

Measuring for privacy: From tracking to cloaking

Nayanamana Samarasinghe

A Thesis
in
The Concordia Institute
for
Information Systems Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of
Doctor of Philosophy (Information and Systems Engineering) at
Concordia University
Montréal, Québec, Canada

September 2022

© Nayanamana Samarasinghe, 2022

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: **Nayanamana Samarasinghe**

Entitled: **Measuring for privacy: From tracking to cloaking**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Information and Systems Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Sivakumar Narayanswamy

_____ External Examiner
Dr. Rei Safavi-Naini

_____ External to Program
Dr. Emad Shihab

_____ Examiner
Dr. Amr Youssef

_____ Examiner
Dr. Walter Lucia

_____ Thesis Supervisor
Dr. Mohammad Mannan

Approved by _____
Dr. Zachary Patterson, Graduate Program Director

Nov 17, 2022 _____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Measuring for privacy: From tracking to cloaking

Nayanamana Samarasinghe, Ph.D.

Concordia University, 2022

We rely on various types of online services to access information for different uses, and often provide sensitive information during the interactions with these services. These online services are of different types; e.g. commercial websites (e.g., banking, education, news, shopping, dating, social media), essential websites (e.g., government). Online services are available through websites as well as mobile apps. The growth of web sites, mobile devices and apps that run on those devices, have resulted in the proliferation of online services. This whole ecosystem of online services had created an environment where everyone using it are being tracked. Several past studies have performed privacy measurements to assess the prevalence of tracking in online services. Most of these studies used institutional (i.e., non-residential) resources for their measurements, and lacked global perspective. Tracking on online services and its impact to privacy may differ at various locations. Therefore, to fill in this gap, we perform a privacy measurement study of popular commercial websites, using residential networks from various locations.

Unlike commercial online services, there are different categories (e.g., government, hospital, religion) of essential online services where users do not expect to be tracked. The users of these essential online services often use information of extreme personal and sensitive in nature (e.g., social insurance number, health information, prayer requests/confessions made to a religious minister) when interacting with those services. However, contrary to

the expectations of users, these essential services include user tracking capabilities. We built frameworks to perform privacy measurements of these online services (include both web sites and Android apps) that are of different types (i.e., governments, hospitals and religious services in jurisdictions around the world). The instrumented tracking metrics (i.e., stateless, stateful, session replaying) from the privacy measurements of these online services are then analyzed.

Malicious sites (e.g., phishing) mimic online services to deceive users, causing them harm. We found 80% of analyzed malicious sites are cloaked, and not blocked by search engine crawlers. Therefore, sensitive information collected from users through these sites is exposed. In addition, underlying Internet-connected infrastructure (e.g., networked devices such as routers, modems) used by online users, can suffer from security issues due to non-use of TLS or use of weak SSL/TLS certificates. Such security issues (e.g., spying on a CCTV camera) can compromise data integrity, confidentiality and user privacy.

Overall, we found tracking on commercial websites differ based on the location of corresponding residential users. We also observed widespread use of tracking by commercial trackers, and session replay services that expose sensitive information from essential online services. Sensitive information are also exposed due to vulnerabilities in online services (e.g., Cross Site Scripting). Furthermore, a significant proportion of malicious sites evade detection by security/search engine crawlers, which may make such sites readily available to users. We also detect weaknesses in the TLS ecosystem of Internet-connected infrastructure that supports running these online services. These observations require more research on privacy of online services, as well as information exposure from malicious online services, to understand the significance of privacy issues, and to adopt appropriate mitigation strategies.

Acknowledgments

While I was pursuing the *Master of Engineering, Information Systems Security* degree program at Concordia University (Montreal, Canada), I happen to work on a course project under the supervision of Dr. Mohammad Mannan. Due to his meticulousness, support and guidance, the outcome of this work was published during a very short time. This initial interactions I had with Dr. Mohammad Mannan, inspired me to believe in myself, and decide to pursue my Ph.D. studies under his supervision. Therefore, I would like to express my deepest gratitude and appreciation to Dr. Mohammad Mannan, for providing me an opportunity to move into academic research (after having worked many years in the industry).

I am also grateful for all members of the *Madiba Security Research Group*, and all my colleagues at *Concordia Institute for Information Systems Engineering (CIISE)*, for their support during my tenure at Concordia University.

I also would like to express my gratitude to all my family members and friends, who have been supporting me in many forms, and understanding the devoted nature of Ph.D. studies, without which this thesis would not have been possible.

Contents

List of Figures	xii
List of Tables	xv
1 Introduction	1
1.1 Motivation	1
1.2 Thesis statement	4
1.3 Objectives and contributions	4
1.4 Ethical considerations	7
1.5 Related publications	8
1.6 Outline	9
2 Background	10
2.1 Technologies used in tracking online services	10
2.1.1 Tracking technologies	10
2.1.2 Content included by first and third party domains	12
2.2 Techniques of tracking websites	12
2.3 Tracking detection and privacy measurement tools	14
2.4 Luminati - Residential proxy service	15
2.5 Web cloaking	16

3	Tracking on popular sites from a global perspective	18
3.1	Introduction	18
3.2	Related work	21
3.3	Methodology	25
3.3.1	Country and first-party site selection	26
3.3.2	Tracker identification and prominence	28
3.3.3	Dynamicity of trackers	29
3.3.4	Ethical issues	31
3.3.5	Limitations	31
3.4	Trackers vs. geolocation	33
3.5	Overall tracker prominence	35
3.6	Cookie validity durations	37
3.7	Factors other than geolocation	39
3.7.1	Internet speed	39
3.7.2	Censorship	41
3.7.3	Browser user-agents vs. tracking	42
3.8	Data protection laws vs. tracking	43
3.9	Recommendations	45
3.10	Summary	45
4	Privacy analysis of government websites and mobile apps	47
4.1	Introduction	47
4.2	Related work	51
4.3	Methodology	54
4.3.1	Collecting government sites and apps	54
4.3.2	Measurement of trackers on government sites	58
4.3.3	Malicious government and tracking domains	59

4.3.4	Android apps analysis	59
4.3.5	Ethical considerations and limitations	61
4.4	Results: Government websites	62
4.4.1	Third-party tracking scripts	62
4.4.2	Trackers on EU and California government sites	66
4.4.3	Third-party cookies	68
4.4.4	Fingerprinting APIs	69
4.4.5	Government sites and tracking domains flagged as malicious	70
4.4.6	Privacy policies in government websites with trackers	70
4.4.7	Foreign-hosted government sites	71
4.5	Results: Government Android apps	73
4.6	Discussion	75
4.7	Recommendations	77
4.8	Summary	78
5	Privacy analysis of hospital websites	79
5.1	Introduction	79
5.2	Related work	82
5.3	Methodology	84
5.3.1	Collecting hospital websites	85
5.3.2	Web privacy measurements	86
5.3.3	Session replay scripts	87
5.3.4	Detecting malicious domains	88
5.3.5	Limitations	88
5.4	Results	88
5.4.1	Session replay	89
5.4.2	Domains flagged as malicious	90

5.4.3	Websites using HTTP and login forms	94
5.4.4	Third-party tracking scripts	94
5.4.5	Third-party tracking cookies	97
5.4.6	Fingerprinting APIs	99
5.5	Recommendations	99
5.6	Summary	101
6	Privacy analysis of religious websites and mobile apps	102
6.1	Introduction	102
6.2	Related work	106
6.3	Methodology	107
6.3.1	Collecting religious websites and Android apps	107
6.3.2	Web privacy measurements	109
6.3.3	Session replay scripts and chatbot services in religious websites	109
6.3.4	Security issues in religious websites	110
6.3.5	Android app analysis	111
6.3.6	Ethical considerations and limitations	112
6.4	Results: Religious websites	113
6.4.1	Session replay and chatbot services	113
6.4.2	Religious sites with security issues	115
6.4.3	Religious sites flagged as malicious	116
6.4.4	Analysis HTTP/HTTPS traffic from religious websites	117
6.4.5	Third-party tracking scripts	118
6.4.6	Third-party tracking cookies	120
6.5	Results: Religious Android apps	120
6.6	Recommendations	122
6.7	Summary	123

7	Cloaking behaviors of malicious websites	124
7.1	Introduction	124
7.2	Related work	128
7.3	Methodology	130
7.3.1	Generating squatting domains	130
7.3.2	Our crawler	134
7.3.3	Analyzer	135
7.3.4	Limitations	140
7.4	Issues during crawling	140
7.5	Ground truth	143
7.6	Dissimilarities	146
7.6.1	Link dissimilarities	146
7.6.2	Header dissimilarities	146
7.6.3	Content dissimilarities	147
7.6.4	Image dissimilarities	150
7.6.5	Comparison of results of cloaking detection techniques	151
7.7	Discussion	153
7.7.1	Dynamicity in squatting sites	153
7.7.2	Malicious squatting domains generated from DNSTwist	153
7.7.3	Relevance of seed domains	154
7.7.4	Detection of cloaked sites by blacklists	154
7.7.5	Variations of cloaking in different device types	156
7.7.6	User-agent vs. referrer cloaking	157
7.7.7	Relevance of type of squatting domains for cloaking	158
7.7.8	Relevance of cloaking by other factors	158
7.8	Recommendations	159

7.9	Summary	159
8	Longitudinal study of the TLS ecosystems in networked devices	162
8.1	Introduction	162
8.2	Related work	165
8.3	Methodology and device info	168
8.4	Analysis and results	171
8.4.1	Prevalence of weak security practices	171
8.4.2	Changes in the use of weak cryptographic primitives	179
8.4.3	Changes in the use of strong cryptographic primitives	181
8.5	Disclosure	183
8.6	Limitations	186
8.7	Recommendations	187
8.8	Summary	189
9	Conclusion and future work	191
	Bibliography	196

List of Figures

1	Our system setup - Measurement of tacking residential users from a global perspective.	26
2	First-party percentages for script-based trackers across 15 countries.	33
3	First-party percentages for tracking org.	34
4	Prominence of tracker scripts across 56 countries.	35
5	Average of top trackers in different regions.	36
6	Prominence of tracker scripts: Alexa global sites vs. Twitter URLs.	36
7	The number of cookies vs. validity period.	38
8	Internet speed vs. tracker prominence.	40
9	KW Ranks highlighting errors of HTTP/S requests.	40
10	Censorship rating vs. tracker prominence.	41
11	Tracking scripts vs. user-agents.	43
12	Tracking cookies vs. user-agents.	43
13	Overall methodology - Government websites/Android apps.	56
14	Top-10 known third-party tracking script sources on government sites.	63
15	Heatmap of percentage of government websites with known tracking scripts in different countries.	64
16	Heatmap of percentage of government websites with known tracking cookies in different countries.	64

17	Proportions of third-party scripts (known trackers vs. unknown trackers) on government sites per region.	65
18	Known trackers on government sites by region.	68
19	Overview of our methodology - Hospital websites.	84
20	Percentage of hospital websites with known tracking scripts/cookies.	95
21	Top-10 known tracking scripts on hospital sites.	96
22	Proportions of third-party scripts in different categories included on hospital websites by region.	96
23	Top-10 known tracking cookies on hospital sites.	97
24	Proportions of third-party cookies in different categories set on hospital websites by region.	98
25	Overview of our methodology.	108
26	Top-10 known third-party tracking script sources on religious sites.	119
27	Top-10 known third-party tracking cookies set on religious sites.	119
28	Our system setup - Measurement of cloaking behaviours in potentially malicious websites.	132
29	Cloaking differences for site: 000webhostapp	149
30	Cloaking differences for site: homdedepot.com	150
31	Cloaking differences for site: bodybuildinh.com	150
32	Top 7 seed domains of the corresponding cloaked domains with 8-13 permutations.	155
33	Cloaking by type of squatting domain.	158
34	Hashing algorithms	173
35	Signature algorithms	173
36	Key lengths (RSA)	173
37	Encryption algorithms	173

38	SSL/TLS protocol versions	173
39	Unique certificates: Alexa-1M vs. devices.	177

List of Tables

1	List of regions and countries.	27
2	Number of tracking cookies with validity periods (EU).	38
3	Domains of top-10 tracking cookies registered in EU countries.	39
4	List of regions and government website counts.	55
5	Cookie validity periods on EU government sites.	67
6	Cookie validity periods on California government sites.	67
7	The top-10 known tracking cookies and their expiry periods.	69
8	Tracking scripts included from potentially malicious domains.	71
9	Tracking cookies set by potentially malicious domains.	71
10	Exposure of sensitive information from Android apps.	75
11	Examples of private/sensitive information collected by <i>Yandex</i> session replay service.	91
12	Examples of private/sensitive information collected by session replay service in EU Countries.	91
13	Session replay services on hospital websites.	92
14	Examples of hospital websites with session replay services.	92
15	Known tracking scripts hosted on potentially malicious domains that are flagged by VirusTotal.	93
16	Known tracking cookies set by potentially malicious domains that are flagged by VirusTotal.	93

17	The top-10 known tracking cookies and their expiry periods.	98
18	Use cases for information leakage with session replay services (SRS) on religious sites.	114
19	Examples of security issues in religious websites.	116
20	Top-5 religious websites with most leakages of personal/sensitive information over HTTP	117
21	The top-10 known tracking cookies and their expiry periods	120
22	Squatting domain lists used in our experiments	130
23	Top 5 errors encountered during crawling	141
24	Failures while crawling	142
25	Sites blocked from GooglebotUA	143
26	SiteReview categorization of malicious squatting domains - cloaked vs. non-cloaked	144
27	Header dissimilarities.	147
28	Failures from GooglebotUA	147
29	Combo-squatting domains served via HTTP/HTTPS	148
30	Projecting results of cloaked squatting domains to corresponding seed domains (i.e., squatting domain vs. seed)	155
31	Variation in cloaking between device types	156
32	Variation between user-agent vs. referrer cloaking	156
33	Type-wise device distribution	170
34	Percentages of weak cryptographic primitives in devices.	174
35	Top-10 organizations issuing device certificates.	176
36	Changes in weak cryptographic primitives in devices/Alexa-1M sites. . . .	178
37	Changes in vulnerability – an increase in devices supporting vulnerable ICS protocols is apparent with time.	180

38 Top-10 countries with known private keys included in devices 181

39 Devices groupings with a known private key as tagged in Censys 182

40 Top-10 manufactures of devices with a known private key as tagged in
Censys. 182

41 Changes in strong cryptographic primitives in devices/Alexa-1M sites. . . . 184

42 Top-5 manufactures with vulnerable devices. 186

Chapter 1

Introduction

1.1 Motivation

The early web only allowed access to static content that did not maintained state between multiple client requests; i.e., web pages containing static HTML. Over time, the web re-shaped to be more consumer-oriented with the introduction of web technologies that facilitate rendering dynamic content (e.g., cookies, JavaScript), and various browsers (e.g., Netscape, Internet Explorer, Firefox) started supporting those technologies. Subsequently, with the commercialization of the web, various online services were created for financial gain [279, 199] — e.g., e-commerce platforms created as a result of the dot-com bubble (e.g., Amazon), social media platforms (e.g., Facebook), micro-blogging and social networking services (e.g., Twitter). The commercialization of the web have also allowed a personalized experience for its users, which in turn provide third parties (e.g., ads/analytics services) to infer browsing behaviours for tracking users. Mobile apps have expanded the landscape of online services. As more services are transitioned to online space, it created more opportunities to track users, and to build better user profiles by correlating users' browsing behaviours across various types of online services. Tech giants (e.g., Google, Facebook) track users to provide a better user experience [102], but at the same time they

also profile users for monetization purposes. Although users are aware of such practices, they still accept such inevitable tracking, to consume services (e.g., Google maps) that are offered at no cost. Tracking online services is not only limited for rendering advertisements that are used to target users with personalized content, but have also used for government surveillance programs [98].

Lack of global perspective in past privacy measurement studies. Several past measurement studies [97, 98, 173, 34] have focused on various aspects of tracking on commercial online services, and its impact on privacy of users. Englehardt et al. [97] measured the extent of third-party trackers (e.g., script/cookies) on Alexa Top-1M websites, and found many trackers on these sites, where most of those trackers were from market leaders (e.g., Google, Facebook). In addition, advanced fingerprinting methods were used to passively track users of these websites. Lerner et al. [173] performed a longitudinal study of web tracking behaviours (from 1996 to 2016), and found an increase in third-party tracking on the web in prevalence and complexity, and their findings can trigger technical and policy discussions surrounding tracking. Binns et al. [34] studied the prevalence of third-party tracking on 959,000 apps from US and UK Google Play stores, and found most apps included third party tracking; their distribution of trackers were long-tailed with several highly dominant trackers. Englehardt et al. [97] and Binns et al. [34] observed a large number of trackers, on news related websites and apps, respectively. Privacy measurements of these studies focused on tracking of institutional users who typically do not use residential networks to interact with online services. Also, past privacy measurement were carried out from a particular location [97] or from a few jurisdictions [113]. Therefore, a clear understanding on web tracking from a global perspective is still lacking.

Lack of privacy measurements of essential online services. Following the commercialization of the web, various essential services (e.g., governments, health services, religious communities) have transitioned to online space, to expand their services, increase efficiency

and to reach a larger audience. The recent COVID-19 pandemic also accelerated the moving of these essential services to online space, due to limitations in physically consuming such services. In contrast to tracking in commercial online services, users do not expect commercial trackers on these essential online services. Such tracking on essential online services is quite revealing, due to the sensitive nature of information that is used to interact with these online services. Disclosure of sensitive information from these essential online services (if any), can even lead into risks such as discrimination and social stigma. However, tracking on these essential online services has not been comprehensively studied. If user profiling from tracking of commercial online services are integrated with that of these essential online services, it can cause adverse consequences to users — Canadian commercial data brokers collect deidentified patient data from pharmacies, private drug insurers, federal government and medical clinics without the consent of patients [47]. This practice of commercializing unconsented patient data, risks the loss of anonymity of data, use of patient data for surveillance/marketing and discrimination.

Cloaking behaviours of malicious websites. As traditional services are transitioned to online space, adversaries launch malicious sites to collect sensitive information of users (e.g., user credentials, personal information) — e.g., phishing sites that mimic web sites of popular services (e.g., financial institutions, governments), spyware to gather browsing activities and login credentials of users. These adversaries use various techniques (e.g., cloaking) to avoid detection of these sites by security/search engine crawlers. If the mechanisms in place to detect these malicious sites are evaded, adversaries will be successful in collecting sensitive information of users by deceiving them, impacting the privacy of users.

Security issues leading to privacy exposures. Online services can also suffer from various security issues. These security issues can lead into jeopardizing the protection of data collected by various automations, or provided by users during interactions with those online services. This will result in users losing control of their own data, impacting their privacy.

These security issues are of many forms — e.g., malicious dependencies such as JavaScript libraries, Android SDKs; security issues in online services due to bad coding practices; vulnerabilities in the underlying infrastructure (comprised of various Internet-connected devices).

1.2 Thesis statement

Our research is primarily focused on privacy measurements of online services that are not comprehensively explored in the past (e.g., essential online services), or studied only from a fixed or limited number of geographical locations. In addition, we also look into privacy issues that are often caused by various security issues (e.g., web cloaking, weaknesses in the use of TLS of the underlying infrastructure). As part of our research focus, we explore the following research questions:

Question 1. Is there a variation in tracking residential users (cf. institutional users) by commercial trackers on popular websites, from a global perspective?

Question 2. Do commercial trackers perform tracking on essential online services used by users, and what is the significance of such tracking?

Question 3. What techniques can be used to identify cloaked malicious websites that mimic legitimate popular websites (e.g., websites of popular brands)?

Question 4. How vulnerable is the TLS ecosystem for Internet connected devices (due to weaknesses in TLS certificates), that support the running of various online services used by end users (cf. popular web sites)?

1.3 Objectives and contributions

This research aims in providing multiple frameworks for performing privacy measurements by extending the contributions made from past work [97, 282]. The privacy measurements

in this thesis are related to profiling of users, with tracking on online services, and leakage of personal information. We make the following contributions; contributions 1, 2, 3 and 4, assist in answering research questions 1, 2, 3 and 4 (see Section 1.2), respectively.

1. We build a framework to perform privacy measurements of websites by proxying residential networks in various locations around the world, to measure tracking of *residential users*. Using this framework, we collect metrics pertaining to stateless (cookies, JavaScript) and stateful (fingerprinting) forms of tracking, and analyze the prominence of commercial trackers on popular commercial websites from a global perspective. We observe a significant variation of trackers on first party websites between the analyzed countries — websites in United Kingdom and Armenia have a higher prominence of trackers, while Ethiopia and Iran have the least. Countries subjected to censorship, attract less trackers on popular websites. Also, a significant number of cookies have a larger validity period (i.e., greater than 1 year) across the analyzed countries.
2. We implement a framework to perform privacy measurements of essential online services (e.g., websites and Android apps) used by online communities. This framework can be customized to adapt various types on essential online services. The essential online services that we collect for privacy measurements, are done on a best effort basis, and the sources from which those online services are extracted, may not cover all available online services, of the selected essential service types. It can instrument both stateful and stateless tracking metrics from privacy measurements of websites, identify information exposure from session replay, perform static and dynamic analysis of Android apps, scan websites/Android apps and included third party domains with VirusTotal to find those that are flagged as malicious. The information collected from this framework is analyzed to identify tracking and exposure of sensitive information. We observe that some analyzed essential websites send sensitive

information (e.g., user credentials, email address, phone number) to remote session replay providers. A large number of third party scripts and cookies are set on essential websites by Google. Some of the cookies set on these websites expire after many years (e.g., in year 9999). Similar to the essential websites, SDKs included on a essential Android apps to track users, are largely dominated by Google. In addition, some essential apps send sensitive information (e.g., login information, device identifiers) over plain HTTP, to remote third parties.

3. We develop a crawler to identify cloaking in malicious websites. This crawler uses VirusTotal to identify malicious domains, and a set of heuristics to determine the presence of cloaking (to avoid detection by security/search engine crawlers) in websites. The crawler supports websites with both static/dynamic content. As part of this work, we also build an automated framework to extract live typo-squatting and combo-squatting domains that are fed to the crawler, to determine the extent of cloaking in the malicious websites hosted on the extracted typo-squatting/combo-squatting domains. Our crawler extracts features based on links, headers, content and screenshots of analyzed web pages, and formulate dissimilarities between a site viewed from a Chrome browser and Googlebot. Using our heuristics, we found 80% of cloaked sites were malicious — 46% of the cloaked malicious sites, persist even after 3 months.
4. We study the vulnerabilities in the TLS ecosystem for Internet connected devices (specifically on SSL/TLS certificate weaknesses). For this work, we collect certificates pertaining to TLS deployments of networked devices from the Censys search engine [88, 51]. We identify different types of networked devices using annotations in Censys search engine results. Thereafter, we extract SSL/TLS parameters (e.g., cipher suite, SSL/TLS protocol version, RSA key length) from the collected certificates, to identify those devices that are prone to TLS vulnerabilities. From our

findings, we observe that, despite the sharp increase of devices that continue to adopt TLS, still a large number of devices use weak TLS parameters.

