 before we conclude in Sect. 7.

2 Related Work

Most existing scanning services focus on *security issues* which may allow malicious adversaries to compromise a web server or to eavesdrop on encrypted traffic. Prominent examples are the SSL scanners by Qualys [28], High-Tech Bridge [14], and Mozilla [26]. Some of these services also check for the presence of relatively new HTTP headers [31], which protect against selected attacks.

On the other hand, there are only very few scanning services that focus on *privacy issues*, i. e., design decisions allowing site owners or third parties to track users. A popular service is “Webbkoll” [9], which is offered by *dataskydd.net*, a Swedish non-profit data protection organization.

Besides scanning services that allow users to scan arbitrary websites, there have been several efforts to scan pre-defined lists of websites and IP addresses. For instance, Helme regularly publishes a dataset containing an analysis of the HTTP security headers for the “Alexa Top 1 Million” list [30]. Many more scans are available in the *scans.io* repository.

Furthermore, there are numerous scientific studies that have analyzed security (e. g., [15,23]) and privacy aspects (e. g., [12]) of popular websites. These studies provide insights about the overall state of privacy and the adoption of security technologies on the web. However, as they typically focus on aggregate statistics, they do not create strong incentives for individual site owners to improve. Moreover, they rarely provide sector-specific insights ([12] is one of the few exceptions). And finally, as the published data is typically not updated after the publication, the results become outdated rather quickly. This is also true for the 2016 municipality survey [8] of *dataskydd.net*, that inspired us to create PrivacyScore.

More distantly related to our work are browser add-ons such as Lightbeam [25] and EFF’s Privacy Badger [11]. These tools analyze visited sites on the fly and allow users to block dedicated tracking services. However, some add-ons have been shown to track their own users [32,16].

With the existing projects and tools checking sites for legal compliance and comparing multiple sites are tedious processes. At the moment these tasks involve substantial amounts of manual work, because various results have to be obtained from multiple sources. With PrivacyScore we want to unify and simplify this process so that it becomes easily repeatable.

3 PrivacyScore Overview

In this section we describe the main features of the PrivacyScore platform. Figure 1 provides an overview of the use cases and data structures.

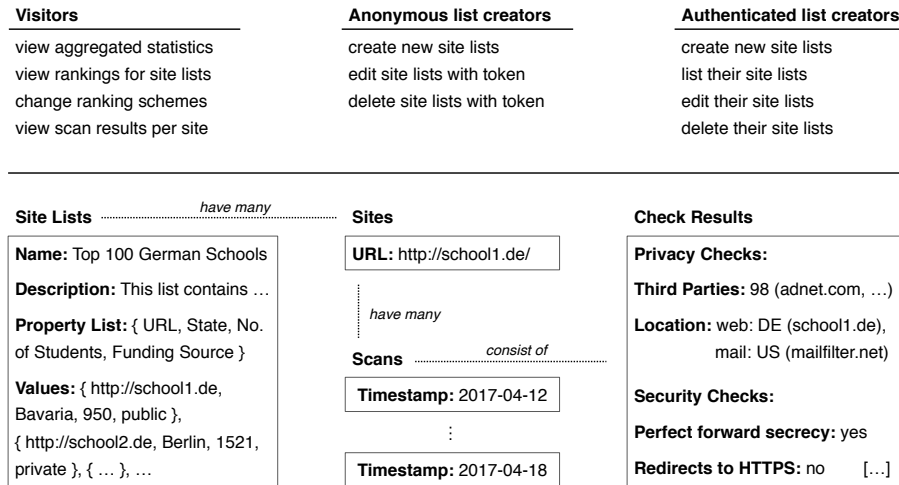


Fig. 1. Use cases and data structures

3.1 Main Use Cases

Like other website scanning services, PrivacyScore allows users to submit URLs of websites which are then analyzed in terms of security and privacy features. While one-off single-site scans are supported, the main purpose of PrivacyScore is the creation and publication of site lists that comprise multiple websites that share a common feature. For instance, a list could consist of the websites of all schools of a country, of major newspapers, or popular health portals.