The source code of frameworks and data sets pertaining to the different studies of this thesis are shared at <https://github.com/nayanamana/PhD>, for the benefit of future research.

1.4 Ethical considerations

Prior to the commencement of all studies that pertain to tracking and privacy measurements, we reach out to the internal Research Ethics Unit of our university (Concordia University, Montreal, Canada), and explain them our studies (including test methodologies). In addition, during our experiments of different studies, at each stage, we keep the Research Ethics Unit informed of our findings, and how we handle sensitive data.

During our study relating to tracking on popular sites from different residential machines accessed through a proxy, we were unaware of sites blocked/censored from the respective jurisdictions, which may cause legal problems to users (of whom, we do not have contact information). With studies relating to essential Android apps, we do not use the sensitive information (e.g., user identifiers and passwords) extracted from static and dynamic analyses of Android apps for any intrusive validations that may have an impact to the privacy of users. In addition, we did not retain any data from exposed Firebase databases. Furthermore, we limit the security evaluation of religious online services due to possible legal and ethical issues.

For each of the studies, the Research Ethics Unit of our University, did not object to our test methodologies, and did not require us to go through a full ethics evaluation.

1.5 Related publications

All research topics in this thesis have been peer-reviewed. The corresponding publications are listed below; (1), (2), (3), (4) and (5) are also co-authored by other students; (1) and (2) are not included in this thesis. My contributions for (5) include assisting with setting up the test environment, synthesizing the results and writing. For (4), I was involved in identifying session replay services included in hospital websites, synthesizing results collected from privacy measurements, and writing. For (3), my contributions include, all the work other than the experiments pertaining to religious Android apps. Contributions for (1) and (2) include the literature review on related work, synthesizing results from information collected, and writing.

1. B. Tejaswi, N. Samarasinghe, S. Pourali, M. Mannan, and A. Youssef. Leaky Kits: The Increased Risk of Data Exposure from Phishing Kits. In APWG Symposium on Electronic Crime Research (eCrime'22), Online, Nov. 2022.
2. S. Pourali, N. Samarasinghe and M. Mannan. Hidden in plain sight: Exploring encrypted channels in Android apps. In ACM Conference on Computer and Communications Security (CCS'22), Los Angeles, USA, Nov. 2022.
3. N. Samarasinghe, P. Kapoor, M. Mannan, and A. Youssef. No salvation from trackers: Privacy analysis of religious websites and mobile apps. In International Workshop on Data Privacy Management (DPM'22), Copenhagen, Denmark, September 2022.
4. X. Yu, N. Samarasinghe, M. Mannan, and A. Youssef. Got sick and tracked: Privacy analysis of hospital websites. In International Workshop on Traffic Measurements for Cybersecurity (WTMC'22), Genova, Italy, pp. 278-286, June 2022.
5. N. Samarasinghe, A. Adhikari, M. Mannan, and A. Youssef. Et tu, brute? Privacy

- analysis of government websites and mobile apps. In Proceedings of the ACM Web Conference (TheWebConf'22), Online, pp. 564-575, Apr. 2022.
6. N. Samarasinghe and M. Mannan. On cloaking behaviors of malicious websites. *Computers & Security*, 101:102114, 2021.
 7. N. Samarasinghe and M. Mannan. Towards a global perspective on web tracking. *Computers & Security*, 87:101569, 2019.
 8. N. Samarasinghe and M. Mannan. Another look at TLS ecosystems in networked devices vs. web servers. *Computers & Security*, 80:1–13, 2019.
 9. N. Samarasinghe and M. Mannan. Short paper: TLS ecosystems in networked devices vs. web servers. In *Financial Cryptography and Data Security (FC'17)*, Malta, pp. 533-541, Apr. 2017.

1.6 Outline

The rest of the thesis is organized as follows. Chapter 2 introduces background material relating to privacy measurements. Chapter 3 explores tracking of popular commercial sites from the perspective of residential users in a global perspective. Chapters 4, 5, 6 contain privacy measurement studies, for essential online services pertaining to governments, hospitals and religions, respectively. Chapter 7 (cloaking in malicious websites to evade detection) and Chapter 8 (a longitudinal study of vulnerabilities in TLS deployments of networked devices compared to that of popular websites) focus on security issues that will eventually have an impact on the privacy of online users. Because of the wide scope of research topics covered in this thesis, we have the discussion on related work in each individual chapter instead of a dedicated chapter of the thesis. Chapter 9 mentions future work and highlights the concluding remarks of this thesis.

Chapter 2

Background

This chapter presents different concept and technologies used throughout the subsequent chapters of this thesis.

2.1 Technologies used in tracking online services

In this section, we describe web technologies that make tracking on online services possible.

2.1.1 Tracking technologies

Various technologies that are used in websites and mobile apps to track users are discussed in this section.

JavaScript. Tracking scripts written in JavaScript are included in web pages to provide a dynamic browsing behaviour with websites (e.g., loading content without reloading a web page, web animation, input validation) for users. Besides, JavaScript can also log data about user's browsing behaviours that can be used for personalization, analytic and ad tracking. Possible functions of JavaScript tracking code include event tracking (e.g.,

keystrokes, mouse clicks) with web page elements, control of clicks through advertisements and tracking of marketing campaigns.

Cookies. HTTP protocol is stateless and handles each HTTP request independently from other HTTP requests. However, a web site requires the capability to identify a user having multiple interactions with it through a specific browser. This is necessary for both legitimate (e.g., user authentication) and tracking purposes. In order to serve these purposes, cookies are used. HTTP cookies are a small piece of information stored in the user browser. There are two types of cookies [80] — HTTP cookies and JavaScript cookies. Using the *Set-Cookie* HTTP response header, a domain can set a HTTP cookie in the user's browser. This cookie is defined by the triplet (host, key, value), where host refers to the domain that sets the cookie. The browser may store the cookie and send it back to the same server with later requests (i.e., server can distinguish if two requests come from the same browser). A time period can be set after which the cookie should not be sent. Also, additional restrictions [80] are allowed to set a cookie for a specific domain and a path, to limit where the cookie is sent. Similarly, *document.cookie* property is used to create JavaScript cookies (programmatically). Every cookie is stored in the browser with an associated domain and path, so that every new HTTP request sent to the same domain and path gets a cookie associated to it, that is attached to the request. Based on the life time of a cookie, it can be categorized as a *session cookie* (cookie is deleted when the current browser session ends) or a *persistent cookie* (cookie is valid until the date specified by the *Expires* attribute, or after a period of time specified by the *Max-Age attribute*).

Referer header. The HTTP referer [79] is used to identify the address of the webpage from which the web resource is requested (i.e., from where the request was originated). Instead of using third party trackers to uniquely identify a user across websites, it is possible to identify which website the user is visiting, using the HTTP referrer header. This also allows recreating user's browsing history. By default, the browser sends the Referer field

in every HTTP request. Third parties may also use other techniques (e.g., JavaScript calls to `document.location`) to identify the visited page.

Tracking SDKs. Software Development Kits (SDKs) [127] are Application Programming Interfaces (APIs) that are in the form of on-device libraries of reusable functions used to interface to a particular programming language. These SDKs are included as small code libraries in mobile apps for serving advertisements, provision of analytics, push notifications and tracking user activity in mobile devices. Tracking SDKs are developed for different mobile platforms (e.g., Android, iOS).

2.1.2 Content included by first and third party domains

Websites are comprised of first party content and third party content. The latter includes content from advertisements, web analytic scripts, social widgets and images. We define a *first party domain* as the domain of the website, while *third party domains* are domains that service third party content on the website. First party cookies are set by the first party domains or programmatically via scripts running in the context of the website. These first party cookies are used to track users within the same website. In contrast, third party cookies set by third party domains, allow third parties to track users across websites [235].

2.2 Techniques of tracking websites

In this section, we present different tracking techniques used in websites.

Cross-site tracking. Third party cookies set by a tracking domain can be used to track user's activity across websites [36]. When a user directly visits a website that includes content from a third party tracker (e.g., a social share plugin from *Facebook*), the browser will send a request to fetch its content. The third party tracker will then set a third party cookie as part of the response on user's browser. When the user visits a different website

that include the content of the same third party tracker, the other website will receive the cookie set on the original website, linking the user's activity across both websites.

Cookie syncing. According to the Same Origin Policy (SOP) [300], cookies set by user's browser is only accessible to the domain that sets it. However, third parties can leverage *cookie syncing* [3] to merge information collected that pertain to users, and to recreate a more comprehensive history of user browsing behaviours. To perform cookie syncing, third parties share identifiers of the same user among different third parties. Cookie syncing is often used in real time bidding actions used for targeted advertisements [148].

Cookie respawning. With cookie respawning [259], a cookie that is deleted by a user is automatically respawned. Several techniques are used to respawn a cookie. When a user visits a website that supports cookie respawning, the website generates a user identifier (included in cookies) that stored in multiple storages. Consequently, when the use deletes the corresponding cookie, the website can recover it from a backup storage. The backup storage from which the cookies are recovered and respawned can be Flash [259], ETags, browser cache [260] and IndexedDB.

Session replay. In order to reconstruct the presentation of how a user experiences a website or a mobile app, *session replay* [19] is used. It captures keystrokes, clicks, mouse movements and page scrolls while a user browses through a website. Then, it creates a video of a walk-through to show what the user performed, while he was on the website/mobile app. Unlike traditional analytic tools, session replay provides a more visual way of determining user's browsing behaviours. Session replay tools should be properly configured to ensure possible user privacy issues are addressed — e.g., no private and sensitive information are collected during session replay. In addition, private and confidential user data (e.g., email address, date of birth) should be masked.

Browser fingerprinting. This is a stateless tracking mechanism used to collect information relating to the browser (e.g., browser type and version), operating system, active

plugins, timezone, language, screen resolution and other device/hardware specific characteristics. Then, the collected information are used to build a unique identifier that relies on several browser (e.g., user agent, WebGL) and machine level (e.g., AudioContext, AudioWorkletNode) characteristics [157]. Past studies [94] mention that there is significantly a smaller chance for any two users to have the exact unique identifier derived from the characteristics used for browser fingerprinting.

2.3 Tracking detection and privacy measurement tools

In this section, we present tracking detection methods and privacy measurement tools that are used in past studies to measure tracking in online services.

Tracking filter lists. Filter lists contain a list of regular expressions or domain names that need to be blocked. The most widely used filter rules are EasyList and EasyPrivacy [92], that are based on a set of rules originally designed for *Adblock* browser extension. EasyList is an ad-blocking list used to remove ads on websites. In contrast, EasyPrivacy is a tracker-blocking list used to remove tracking behaviours. EasyList also provides supplementary filter lists [93] for a limited number of other languages (e.g., German, Italian, Arabic, Chinese).

OpenWPM web privacy measurement framework. OpenWPM [223] is a privacy measurement framework for web sites. It supports the collection of instrumented data for millions of websites (provided as input). Currently, OpenWPM provides an automation using Selenium, and allows websites to launch in headless mode or through the Firefox browser. OpenWPM includes several configurable features to facilitate its instrumentations — e.g., HTTP request/response headers, redirects, POST request bodies; saving properties accessed/method calls (with arguments) of JavaScript APIs (includes APIs to extract potential fingerprinting attributes); page navigations; callstack details; DNS instrumentations; cookie instrumentations (i.e., HTTP cookies and JavaScript cookies). OpenWPM can be

configured to perform both stateless and stateful crawls. It can also save screenshots of pages crawled. Also, OpenWPM can save response content of web pages crawled to a *LevelDB* database.

Mobile Security Framework (MobSF). MobSF [196] is an automated framework that is capable of doing pen-testing, malware analysis and security assessments of mobile apps running on different mobile platforms (i.e., Android, iOS, Windows). MobSF can perform both static and dynamic analysis with app binaries (e.g., APK files). MobSF provides REST APIs to upload, scan and generate reports on the analysis of APK files. The *dynamic analyzer* in MobSF, performs runtime security assessments and interactive instrumentations.

LiteRadar. With LiteRadar [182], various instrumentations of tracking data from APK files of Android apps can be extracted — i.e., tracking SDKs included in an app, the use of tracking SDKs and permissions required to use an app.

2.4 Luminati - Residential proxy service

Luminati [179] is a commercial HTTP/S proxy service provider that routes traffic through 35 million residential IPs worldwide. The service operates over Hola [139] (installed as a browser extension) and applications built using the Luminati Monetization SDK [180]—residential users without a paid subscription. Luminati gradually transitioned from Hola to a SDK model. However, at the time we ran our experiments (see Chapter 3), Hola was used comprehensively for Luminati’s exit node infrastructure (i.e., HTTP requests are served over a browser installed on an exit node of a residential user). Routing in Luminati goes as follows: a Luminati client makes a proxy connection to a Luminati proxy server (super proxy); the server forwards the request to an exit node (peer proxy); and the exit node forwards the response to the super proxy, which in turn is sent back to the Luminati client. Luminati enables selecting exit nodes by country (or city/ASN at a higher cost), and

allows the same exit node to be used in subsequent requests by using the “sequential session (IP) pool” option. Switching the IP address of an exit node in the pool can be configured based on the number of maximum requests and session duration parameters, or at random. Luminati also allows controlling DNS resolution to happen at the super proxy (Google Public DNS), or the exit node. We choose a sequential pool of pre-established sessions to run a group of requests to target sites. Also, we configure DNS resolution to happen at a super proxy (US), to prevent DNS localization of web site domains at exit nodes so that trackers of the same first-party site are comparable between countries; e.g., when crawling `amazon.com`, we do not want the exit node to retrieve content from a regional first party site e.g., `amazon.com.mx` due to DNS localization. This is unlikely to influence the comparison of regional trackers as their DNS resolution remains unaffected. Luminati supports super proxy IP caching where three super proxy IPs in the cache are available to service requests, eliminating unnecessary timeouts due to distant super proxies.

The IP address of a user, connecting to a websites through a proxy can be identified using the *X-Forwarded-For* [188] request header. Adding the IP address of the user to *X-Forwarded-For* request header by a proxy defeats the purpose of being anonymous [146] with the connecting server. Luminati has been adding this header to requests in the past [178].

2.5 Web cloaking

Web cloaking is a technique in which the content presented to a security/search engine crawler is rendered as benign, and different from the malicious view presented to the user’s (i.e., the victim) browser. Adversaries who host phishing and malware services want to hide their activities from the search engine crawler [156]. In addition, adversaries leverage different search engine optimization (SEO) techniques when showing fake content to search engine crawlers compared to users who use browser clients. These SEO techniques are used to increase the ranking of illicit sites [55]. Adversaries can also pay advertising networks

to show benign advertisements to crawlers, while users view deceptive advertisements that lead to scams and malware [156].

In order to cloak content, the adversary's web server needs to distinguish the type of client (i.e., crawler vs. browser) based on an identifier [301], and the choice of the identifier depends on the cloaking technique as described below.

1. In *user-agent cloaking*, the type of client of an incoming request is identified by inspecting the user-agent string. If the user-agent belongs to a crawler, benign content is shown, otherwise malicious content is displayed.
2. With *IP cloaking*, the user is identified using the client IP address of the incoming request. If the IP address of incoming request is within a well known range of public IP addresses of a search engine crawler, benign content is rendered. Otherwise, the IP address most likely belongs to a user/enterprise, in which case malicious content is displayed.
3. *Repeat cloaking* is used to victimize a user on the first visit to the website (but not on subsequent visits). In this case, the state of the user is saved at client side (e.g., cookie) or server side (e.g., client IP) to determine a new user visit.
4. *Referrer cloaking* uses the *Referrer* field of the request header to determine if the user clicked through a search engine query result, in which case, the user can be redirected to a scam web page. In Referrer cloaking, adversary's objective is to target search engine users.

In practice, different types of cloaking are combined and used together.

Chapter 3

Tracking on popular sites from a global perspective

3.1 Introduction

Several past measurement studies uncovered various aspects of web-based tracking and its serious impact on user privacy. Most studies used institutional resources, e.g., computers hosted at well-known universities, or cloud-computing infrastructures such as Amazon EC2, confining the study to a particular geolocation or a few locations. Would there be any difference if web tracking is measured from actual user-owned residential machines? Does a user's geolocation affect web tracking? Past studies do not adequately answer these important questions, although web users come from across the globe, and tracking primarily targets home users. Therefore, in this study, we explore variations in tracking as experienced by residential users, from a global perspective.

Third-party web tracking based on user-behavioral profiling has become a major enabling technique for online targeted ads (for business impacts see e.g., [230]; see also Mayer and Mitchell [187] for a discussion on economics and tracking). Tracking is generally performed using cookies, scripts and browser/traffic fingerprinting (see e.g., [97, 187]).

Beyond ads/analytics, tracking can also be effectively exploited by government surveillance programs [98]. Indeed, the US NSA has reportedly used Google cookies for targeted hacking/surveillance [304, 106].

Several past studies explore the extent of tracking, evolving techniques used for tracking, and privacy/business implications of tracking. The literature on tracking is rich and becoming very useful to researchers and regulatory bodies. Englehardt and Narayanan [97] recently measured the extent of third-party trackers on Alexa Top-1M websites using the OpenWPM framework [97]. They run their crawler from an Amazon EC2 instance. Fruchter et al. [113] performed another study, albeit at a much smaller scale (Alexa Top-250 country-specific websites), to uncover variations in tracking in four geographical locations (US, Germany, Australia and Japan) of varying policies/laws/cultures. They also used Amazon EC2 machines from different locations. Falahrastegar et al. [104] studied web tracking using Alexa Top-500 country-specific websites for seven countries (USA, UK, Australia, China and Egypt, Iran, and Syria) from a single location in the UK. To evaluate the possibility of surveillance via (third-party) cookies and (first-party) plaintext user-identifiers, Englehardt et al. [98] used Amazon EC2 instances from three geolocations: US (Northern Virginia), Ireland (Dublin), Japan (Tokyo).

Although the study by Fruchter et al. [113] indicates that there are significant differences between countries (four in the study), all past studies lack a global perspective, in terms of the number of locations used to measure tracking (1 to 4 countries). Also, all studies were conducted from institutional machines and known IP ranges (university/Amazon), although tracking primarily targets home users (residential machines). Institutional/data center proxies are also prone to be blocked or challenged with CAPTCHAs, to mitigate potential abuse (see e.g., [121]). Therefore, for tracking measurements, the use of residential machines appear to be more appropriate. In some other security-related studies, such as censorship [214], end-to-end connectivity violation [58], and DNSSEC infrastructure

management [60], geolocation and/or the use of residential machines have been taken into more serious consideration.

We focus on exploring the effects of geographical variations in tracking as experienced by residential users in various parts of the world. Considering differences in political, social, and cultural factors, we choose 56 countries from across the world for crawling a selected set of web sites, using the Luminati HTTP/S proxy service [179].¹ Using OpenWPM, we automatically crawl different types of website URLs (first parties) including the Alexa Top-1000 global sites (home pages), 1000 URLs hosted on the selected Alexa web domains that were shared via Twitter, and Alexa Top-50 country-specific sites (home pages). Subsequently, we extract third-party information of scripts and cookies from the OpenWPM database, and process them using EasyList rules [92] with BlockList-Parser [253] to perform privacy-related tracking measurements.

Our results show that the prominence of trackers varies significantly between countries – not only in the country-specific sites, but also for global sites. Furthermore, tracker prominence of inner links of a website appears to be higher than its home page. A significant number of third parties place cookies on websites with long validity periods (e.g., >20 years), egregiously violating any reasonable use scenario, and in some cases existing laws/regulations (e.g., the EU cookie law). Although most trackers are global in nature (mostly owned by US companies), top trackers from countries such as China and Russia appear to operate only within the same country.

Contributions.

1. We extend existing tracking measurement studies in three important directions: (i) crawling websites from 56 countries around the world, representing different political, cultural, regulatory, and Internet speed and freedom situations (cf. four countries

¹Luminati is a commercial network proxy service, providing residential exit nodes in many countries. Recent work by Mi et al. [191] raises serious doubts about how these machines are recruited (e.g., possibility of compromised machines). However, their methodology for characterizing Luminati nodes appears to be unclear—discussed more in Section 8.2.

used in [113]); (ii) the use of residential computers via the Luminati proxy service as opposed to institutional/data center machines; and (iii) analyzing web content from home pages and inner links of selected Alexa domains (also studied in [97] for a single location). Our methodology provides a more bona fide, global perspective on tracking.

2. We find that for most cases, a tracker’s prominence changes significantly with the geographic location, beyond the dynamic nature of current advertisement/tracking ecosystems (which we also measure separately from Montreal, Canada).
3. We also confirm the findings from existing studies and extend them; e.g., similar to Trevisan et al. [287], we also found that the EU cookie law [276] is violated by most tracking companies/sites. Forwarding web requests to local IP addresses through DNS hijacking was reported for Iran [25]; we also observe similar behavior in Saudi Arabia and Uzbekistan in significant numbers.

3.2 Related work

Our work provides a more inclusive, global perspective on tracking, by leveraging existing tools and methodologies from several past studies. Here we summarize a few such efforts. Fruchter et al. [113] measure tracking variations in four countries with different privacy models (as categorized in [270]): (1) *comprehensive*, protecting all digital data (Germany); (2) *sectoral*, protecting certain types of data such as health-care (USA, Japan); (3) *co-regulatory*, similar to (1), but enforcement is done by industry (Australia); and (4) *mixed/no-policy*, no protection for digital privacy (China/Russia, not studied in [113] due to the non-availability of AWS EC2 instances in those countries). They use Alexa Top-250 country-specific sites, and report significant differences in tracking activities between the countries. For example, the number of third-party cookies in news sites are considerably

more in the USA, Japan and Australia, compared to Germany. They further conclude that tracking differences in countries may not solely depend on their privacy models, but also on factors such as policy, regulations and culture.

Tracking primarily leverages third-party scripts and cookies, but other advanced/subtle techniques are also used, e.g., evercookies, cookie syncing, and fingerprinting of browser type, canvas/font, web traffic, and WebRTC, AudioContext and battery-level APIs; cf. [3, 187, 209]. In a comprehensive measurement study, Englehardt and Narayanan [97] recently measured the extent of third-party on Alexa Top-1M websites using the OpenWPM framework [97]. They make 15 types of measurements of stateless and stateful tracking techniques. Their results include many important findings: only few third-parties are present in most sites, news sites hosting the most number of trackers, the use of advanced stateless fingerprinting techniques in the wild, and effectiveness of anti-tracking measures (addons and browser features). They also crawl four internal pages of Alexa Top-10K domains; top 20 trackers are found more prominently on the internal pages compared to the home pages.

Tyson et al. [288] analyze the degree of HTTP header manipulation by middleboxes across ASes in different networks and regions around the world, using Hola [139] (a peer-to-peer VPN service operated by Luminati). They report that 25% of the ASes modify HTTP headers, and the level of manipulation depends on the region and AS type: well-connected regions have fewer caching headers than less-connected regions with costly transit. However, the frequent use of cached data from legacy middleboxes can be exploited.

Using Luminati, Chung et al. [58] propose a novel approach to identify end-to-end violations in HTTP, HTTPS and DNS protocols. They observe that web content sent over HTTP is compressed in flight by some ISPs. They identify a vulnerability where HTTP requests from users are recorded at ISP middleboxes, and the same content is fetched later by third party servers. This allows adversaries to monitor HTTP responses, raising privacy implications.

Pearce et al. [214] designed a measurement platform to assess DNS manipulation attempts for imposing Internet censorship, by leveraging OpenDNS resolvers hosted by ISPs and cloud service providers from 151 countries. They reported that DNS manipulation is heterogeneous across countries, domains and DNS resolvers. Several countries such as Iran, Pakistan, China are found to use DNS manipulation for censorship.

Merzdovnik et al. [190] analyze the effectiveness of current anti-tracking privacy tools on more than 100,000 websites from Alexa Top-200K domains; some of these tools are very effective (over 90% success rates) against stateful trackers, and less successful against stateless fingerprinting trackers. They also report that over 60% of the third-party requests didn't use TLS, which makes it possible for adversaries to passively analyze the unencrypted traffic (i.e., third-party requests and responses). They also highlight the danger of over-reliance of a specific third-party tracker being used in a large number of first-party sites (cf. NSA's alleged exploitation of Google cookies [304, 106]).

According to the EU Internet Handbook [276], the use of *profiling/tracking* cookies require explicit user-consent; session cookies and cookies that are required for essential functionality are exempted. Trevisan et al. [287] use 35,862 popular sites from 25 countries (21 EU and 4 non-EU) to measure the compliance of the EU ePrivacy Directive (also known as the EU cookie law). They also use proxy services from eight EU countries to check variations of tracking cookies based on browsing locations (the EU cookie law's enforcement varies across member states). The authors identify cookies in trackable context by comparing them with a public list of web tracker domains.² They find 65% of the web sites fail to comply with the cookie law (i.e., a cookie is set before a cookie accept bar is even displayed to the user). They also observe that 80% of the third-party cookies last more than a month, and approximately half of those cookies remain valid for more than a year. We find 22% of the cookies remain valid over a year across EU countries (vs. 23% across

²<https://better.fyi/trackers/alexa-top-500-news/>

all 56 countries).

Mayer et al. [187] observe that third-party web tracking is transitioning from a regulatory vacuum to regulatory frameworks, implemented by government organizations (e.g., US FTC, EU ePrivacy Directive, and self-regulatory programs such as Network Advertising Initiative, Interactive Advertising Bureau).

Degeling et al. [78] analyzed GDPR's impact on Top-500 country specific sites in 28 member states in the EU. They found that GDPR made the majority of companies to make adjustments to accommodate the new regulations. Despite, the authors claim, we find that tracking activities have not changed and most cookie consenting libraries are not meeting the requirements of the GDPR.

Schelter et al. [244] performed a large scale analysis of third-party trackers using the *Common Crawl 2012* corpus. The corpus may contain tracking information of residential as well as institutional users. Since third parties are extracted from static embedding of web pages, transient trackers having dynamic content are not considered. In contrast, our study includes mostly residential computers, and the content we collected is not limited to static trackers.

Web services may be divided into categories e.g., culture, religion, news, sports, etc. To measure tracking variation across different categories, Falahrastegar et al. [104] study seven countries from all continents with different languages using 500 most popular country-specific web sites (crawled from a UK location). Their findings show that some of the top trackers are local to the hosting country of the corresponding first-party website (e.g., websites from China and Iran).

Mi et al. [191] use five residential proxy services including Luminati, for illegal/unwanted/malicious nodes in these ecosystems. They claim that Luminati runs many IoT devices although most exit nodes are indeed residential. However, their methodology for detecting IoT devices inside a NAT requires scanning the internal network (local

subnet), which is disallowed by Luminati; thus, such device characterization for Luminati seems to be flawed (also confirmed by Luminati). Luminati also informed us that their proxy software is not supported on any IoT devices (available only for desktop and mobile OSes). Also, Luminati software is installed with explicit user consent, in contrast to the claim by Mi et al. [191]—see Section 3.3.4 for more issues related to ethics.

3.3 Methodology

We use the Luminati proxy manager [179] to run experiments from 56 countries, and the OpenWPM privacy measurement framework [97] for automating browser data collection and tracker analysis on a selected list of URLs. With OpenWPM [97], we automate the crawling of a large number of URLs. The built-in proxy that is available in OpenWPM is replaced with Luminati. We configure OpenWPM (ver: 0.7.0) for stateless crawling (each new page-visit uses a separate browser profile), as we are primarily interested in the location-related aspects of tracking. Instrumentation results are stored in a local SQLite database; we modify the database schemas to record additional information, e.g., the exit node’s IP address, AS details, and location (country). We launch three browser sessions simultaneously through OpenWPM; we could not further increase the number of parallel sessions due to performance issues which would crash the crawling sessions (system configuration: AMD FX8350, 8GB RAM, Ubuntu 16.04, Gigabit Internet).

The requests from OpenWPM crawls are proxied via Luminati, so that they go through exit nodes in the country of our choice. Luminati passes the response from exit nodes back to OpenWPM, which processes the response data, extracts privacy related measurements, and stores them in a database. We then query the database to analyze the measurement data and compute various metrics.

In this work, we expand on our country/URL selection, and define trackers and their prominence (largely based on [97]). Through Luminati, we process a total of 68,800 URLs.

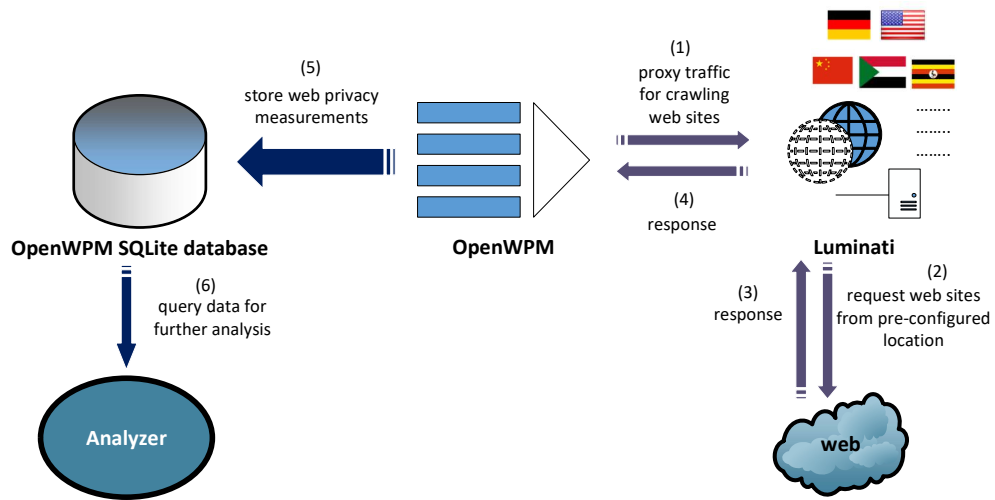


Figure 1: Our system setup - Measurement of tacking residential users from a global perspective.

These sites include, Alexa Top-1000 (global) and Top-50 (country specific) URLs from 56 countries, and 1000 URLs shared via Twitter for 10 selected countries. Each URL request takes 1.16MB of bandwidth on average (including repeated attempts for failed/timed-out requests). We run the experiments between June 1 and July 8, 2017. Using Luminati is expensive.³

We only considered successful third party requests for our analysis. Those URL responses with client errors (4xx status code) and server errors (5xx status code) are eliminated from the analysis. Such failures are attributed to many reasons (e.g., authentication issues, timeouts, censorship).

3.3.1 Country and first-party site selection

The use of residential machines from all countries/regions/cities would be ideal for our goal. However, using Luminati is costly, and it also lacks exit nodes in certain countries (e.g., North Korea). Covering several regions with various political and socio-economic

³With the cheapest Luminati *Starter residential* package, it costs USD 12.50/1GB (for 40GB, with a minimum monthly commitment of USD 500). Hence, we incurred USD 14.50 for 1000 URLs. Therefore, for 56 countries, it will cost USD 812 to process 1000 URLs. For 1 million URLs (cf. [97]) from 56 countries, the cost will be USD 812,000. We thus had to limit the number of crawled URLs.

situations, we select 56 countries. We list in Table 4 the countries in various regions used in our experiments. Our selection is influenced by Freedom House [111], and Swire and Ahmad [270].

Asia-Pacific	Australia (AU), Bangladesh (BD), China (CN), India (IN), Japan (JP), Malaysia (MY), Myanmar (MM), Pakistan (PK), Philippines (PH), Singapore (SG), South Korea (KR), Sri Lanka (LK), Vietnam (VN)
Americas	Argentina (AR), Brazil (BR), Canada (CA), Colombia (CO), Cuba (CU), Ecuador (EC), Mexico (MX), United States (US), Venezuela (VE)
Europe	Estonia (EE), France (FR), Germany (DE), Great Britain (GB), Hungary (HU), Iceland (IS), Italy (IT), Turkey (TR), Switzerland (CH)
Eurasia	Armenia (AM), Georgia (GE), Kazakhstan (KZ), Russia (RU), Ukraine (UA), Uzbekistan (UZ)
Middle East and North Africa	Bahrain (BH), Egypt (EG), Iran (IR), Israel (IL), Jordan (JO), Lebanon (LB), Libya (LY), Morocco (MA), Saudi Arabia (SA), Tunisia (TN), United Arab Emirates (AE)
Sub-Saharan Africa	Ethiopia (ET), Kenya (KE), Nigeria (NG), Rwanda (RW), South Africa (ZA), Sudan (SD), Uganda (UG), Zimbabwe (ZW)

Table 1: List of regions and countries.

The 2050 distinct URLs that we use for crawling include: (1) home pages of Alexa Top-1000 global domains; (2) 1000 popular URLs that are shared via Twitter from the Alexa Top-1000 domains, excluding home pages, and links to media (e.g., images, audio and video) and text files (which may not host any tracker); and (3) home pages of Alexa Top-50 country-specific domains.

We extract Twitter URLs using Tweepy [159] that internally uses the Twitter streaming APIs to access the global stream of Twitter data. Twitter mandates that a client filter the streamed data according to a specific criterion. To not omit streams from any parts of the world, we use: `twitterStream.filter(locations = [-180, -90, 180, 90])`. Assuming geotagging is turned on, this filter selects tweets

from all around the world using the *locations* filter. We select the most shared URLs from the Alexa Top-1000 domains.

3.3.2 Tracker identification and prominence

We define the third parties as follows. (1) Third-party scripts: the domain on which the third-party script runs is different from the domain of the first-party site. (2) Cookies: the cookie’s domain is different from the domain of the first-party site.

Not all identified third parties may necessarily be trackers. Third party domains can be trackers, advertisers, or simply content embedded on a first-party site. We use BlockList-Parser [253] to filter third parties in a tracking context with the aid of a set of ad-blocking filtering lists as used by the Adblock browser extension: *EasyList* and *EasyPrivacy* [92]. *EasyList* tracking protection lists contain rules to identify trackers which are also advertisers, while *EasyPrivacy* identifies non-advertising trackers [97]. This filtering is in line with previous studies; cf. [97, 113]. For analysis, we keep trackers that exist on at least two first-party sites (similar to [97]). Since advertisers in certain circumstances can play a dual role as trackers, we emphasize that the identified trackers in our analysis may fall into a lower bound of trackers in reality; more sophisticated filtering is difficult (e.g., some third parties directly, or through their parent organizations, may act as genuine content providers [119]). For example, Google receives a large proportion of content related third-party requests that do not fall into the categories of tracking or advertisements.

To identify tracker domains based on third-party scripts or cookies, we use *public suffix + 1 (PS+1)* of the script URL or the cookie domain (along with Mozilla’s Public Suffix List⁴ as in [97]). For example, if the script URL is `http://tpc.googlesyndication.com/sodar/d5qAyLYU.js`, then PS+1 of the domain is `googlesyndication.com`; if the

⁴Hosted at: <https://publicsuffix.org>; a public suffix is defined as “one under which Internet users can (or historically could) directly register names.”

script is included as a dependency in <http://oneindia.com>, then googlesyndication.com is a third-party tracker.

Tracker prominence. A possible limitation of measuring a tracker’s activity using the number of first parties on which the tracker is present is that the tracker may have a low first-party count, when a limited number of first-party sites are used. To properly rank trackers’ prominence, we use the following metric from Englehardt and Narayanan [97]: $Prominence(t) = \sum_{edge(s,t)=1} \frac{1}{rank(s)}$; $edge(s, t)$ indicates third-party t ’s presence on site s . This metric mitigates the distortion of a tracker’s importance due to the selection of a small set of first-party sites (as in our case, 1050- 2050 URLs per country). Such a small set of first-party sites may not include all first parties where a particular third-party is highly prevalent.

Comparing countries. To compare the extent of tracking between countries, we treat the prominence values of trackers of each country as a group, and we compute non-parametric Kruskal-Wallis (KW) rank averages (assuming groups are independent). Countries with a higher rank average should have a higher level of tracking and vice-versa. Furthermore, the rank averages of all the countries can be used to perform the *KW test* to determine if the level of tracking between countries is independent of each other or not. In a KW test, a null hypothesis is initially assumed where all samples (i.e., groups) come from identical populations. If the KW test value is greater than the critical chi-square value, the null hypothesis is rejected, proving at least one group comes from a different population. A similar approach was adopted by Fruchter et al. [113] for comparing tracking activities between four countries.

3.3.3 Dynamicity of trackers

Since ad exchanges leverage a Real-time Bidding (RTB) auction based model where only winning bidders are allowed to serve content to users [28, 128], web trackers are also

expected to be dynamic in nature. However, dynamicity of trackers have not been discussed in previous large scale measurement studies (e.g., [97, 113, 104]). To establish ground-truth on the limits of dynamic behaviors of trackers, we conducted several experiments with Alexa Top-1000 sites. We calculated the difference of the number of first parties for each tracker as observed from two different ISPs within Montreal, Canada; we performed 12 tests simultaneously from both ISPs, and at different times of the day, over a period of two months, where each test took approximately 4 hours to complete.

We use *z-score*⁵ to assess the variation of trackers; *z-score* measures the number of standard deviations of the signed distance between a data point and the mean of a distribution.⁶ If the data point is greater than the mean, the *z-score* is positive, otherwise it is negative. Overall, *z-scores* for our observations lie between -0.4 and +0.4. For simultaneous runs from both ISPs, the differences and *z-score* values of the number of first parties for the Top-5 trackers are: advertising.com (223, 0.36), pubmatic.com (192, 0.27), adsafe-protected.com (140, 0.11), moatads.com (75, -0.09), scorecardresearch.com (68, -0.11). Similarly, when measured from a particular ISP at different times, the differences and *z-score* values for the Top-5 trackers are: openx.net (217, 0.39), googlesyndication.com (114, 0.05), adnxs.com (109, 0.03), gstatic.com (73, -0.09), yandex.ru (73, 0.09). These values change when measured at different times, although the *z-scores* always remain within -0.4 and +0.4. We also computed the Pearson correlation coefficient for the number of first parties that trackers are found in different runs (different ISPs, and at different times of the day); our Pearson coefficient turned out to have a highly positive linear correlation (0.9), implying that the trackers identified in independent runs follow a strong linear relationship. Therefore, for the overall tracking ecosystem, the dynamicity of trackers does not appear to have adversely impacted the interpretation of results of our measurement study.

⁵https://en.wikipedia.org/wiki/Standard_score

⁶Unlike standard deviation, *z-score* is used to compare scores from different distributions [245]. Also, *z-score* determines whether a given value is typical in a data set.

3.3.4 Ethical issues

We access residential users' Internet connection through Luminati, which is a paid service. We do not compromise the security or privacy of users (of exit nodes) beyond using their internet connection, which they have agreed to when signing up with Luminati. These users include those using Hola [139] clients and applications built leveraging the Luminati monetizing SDK [180] without a subscription. Hola and Luminati explicitly mention the sharing of internet connection to their users. Furthermore, we do not store the response content returned by the websites, except the measurements for trackers.

Some websites that we crawl (Alexa top sites and Twitter-shared URLs) may be censored in a few countries. Other than Egypt,⁷ we are unaware of any place where attempts to access blocked/censored content will trigger legal problems for a user. During our tests, the new law in Egypt that threatens imprisoning those browsing censored web sites did not exist. Besides, the sites crawled from Egypt are not subjected to censorship according to the Citizen Lab dataset [62]. We are unable to get the consent from targeted users owning Luminati exit nodes, as we do not have their contact information. However, we reached out to the internal Research Ethics Unit of our University, and explained our experiments; they did not object to our methodology and did not require us to go through a full ethics evaluation.

3.3.5 Limitations

We use Luminati's residential exit nodes for measuring web tracking from a home user's perspective. However, we have no control over such nodes (compared to using more reliable university/EC2 infrastructures). Here we list some issues that may affect our results.

(1) Web tracking may depend on the browsing history of a specific client as identified by its

⁷News article (Aug. 19, 2018): <https://www.telegraph.co.uk/news/2018/08/19/egyptians-face-jail-accessing-banned-websites/>

IP address. Thus our results may be influenced by the browsing history of the Luminati exit nodes, which is beyond our control. This is an inherent limitation of using residential IPs as opposed to university/Amazon IPs. However, our connections are not effected by local cookies or other browsing data (only share the same IP address). (2) We crawl websites via OpenWPM in a sequential order over the period of five weeks. Hence, time dependent trackers (if any) may affect our results. Furthermore, the number of trackers on first party sites may grow or shrink with time. This is due to many reasons, including technological advancements of tracking techniques [173], outages, performance issues with tracking services, and ISP filtering [31, 48]. A comprehensive study of such dynamic behaviors and uncertainties of tracking at a global scale is beyond our scope as it requires repeated tests, which is not pragmatic due to the high cost of using Luminati. However, we measured dynamicity of trackers via two ISPs from two locations, and found the impact to be limited to our measurement criteria (see Section 3.3.3). (3) Tracking context of some Google trackers (e.g., google.com, gstatic.com, youtube.com) are omitted from our work as they are not proxied by Luminati. We realized this limitation during our experiments.⁸ However, most Google-owned tracking domains remain unaffected, i.e., proxied through Luminati. We manually verified this for all top tracker domains in our list. (4) The EasyList [92] filter that we use to identify third party domains participating in a tracking context may not have adequate coverage in all countries, although it can filter most trackers from international web pages. Therefore, our results may not include trackers that are not identified by EasyList rules. However EasyList offers several supplementary filter lists [93] to support several non-English domains (e.g., German, Italian, Dutch, French, Chinese, Bulgarian, Arabic, Czech, Slovak, Lithuanian and Hebrew). The coverage of these supplementary lists are still unknown. (5) If a first party website intentionally uses a third party for tracking the site visitors, we do not distinguish such trackers from other third parties participating in a

⁸Luminati prohibits “Any form of outbound automated Google search queries”, but mentions no other Google related restrictions.

tracking context.

3.4 Trackers vs. geolocation

In this section, we explain the analysis process followed by the results. Unless otherwise stated, the tracking context is measured using third-party scripts on the Alexa Top-1000 global domains.

We first check the presence of top-10 trackers in Alexa Top-1000 domains in all countries. For brevity, we highlight the results from 15 countries with most significant differences across regions; see Fig. 2. The top-10 trackers are determined based on the average percentage of first-party sites across 15 countries. If multiple instances of the same tracker are found on a particular first-party site (e.g., several scripts from a single tracker domain), we count them separately.

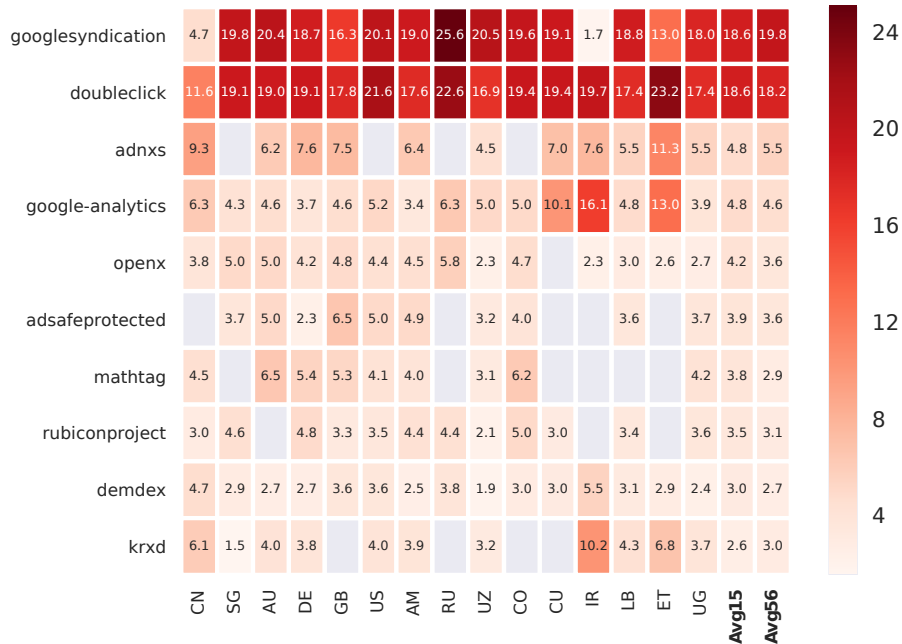


Figure 2: First-party percentages for script-based trackers across 15 countries; *Avg15* and *Avg56* represent average percentages for the selected 15 countries and all 56 countries, respectively.

In Fig. 2, darker-shade trackers have more presence in first-party sites compared to

lighter shade ones. We calculate the tracker percentages for each country based on the first party count for the specific tracker over the total first-party count of the specific country. Highest percentages for googlesyndication (25.6%) and doubleclick (23.2%) trackers are observed in Russia and Ethiopia respectively; the percentages are relative to other trackers observed from the same country. These two trackers are also prominent in all other countries. In contrast, some trackers are not seen in certain countries (blank cells in Fig. 2).