Users can browse the database of existing site lists (using tags and a full-text search) or create new site lists. New site lists can be created by anyone at any time. A site list is created by supplying a *list of website URLs*. In addition, the creator of a site list can supply metadata such as a title for the list, a description of the methodology used to select the URLs, and a set of tags. Once all relevant data has been entered (or uploaded using a CSV file), it is submitted to the scanning engine, which retrieves each site and gathers a number of facts using multiple scan modules. The gathered facts are evaluated by checks that assess specific security and privacy properties (cf. Sect. 4). The results can be displayed at varying levels of detail, from aggregate statistics over a tabular ranking of all sites to detailed results for each individual check and site.

Site lists and results are retained in the system. All lists are *re-scanned on a regular basis* by default in order to document when site operators make changes to their sites (site list creators can disable automatic re-scans).

The system supports both incidental and professional use. Occasional users can create site lists or scan individual websites without registration. They receive a randomly generated *access token* that allows them to change and delete their site list at a later time. Professional users like DPAs can create a *user account* to manage their site lists without having to keep track of their access tokens.

3.2 User-centric Results

PrivacyScore has two features that enable users to create meaningful assessments: *user-defined properties* and *user-defined ranking schemes*.

User-defined Properties A set of properties defined by the creator of a list can be stored for each site within a list. These properties can provide additional insights. The set of properties and their initial values are supplied by the creator of a site list, but they can also be refined and updated at a later time. In the example of school websites, the following properties might be of interest: location (federal state), number of students, and whether it is publicly or privately funded. When users view the results of the assessment, they can use the properties to perform comparisons like “is there a difference between public and private schools?”.

User-defined Ranking Schemes The ranking of the scanned websites is obtained by aggregating the results of the individual checks using a user-defined ranking scheme. Most existing scanning services use predefined weights to model the fact that some vulnerabilities are more critical than others. In contrast to existing scanning services, where the scanning service imposes its ranking methodology on all scans, PrivacyScore gives its users more control. Our user-defined ranking schemes make the results useful for different audiences. For instance, DPAs may solely be interested in features indicating non-compliance with data protection law, while privacy activists may want a more strict rating for the purpose of “naming and shaming” websites with excessive data collection practices.

List creators can choose from a set of pre-defined ranking schemes. Moreover, users who view a site list can switch to a different ranking scheme or define their own on the fly. Our framework allows to calculate (potentially weighted) total scores or to define different types of ratings, for instance based on school grades (e.g., A⁺ to F) or by color-coding the result (e.g., red, yellow, green).

In order to make the results more easily accessible, checks are organized in *check groups*. A ranking scheme defines how checks are organized into groups, how each possible outcome of a check is supposed to be rated, and which importance is associated to each check and group.

Rating and Ranking in the Beta In the following we explain the ranking scheme that is available at the start of the beta phase (cf. Fig. 2). This is subject to

#	URL	Name	Type	NoTrack	Attacks	EncWeb	EncMail	Rating
1	http://www.ibb.de/ (1 failure) / 2017-06-24 @ 18:29:42	IBB Berlin	public	✓	!	!	?	!
2	http://www.helaba.de/ / 2017-06-24 @ 18:26:12	Hessische Landesbank	public	✓	!	!	!	!
3	http://www.berlinhyp.de/ / 2017-06-24 @ 18:24:42	Berlin Hyp AG	public	✓	!	✘	!	✘
4	http://www.bayernlb.com/ / 2017-06-24 @ 18:24:26	Bayerische Landesbank	public	✓	!	!	!	!
5	http://www.bhw.de/ / 2017-06-24 @ 18:23:35	BHW Bausparkasse AG	private	!	!	!	?	!
6	http://www.pfandbriefbank.com/ / 2017-06-24 @ 18:22:38	Deutsche Pfandbriefbank AG	private	!	!	!	?	!
7	http://www.hypovereinsbank.de/ / 2017-06-24 @ 18:22:57	Unicredit Bank AG	private	!	!	!	!	!

Fig. 2. Ranking for a site list that contains home pages of German banks

change and we will provide a more comprehensive description of the methodology at a later time. For now the mapping of checks to check groups cannot be influenced by users and there are four check groups: *NoTrack*, *Attacks*, *EncWeb*, and *EncMail* (cf. Sect. 4).

The currently implemented ranking scheme is based on the user-defined *order of the check groups*. Users can manipulate the order according to their personal preferences. Sites with a “good” (green) rating in the highest-priority check group (leftmost column) are pulled to the top of the table, followed by sites with a yellow and red rating in the first check group, respectively. All sites with the same rating (color) in the first check group are further sorted according to their rating in the next check group – and so on until all check groups have been considered.