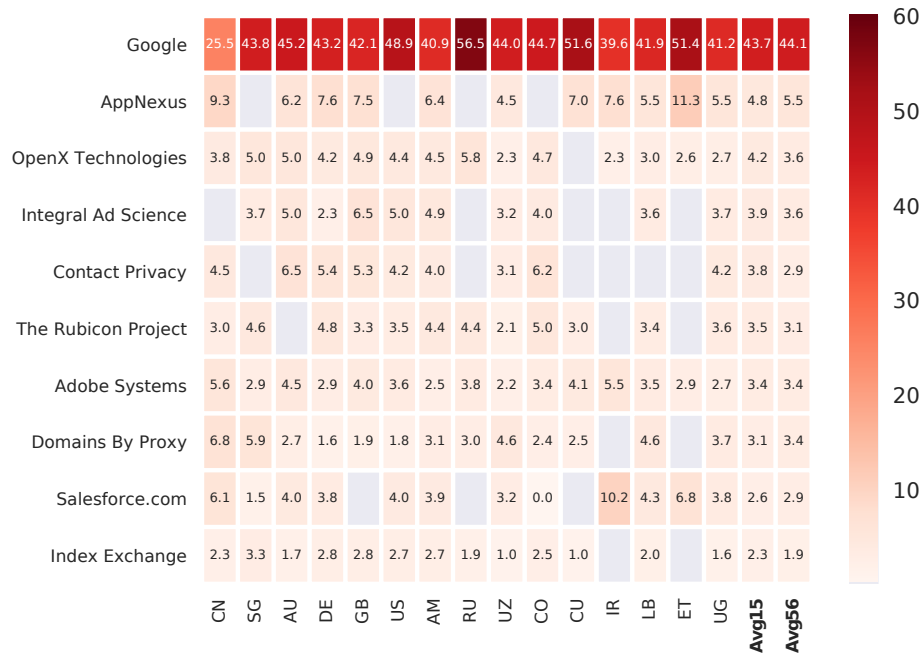


Figure 3: First-party percentages for tracking org.

China and Iran have a relatively low percentage of trackers. Google advertisements are sanctioned in Iran by United States Office of Foreign Assets Control (OFAC) [129]. Schelter et al. [244] observed a similar pattern in their study, and they justify this behavior due to political factors including lack of democracy and freedom of the press.

We also identify the top-10 tracking organizations; we use *pywhois* [192] to locate organizations from corresponding domains; see Fig. 3. Google has a clear domination across the world. Note that despite Google services, e.g., Google Search, Maps, Docs, Mail being censored in China [158], Google trackers remain active in China on uncensored websites.

3.5 Overall tracker prominence

In this section, we analyze the differences in tracking, using prominence and KW rank metrics (Section 3.3.2), and compare 56 countries; see Fig. 4. UK and Armenia have the highest prominence values, while Cuba, Ethiopia and Iran have the least. The latter countries are known to have less media/Internet freedom [25, 111]. Cuba [18], Ethiopia [65] and Iran [52] are subjected to online censorship, where popular third parties (e.g., social media resources such as Facebook) are blocked. Therefore, these censorship practices may have contributed to the low number of trackers on sites from these countries. Countries such as Morocco, Singapore, Venezuela, Mexico and Rwanda have relatively higher prominence values although they rank low in media/Internet freedom, showing the presence of other factors influencing tracking. We discuss the impact of few of those factors such as Internet speed, censorship and browser user agents in Section 3.7.

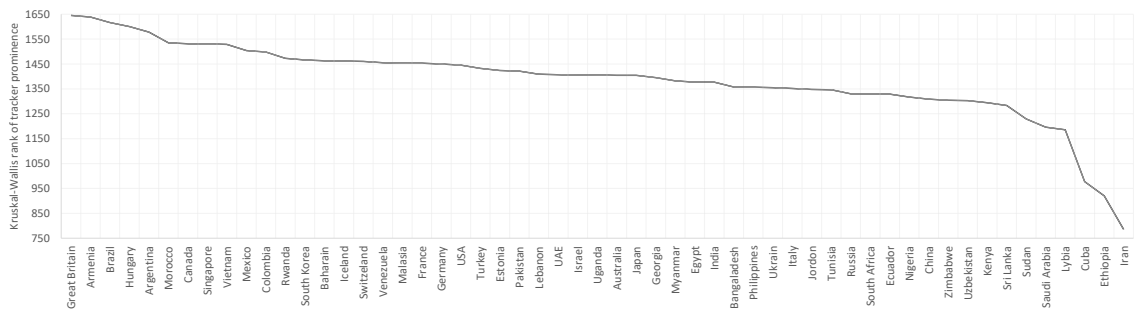


Figure 4: Prominence of tracker scripts across 56 countries.

We summarize prevalence of top trackers (in terms of the average of raw count in countries) in different regions in Fig. 5. In general, Europe has the highest count compared to others, despite the EU cookie law. Degeling et al. [78] claims, GDPR didn't have a noticeable change in tracking although it made the web more transparent by having the website owners updating their privacy policies.

Our results from the KW test show that the tracker prominence among different countries are independent of each other ($\chi^2 = 83.64, df = 55, p = 0.05$). This is because the

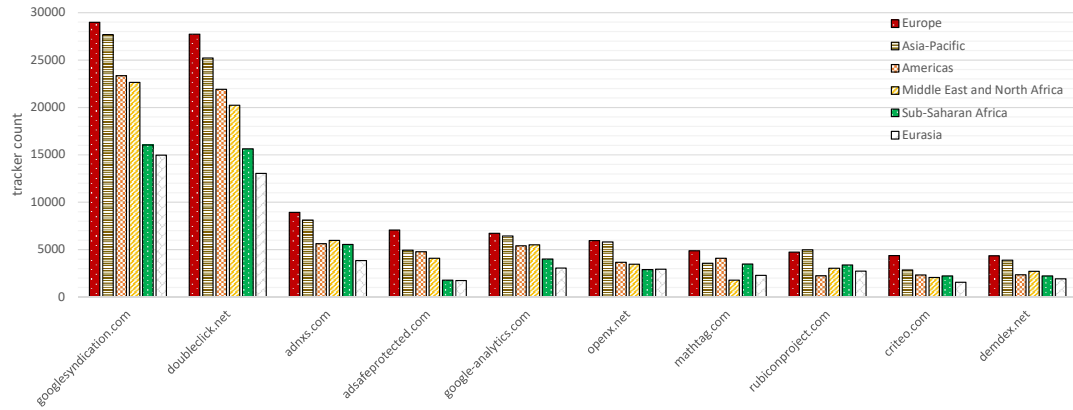


Figure 5: Average of top trackers in different regions.

null hypothesis of the KW test is rejected as the KW test value ($x^2 = 83.64$) is greater than the critical chi-square value (73.311) [189] with 55 degrees of freedom (df), where p -value (used to accept/reject the null hypothesis) is 0.05. Therefore, the prominence of trackers varies with different browsing locations that are independent of each other.

Comparing prominence between home pages and Twitter URLs. We also compare countries based on tracker prominence in Alexa Top-1000 home pages and Twitter-shared URLs. For this experiment, we consider 10 countries across all regions. We calculate the prominence values of trackers in each country for the home pages and Twitter URLs; see Fig. 6.

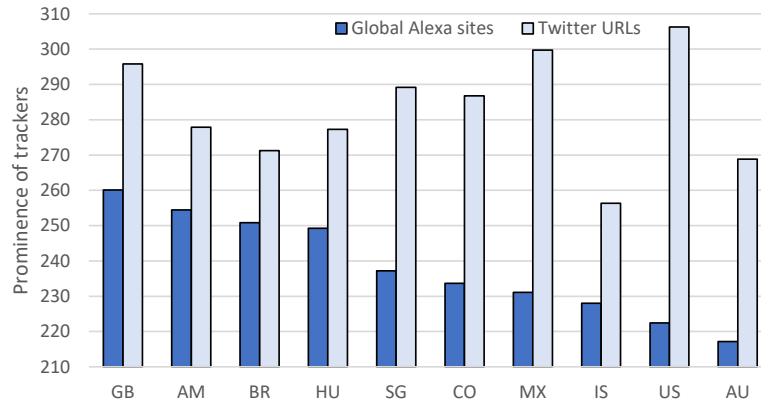


Figure 6: Prominence of tracker scripts: Alexa global sites vs. Twitter URLs.

It is apparent that the prominence values of trackers in Twitter URLs are significantly

higher compared to home pages in *all* selected countries. Englehardt et al. [97] noticed a 6%–57% increase of third-party presence on first parties (for top 20 third parties) with inner URLs as opposed to their home pages. In our experiments, the increase of prominence in Twitter URLs is between 7%–28%. However, we take into account the prominence of all third parties available. Therefore, it appears that increase of third party presence in inner URLs is relatively higher for the top trackers.

Trackers in country-specific sites. Most trackers across the world are hosted from US domains. However, similar to the observations in Falahrastegar et al. [104], we note an exception from Top-50 country-specific first-party sites in China and Russia, where the top trackers for both third-party scripts and cookies originate from the same country. In China, baidu.com tops the first-party count in both tracking scripts (93) and cookies (5). Similarly, in Russia, yandex.ru is a leading tracker having the highest first-party count for tracking scripts (427) and cookies (25). The difference in approaches between Falahrastegar et al. and ours is that in the former, 500 country-specific first party sites are used (from the same location), while we use Top-50 country specific sites (from 56 countries); they report more baidu.com count (approx. 2000), although they do not clarify the tracking context.

3.6 Cookie validity durations

Similar to Trevisan et al. [287], we also found that the EU cookie law is not complied by most tracking companies in the EU and non-EU countries; see Fig. 7. Many cookies have a validity period over 20 years, and some up to 7988 years (e.g., rubiconproject.com, rfihub.com). Overall, UK and US have the highest counts of these cookies, while Iran, Cuba, Ethiopia and Libya have the least. We did not use the data collected from Kazakhstan for our analysis, as it is impacted due to slow connections at the respective exit nodes.

Trevisan et al. [287] found 49% of the websites in 25 countries (21 from EU) install tracking cookies. In comparison, we found that, for the nine European countries, 60% of

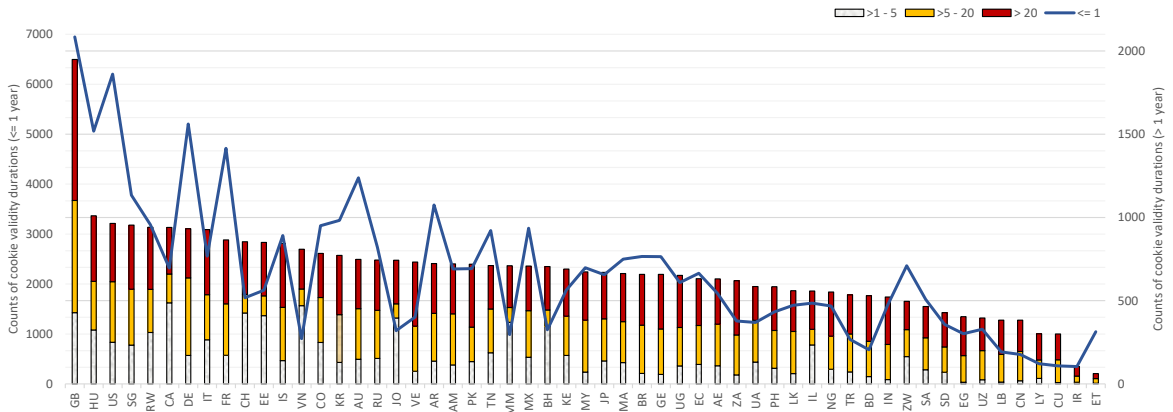


Figure 7: The number of cookies vs. validity period: counts of cookie validity periods ≤ 1 year are shown by the line (left y-axis), and the rest (> 1 year) are shown with bars (right y-axis). All these cookies are set without user consent.

Access country	>1year	>20 years
Great Britain	10,516	3618
Germany	5047	1956
Hungary	5071	1866
Italy	5000	1853
France	4250	1801
Estonia	2692	1267

Table 2: Number of tracking cookies with validity periods (EU).

first-party sites set tracking cookies without consent, which is even higher than our global average (56.2% sites in the 56 countries; see Degeling et al. [78] for technical issues in GDPR compliance and common cookie consent implementations).

In addition to third party cookies, some first party cookies (e.g., doubleclick.net, paypal.com) contain unique pseudonymous identifiers, although they do not include Personally Identifiable Information (PII) [20]. We did not find attributes in first party cookies containing any identifiable PII

The top-5 domains of tracking cookies with over a year validity are as follows (the number of cookies, first-party percentages): scorecardresearch.com (23,171, 0.015%); rubiconproject.com (12,680, 0.008%); rfihub.com (12,105, 0.008%); advertising.com (11,042, 0.007%); and adtechus.com (9940, 0.006%). We also checked their privacy policies (Sept. 2, 2018). They do not mention their cookie validity periods, but claim to

Tracker domain	Access country	Reg.	Count
smartadserver.com	DE, EE, FR, GB, HU, IT	FR	4005
angsrvr.com	DE, FR, GB, HU, IT	DE	1570
criteo.com	EE, FR, GB, HU, IT	FR	1461
ml314.com	DE, EE, FR, GB, HU	FR	920
theadex.com	DE, EE, GB, IT	FR	665
yieldlab.net	DE, GE, HU	DE	420
visualdna.com	DE, GB, IT	GB	417
semasio.net	DE, GB	DE	392
switchadhub.com	EE, FR, GB, HU, IT	GB	303
ligadx.com	HU	DE	280

Table 3: Domains of top-10 tracking cookies registered in EU countries.

be in compliance with EU privacy laws (including GDPR [100]). The opt-out mechanism of scorecardresearch.com is also cookie based [247], i.e., opt-out is not possible when cookies are blocked or deleted. Top-6 EU countries with the most number of cookies with long validity periods are listed in Table 2. Top-10 EU specific domains with the highest number of tracking cookies are listed in Table 3; for each domain, we also list the countries from which the requests are originated, and the country where the third-party domain is registered (most registered in France and Germany).

3.7 Factors other than geolocation

3.7.1 Internet speed

Tracking appears to vary proportionally with Internet speed in a country; see Fig. 8 (we use Akamai’s report [8] on global Internet speed as of June 1, 2017). The countries right to the vertical dotted line in Fig. 8 have a higher tracker prominence (marked red ▼). Tracker prominence values are high for Armenia, Colombia, Lebanon and Uganda, although they have relatively low Internet speed. Resource intensive tracking sources included in first-party sites may not completely load with a slow connection, increasing the rate of request

failures.⁹

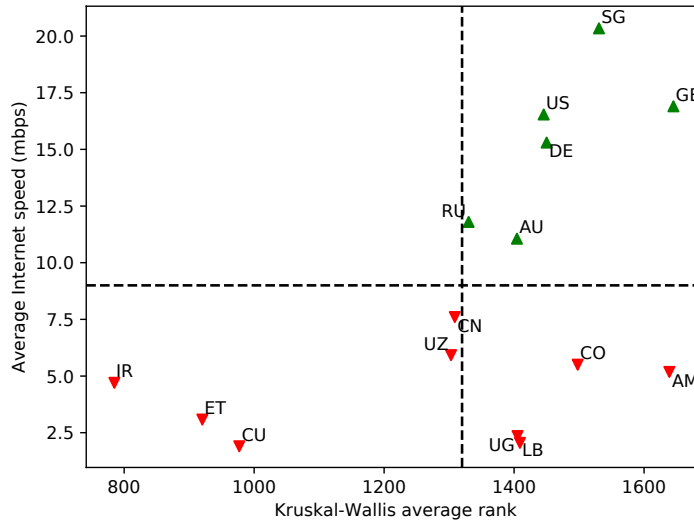


Figure 8: Internet speed vs. tracker prominence.

The failure rates of OpenWPM requests (following its re-connection attempts) vary between 11.21% – 19.04%. The highest failure rates are in United States (19.04%), Great Britain (18.26%) and Germany (18.13%). These countries still rank high in tracking prominence. We also checked the connection failures more closely, and observed that such failures are more common for trackers than the first-party sites.

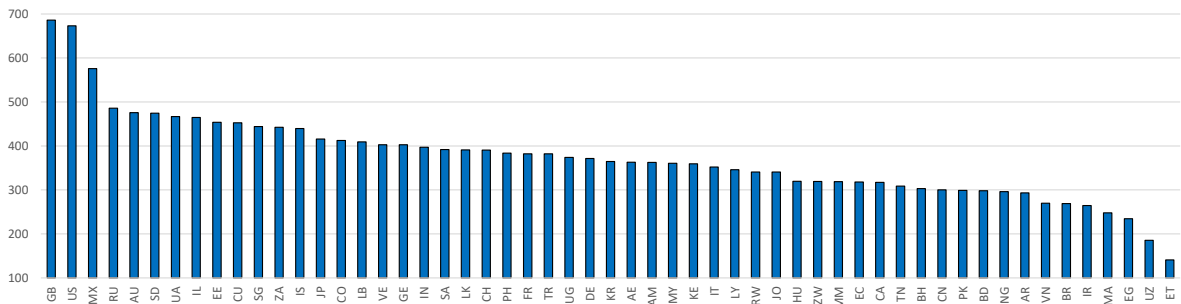


Figure 9: KW Ranks highlighting errors of HTTP/S requests.

To understand the impact of HTTP/S errors on tracker prominence, we calculated the difference of KW ranks of all HTTP/S requests vs. requests without client/server errors.

⁹The average webpage size is growing significantly, every year; in 2017, it is approximately 2.5MB, part of which is attributed to trackers, see e.g., KeyCDN (<https://www.keycdn.com/support/the-growth-of-web-page-size/>). For example, CNN’s home page size is 4.7MB and the page creates 349 HTTP requests (as of Sept. 25, 2018; tested using tools.pingdom.com).

Fig. 9 shows the KW rank of requests with errors. Although the tracker prominence and the rate of failures are not proportional in all 56 countries, the KW rank of request errors are high in Great Britain (686), United States (673) and Mexico (576), while they are lower in Uzbekistan (186) and Ethiopia (141).

3.7.2 Censorship

Apparently, there is a direct relationship between Internet/media freedom and tracking prominence—more open countries seem to attract more trackers; see Fig 10 (for clarity, we show only 15 countries, but a similar trend is observed for all 56 countries). We divide the countries into three categories based on the 2017 Freedom of the Press rankings [111]; countries marked in red (▼) are considered to be *free*, amber ones *partially-free*, and green ones *not-free*. All the not-free countries have a lower tracker prominence. Note that, although Ethiopia shows a higher percentage of trackers in Fig 3, those values are relative to the country.

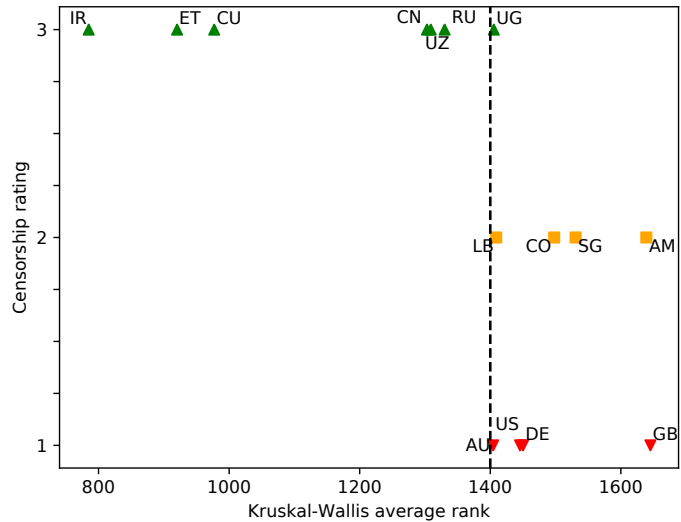


Figure 10: Censorship rating vs. tracker prominence.

We also analyzed HTTP response codes. While there were many codes other than

200 (OK), those with 403 (Forbidden Host) are interesting: majority of third parties included in these first party sites appear to be hosted on local IP addresses (e.g., 10.10.34.34, 192.168.1.1). Other studies [25, 214] also reported similar behavior in Iran as ours (87 occurrences), where DNS hijacking is used for censorship; a blocked site is redirected to a web page running on a local IP address that is accessible within Iran. In addition to Iran [25], we also observe the same behavior with SA (173), UZ (46), NG (28), GB (22), PK (19), TN (11), and US (9). Note that the 403 response code is also returned when appropriate authorization is not provided (e.g., a non-public page).

3.7.3 Browser user-agents vs. tracking

A user-agent, as sent with an HTTP request, can help identify a user’s device, browser/OS versions, and even a specific user (although not very accurately) [168]. Currently, OpenWPM supports only Firefox. We modify OpenWPM with a list of user agents¹⁰ supporting different browser/platform types. Considering four popular browsers—Chrome, Firefox, IE and Safari, we use a total of forty user agents with different desktop OSes (Windows, Mac OS X, and OpenBSD); a random user-agent is picked for each crawl. This allows an unbiased approach in simulating requests made from different browsers (instead of sending a series of requests with the same user-agent). We run the tests for each browser type at a time (i.e., each browser type is tested equally).

We summarize prevalence of top trackers (scripts vs. cookies) for common browser user-agents; see Figs. 11 and 12. Some trackers appear significantly more than the rest across all user agents for Chrome, Firefox, IE and Safari—e.g., googlesyndication.com and doubleclick.net in tracking scripts, and adnxs.com and rubiconproject.com in tracking cookies. Surprisingly, some trackers do not appear at all for certain browser types. We validated such unusual cases with manual inspection using *Chrome DevTools*¹¹, and similar

¹⁰Extracted from: <http://www.useragentstring.com/pages/useragentstring.php>

¹¹<https://developers.google.com/web/tools/chrome-devtools/>

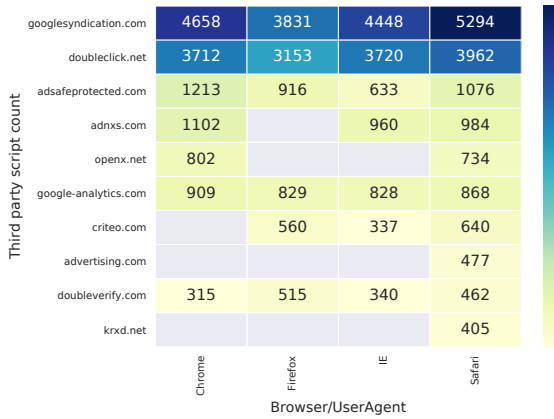


Figure 11: Tracking scripts vs. user-agents.

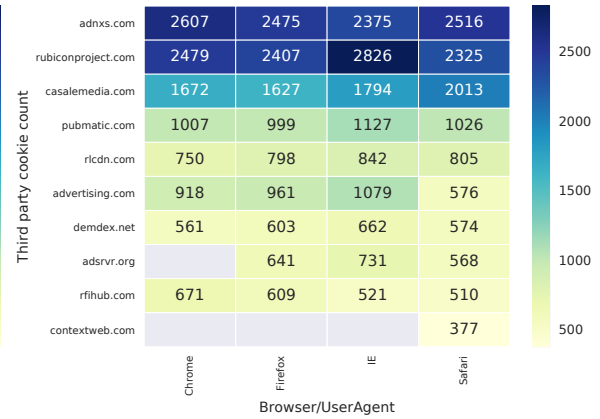


Figure 12: Tracking cookies vs. user-agents.

tools in other browsers (e.g., F12 Developer tools ¹²).

3.8 Data protection laws vs. tracking

We summarize below data protection laws in different regions and explain their relevance to tracker prominence (see Section 3.5), based on DLA Piper [84]. Overall, countries with higher tracker prominence also have relatively tougher data privacy regulations, implying whether such regulations are properly enforced.

Asia Pacific. No specific laws or regulations exist relating to data privacy except in South Korea (prominence score: 1466), with a fairly higher tracker prominence. In South Korea, cookie, log and IP information usage is governed by *IT Network Act*, and requires to get opt-out consent from users. Location information of users is regulated by the *LBS Act*. Australia (1404) leverages its *Privacy Act*, state and privacy laws to regulate e-privacy and the collection of location data to some extent.

Americas. Canada (1531), United States (1446) and Mexico (1504) have a higher tracker prominence. On top of provincial laws, Canadian *Personal Information Protection and*

¹²<https://docs.microsoft.com/en-ca/microsoft-edge/devtools-guide>

Electronic Document Act (PIPEDA) applies to consumer and employee personal information. In the US, *Federal Trade Commission (FTC)* ensures businesses take reasonable minimal data security measures to ensure consumer privacy. In contrast, South American countries (only Argentina and Uruguay are covered by DLA Piper) lack privacy laws, which also have relatively higher tracker prominence values.

Europe. EU's General Data Protection Regulation (GDPR) [100] is in effect since May 2018, governing all its member states alike. While the existing e-privacy directives in EU are complied by its member states, there is no clear indication of any reduction in tracking activities due to these regulations. Although, France, Switzerland and Italy are more strict in applying e-privacy laws compared to Germany, our results indicate France (1453) and Switzerland (1460) have comparable tracker prominence as Germany (1450). Switzerland requires explicit consent from users before data is collected, and personal data (e.g., stored in cookies) is deemed to be sensitive. France requires traffic data to be anonymized or erased, and not use location data without explicit consent. In Italy (1352), traffic data is supposed to be removed when no longer required, and cannot be held for more than 6 months. According to UK's (1645) Privacy and Electronic Communications (PEC) act, traffic data needs to be erased when not required and can be used with consent for value added services; nevertheless, UK has the highest tracker prominence.

Eurasia. Russia (1330) and Ukraine (1355) do not have specific privacy legislations, but their tracker prominence values are lower than most countries.

Middle-East. Baharain's (1463) tracker prominence is relatively high and it lacks any privacy laws. Saudi Arabia (1196) also has no privacy rules. In UAE (1355), although its penal code does not provision regulations for Internet privacy, the general laws contained therein can be applied for online privacy. Both Saudi Arabia and UAE have lower tracker prominence values. Tracker prominence in Egypt (1377) is also low; its 2014 constitution provides clear guidelines on Internet security, but not about privacy. In 2017, the Egyptian

government cracked down on encryption and circumvention tools [112].

Sub-Saharan Africa. Most countries in this region have a lower tracker prominence. South Africa (1330) doesn't appear to have laws to regulate privacy. However, Nigeria (1317) has regulations for electronic communication/privacy rights with respect to cookies and location data.

3.9 Recommendations

Although different countries may have regulations to protect privacy of users, there seem to not have a process to effectively verify compliance to such regulations. In addition, users can use privacy related browser extensions (e.g., ad blockers), or privacy browsers (e.g., Tor [285]), to safeguard their privacy from tracking, although they may not guarantee perfect privacy. The users in countries subjected to censorship, can use Virtual Private Networks (VPN), proxy servers and privacy browsers, to avoid from being eavesdropped or tracked from state actors or commercial entities.

3.10 Summary

We observe a significant variation of trackers on first-party sites between countries. Some Google trackers (e.g., doubleclick and googlesyndication) on average have an extensive presence compared to other trackers (cf. [166]). The UK and Armenia have the highest tracking prominence, while Ethiopia and Iran have the least. We observe a significant number of cookies valid for many years (>20) in EU countries and elsewhere. Several other factors also influence tracking beyond location. The countries that enjoy a greater freedom of expression and information flow show a stronger presence of trackers. We also noticed several third-party requests are censored in Iran and few other countries. Countries that are subjected to censorship, attract less trackers on websites browsed from those countries. We

also observe inner URLs of websites, have a higher tracker prominence (7-28%) compared to that of corresponding home pages, and the prominence of trackers in inner URLs varies with geolocation. In addition, countries in the European Union have the highest count of trackers, compared to other regions. Also, in general, having stronger privacy regulations does not limit tracking in any significant way — e.g., influence of GDPR on tracking has not been completely effective.

Chapter 4

Privacy analysis of government websites and mobile apps

4.1 Introduction

Tech giants such as Google, and Facebook constantly track online user behaviors to provide a better user experience, and more importantly, to improve user profiles that they curate for monetization, e.g., via advertisements. Users are aware of, and to some extent reluctantly submit themselves to such inevitable tracking on commercial websites, to get the so-called “free” services. In contrast, one may not expect commercial trackers on government web services as those are directly funded by the tax-payers’ money. Indeed, government sites are frequently used and highly trusted by users [218, 283]. Citizens use government websites and mobile apps to perform their civic obligations (e.g., taxes), query public services (e.g., waste disposal), and browse for important information (e.g., HIV, pregnancy, COVID-19). Tracking on these services could be quite revealing due to their sensitive nature [200]. If combined with user profiles on commercial sites, such tracking can make it easier to manipulate real-world user behaviors (cf. voting as targeted by Cambridge Analytica [309]). security of users; e.g., past attacks used government sites to distribute malware, ransomware,

run a botnet and cryptocurrency mining [220, 257, 295, 150, 277].

Privacy implications of web tracking have been extensively studied. Englehardt et al. [97] developed OpenWPM for large-scale evaluation of web tracking, and found Google, Facebook, Twitter and AdNexus trackers on more than 10% of Alexa top 1 million websites [13]. Privacy and security measurement studies (e.g., web tracking, HTTPS) also used sites published by Alexa, Tranco [170] and Cisco Umbrella [61], obviously due to the popularity of those top sites. However, only 9.07% government sites are present in commonly used top-million website lists [256], as government sites are used by a geographically-confined population.

Studies specifically targeting government websites (e.g., [99, 284, 14]), either focused on a specific country, or did not consider security and privacy issues in their evaluation. Recently, Singanamalla et al. [11] measured the HTTPS adoption errors and misconfigurations on government websites. In terms of tracking, a 2019 Cookiebot report [66] identified that 89% of EU government websites (e.g., nhs.uk, gov.uk) from 28 countries contained ad trackers (82% of which were from Google). However, a global perspective on commercial trackers on government services is still missing, even though governments across the world are increasingly making their services available online, especially during the current COVID-19 pandemic situation.

In this work, we perform a large-scale privacy and security measurement study on government services, using 150,244 unique government sites from 206 countries [307] and 1166 Android apps from 71 countries. We consider a website belongs to a government, if the domain name of the corresponding nameserver (`ns`) or mail exchanger (`mx`) record pertains to that government. We use a semi-automated methodology to identify 121,846 unique government domains, which we then complement with an additional 109,603 government domains from Singanamalla et al. [256, 151] (totalling 231,449 distinct domains). We then crawl the landing pages from these domains using OpenWPM [97] and measure

tracking prevalence on them; a total of 150,244 domains were successfully crawled (81,205 domains were inaccessible or inactive at the time of our crawl). We leverage the content saved from government websites to identify Google Play URLs and download 1166 government Android apps (after filtering non-government apps). To understand security and privacy exposure of these apps, we use both static analysis (MobSF [196], LiteRadar [182] and Firebase scanner [238]), and dynamic analysis techniques (using a Samsung S5 phone, with Google UI/Application Exerciser Monkey). However, we limit the security evaluation of government services due to possible legal and ethical issues. In addition, we scan all government and tracking (script/cookie) domains, and government Android APKs using VirusTotal [298] to determine the possible inclusion of malicious content in government websites/apps.

We characterize widespread tracking on government services as a *betrayal* by the governments, specifically when some jurisdictions (e.g., EU, California) have explicit laws (GDPR [100], CCPA [263]) to restrict tracking on commercial sites. Similarly, the breach of trust from compromised/malicious government apps [75] is also real.

Contributions and notable findings. We develop a comprehensive framework for collecting government sites and Android apps, and a test methodology for evaluating them, primarily focusing on privacy exposure. Our main findings include:

1. Although governments can completely prevent tracking on their online services, we found widespread use of commercial trackers on government websites and apps. Unsurprisingly, major trackers that are present on the regular web (cf. [243]) also dominate on government services—e.g., Google trackers are on both government sites (17%) and Android apps (37%). There were tracking cookies set to last for a long time—13% (19,566) of government sites contain YouTube cookies with an expiry

date in the year of 9999. These trackers are primarily due to the inclusion of commercial content (e.g., Google maps) on government sites, and the use of analytic libraries in apps. Privacy policies of 23 (out of a selected set of 227) government sites do not mention the use of any tracker. Whether explicitly mentioned or not in policy documents, these trackers can definitely correlate user activities across commercial and government services.

2. Government services from regions with strong privacy regulations such as the EU countries and the state of California (crawled from a localized VPN), also contain a lot of trackers, apparently violating their own regulations (GDPR and CCPA, respectively). For example, 49% (953/1942) and 69% (306/444) of EU and California government websites include known tracking scripts, respectively. These sites also include known tracking cookies with long validity periods; e.g., a total of 35 sites from both regions include known tracking cookies that are valid for 7984 years. Note that our crawler does not click on the cookie consent prompts, if present.
3. Surprisingly, there are government services that are potentially malicious, or load content from domains labelled as malicious as per VirusTotal (see Sec. 4.4.5); 304 government sites and 40 governments apps are labelled as malicious. In addition, 21 tracking domains (19 included in 377 sites, 2 included in 2 apps) are labelled as malicious.
4. Several government apps leak privacy/security sensitive information to trackers, or any network attacker. Examples: 23.1% (269/1166) of the apps expose device data (e.g., device model, device ID) to trackers; 7 apps send user login information in cleartext; 11 apps include hard-coded user/admin credentials and API keys; and 30 apps expose their unprotected Firebase datastores (apparently including confidential and personally identifiable information).

5. Sensitive user or government data may cross jurisdictional boundaries due to the use of CDNs and hosting providers. Notable examples: US/Delaware’s election website `elections.delaware.gov` is hosted in the UK, Australia’s `army.defencejobs.gov.au` and Somalia’s `centralbank.gov.so`, `as.parliament.gov.so` are hosted in the US.
6. We found 23 government sites from 7 countries include FullStory [114, 2] third-party script, which is used to collect the full user session (e.g., for debugging, replaying). Moreover, 5 sites expose user information (e.g., email address, search terms) to FullStory, although FullStory can be configured to limit such exposure.

We disclosed our findings on the leakage of user/admin credentials and API keys to the developers of those 11 government Android apps, but received only one response after several months (we also made several follow ups). We also reported 8 government websites flagged as malicious (by at least 5 VirusTotal engines) to site administrators/contacts of those sites, but received no response. Furthermore, we reported 38 government Android apps flagged as malicious (by at least by 1 VirusTotal engine, as the number of apps is smaller compared to government sites) to its developers, but only one developer reached out to us.

4.2 Related work

Tracking on popular websites. There exist a significant number of papers (e.g., [97, 241, 113, 183, 104, 78, 28]) on web tracking on popular websites. Englehardt et al. [97] developed the OpenWPM framework [223] to measure the prevalence of tracking on websites at a large scale. OpenWPM can measure both stateful (third-party scripts and cookies), and stateless (fingerprinting) tracking. Englehardt et al. found that only a few third-party

tracking and advertising scripts (i.e., Google, Facebook, Twitter, Amazon, AdNexus, Oracle) were present in more than 10% of the top-1M Alexa sites. Their findings also include the use of sophisticated fingerprinting techniques (e.g., WebRTC-based, AudioContext, Battery API) in top-1M Alexa sites. The additional functionalities offered by HTML5 APIs increased the effectiveness of browser fingerprinting techniques [122]. Previous work [241, 113, 104] has also studied web tracking using popular Alexa sites from a global perspective, and found differences based on geo-location and other factors (e.g., availability of data privacy policies, laws, censorship, surveillance). Hu et al. [145] found 80% of Alexa top-2K global sites contained Google trackers. Karaj et al. [160] found third-party Google scripts in 82% of web traffic (measured using crowd-sourcing efforts). Sanchez-Rola et al. [242] observed Google tracking cookies on 93% of popular sites (on the Tranco list). We use existing methodologies and tools (e.g., OpenWPM) to specifically study commercial trackers on government sites from across the world; 91% (123,115/135,408) of these sites are not ranked in popular lists (e.g., Alexa, Cisco, Tranco).

Tracking consent solutions. Online tracking consent solutions, such as Cookiebot [66], assist website owners to manage tracking activities (i.e., detect and block trackers until a user grants consent), and ensure that web tracking complies with existing data protection regulations such as the EU GDPR. Websites integrated with Cookiebot present cookie consent banners to record user preference (accept/reject cookies). Cookiebot can also measure tracking on a given website (without an integration), and was used to analyze government websites from the 28 EU member countries; over 100 unique trackers were found. Many of these trackers were from Google (82%); only Spanish, German and Dutch government sites did not contain any tracker [29]. We found that all countries in the European Union had known tracking cookies on the analyzed government websites (291 unique trackers in total). Websites also actively take measure against users who choose not to allow cookies, e.g., by deploying aggressive browser fingerprinting techniques (see e.g., [212]). We focus

on governments across the globe, and study the presence of commercial trackers on government sites, and also evaluate privacy and security issues in government Android apps.

Tracking in mobile apps. Due to the popularity of mobile apps, they also have been analyzed for privacy and security issues in the recent past, with a focus on the increasing use of tracking SDKs. Reuben et al. [34] studied 959,000 apps from US and UK Google Play stores, and found that third party tracking follows a long tail distribution dominated by Google (87.75%). Nguyen et al. [199] performed a large-scale measurement on Android apps (no mention of government apps) to understand violation of GDPR’s explicit consent. The authors found 28.8% (24,838/86,163) of apps sent data to ad-related domains without explicit user consent. Several recent studies (e.g., [237, 56]) also analyzed COVID-19 tracing apps, and highlighted privacy and surveillance risks in these apps. In contrast, we target 1166 government apps of various types (including COVID-19 tracing apps) from 71 countries and territories around the globe.

HTTPS inconsistencies on government websites. There have been numerous large-scale studies on HTTPS/TLS in general. Singanamalla et al. [256] conducted the first measurement study on the HTTPS adoption in 135,408 government websites, and found a lower adoption rate (39%) compared to commercial websites; we also found similar results (61,679/150,244, 41%).¹ They also observed the prevalence of HTTPS adoption errors (e.g., the use of insecure cryptographic protocols and keys) on these sites.

Privacy and security issues on government websites. Lapses in government websites that lead into privacy and security issues have been studied for specific countries. Csonotos et al. [70] found 52% of the analyzed Hungarian public sector websites used outdated server software versions and programming language releases; less than half of those websites used HTTP. The office of the auditor general in Western Australia [205] found 328

¹Note that as of September 2021, according to the Google Transparency Report (<https://transparencyreport.google.com/https/overview?hl=en>), 95% sites are now loaded over HTTPS on Chrome.

weaknesses in information technology processes (e.g., information security, IT operations, business continuity) used by 50 local government entities, out of which 10% were rated as significant. We focus on finding privacy and security issues (e.g., third party tracking, inclusion of content from malicious domains) of government services across the world.

4.3 Methodology

In this section, we first provide details of our government website and app collection methodology. Then, we detail our privacy analysis and measurement techniques for the collected websites and Android apps; see Figure 25 for an overview of our methodology.

For websites, we define *known trackers* as the third parties (e.g., script/cookies on first-party websites) blacklisted by EasyList and EasyPrivacy [92] filtering rules; we define the rest as *unknown trackers*. We count trackers sharing the same domain name with different sub-domains separately. Furthermore, we define Android SDKs identified as trackers by MobSF [196] as *known trackers*.

4.3.1 Collecting government sites and apps

We compile a list of government websites from 206 countries and territories by initially using a seed list, and then refining and extending it via automated searching and crawling (between July and October, 2020). We then augment our list with the website dataset from Singanamalla et al. [256, 151]; note that our site collection methodology was developed independently. We list the regions, and the count of government websites (in countries/territories of the corresponding regions) used in our study in Table 4.

Preparing the seed list. We begin by extracting an initial seed list of 14,861 government websites using several known sources [116, 83, 181], after removing obvious non-government entities (e.g., political parties). To eliminate any remaining non-government

Region	# websites
Africa	4586
Asia	60,357
Central America	2506
Europe (Non-EU)	15,148
European Union	16,681
Middle East	3209
North America	23,934
Oceania	6588
South America	15,939
Caribbean	1296

Table 4: List of regions and government website counts (countries are grouped in regions based on the categorization in [154]).

sites, we use `nslookup` [81] to query the nameserver (`ns`) and mail exchanger² (`mx`) records for each site. We then check for unique top-level domains and second/third level domains as used by various governments [306]; we then eliminate the sites that do not contain these domain suffixes in `ns` and `mx` records.

Extending the seed list. We extract the suffixes from the seed list and prepare a Google dork [250] (e.g., `site:"gov.uk"`) for each country. Then we use `googler` [315] (a command line Google search tool) to perform Google search on each Google dork and extract the search results, which may contain new domains and sub-domains. Then, we remove non-government domains from search results as explained in the previous step. We collected a total of 56,766 unique government domains/sub-domains at the end of this step.

Deep crawling to scrape inner-links. Since landing pages and inner pages of government domains collected in the previous step may contain links to other government sites, we perform a deep crawl to scrape links in the HTML page source, up to a depth of 4 levels. For this purpose, we use `Hakrawler` [167], that can find links in page source and the associated JavaScript files of crawled URLs. We randomize the URLs fed to `Hakrawler` to avoid generating a large amount of traffic to any particular web server hosting government sites. `Hakrawler` crawls only the web content hosted on government domains/sub-domains —

²Some government domains appear only in `mx` records.

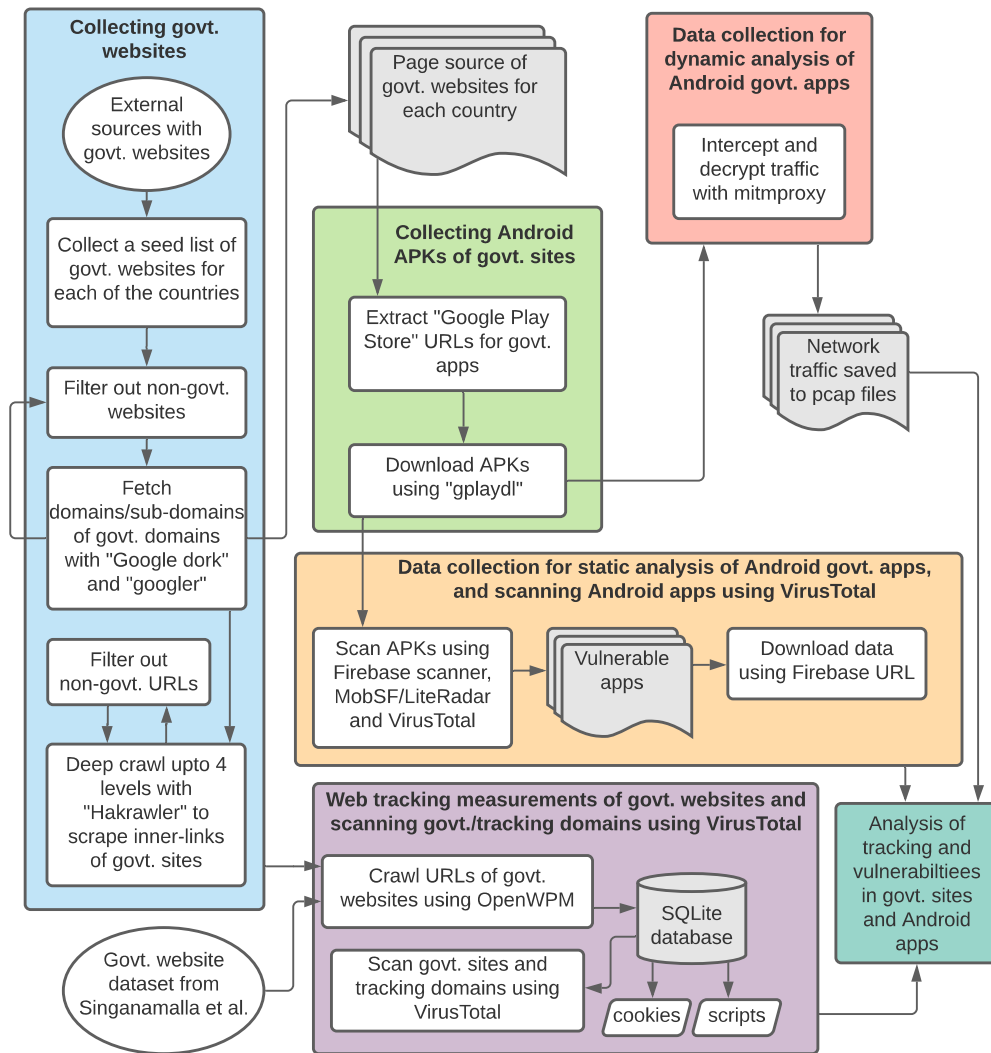


Figure 13: Overall methodology: website and Android app collection, tracking measurement on websites, and privacy and security analysis techniques used on apps.

i.e., it does not crawl any external websites (e.g., social media sites). For all links collected up to a depth of 4 levels, we filter out the following: links to common file extensions (e.g., docx, pdf, xls); links to social media websites; non-responsive links using *curl* [82]; domains not ending with known suffixes of government domains. After filtering, we obtained 15,214,100 URLs from 121,846 unique government domains. For each domain's landing page, we use *curl* to save the page source, which is later used to extract Google Play store URLs of government apps.