The rating of a site in a specific check group is determined by the checks belonging to that group. In the beta phase, a *green* rating is obtained if all checks of a group succeed, while a *red* rating is obtained if one of the critical checks within a group failed. Otherwise the site obtains a *yellow* rating. The criteria for each check are documented on the PrivacyScore website (cf. Fig. 3).

Finally, the *overall rating* of a site is given by the rating of the worst group. For instance, a site that has only green group ratings gets a green overall rating, while a site with at least one yellow or red group rating gets a yellow or red overall rating, respectively.

3.3 Open Data versus Privacy

We have designed PrivacyScore with the intention to improve transparency for end users by creating awareness for poor security and privacy practices of site

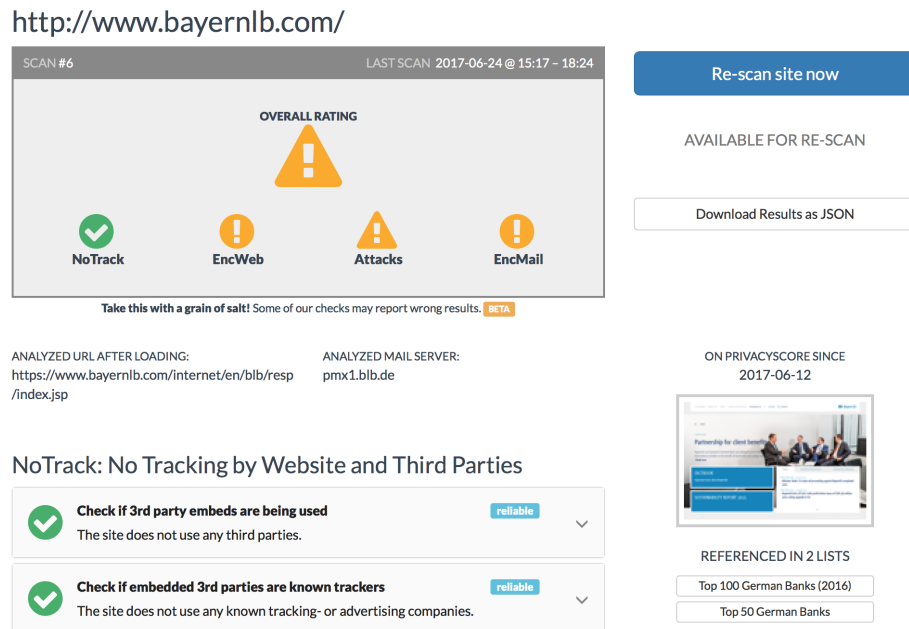


Fig. 3. Scan results for individual sites contain a list of all checks.

operators. Therefore, all data that is generated on the platform is available publicly via an open RESTful API.

However, we recognize that some professional users have special privacy requirements. Therefore, all of the scan modules implemented on PrivacyScore run locally, i. e., the scanned URLs are not leaked to third parties. In addition, in order to support private investigations, site list can be marked as *private*. This allows corporate users and DPAs to use PrivacyScore without disclosing this fact to the public. Furthermore, site lists can be deleted by the list creator or a system administrator upon request.

We also support users with even stricter privacy requirements, who can run their own (in-house) instance of PrivacyScore, as we release our source code under the GPLv3+ license.⁴

The open source nature also makes it easy to add additional scan modules, which are Python modules providing a specific interface. That way, scan modules can either be implemented directly in Python or just use a simple Python wrapper calling any external executable.

⁴ See <https://github.com/privacyscore>

4 Privacy and Security Checks

PrivacyScore is designed to perform various checks on each website. In the following we describe our roadmap, i. e., the checks that we plan to support in PrivacyScore. We distinguish two types of checks: *Security checks* analyze whether site operators follow best practices that protect against malicious attacks by outsiders. These checks are relevant because successful attacks may infringe the privacy of users. We also perform various *privacy checks* that determine whether the owners of a site designed it in such a way that it infringes the privacy of users.

4.1 Privacy Checks

The privacy checks are reported in the *NoTrack* check group in the beta phase (cf. Figs. 2 and 3). The most prevalent privacy problem on modern websites is the plethora of tracking, analysis, and advertising services. These services are usually embedded as JavaScript files that are retrieved from a web server run by the service provider (the so-called “third party”). The privacy and security implications of including third-party services in a website have been widely discussed [22]. Nevertheless, their use is ubiquitous on the modern web.