Complementing websites from Singanamalla et al. We add 135,416 government websites from Singanamalla et al. [256, 151] (collected using a different methodology including crowd sourcing via Amazon MTurk). After eliminating the overlaps, we had a total of 231,449 government websites, and we finally used 150,244 websites as the rest were unreachable (for various reasons, including unresponsive or unreachable servers). The top-10 countries with the highest number of websites in our dataset have a cumulative of 60% (90,047/150,244) — e.g., US (22,506, 15%), China (12,583, 8.4%), Bangladesh (12,258, 8.2%). We observe that 8.6% (12,873/150,244) domains of government websites make it into the Tranco [170] top-1M websites (cf. 9.07% in [256]). We manually verify our government website dataset (with a limited sample size of 100, selected randomly) to ensure false positives are eliminated. We summarize the regions and website counts in Table 4.

Government Android apps. Government apps do not follow a common package naming convention. Therefore, we look for URLs relating to Google Play store (i.e., `https://play.google.com`) in the page source of government URLs saved for each country. However, not all such Google Play store URLs point to government apps (some third-party apps are also linked). We run each Google Play URL with the `curl` command to fetch developer email, developer website and privacy policy website URLs. We label a Google Play URL as a government app URL in the following cases: (i) the developer email, or developer website/privacy policy URL contains `.gov.`; (ii) the developer website/privacy policy URL appears in the list of our government websites. Then for each of the government Google Play store URLs (a total of 1566), we attempt to download the app using `gplaydl` [228]. A significant number of Android apps failed to download as they are region-locked. In the end, we collected 1166 government Android apps from 71 countries. The top-10 countries with the highest number of government Android apps in our dataset have a cumulative of 641 (out of 1166, 55%) apps — e.g., India (95, 8.1%), Australia (92, 7.9%), Indonesia (91, 7.8%).

4.3.2 Measurement of trackers on government sites

We configure the OpenWPM [97] web privacy measurement framework to launch 15 parallel browser instances in headless mode. To simulate the first visit to a website, we clear the browser profile after each URL visit. We use two Azure VMs running Ubuntu server 18.04 LTS, 4vCPUs, 16GB RAM, 30GB SSD, and a physical machine running Ubuntu 18.04 LTS, Intel Core i7-9700K, 8GB RAM, 1TB HDD for our OpenWPM measurements between Nov. 5–9, 2020. A total of 150,244 websites were successfully crawled by OpenWPM (out of 231,449), and the rest (81,205) were unreachable during our crawl (e.g., website no longer exists, SSL/TLS errors, name resolution failure, disconnection by the remote-end, timeout).

The instrumented tracking metrics from OpenWPM which include HTTP request/response of both the landing page and associated third party scripts, third party cookies, fingerprinting API calls, call stack information of web requests, and DNS resolution information are saved to a SQLite database. The saved information in OpenWPM contains both stateful (i.e., scripts and cookies) and stateless (fingerprinting) forms of metrics. We then check the saved tracking scripts and cookies for third party domains; i.e., domains of scripts/cookies that do not match the domain of the government site that they are on. We also study the known tracking scripts to find techniques used for other purposes such as session replaying and web analytics (which also could directly aid user tracking).

In order to find the correlation between privacy regulations (i.e., GDPR [100], CCPA [263]) and tracking, we separately run OpenWPM with 444 California government websites (from a VPN in California), and 1942 European Union government websites (from a VPN in the Netherlands). Note that our initial OpenWPM measurements are not done using VPNs. Our OpenWPM automation does not interact with crawled government websites, e.g., to accept or reject the cookie banners on EU sites. Therefore, our automation does not accept cookie banners on sites crawled.

4.3.3 Malicious government and tracking domains

We scan domains of all known tracking scripts/cookies in government domains (150,244), and government domains with VirusTotal to check if any of these domains are labelled as malicious. Note that, at least in some cases, VirusTotal engines³ may misclassify or delay in updating domain categorization labels [217]. Therefore, to improve our labelling, we also automatically collect and use domain categories (e.g., phishing, malicious, spam, and advertisements, as assigned by different anti-virus engines), and community comments in VirusTotal⁴ (sometimes with links to detailed analysis).

4.3.4 Android apps analysis

Tracking SDK detection. We use Mobile Security Framework (MobSF [196]) to find tracking SDKs embedded in government apps (via static analysis). We load each app to the MobSF server, scan it using the MobSF REST API, and download the JSON formatted results, which include known tracking SDKs, and strings with sensitive data and dangerous permissions [21] (e.g., camera, contacts, microphone, SMS, storage and location) used by the apps. We then use LiteRadar to find the purpose of the included tracking SDKs (e.g., Development Aid, Mobile Analytics). Finally, we store these results in a local database for our analysis.

Misconfigured Firebase database. Many Android apps, including government apps, use Google Firebase [126] (a widely used data store for mobile apps) to manage their back-end infrastructure. However, due to possible misconfiguration, Android apps connected to Firebase database can be vulnerable (see e.g., [41]). Exposed data from Firebase vulnerabilities includes personally identifiable information (PII), private health information and

³<https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors> (we exclude CRDF and Quttera for their unreliable results as we observed).

⁴We used the VirusTotal API to extract community comments — see <https://developers.virustotal.com/v3.0/reference#comments>. To analyze the the community comments on malicious behaviour, we matched them with pre-determined keywords (e.g., phishing)

plain text passwords [40]. Firebase scanner [238] is used to find Firebase vulnerabilities of an app (if exists). We run the Firebase scanner [238] on each APK file, which identifies the vulnerable Firebase URLs; we then download the exposed data from the Firebase datastore URL⁵ and check for apparent sensitive and PII items, including: user/admin identifiers, passwords, email addresses, phone numbers. However, for ethical/legal considerations, we do not validate the leaked information (e.g., login to an app using the leaked user/admin credentials). Then we remove the downloaded Firebase datastore. We also promptly notify the developers of affected apps.

Dynamic analysis. We use a rooted Samsung S5 neo mobile phone with Android 7. We restrict only newly installed apps to proxy the traffic via mitmproxy [194] using Proxy-Droid [131], to avoid collecting traffic from system and other apps. A mitmproxy root certificate is installed on the phone. We also installed mitmproxy on a separate desktop machine to collect and decrypt HTTPS traffic. Both the desktop machine and phone are connected to the same Wi-Fi network. We use adb [123] to automate the installation, launch, and uninstallation of the apps. We also use Monkey [124] with 5000 events (e.g., touch, slide, swipe, click) for each app. The network traffic is captured and stored in pcap files. We use the captured network traffic to determine sensitive information (e.g., device identifiers sent to trackers, leaked hardcoded user/admin credentials and API keys) sent to external entities. We close mitmproxy and uninstall that government app before moving to the next app.

Malicious domains and apps. We scan the APK files of 1166 government Android apps with VirusTotal. We also scan domains included in apps (as found in the network traffic) with VirusTotal.

⁵The URL is of the form `<Firebase project name>.firebaseio.com/.json` (e.g., `https://mi-senado-colombia.firebaseio.com/.json`).

4.3.5 Ethical considerations and limitations

During deep crawling to scrape inner-links to other government sites, we randomize the URLs fed to the crawler, to avoid generating a large amount of traffic to any web server hosting a government site. We do not use the sensitive information (e.g., user identifiers and passwords) extracted from static and dynamic analyses of Android apps for any intrusive validations that may have an impact to the privacy of users. In addition, we did not retain any data from exposed Firebase databases. We also reached out to the internal Research Ethics Unit of our University, and explained our experiments. They approved our methodology without requiring a full ethics evaluation. We also kept them informed about our findings and contact attempts with app developers.

Obviously, our dataset does not include all the government websites and apps available throughout the world. Furthermore, during our crawling process, we may not have encountered all trackers that are time dependent [241]. We use EasyList/EasyPrivacy [92] to filter third parties (e.g., trackers, advertisers) in government websites. Some of these filtered third parties may operate in an advertising context and may not necessarily track users, or vice-versa. It is also possible that third parties blocked by EasyList rules perform the dual role of advertising and tracking. However, the presence of third-party ad/annoyance domains is not expected on government sites as government services do not rely on ad revenue. Also government websites may intentionally use third-party scripts for tracking/analytics, and we still label such activities as tracking, as there is no technical barrier for these third-parties to use analytics data also for tracking/profiling. Determining the geolocation using IP address (see Section 4.4.7) may not be accurate in some cases (e.g., CDN-fronted websites, non-CDN websites with multiple regional servers behind a load-balancer). However, this is less of a concern for our country-level attribution; e.g., Gharaibeh et al. [120] reported 95.8% accuracy for country-level IP-geolocation. We

crawled government sites from a location outside of their home countries, except for government sites pertaining to the country where the crawler is located (i.e., Canada, the Netherlands, California). Government sites of some countries (e.g., Egypt, Iran), may not properly function when accessed from outside of the country. Also, we particularly focus on Android apps due to its larger market share, and do not consider iOS apps for this work.⁶ Android apps with obfuscated code may have impacted our static analysis, but not so on our dynamic analysis. In addition, during the dynamic analysis of apps, we did not collect traffic for those apps using SSL pinning (as we could not automatically perform un-pinning).

We involve manual steps in our methodology for verification, only when automation is unreliable or challenging (e.g., verify websites crawled pertain only to governments), to ensure that our results are reliable.

4.4 Results: Government websites

In this section, we summarize our main findings on tracking and security issues on government sites.

4.4.1 Third-party tracking scripts

We found 29.9% (44,880/150,244) of government websites had one or more known trackers on their landing pages, and a total of 748 unique known trackers (524,906 total known trackers). The most common known trackers were youtube.com (19,565, 13% of websites), doubleclick.net (19,339, 12.9%) and google.com (5478, 3.6%), all owned by Alphabet; see Figure 14 for the top-10 known trackers. Note that YouTube videos and Google maps are often present on government sites.

⁶As of August 2021, according to one estimate, Android has 72.7% market-share worldwide (<https://gs.statcounter.com/os-market-share/mobile/worldwide>).

We also compared the presence of third party scripts (known trackers) by country; see Figure 15. China had a high number of government sites with known trackers (5394 sites with known trackers, out of a total of 12,583 sites, 42.9%). Russia (1623/1818, 89.3%) and Tajikistan (10/11, 90.9%) also had a high percentage of government websites with known trackers.

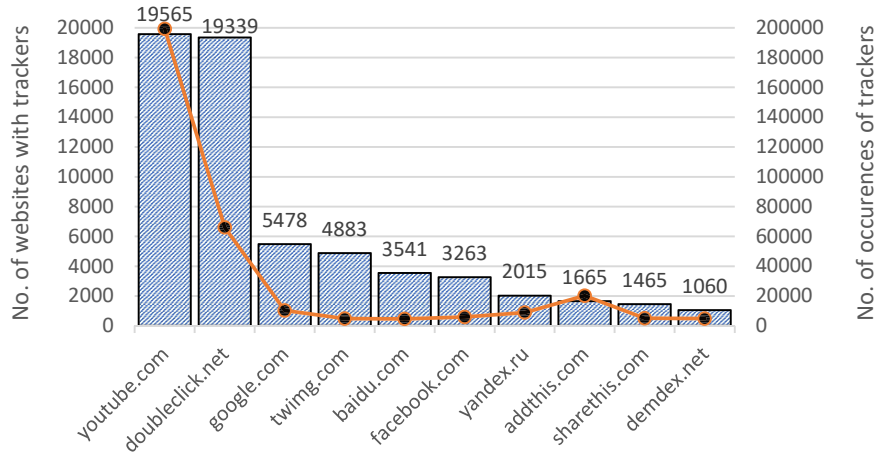


Figure 14: Top-10 known third-party tracking script sources on government sites — the bars show the number of government sites with trackers (vertical axis to the left), while the line chart shows the number of occurrences of trackers (vertical axis to the right).

We evaluated the percentages of government websites with known third-party tracking scripts for countries in different regions — see Figure 18. Notably, government websites in countries in the European Union have a relatively low percentage (14.1%) of tracking scripts, perhaps due to GDPR (although this number should be zero as per GDPR requirements). However, 49% (953/1942) European Union government websites include known tracking scripts, compared to 69% (306/444) of that of California government websites — see Section 4.4.2. We also compared the known trackers and unknown trackers hosting tracking scripts per geographic region — see Figure 17. The proportion of known trackers is high in Africa (25,394 from a total of 27,941 trackers, 90.9%), while in South America, the proportion of unknown trackers is high (40,247/101,259, 39.7%).

Session replay by FullStory third-party script. We found some government sites in Poland (11), Mexico (1), New Zealand (1), Saudi Arabia (2), Australia (3), United States

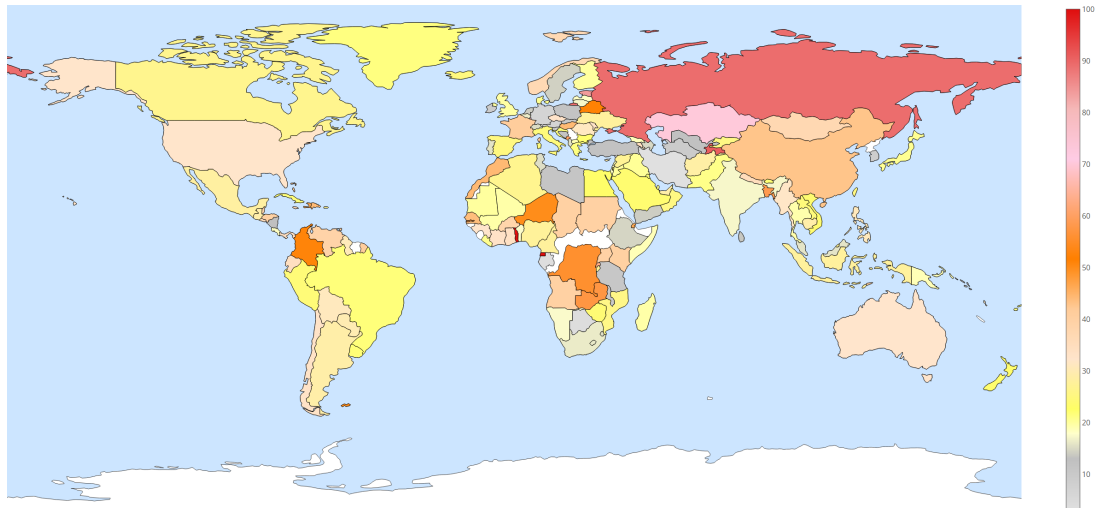


Figure 15: Heatmap of percentage of government websites with known tracking scripts in different countries (countries in white had no trackers).

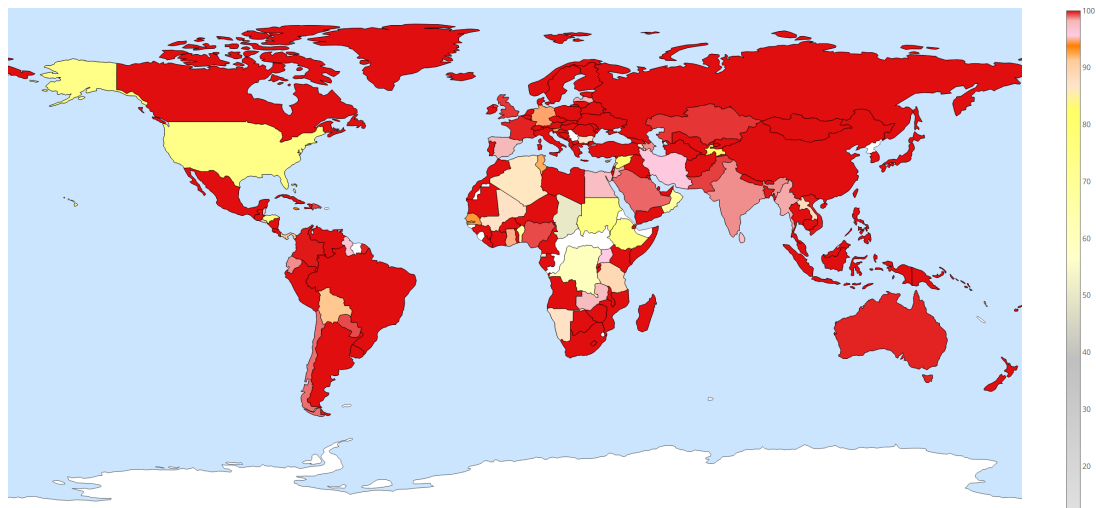


Figure 16: Heatmap of percentage of government websites with known tracking cookies in different countries (countries in white had no trackers).

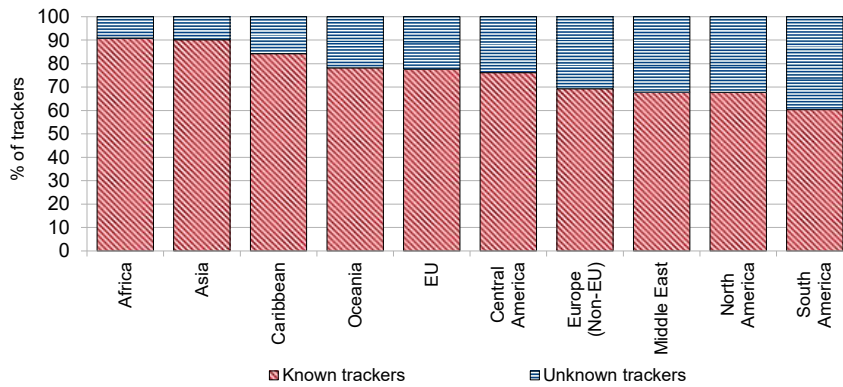


Figure 17: Proportions of third-party scripts (known trackers vs. unknown trackers) on government sites per region.

(4) and Ukraine (1) include the FullStory [114, 2] third-party script (fs.js). This script attaches event listeners to capture various events, including: button clicks, mouse movements, scrolling/resizing of windows, touch events in mobile browsers, key presses, page navigations, and network requests; all recorded events are then sent to FullStory servers. The script offers privacy options to exclude specific page elements with sensitive information (e.g., passwords, credit card numbers) to be collected/sent to FullStory servers. However, several government sites do not leverage these options, and thereby expose sensitive user information to FullStory. Examples include: `my.nzte.govt.nz` exposes a user's first/last name, email address during account creation; `durangodigital.gob.mx` exposes email address during login; several sites (e.g., `rockvillemd.gov`, `connection.homebaseiowa.gov`, `nassauparadiseisland.com`) expose search terms to FullStory; and several sites (e.g., `eservice.sba.gov.sa`, `mybusiness.service.nsw.gov.au`) also send browser fingerprinting information (e.g., ScreenWidth, ScreenHeight), and links clicked by users to FullStory. In contrast, `parliament.vic.gov.au` blocks sending search terms to FullStory.

4.4.2 Trackers on EU and California government sites

As more services are going digital, and many commercial entities' sole business model is based on profiling users, at least some governments are apparently starting to take user privacy more seriously. They are also enacting regulations to impose significant penalties to commercial online service providers for the violation of data privacy and security measures, which include: unnecessary data collection, tracking without consent, and failing to protect personal data. Prominent regulations include: the EU General Data Protection Regulation (GDPR) [100], California Consumer Privacy Act (CCPA) [263], Virginia Consumer Data Protection Act (CDPA) [297], Personal Data Protection Guidelines for Africa [153], Canadian Personal Information Protection and Electronic Documents Act (PIPEDA) [135] (and the newly proposed legislation [134]). Ironically, many governments fail to lead by example as apparent from our results. In this section our emphasis is on the impact of GDPR/CCPA on tracking.

European Union. All websites must comply with GDPR [100] when accessed from any EU member state. GDPR is an opt-in privacy regulation (e.g., user consent must be obtained before tracking them). We found 49% (953/1942) EU government websites include known tracking scripts; note that we visit these sites via OpenWPM from a VPN in the Netherlands. Most tracking scripts (524, 27%) on these sites are served by Google, followed by Facebook (54, 2.8%), Cloudflare (24, 1.2%), and Twitter (23, 1.2%). We also observed companies (e.g., CookieLaw and Cookiebot) that provide solutions (e.g., provision of cookie banners) to adhere to GDPR, included scripts on EU government websites that are categorized as trackers by EasyList/EasyPrivacy [92]. Notably, 24 (out of 1942) government sites (e.g., Germany, Lithuania, Denmark) include tracking cookies that are valid for 7984 years; see Table 5.

California websites. Websites accessed from California are subjected to CCPA [263, 63],

Validity period	# sites	Example trackers
7984 years	24	iteimproveanalytics.io, snoobi.com, nr-data.net
16 years	1	trafic.ro
1 – 5 years	27	statcounter.com, omtrdc.net, adverticum.net
3 – 6 months	11	pubmatic.com, innovid.com

Table 5: Cookie validity periods on EU government sites.

which is an opt-out privacy regulation. For example, CCPA does not require websites accessed from the state of California to provide explicit cookie consent (unlike GDPR). We observed 306/444 (69%) California government websites include known tracking scripts, mostly from Google (163/444), followed by CivicPlus and Microsoft (each 22, 5%), Siteimprove (13, 2.9%), and Facebook (11, 2.5%). Note that we crawled these sites from a VPN located in California. We also found website design companies serving governments (e.g., CivicPlus, Revize) included tracking scripts in government websites. In addition, 11 (2.5%) California government sites set cookies that are valid for 7984 years; see Table 6.

Validity period	# sites	Example trackers
7984 years	11	siteimproveanalytics.io, rfihub.com, nr-data.net
10 years	1	webtrends-live.com
1–2 years	15	stackadapt.com, scanscout.com, rubiconproject.com
1–6 months	7	krxd.net, demdex.net

Table 6: Cookie validity periods on California government sites.

4.4.3 Third-party cookies

We found many third party persistent cookies (i.e., cookies that do not expire after a session) set by known trackers, with varying validity periods; see Table 7. YouTube is the most common tracking cookie set in a large number of government sites (56,444 out of 150,244 government sites, 37.6%). About 11.5% (17,312) of government sites included cookies set by YouTube that expired within a month. YouTube cookies on 13% (19,566) of government sites are set to expire in the year 9999. Cookies set by YouTube are used to associate site visits with a Google account (if logged in) and contain information on browsing behaviours of users [312]. Also, doubleclick.net cookies on government sites (18,219, 12%) were set to expire between 1-5 years. 14 known trackers set cookies with over 5-year expiry periods; these trackers provide services in sectors including: ads/analytics (nr-data.net, cnzz.com, rezync.com, bitrix.info, 51.la), business (gemius.pl, pixlee.co), social networking (twimg.com, ok.ru), travel (sinoptik.ua), news (cctv.com) and file sharing (radikal.ru).