PrivacyScore enumerates the *hostnames of third parties* that are included when a website is visited. The hostname of every third party is checked against a list of known advertisers and trackers (extracted from the EasyList [4]) to determine which of them are trackers.

A common method for tracking users across multiple sites or over multiple visits to the same site are *HTTP cookies*. PrivacyScore measures how many cookies are being set, and how many of these are owned by third parties (which could use them for cross-page tracking of users). A more modern technique for tracking and re-identifying users is *browser fingerprinting*. Here, characteristics of the browser (e.g., installed plugins and their version) and the device (e.g., available fonts and screen resolution) are being measured to compute a fingerprint. Studies have shown that this fingerprint can often uniquely identify a browser [10,19]. PrivacyScore will check whether the source code of a website contains known patterns of browser fingerprinting.

Furthermore, many websites rely on content distribution networks (CDNs) or load balancing services run by third parties, either for the site itself or in order to include software libraries such as jQuery. CDNs reduce page load times and improve scalability. However, they also pose a risk to the privacy and security of the users because the CDN operators have full access to the unencrypted traffic. This was demonstrated by a recent vulnerability in the Cloudflare denial of service (DoS) protection service, which exposed private information from thousands of websites [6]. Additionally, the CDN companies themselves could track users on all websites that use their services. Therefore, PrivacyScore checks whether the website itself or content from third parties is served from popular CDNs.

Finally, the *geographic location of servers* may be of interest. If the server is hosted under a different jurisdiction than the company itself, additional data

protection rules may apply. While the prevalence of CDNs with colocations in many countries makes determining the geographic location of servers and their associated jurisdictions complicated, a first approximation can be made using a GeoIP database. PrivacyScore checks the location of the web-, mail-, and nameservers used by a website.

4.2 Security Checks

Insecure websites are more likely to suffer from breaches, unintentional data leaks, or data monitoring by rogue hotspots or ISPs. PrivacyScore includes checks for security issues that may indicate privacy problems.

The most straightforward security feature a website can offer is to encrypt HTTP connections using TLS. PrivacyScore checks if the website offers TLS, and if yes, whether unencrypted connection attempts are automatically forwarded to the HTTPS version of the site. It also checks if the website follows established best practices for TLS deployment, if it is vulnerable to known attacks (such as Heartbleed and POODLE), and if the website contains unencrypted content on an encrypted page (*mixed content*). These checks are part of the *EncWeb* check group. Similar checks are run for the primary *mail server* that is listed in the MX domain record of the site, if one exists (*EncMail* check group).

We also check whether a website sets *HTTP security headers* like Content-Security-Policy and X-XSS-Protection that protect users from certain attacks. Together with the checks for unintended information leaks (described in the next paragraph) these checks make up the *Attacks* check group.

The checks for unintended information leaks try to retrieve content from various well-known locations on a web server. Leaks may occur at the locations */server-status/* and */server-info/*, where the Apache server software publishes details about recently served requests (including source IP address) and the current load. We also check whether the operators have forgotten to remove frequently used test scripts (*test.php* and *phpinfo.php*) in the root directory of the server. These scripts may disclose the version and configuration of the server. We also check for the presence of */.git/* and */.svn/* directories. Operators that use version control systems to manage their sites should prevent unauthorized access to these locations. Otherwise, adversaries could retrieve the data stored in these locations to inspect the source code of server-side scripts, which may contain sensitive information such as database access credentials. Finally, we check for the presence of *core dumps*, which are located at */core* and contain the memory of a process at the time it crashed. Core dumps can contain private information and should not be exposed publicly.

Another aspect that is often overlooked in practice is the security of the underlying Domain Name System (DNS) records. If the DNS entries can be compromised, sophisticated phishing attacks may be possible, regardless of all other security features of a website. Accordingly, PrivacyScore checks if the DNS records of a site are protected with *DNSSEC*. PrivacyScore also checks if the mail server of a site uses state-of-the-art authentication techniques by consulting

Sender Policy Framework (SPF) and Domain-based Message Authentication, Reporting and Conformance (DMARC) records in the DNS [17,18].

The final building block in website security is keeping the software up to date. When PrivacyScore retrieves a website it searches for *outdated software* by looking at version banners in headers sent by a server (“Server” HTTP header, SMTP version banner string). It also tries to detect the version of the content management system (CMS) that is used to build the website (“generator” attribute in HTML, and potentially file fingerprinting). Finally, PrivacyScore attempts to detect outdated client-side JavaScript libraries, which has been shown to be a surprisingly prevalent problem today [20].

4.3 Incentives and Actionable Advice for Operators and Users

PrivacyScore increases the visibility of security and privacy issues found on the scanned websites. However, we also want to increase the incentive for operators to improve their systems. Therefore, we plan to enrich the checks with explanations of the resulting security and privacy risks. For instance, a missing or incorrectly set HSTS header means that malicious Wi-Fi access points can eavesdrop on traffic by performing an SSL stripping attack [24].

However, besides creating an incentive for site owners, our advice may also help attackers. For instance, we could create a tangible illustration of the risks resulting from outdated software by listing all relevant CVE entries [2] and reporting whether ready-to-run exploit modules for the Metasploit framework [5] are available. This example demonstrates that legal and ethical ramifications have to be considered to find an acceptable trade-off (cf. Sect. 6).

Furthermore, we want to support operators by offering actionable advice. This includes suggesting alternative software (e.g., using a self-hosted analytics platform like Piwik [27] instead of Google Analytics) and recommending specific configurations for popular operating systems, web servers, and CMSs.

Finally, we will support end users in protecting their privacy by suggesting protection strategies like installing browser add-ons such as PrivacyBadger [11] or uBlock Origin [29] that block advertisements and trackers.

5 Implementation

At the time of writing PrivacyScore is in a public beta phase and under active development (the benchmarking portal is available at <https://privacyscore.org/>). We are implementing the system using a multi-tier software architecture (cf. Fig. 4). Data is collected by a number of dedicated scanning machines (workers), each running a scanning system implemented in Python. Scanning jobs are managed using the distributed task queue Celery [1].

This architecture allows us to scan multiple websites concurrently and to delegate individual scan modules to different machines, making the system horizontally scalable. New machines can be dynamically added to decrease the time that is needed to collect the data for a scan of a site.

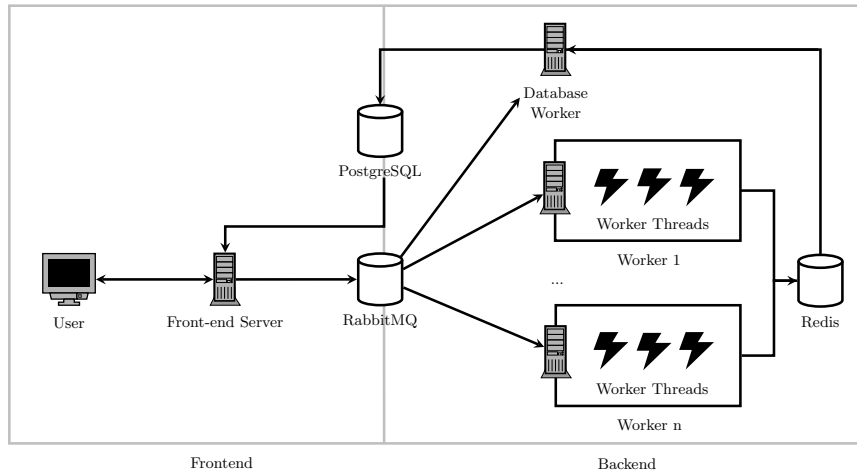


Fig. 4. System Architecture of PrivacyScore

New scanning tasks are queued by the front-end server (written in Python), while the results are aggregated and stored in a PostgreSQL database by a back-end server. For security reasons, the database is not publicly accessible. Instead, the back-end offers selective access to the data via a RESTful API. The front-end views are implemented in Python using the Django web framework [3].

Implemented Checks At the time of writing, we have implemented four scan modules. The first scan module uses the *OpenWPM* [12] framework to collect information directly from a website by visiting it with a remote-controlled Firefox browser (we plan to support different browsers as well as browsers running on mobile operating systems in the future). OpenWPM allows us to observe requests to third-party hosts as well as the cookies they set. We also check whether the web server sent any HTTP security headers.

The second scan module analyzes the security of the encrypted connections to the web server and (if available) to the mail server of a website using the *testssl.sh* script developed by Dirk Wetter [7].

The third scan module performs GeoIP lookups to determine the geographic location of the web and mail servers and the fourth scan module checks whether a website leaks information about its internal configuration.