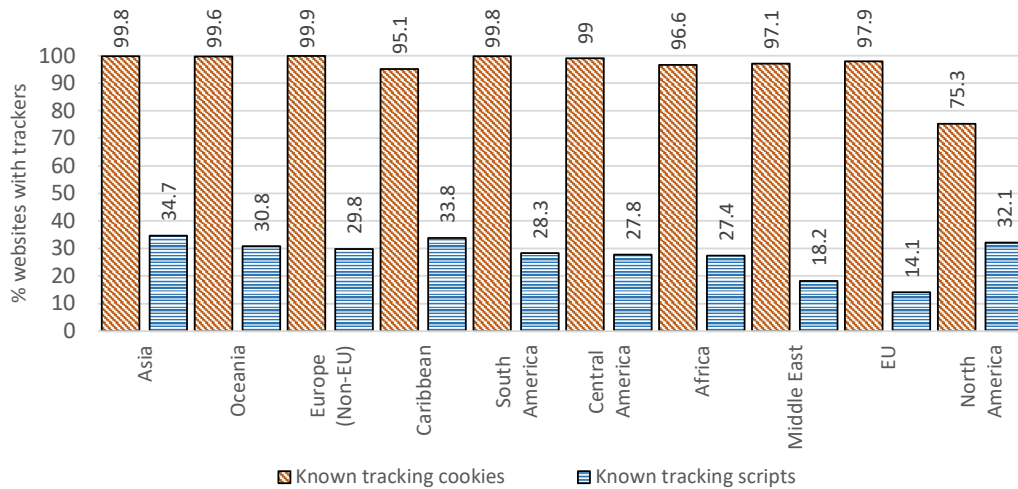


Figure 18: Known trackers (third-party scripts/cookies) on government sites by region.

We found government websites in 112 countries set known tracking cookies on all of its websites (20,558/150,244, 13.7%). The percentage of government websites setting known tracking cookies is over 80% in 170 (out of 206) countries; see Figure 16 (also Figure 18 for region-specific prevalence of these tracking cookies). The lowest percentage of

government websites with known tracking cookies was from North America (5783/7681 websites, 75.3%). The US government sites had the lowest proportion known tracking cookies (5417/7314, 74.1%), in part possibly due to California Consumer Privacy Act (CCPA) [263]. In contrast, despite GDPR [100], the percentage of government websites with known tracking cookies in the European Union was very high (2306/2355, 97.9%).

Tracker	# sites	Cookie expiry		
		> 1m & ≤ 1y	> 1y & ≤ 5y	> 5y
youtube.com	56,444	19,566	0	19,566
doubleclick.net	37,632	50	18,219	18
google.com	7731	5439	130	1
yandex.ru	4113	1995	81	2005
addthis.com	2589	921	1665	0
adsvr.org	1045	1045	0	0
rlcdn.com	793	793	0	0
bluekai.com	779	779	0	0
tapad.com	626	626	0	0
id5-sync.com	559	278	0	0

Table 7: The top-10 known tracking cookies and their expiry periods (m=month, y=year).

4.4.4 Fingerprinting APIs

We found many instances of calls to various fingerprinting APIs on government websites. Examples include: Storage (5,355,626), CanvasRendering2D (7,615,438), window.navigator (3,349,296), HTMLCanvasElement (1,102,482), hardware related APIs⁷ (230,426), window.screen (99,504), audio related APIs (16,274), window.navigator.geolocation (8334), RTC (2655). APIs related to Audio, hardware, RTC and window.screen can track users for a relatively longer period as the characteristics of those fingerprints generally remain static for a long time [225, 258]. We found other privacy implications from the fingerprinting APIs: Window.navigator.geolocation gives a website access to the location of user device (called 8334 times), and RTC is used to discover local

⁷Hardware fingerprinting APIs include: window.navigator.hardwareConcurrency, window.navigator.mediaDevices, window.navigator.getGamepads, window.navigator.oscpu, window.navigator.platform, window.navigator.vibrate and window.navigator.maxTouchPoints.

IPs without user permission [97] (called 498 times). Such a combination of multiple fingerprinting APIs can be used to identify a user with a high precision [97], and reportedly being used to bypass EU GDPR cookie restrictions [212].

4.4.5 Government sites and tracking domains flagged as malicious

We found 0.2% (304) government sites were flagged as suspicious or malicious by VirusTotal (at least by one engine). We skipped the sites flagged as malicious by *Quttera* and *CRDF* VirusTotal engines, as the categorization returned by those engines were inconsistent. In addition, we only considered the sites that apparently were used for malicious purposes according to VirusTotal category labels and community comments, containing keywords, including: malware (41 domains), compromised (51), infection (71) spyware (36), fraud (6), weapons (3), command and control (5), bot networks (2), and callhome (4). Top 3 countries with sites flagged as malicious include Indonesia (112 out of 304), China (30) and the US (14); example sites include Royal Thai air force (`rtaf.mi.th`), Palestine civil defence (`pcd.gov.ps`), Iran health insurance organization (`ihio.gov.ir`) and Yemen parliament (`yemenparliament.gov.ye`).

We also found 15 malicious domains host known tracking scripts in 377 government sites as per VirusTotal (at least by one engine); see Table 8. We used the same procedure as for government sites to scan tracking domains with VirusTotal. 8 (out of 51) malicious domains set cookies on 311 government sites; see Table 9. We observed `50bang.org` set cookies on 299 government sites.

4.4.6 Privacy policies in government websites with trackers

For this analysis we leveraged 551 privacy policy URLs extracted from the government Android apps (see Section 4.3.1). We found only 41.2% (227/551) of the corresponding government sites included trackers (scripts/cookies). 23/227 sites do not mention the use

Malicious type	Tracking domains	# Govt. sites (example countries)
Malware, malnets, malvertising	iclickcdn.com, qdatasales.com, graizoah.com, 50bang.org, popcash.net	320 (China, India, Pakistan)
Browser hijacking	otrware.com	1 (Brazil)
Adware, unwanted redirects	newrrb.bid, supercounters.com, tradeadexchange.com	43 (Indonesia, Myanmar, Vietnam)
Potentially unwanted program	coinpot.co	4 (Bangladesh, Kyrgyzstan)
Spam	freecounter.ovh	3 (Colombia, Malaysia, Pakistan)
Suspicious	ufpcdn.com, dprtb.com, loulouly.net, adhitzads.com	6 (Indonesia, Malta, USA)

Table 8: Tracking scripts included from potentially malicious domains.

Malicious type	Tracking domains	# Govt. sites (example countries)
Malware and malnets	pingclock.net, qdatasales.com, 50bang.org	303 (China, Malaysia)
Potentially unwanted programs	iyfsearch.com, coinpot.co, rtmark.net	4 (Bangladesh, Kyrgyzstan)
Suspicious	ufpcdn.com, remarketingpixel.com	4 (Kenya)

Table 9: Tracking cookies set by potentially malicious domains.

of tracking services in their privacy policies — based on matching the policy content with a set of predefined keywords (e.g., analytics, 3rd party, Google, Facebook, Twitter, LinkedIn) using *Policy Highlights* [289]. Government sites with top unique tracking domains, but not emphasizing the use of tracking services in their privacy policies include `privacy.gov.ph` (8) — national privacy commission of Philippines, `fsq.moh.gov.my` (5) — food safety and quality division of Malaysia. The unique tracking domains in these government sites include `facebook.com`, `facebook.net`, `google.com`, `google-analytics.com`, `googletagmanager.com`, `gstatic.com`, `youtube.com`, `ytiming.com`, `wp.com`. There were 9.3% (21/227) privacy policies of government sites that are not written in English (we could not translate 6 of them). There were also 11 privacy policy URLs of government sites that no longer exist.

4.4.7 Foreign-hosted government sites

We extract the DNS resolution information of the crawled government sites from OpenWPM to find the IP of each domain. Then using *geoipllookup*,⁸ we determine the geolocation and Autonomous System Number (ASN) details of each IP address. Singanamalla

⁸<https://linux.die.net/man/1/geoipllookup>

et al. [256] found 94.5% (127,327/134,685) of government sites are either hosted privately or by an unknown hosting provider. In contrast, our analysis focused on government sites hosted in foreign countries. We observed 194 countries host their site content using services from a foreign country; e.g., 2.2% (489/22,506) websites from the United States and 2.9% (370/12,583) websites from China are hosted outside these countries. These sites are hosted by cloud providers (i.e., hosting/CDN providers) with data centers around the globe; Wix (102) and Akamai (67) host most of these sites for the United States, while Quantil (202), Cloudflare (39) and Alibaba (25) hosted most sites for China. Some countries in Africa host all their government sites (in our dataset) outside: Chad (5), Congo (9), Equatorial Guinea (2), Somalia (16), Togo (3). Most prominent government sites (10) in Somalia (e.g., `centralbank.gov.so`, `as.parliament.gov.so`) were hosted by a provider (Unitedlayer) in the US.

We analyzed 1466 government websites, which are likely to be hosted at a foreign provider, not at CDNs due to the fact that ASN names of these websites did not contain a CDN listed in [50], and their IP addresses remained static and at a foreign geolocation when accessed both from IP addresses in Canada and in the Netherlands. We also found the categories [201] of these websites by parsing the text within meta tags of request headers—to determine if these sites serve any sensitive/critical purposes. Notable categories of these sites include: election (e.g., US/Delaware’s election website `elections.delaware.gov` hosted in the UK); defence (e.g., Australia’s `army.defencejobs.gov.au` hosted in the US); police (e.g., Australia/Victoria’s `policecareer.vic.gov.au` hosted in the US); courts (e.g., a New Zealand district court website: `districtcourts.govt.nz` hosted in the US); immigration (e.g., Papua New Guinea’s `immigration.gov.pg` hosted in Australia); and airports (e.g., Kenya’s `kaa.go.ke` hosted in the Netherlands).

4.5 Results: Government Android apps

In this section, we present privacy and security issues found in government Android apps using static and dynamic analysis methods.

Static analysis results: Tracking SDKs and exposed databases. From MobSF, we found a total of 1647 tracking SDKs (59 unique) in 1166 apps. With LiteRadar, we checked the usage types of these SDKs (e.g., *Google Mobile Services* is used as a development aid). Similar to government websites, most tracking SDKs were also from Google (611/1647, 37.1%). Other tracking SDKs include Facebook (105/1647, 6.4%), Microsoft (34/1647, 2.1%) and One Signal (48/1647, 2.9%). Note that Google tracking SDKs are used for ad and mobile analytics. Although the collection of analytics can help provide a better user experience and improved protection (e.g., fraud detection [206]), it can also be effectively used for tracking/profiling.

We found that 2.57% (30/1166) government Android apps possibly exposed their Fire-base databases with sensitive user information (as apparent from the data types); however, we did not verify/use/store this info (deleted immediately after checking the data types). Notable examples: an official app of the Colombian senate (*gov.senado.app*), and a real-estate regulation app from the government of Saudi Arabia (*sa.housing.mullak*) apparently leak user names and passwords.

Dynamic analysis results. Here we report the vulnerabilities found by inspecting the pcap files collected from our dynamic analysis (see Section 6.3.5). 7 apps from Bangladesh, Brazil, India, Malaysia, Nigeria, Palestine, United Arab Emirates sent login information over clear text via HTTP. These apps provide various services, including: crowd funding (*com.synesis.donationapp* in Bangladesh); provisioning birth/death/marriage certificates, and property tax details (*in.gov.lsgkerala.mgov*, in Kerala, India); services for teachers (*com.trcn.teachers*, in Nigeria); anti-drug volunteer management (*my.gov.onegovappstore.skuadaadk*, in Malaysia); and salary payments and other

services for government employees (ps.gov.mtit.mservices, in Gaza, Palestine). One of these applications (com.trcn.teachers) sent traffic in the clear to an IP address belonging to an advertising/marketing service.

From the decrypted traffic from our mitmproxy, we observed 11 apps leaked hard-coded (default) user/admin credentials and API keys; see Table 10. We disclosed our findings to the app developers, and one replied mentioning that the credentials we observed were for an experimental feature which is now discontinued. For ethical/legal considerations, we did not use the leaked passwords observed for any form of validation. The services offered by these apps include crowd funding, information of leisure activities at beaches and parks, driver training and road rules, lodging of complaints, and provision of various digital resources.

We also found 23.1% (269/1166) government Android apps sent device data such as device model, and device ID to known trackers. Such device data can be used to passively track users by fingerprinting their devices. Most data types used for tracking were collected by *Branch Metrics* (device ID, device model, IPV4, screen DPI, height, and width) and *Unity Technologies* (device ID, device model, hardware name, screen DPI, height, and width).

Government apps and 3rd-party domains flagged as malicious. 40/1166 government apps from 22 countries were flagged as malicious by VirusTotal (at least by one engine). 10 of these apps contained a stealthy malware [142] disguising as a legitimate process executing harmful tasks (one of these apps removed the malware in a newer version), 3 apps included a stealthy adware showing as an ad blocker for Android devices (Android.WIN32.FakeAdBlocker.a), 2 apps included obfuscated malicious software

Info leak	Country	App type
Default/admin user ID and password	Australia	Parking info/directions to public hospitals
	Bangladesh	Crowd funding platform for nation building (Ek Desh)
	Brazil	Quality information of beaches
	Cambodia	Info on new driver training and road rules
	Pakistan	Communicate and provide information to public on natural disasters
	Pakistan	Lodge complaints against federal government agencies
	Portugal	The European Economic Area (EEA) program
	Singapore	A citizen-science platform for the National Parks Board (NParks)
	UK	Access services offered from local council
	API key	Afghanistan
Bangladesh		Used for the “Digital Bangladesh” initiative

Table 10: Exposure of sensitive information from Android apps (observed in the decrypted traffic via mitmproxy).

that installs other malware (Trojan.Trojan.Dropper.AndroidOS.Hqwar.bb). We also observed calls to 2 malicious 3rd-party domains by government apps. According to VirusTotal community comments, 2 apps (com.linkdev.dhcc.masaar and com.rajerawanna offered by United Arab Emirates and India, respectively) made calls to a malicious domain (`api.ipify.org`) that is infected by Cobalt Strike [195].

4.6 Discussion

In this section, we discuss privacy implications of our findings, and list a few recommendations to mitigate these issues.

Commercial trackers. Commercial websites are heavily tracked by the top tech giants such as Google, Facebook (see e.g., [160, 242]). Both government websites and Android

apps contain a significant number of such trackers; e.g., 17% of government sites and 37% of government apps contain Google trackers. Such commercial tracking is unexpected and may surprise many privacy-conscious users. Governments may want to engage citizens more actively by integrating social media resources on their websites, or attempt to understand their users' needs through the use of commercial analytical services; however, exposing their users to commercial trackers should be taken more seriously. We found 10% of the analyzed privacy policies did not even mention the use of tracking services in the corresponding government sites (see Section 4.4.6).

Out-sourcing app development. We found 19.8% (231/1166) apps were built by developers with non-government email addresses (137 with Gmail), indicating that at least some of these apps were developed by third-parties. Such out-sourcing may introduce the risk of leaking sensitive information, supply-chain attacks.⁹

CDNs and foreign hosting providers. Many web services, including some government services, are adopting cloud platforms (e.g., Microsoft Azure) for scalability and cost reduction. We observed several government sites that supposedly deal with sensitive user information (e.g., election, police, courts, defence, immigration, airports) were hosted in a foreign country. Privacy policies of these government sites (e.g., `elections.delaware.gov` — see Section 4.4.7) do not mention anything about such outsourcing. The use of foreign hosting providers and CDNs undermine the control of the hosted data; even if the backend databases remain at a government-owned facility, user data may still be (temporarily) available to the server admins of the CDNs/hosting providers, and violate data sovereignty.¹⁰ Although CDN hosting providers allow choosing a particular location to serve traffic, the closest location of the edge server/data center may not be within the country owning the site. There are many countries where CDNs have no data centers [50].

⁹Cf. the recent SolarWinds incident: <https://www.cisecurity.org/solarwinds/>

¹⁰Several governments are considering legislation on these issues—[wikipedia.org/wiki/Data_sovereignty](https://en.wikipedia.org/wiki/Data_sovereignty); see also the French government agreement with Google and Microsoft [232].

Malicious domains. Government sites and apps that are flagged as malicious, or include content from third parties (e.g., scripts, cookies) labelled as malicious, can harm users and diminish their trust. Unfortunately we found such malicious sites/apps on government services (304 government sites and 40 apps were flagged as malicious by VirusTotal). Governments should scan their websites/apps regularly to detect such domains.

App vulnerabilities. We found 7 government apps expose cleartext user login information, 11 apps include hard-coded (possibly admin) credentials and API keys, and 30 apps expose their unprotected Firebase datastores — all of which can enable attackers to harvest PII at a large scale.

4.7 Recommendations

Since many governments continue to move to digital platforms, the relevant government authorities responsible to ensure privacy in each country/region should periodically review government websites and mobile apps for tracking, privacy and security exposures, at least to comply with their own legislation.

We strongly recommend developers to use HTTPS properly (cf. [256]), not to rely on cloud-hosted mobile backends such as Google Firebase (exposing user data to commercial operators), and not to include admin API keys/credentials in the app code (possibly exposing user data to anyone). Security issues regarding the use of cloud-based mobile backends have been analyzed in recent work [321, 17], and developers should check their apps and servers for similar issues.

Government developers also need to be aware of privacy implications of using commercial JavaScript libraries and mobile SDKs, as user tracking is at the core of many of these libraries/SDKs. Clearing the browser history or the use of private browsing mode is not effective against fingerprinting attacks, which are actively being deployed to defeat cookie consent [212]. Thus, third-party scripts should be analyzed to check for the presence of any

fingerprinting APIs, especially if the APIs are not essential for the service’s functionality. Similarly, the use of session replay scripts (e.g., FullStory) should be avoided, or at least configured properly to reduce tracking and data exposure.

4.8 Summary

Despite being publicly funded by tax payers money, government services enable commercial trackers to collect data about citizens virtually everywhere across the globe. From our analysis of 150,244 government websites and 1166 government Android apps, we found Google dominates in tracking, closely resembling the same trend as in the commercial domain, which is largely powered and monetized by tracking/profiling; cf. [145]. Many government sites include Google maps, YouTube videos, analytic services, and social bookmarking services (e.g., *AddThis*), that allow to track users from commercial entities. YouTube cookies on 13% of government sites are set to expire in the year 9999. These YouTube cookies can associate browsing behaviours of users with a Google account (if logged in). Despite GDPR, there were 98% of government sites with known tracking cookies in the European Union. We found government sites in 7 countries use session replay, that can expose sensitive information of users. We also found 59 unique tracking SDKs on the analyzed government apps, and 2.6% of government Android apps exposing Firebase database endpoints with sensitive user information. In addition, 7 apps sent login information over clear text, and 11 apps leaked hard-coded user/admin credentials and API keys. There were 23% apps that leaked device data (e.g., device model, device ID). We also found 0.2% and 3.4% of the analyzed government websites and Android apps, are flagged as malicious by VirusTotal, respectively.

A downside with government services compared to commercial services is that users have no choice in terms of switching to another provider.

Chapter 5

Privacy analysis of hospital websites

5.1 Introduction

Increased tracking of online user behaviours has become the norm for most commercial web services [173], although users can still choose a relatively less privacy invasive service, e.g., between search engines such as Google vs. DuckDuckGo. On the other hand, some websites (e.g., government health and hospital services) do not have any alternatives [239], should a user identifies potential tracking activities. With the COVID-19 pandemic, more health services are being offered online to limit the spreading of the virus—e.g., a general practitioner can be channeled in minutes, around the clock [185], without having to wait for an in-person meeting. As such, patients are able to consume health related services from online services with a few clicks — book appointments, health checkups, and view medical results. Unlike the interactions with other commercial websites, a variety of sensitive information items (e.g., identity information, health status, mental health, reproductive care including abortion, substance abuse) are exchanged with hospital sites. These sensitive information can be leaked to third-parties if trackers/session-replay scripts are deployed on hospital websites. Disclosure risks of such sensitive information may include discrimination, social stigma and physical harm.

Privacy and security of health care systems is paramount, and appropriate policies to safeguard its users needs to be enforced [38]. However, lapses in the deployment of such effective measures are common. For example, a German security firm (Greenbone Networks) found that medical files of 107 million medical images (e.g., X-rays, scans) of Indian patients were leaked and made available online [275]. These medical records happen to contain various sensitive information of patients (e.g., patient name, date of birth, medical institution name, ailment, physician name). In another incident, computer systems of a major hospital chain, with hospitals in more than 400 locations, failed when it was hit by a ransomware attack [198]. Stolen health records may have a higher demand (cf. credit card numbers) in the darkweb [7]. Similarly, the cost to remediate breaches in health care is also high [7].

There are several studies (e.g., [317, 200, 169, 234]) relating to privacy of health services, but they target a specific geographical location. Robinson [234] analyzed 210 public hospital websites in Illinois, USA and found 94% of websites include trackers on them; most common trackers on these websites include Google Analytics (74%), Google (88%), and Facebook (26%). Niforatos et al. [200] analyzed 61 US hospital websites, and found they collect information relating to advertisements (61, 100%), third-party cookies (55, 90%) and session recording (14, 23%) services. Most of these trackers are from Facebook (40, 61%) and Google (54, 89%).

In this work, we perform a large scale web privacy measurement study of hospital websites, using 19,635 hospital websites from 152 countries. We collect hospital URLs from several sources (e.g., [234, 136, 74]) by scraping the source code of the corresponding web pages. Thereafter, we crawl the extracted hospital website URLs using the OpenWPM [97] web privacy measurement framework; 152 sites were unreachable. To the best of our knowledge, this is the first measurement study on the privacy/security of hospital

websites, performed at a global scale. We analyze the instrumented tracking metrics (third-party scripts/cookies, fingerprinting APIs) using the OpenWPM database. We filtered the websites using session replay services, and we inspected the potential sites using session replay with *HTTP Toolkit* [144] to identify specific information leaked (e.g., date of birth). We also use VirusTotal [298] to identify hospital sites and domains hosting scripts/cookies that are malicious.

Contributions and notable findings.

1. We develop a framework to collect hospital websites from various external sources, and a test methodology to evaluate these sites for possible privacy exposures.
2. We found that 699/19,483 (3.6%) hospital websites included session replay services — e.g., *FullStory*,¹ *Yandex*,² *Hotjar*.³ 91/699 (13.0%) of these websites belong to EU hospitals. We observed users' information was sent from these hospital sites to third-party servers (*FullStory*, *Yandex* and *Hotjar*). The information sent to these external servers (owned by session replay services) include sensitive information such as phone number, date of birth, user credentials, residential address, passport information, booked medical services.
3. We found widespread use of commercial trackers on hospital websites. Major known trackers⁴ include Google, Addthis, Facebook and Baidu. We observed 10,417/19,483 (53.5%) hospital websites included tracking scripts/cookies. There were tracking cookies set to last for a very long time — 5.8% (1136/19,483) of sites included 1713 known tracking cookies expiring in the year 9999. These trackers are embedded in analytic services, and other third-party services (e.g., Google maps) on landing pages of hospital websites.