The remaining checks mentioned in Sect. 4 are currently in development. Furthermore, only the supplied URLs (or the final URL, if the browser is redirected) are checked at the moment. In the future, we may instruct the browser to click on a few random internal links to get a more comprehensive picture of each site.

6 Legal and Ethical Considerations

Automated scanning of websites on the internet poses a number of legal and ethical questions. In the following we will mention important issues, which we are currently discussing with legal academics and practitioners.

6.1 Legal Considerations

In some jurisdictions performing an automated security scan of a website without permission granted by its operators may be illegal or constitute a breach of its terms of service.

This issue can be tackled in various ways. We are currently operating with a manual “opt out” policy. Site operators can contact us and declare that they do not want their sites to be scanned in the future. The respective URLs will be added to a blacklist. However, the last known scan results remain visible on the PrivacyScore website, annotated with a note that the operator has disapproved further scanning.

In principle, the scanner could also be programmed to refrain from scanning a website if it encounters a “Disallow” entry in the file *robots.txt* on the web server, which is a commonly used method for site operators to indicate to search engine robots that some or all parts of their site should not be indexed. However, to the best of our knowledge, none of the existing website scanners respect the rules contained in the *robots.txt* file. Furthermore, the Internet Archive has recently announced to ignore *robots.txt* files on purpose [13]. A viable compromise for PrivacyScore might consist in delegating the question whether to honor statements from *robots.txt* to the list creator.

As executing certain scan modules may violate local laws, operators planning to run an instance of PrivacyScore are advised to evaluate applicable laws before deploying a scanner in their country. A more detailed analysis of the legal issues surrounding PrivacyScore is available in a German version of this paper [21].

6.2 Ethical Considerations

It is our intent to help site operators and users, but not adversaries. However, some of our checks and advice provide information that may be useful for attackers, i. e., PrivacyScore is a *dual-use tool*. Therefore, we are carefully designing what checks we implement and how we present the results to the user.

Scanning a website generates traffic (typically less than one megabyte) and temporarily increases its CPU load. Therefore, each scan incurs some cost for the owners of a website. We believe this to be ethically acceptable, because the purpose of having a publicly available website is to have visitors that download it. Furthermore, we consider the public good of providing assessments to outweigh the typically negligible costs of each scan for the website operators.

However, we have to ensure that we do not induce a critical load on a website, which might result in denial of service. This is especially relevant for the SSL checks, which generate a large number of connections. To prevent too frequent

scans we implement rate-limiting controls ensuring that the scanned sites cannot be overloaded maliciously.

7 Conclusion

Running a secure and privacy-respecting website has become a demanding task. On the one hand, website owners must constantly adapt their security measures to novel threats. On the other hand, they have to make the right decisions during the design and operation of their site in order to safeguard the privacy of their users. Today, many sites offer poor security and privacy, either because site owners are unwilling to cover the additional costs or because privacy measures are in conflict with their business model.

We believe that some site owners would make more privacy-conscious decisions if they had an incentive to do so. Our project PrivacyScore aims to create such incentives by generating transparency and publicity. It allows anyone to set up a benchmark for a peer group of websites. Security and privacy features of the sites in a group are automatically analyzed, resulting in a public, regularly updated, and user-controllable ranking.

PrivacyScore is open source software and we plan to release all collected datasets for research purposes. Besides running it as a public service, PrivacyScore can also be deployed in-house. We hope that it will become a useful tool for DPAs that are faced with the task of enforcing a large number of regulatory requirements specified in the General Data Protection Regulation (GDPR).

Acknowledgments This work has been co-funded by the DFG as part of project C.1 within the RTG 2050 “Privacy and Trust for Mobile Users”. The authors are grateful to Marvin Hebisch and Nico Vitt, who implemented a prototype, the attendants of the PET-CON 2017.1 workshop, and members of Digitalcourage e. V. for their valuable suggestions.

References

1. Celery: Distributed task queue (2017), <http://www.celeryproject.org/>
2. Common Vulnerabilities and Exposures (2017), <https://cve.mitre.org/>
3. Django web framework (2017), <https://www.djangoproject.com/>
4. EasyList (2017), <https://easylist.to/>
5. Metasploit Penetration Testing Software (2017), <https://www.metasploit.com/>
6. Cloudflare: Incident report on memory leak caused by Cloudflare parser bug (2017), <https://blog.cloudflare.com/incident-report-on-memory-leak-caused-by-cloudflare-parser-bug/>
7. D. Wetter: testssl.sh (2017), <https://testssl.sh/>
8. dataskydd: Kommunundersökning (2016), <https://dataskydd.net/kommuner-201611/>
9. dataskydd: Webbkoll (2017), <https://webbkoll.dataskydd.net/en/>
10. Eckersley, P.: How Unique Is Your Web Browser? In: Proceedings of the 10th International Symposium on Privacy Enhancing Technologies (PETS 2010). LNCS, vol. 6205, pp. 1–18. Springer (2010)

11. EFF: Privacy Badger (2017), <https://eff.org/privacybadger>
12. Englehardt, S., Narayanan, A.: Online tracking: A 1-million-site measurement and analysis. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). pp. 1388–1401. ACM (2016)
13. Graham, M.: Robots.txt meant for search engines don't work well for web archives (2017), <https://blog.archive.org/2017/04/17/robots-txt-meant-for-search-engines-dont-work-well-for-web-archives/>
14. High-Tech Bridge: SSL/TLS Server Test (2017), <https://www.htbridge.com/ssl/>
15. Holz, R., Amann, J., Mehani, O., Kâafar, M.A., Wachs, M.: TLS in the Wild: An Internet-wide Analysis of TLS-based Protocols for Electronic Communication. In: Proceedings of the 23rd Annual Network and Distributed System Security Symposium (NDSS 2016). The Internet Society (2016)
16. Khandelwal, S.: 'Web Of Trust' Browser Add-On Caught Selling Users' Data (2016), <http://thehackernews.com/2016/11/web-of-trust-addon.html>
17. Kitterman, S.: Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208 (2014)
18. Kucherawy, M., Zwicky, E.: Domain-based Message Authentication, Reporting, and Conformance (DMARC). RFC 7489 (2015)
19. Laperdrix, P., Rudametkin, W., Baudry, B.: Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In: Proceedings of Symposium on Security and Privacy (S&P 2016). pp. 878–894. IEEE (2016)
20. Lauinger, T., Chaabane, A., Arshad, S., Robertson, W., Wilson, C., Kirde, E.: Thou Shalt Not Depend on Me: Analysing the Use of Outdated JavaScript Libraries on the Web. In: Proceedings of the 24th Annual Network and Distributed System Security Symposium (NDSS 2017). The Internet Society (2017)
21. Maass, M., Laubach, A., Herrman, D.: PrivacyScore: Analyse von Webseiten auf Sicherheits- und Privatheitsprobleme. In: Eibl, M., Gaedke, M. (eds.) INFORMATIK 2017. pp. 1049–1060. Gesellschaft für Informatik, Bonn (2017)
22. Mayer, J.R., Mitchell, J.C.: Third-Party Web Tracking: Policy and Technology. In: Proceedings of Symposium on Security and Privacy (S&P 2013). pp. 413–427. IEEE (2012)
23. Mayer, W., Zauner, A., Schmiedecker, M., Huber, M.: No Need for Black Chambers: Testing TLS in the E-mail Ecosystem at Large. In: Proceedings of the 11th International Conference on Availability, Reliability and Security (ARES 2016). pp. 10–20. IEEE (2016)
24. Moxie Marlinspike: sslstrip (2017), <https://moxie.org/software/sslstrip>
25. Mozilla: Lightbeam (2017), <https://www.mozilla.org/en-US/lightbeam/>
26. Mozilla: Observatory (2017), <https://observatory.mozilla.org/>
27. Piwik: Piwik Free Web Analytics Software (2017), <https://piwik.org/>
28. Qualys: SSL Server Test (2017), <https://www.ssllabs.com/ssltest/>
29. Raymond Hill: uBlock Origin (2017), <https://github.com/gorhill/uBlock>
30. S. Helme: Publishing my daily crawler data for wider analysis (2017), <https://scotthelme.co.uk/alexa-top-1-million-analysis-feb-2017>
31. S. Helme: SecurityHeaders.io (2017), <https://securityheaders.io/>
32. Starov, O., Nikiforakis, N.: Extended Tracking Powers: Measuring the Privacy Diffusion Enabled by Browser Extensions. In: Proceedings of the 26th International Conference on World Wide Web (WWW 2017). ACM (2017)