¹<https://www.fullstory.com/session-replay/>

²<https://yandex.com/support/metrica/general/counter-webvisor.html>

³<https://www.hotjar.com/session-replay-software/>

⁴We define a *known tracker* as the third-party (e.g., script/cookie on a first-party website) blocklisted by *EasyPrivacy* [92] filtering rules.

4. We observed hospital websites in Oceania (61.7%, 140/227) and North America (60.1%, 2805/4666) included a large proportion of known tracking scripts, compared to Asian hospital websites (39.6%, 2844/7183). Known tracking cookies were set in less than 15% of hospital websites except for North America (8186/28,960, 28.3%). Known trackers in China are location specific, perhaps due to the use of alternative local services, as foreign web services (e.g., Google, YouTube, Facebook) are mostly blocked in China.
5. We found 33/19,483 hospital websites were flagged as malicious by at least 3 security engines used by VirusTotal (e.g., *cliniqueelmenzah.com*, *mathahospital.org*). Additionally, 11 and 18 domains of known tracking scripts and cookies were flagged as malicious by at least 3 security engines, respectively. We have notified administrators for 18 of these websites about our findings; no contact information were available for the remaining 15 websites.

5.2 Related work

Web tracking measurements. There are many past studies that measured the privacy exposures from a variety of popular web applications and mobile apps. Englehardt et al. [97] implemented the OpenWPM web privacy measurement framework to identify online tracking behaviours of websites, and used their framework to measure tracking in top-1M sites. The authors found Google and Facebook trackers dominate in tracking websites. Samarasinghe et al. [241] measured web tracking in top-1K sites from 56 countries, and found Google trackers are highly prevalent on those sites (irrespective of the location), and many cookies were valid for more than 20 years. Acar et al. [2] extended OpenWPM to investigate attacks that exfiltrate data using third-party scripts (i.e., misuse of browsers' internal login managers, social data exfiltration, whole-DOM exfiltration), and found sites that leak

sensitive user information (e.g., credit card information, medical details, passwords) to session replay services. Xuehui et al. [145] studied tracking in top country specific sites (in Alexa [12] list) from 4 countries (UK, China, Australia, US), and found tracking behaviours that are specific to those countries — e.g., users in China were tracked less than those in the UK. Google Analytic is the most common tracker in 74% of hospital websites. Papadogiannakis et al. [212] found more than 75% of tracking activities happened even before interacting with the cookie banners, or after users reject all possible cookies. We measure tracking in hospital sites from 152 countries around the world, and found level of tracking in countries located in different regions vary — e.g., proportion of third-party scripts in North America is relatively higher compared to that of other regions; i.e., percentage of hospital websites with third-party scripts and cookies in North America was 60% and 29%, respectively. In addition, we observe location specific trackers (e.g., *baidu.com* on Chinese hospital websites).

Privacy and security issues in health related websites. Past studies on privacy and security issues of hospitals targeted hospitals only from a specific or a few jurisdictions. Zheutlin et al. [317] performed a study of patient data tracking on 86 pharmacy websites. The authors found that 76.4% of these websites included ad trackers; other tracking methods used include third-party cookies, session monitoring⁵ (using *Blacklight* [35]), keystroke capturing, sharing data with top tracking entities (e.g., Google, Facebook). Joshua et al. [200] studied tracking on 61 US hospital websites, and found among other forms of tracking, 14 (23%) websites used session recording services to track users. Celine et al. [169] studied how caregivers’ access to patient portals may jeopardize user privacy and security. The authors found 69/102 (68%) hospitals provided proxy accounts to caregivers; 94/102 (92%) hospitals were asked about password sharing between patient and caregiver, and 42/92 (45%) endorsed such practice. Robinson et al. [234] studied 210 public hospital websites

⁵Session monitoring reveals only the use of tracking technologies in a browsing session, but not able to replay a recorded session.

in Illinois, USA, and found 94% of hospital websites included an average of 3.5 trackers. Wesselkamp et al. [305] found advertising cookies performing cross-site tracking in health related websites (i.e., for booking appointments). We found 10,417/19,483 (53.5%) hospital websites included tracking scripts/cookies. Google dominates in tracking hospital websites. Third-party scripts included in 699/19,483 (3.6%) hospital websites sent user information to external session replay servers (*FullStory, Yandex, Hotjar*). In addition, we observed 33/19,483 hospital websites were flagged as malicious by VirusTotal [298].

5.3 Methodology

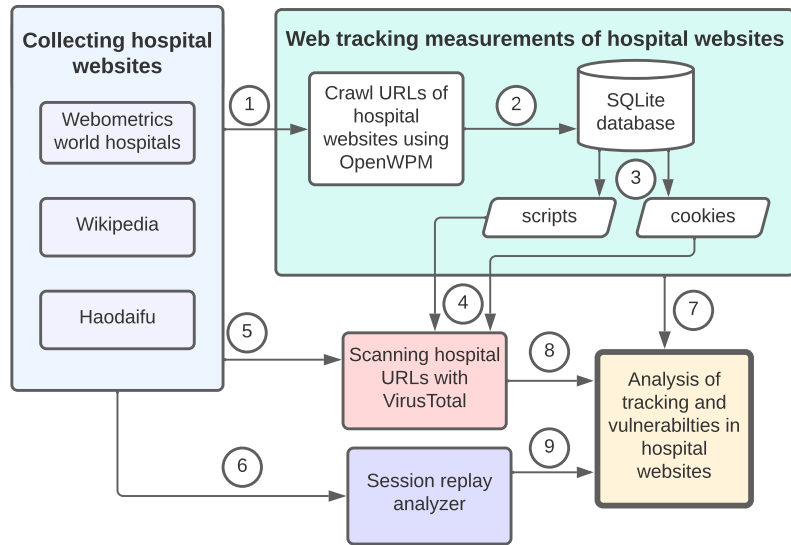


Figure 19: Overview of our methodology: hospital website collection and tracking measurement on websites — steps ①, ⑤, ⑥ represent hospital websites served as input to OpenWPM, VirusTotal scans, session replay analyzer, respectively; step ② is instrumented data saved to OpenWPM database; step ④ is third-party script/cookie domains fed to VirusTotal scans; steps ⑦ (OpenWPM measurement data), ⑧ (malicious domains detected), ⑨ (domains subjected to session replay) are the output for further analysis from OpenWPM, VirusTotal scan results, and websites subjected to session replay, respectively.

5.3.1 Collecting hospital websites

To extract hospital websites, we use *webometric world hospital websites* as the primary source of information. We programmatically parse the content of each of the tabs appearing on the landing page of *webometric world hospital websites* [74] corresponding to different regions. For every hospital website URL, we extract corresponding meta data (e.g., hospital name, country, continent). Since *webometric world hospital websites* is not a complete list of hospital websites, we also complement other available hospital website lists — e.g., for China we use *Haodaifu*.

To collect Chinese hospital websites from *Haodaifu* [136], we crawl the list of hospital names from each of the 31 provinces in mainland China using [136]. Then we extract official names of these Chinese hospitals. These hospitals belong to different tiers (e.g., primary, secondary, tertiary). In order to determine the URL from the official names of these Chinese hospitals, we search each official name using the Baidu search engine. We observe that Baidu search results labels the official name of a hospital website (if exists) with two special Chinese characters — i.e., if a particular hospital does not have a website, Baidu search results will not label the official name of the hospital with the two special Chinese characters. Since the response from Baidu search results is not structured, it is not possible to mechanically parse the output. Therefore, we use the *Baidu Organic Results API* [252] to transform the search results to JSON format, and consider only the top 10 results to collect the hospital websites in mainland China.

We collect 19,635 unique hospital websites from different sources (i.e., *webometrics world hospital websites* [74], *Wikipedia* [308] and *haodaifu* [136]) for our privacy measurements. The hospital websites that we collect are hosted in countries pertaining to different regions — Asia (7183), Europe (5936), North America (4666), Latin America (1362), Oceania (227), Africa (261).

We also identify hospital websites with login forms on landing pages by matching the

corresponding source code with specific keywords (e.g., login, user id, password). This approach will work for any site irrespective of the language of page content, as the source code syntax of a web page is independent of the language of page content.

5.3.2 Web privacy measurements

We configure OpenWPM [223] web privacy measurement framework to run with 15 parallel browser instances in headless mode. We explicitly enable OpenWPM instrumentations for HTTP requests, JavaScript, cookies, DNS requests, callbacks and page navigations. Javascript instrumentation includes passive fingerprinting APIs used in the website. We clear the browser profile after each URL visit, to simulate the first visit to the browser instance, to avoid any influences from past browsing history. We use a physical machine (connected to our university network) running Ubuntu server 20.4 LTS, 32GB RAM, 1TB SSD, Intel Core i7-6700 CPU for our measurements between Sept. 1, 2021–Dec. 31, 2021. A total of 19,635 hospital websites from 152 countries were crawled using OpenWPM from a city in North America; 152 websites failed due to expired domain registrations and unreachable websites. The instrumented tracking metrics extracted from OpenWPM are saved to a SQLite database for further analysis. The saved information in the database contains both stateful (i.e., scripts/cookies) and stateless (fingerprinting) forms of tracking metrics. We then examine the saved tracking scripts/cookies for third-party domains (i.e., domains of scripts/cookies that do not match the domain of the hospital site that they are on).

Categorize third-party scripts and cookies. A third-party is a script/cookie included on a first party website (i.e., hospital website). We use filtering rules [92] that block third-parties on hospital sites to identify 3 categories of third-party domains: *EasyList* rules block ad-related third-parties; *EasyPrivacy* blocks known trackers; third-parties that are not blocked by EasyList/EasyPrivacy filtering rules are treated as unknown trackers.

Identify fingerprinting APIs. We use the instrumented JavaScript data to extract fingerprinting APIs included in hospital websites. Third-party domains hosting scripts that include these fingerprinting APIs are of different types — e.g., `window.navigator`, `window.screen`, `window.document`, `HTMLCanvasElement`, `CanvasRenderingContext2D`, `AudioContext`, `RTC`. These fingerprinting APIs are used to passively track users by leveraging various characteristics of a user’s environment, including hardware, operating system and software characteristics.

5.3.3 Session replay scripts

We extract hospital websites that include scripts (e.g., *fs.js*, *tag.js*, *hotjar-HotjarID.js*) with known session replay functionality [2] from the *javascript* table of OpenWPM SQLite database. These scripts pertain to *Hotjar*, *Yandex* and *FullStory* session replay services. We observe websites with *Hotjar* (but not *FullStory*, *Yandex*) session replay scripts send data over websockets. Therefore, we use *selenium-wire* [227] to automate the crawling of the landing page of 469 hospital websites with *Hotjar* session replay scripts, to identify the sites sending data over web sockets directly from the landing pages.

While existence of the session replay scripts (and the use of websockets by *Hotjar*) can be easily enumerated, it requires some manual effort (e.g., filling out forms) to understand what is leaked to the session replay servers. Therefore, we limit our manual tests to a selected set of hospital websites (183, of which 101 sites with *Yandex* services across multiple continents, 78 EU sites with *Hotjar*, and 4 sites with *FullStory* scripts). We observe 40/183 hospital websites require to create an account prior to booking an online appointment; 74 hospital websites have online forms (without account registration) to book an online appointment; remaining websites (69) do not have functionality to book an online appointment. We created accounts in 40 of hospital websites that require an online registration. Then we use crafted (fake) data (e.g., user name, password, email address) to book online

appointments with 114 (i.e., 40 sites with registration and 74 without registration) hospital websites. Thereafter, for those 114 hospital websites, we use *Chrome DevTools* [57] and *HTTP Toolkit* [144] to identify sensitive information transmitted to remote servers during session replay.

5.3.4 Detecting malicious domains

Potential security issues in hospital websites can lead to privacy issues. Therefore, to determine hospital websites and included third-party script/cookie domains that are malicious, we scan all 19,483 hospital websites, and 3673 third-party domains hosting scripts/cookies using VirusTotal. We report only those domains that are flagged by at least 3 security engines as malicious.

5.3.5 Limitations

Our hospital website collection technique may not find all hospital websites in any given jurisdiction. Additionally, we use filtering rules [92] to identify known advertisers and trackers, which are not comprehensive enough to find all possible tracking domains (especially country specific trackers). Some known advertisers/trackers may operate in a dual role of advertising and tracking. We also involved manual steps in verifying false positives/negatives of hospital websites including scripts pertaining to session replay services, which is non-trivial to automate.

5.4 Results

In this section, we report our findings on privacy issues of hospital websites.

5.4.1 Session replay

Session replay services are used to replay a visitor's session through the browser, to get a deeper understanding of a user's browsing experience; information replayed include user interactions on a website such as typed inputs, mouse movements, clicks, page visits, tapping and scrolling events. During this process, users' sensitive information can be exposed to third-party servers that host session replay scripts. We identified three session replay services in the analyzed hospital websites (19,483): *Hotjar* (469, 2.4%), *Yandex* (226), *FullStory* (4); see Table 14 for examples of hospital websites with session replay services. The regions that have a heavy presence of session replay services on their hospital websites include North America (291/4666, 6.2%) and Europe (299/5936, 5.0%); see Table 13. In total, we found session replay scripts on 699 hospital websites; 91/699 (13.0%) of sites were from EU countries.

Yandex. The session replay scripts hosted by *Yandex* were included in 153/226 (67.7%) hospital websites in Russia. These *Yandex* session replay scripts collect sensitive medical information of users and send them to remote servers (over HTTPS). In addition, sensitive information is exposed while performing common interactions with hospital websites, including booking online appointments, contacting hospital by entering sensitive information (e.g., medical description); see Table 11. There were 24 (out of 101 — see Sec. 5.3.3) Russian hospital websites that leak sensitive information with *Yandex* session replay services — user name, password, phone number, date of birth, address (street, city, country), passport information collected from *lk.baltclinic.ru*; requested medical service and login information collected from *medvedev.ru*, *zdordet.ru*, *vizus1.ru*, *gutaclinic.ru*, *president-clinic.ru*; user comments/messages collected from *alfa-med.ru*, *benefacta.ru*, *gkb12.ru*, *glazalazer.ru*, *presidentclinic.ru*, *onclinic.md*, *vizus1.ru*. We found 13 hospital websites in EU countries include *Yandex* session replay scripts, and 3 of these EU hospital websites (in Greece, Portugal and Czech Republic) apparently violate GDPR [100] privacy

regulation. These 3 EU hospitals leak information to *Yandex* remote servers as follows: *multiscan.cz* (in Czech Republic) leaked search information from the search functionality; *lifeclinic.gr* (in Greece) leaked user name, phone number, email, subject and message sent; and *chpvvc.pt* (in Portugal) leaked name, email, service rendered and message sent.

FullStory. We observed *FullStory* session replay scripts included in *www.mater.org.au*, *www.ramsayhealth.co.uk*, sent visited page, and screen width and height of the user's display to a remote server.

Hotjar. Session replay code from *Hotjar* is included within the *head* tags in the hospital website page source as a JavaScript snippet [141]. The session replay data captured from *Hotjar* scripts is sent to a remote server using websocket connections. From our automation with *selenium-wire*, we found 27/469 (5.8%) hospital websites that include *Hotjar* session replay scripts, sent data over websockets to remote servers — e.g., 3 EU hospital websites and 18 US hospital websites enable such data transmission (apparently, violating GDPR and HIPPA privacy regulations, respectively); the remaining 6 hospital websites are in non-EU countries. In addition, by manually inspecting 78 (see Sec. 5.3.3) EU hospital websites with *HTTP Toolkit/Chrome DevTools*, we found 4 of the inner URLs from those sites, leaked sensitive information through websockets — e.g., user name, email, phone number, medical service are sent from *www.bilicvision.hr* (in Croatia); user name, email, phone number, message and country are sent from *www.reprofit.cz* (in Croatia); see Table 12 for information leaked by hospital websites with session replay services in the EU countries.

5.4.2 Domains flagged as malicious

With VirusTotal, we found 33/19,483 websites were flagged as *phishing*, *malicious* or *malware* by at least 3 VirusTotal engines;⁶ 26 of the flagged sites were part of more than one VirusTotal category; 27 sites were flagged as *phishing*. We did not consider scan

⁶<https://support.virustotal.com/hc/en-us/articles/115002146809-Contributors>

Site	Sensitive Information						Medical service information				
	Name	Phone	Email	Password	DOB	Address	Passport	Specialist	Message	Clinic	Service Date
alfa-med.ru	✓		✓		✓	✓			✓		
bakulev.ru	✓	✓	✓	✓	✓	✓					
lk.baltclinic.ru	✓	✓	✓	✓		✓					
solovevka.ru	✓		✓		✓				✓		
smclinic.ru	✓	✓			✓			✓		✓	✓
rami-spb.ru	✓	✓	✓					✓			

Table 11: Examples of private/sensitive information collected by *Yandex* session replay service — DOB = Date of birth

SRS	Country	Site	Sensitive Information				Medical service information				
			Name	Password	Email	Phone	Country	Specialist	Message	Chat/Search	
<i>Hotjar</i>	Croatia	bilitelevision.hr	✓		✓	✓			✓		
	Czech Republic	reprofit.cz	✓		✓	✓			✓		
	Italy	e-medical.it	✓		✓						
	Portugal	sanfl.pt		✓							✓
<i>Yandex</i>	Greece	lifeclinic.gr	✓		✓	✓			✓		
	Polgual	chpyvc.pt	✓		✓				✓		
	Czech Republic	multiscan.cz								✓	

Table 12: Examples of private/sensitive information collected by session replay service in EU Countries — SRS = Session replay service

Region	FullStory	Hotjar	Yandex
Europe	1	108	190
NorthAmerica	2	282	7
LatinAmerica	-	37	1
Asia	-	20	28
Africa	-	3	-
Oceania	1	19	-

Table 13: Session replay services on hospital websites.

SRS	Hospital domain names	Leaked data
Hotjar	bilicvision.hr, multiscan.cz	name, email, password, phone, chat
Yandex	alfa-med.ru, bakulev.ru	name, email, password, phone, speciality
FullStory	ramsayhealth.co.uk	URL, screen width, screen height

Table 14: Examples of hospital websites with session replay services — SRS = Session replay service.

results from some VirusTotal engines (e.g., *CRDF*, *Quttera*) as the results from those engines were unreliable. Most hospital websites flagged by VirusTotal were in China (10/33, 30.3%) and India (3/33, 9.1%). We also looked into malicious JavaScript files that were included in the 33 flagged hospital websites; *ultramed.pl* (in Poland) and *bcm.es* (in Spain) included 10 and 2 unique malicious JavaScript files, respectively. The common malicious JavaScript files contained *jQuery* keyword in its file name (e.g., *jquery.min.js*, *jquery.themepunch.tools.min.js*), or were part of *WordPress* web applications (e.g., *wp-embed.min.js*, *wp-emoji-release.min.js*). *jQuery* is a commonly used JavaScript library, and it is the base for many add-on scripts/plugins that are also included in platforms such as *WordPress* [71, 177]. Fake *jQuery* scripts with malicious source code [269] can be dangerous for users.

The following 6 hospital websites (in 4 countries) were flagged as malicious by more than 5 security engines: a Tunisian hospital website (*cliniqueelmenzah.com*) was flagged as malicious by 9 security engines; sites from China (*jrszyy.com*, *zyxyfy.com*, *ahzxy.com*), India (*mathahospital.org*) and Brazil (*hsja.com.br*) were flagged as malicious by 6 security engines. The malicious categories of these flagged websites include *known infection*

source, media sharing, compromised websites, malicious, malware and spyware.

We also scanned all 3673 third-party domains (of scripts/cookies) using VirusTotal, and found 27 of them (e.g., *iclickcdn.com*) were flagged by at least 3 VirusTotal engines. For the domains hosting third-party scripts/cookies, 11 and 18 were flagged as malicious and malware, respectively. In Table 15 and Table 16, we list examples of potentially malicious domains hosting tracking scripts and cookies (including the presence of such domains on hospital sites), respectively.

Category	Tracking domains	# hospital sites
Malware, malicious	iclickcdn.com, newrrb.bid, do-hero.com, wek7ipqx359.ru, 51.la, sc-static.net	120 (China, USA)
Malicious, phishing	ignorelist.com, popupsmart.com, leostop.com, popcash.net, secureserver-cdn.net	23 (USA, Chile, Malaysia)
Malware	che0.com, xc7789.top	4 (China, Spain)
Malicious	d10lpsik1i8c69.cloudfront.net, fontawesome.com, bitrix.info	138 (USA, Russia, France, Japan)

Table 15: Known tracking scripts hosted on potentially malicious domains that are flagged by VirusTotal. The countries within parenthesis in the 3rd column of the table are example location(s) of the hospital website(s).

Category	Tracking domains	# hospital sites
Malware, malicious, phishing	cnzz.space, crzenith.com,	2 (China, Saudi Arabia)
Malware, malicious	bedrapiona.com, medreviews.ru, inform-nikolase.live, 04zl.cn, greenclick.biz	85 (China, Mexico, Spain)
Malicious, phishing	onmarshtempor.com, click-matters.biz	2 (Bulgaria, Spain)
Phishing	junmediadirect.com, 123formbuilder.com, app-us1.com	33 (USA, Australia, Belgium)
Malware	fontawesome.com, clarity.ms	124 (USA, Canada, Japan, United Kingdom, Portugal)
Malicious	sc-static.net	32 (USA, Saudi Arabia)

Table 16: Known tracking cookies set by potentially malicious domains that are flagged by VirusTotal. The countries within parenthesis in the 3rd column of the table are example location(s) of hospital website(s).

5.4.3 Websites using HTTP and login forms

Hospital websites served over HTTP may allow an adversary to intercept sensitive information sent over the network traffic. We found 4062/19483 (20.8%) of hospital websites use HTTP. Some sites perform sensitive operations on these HTTP pages. For example, <http://www.bfh.com.cn/Account/Register> allows user registration functionality using HTTP. During user registration, the user is required to enter account information (user name, password) and other sensitive information (official name, national ID, mobile phone number, email, telephone number, province, city, marriage, home address, job, work address, MSN, QQ). Similarly, user registration information (user name, official name, password, national ID, mobile phone number and medical card ID) entered through <http://www.zbdyyy.com/usersys/regist.aspx>, is sent over HTTP, and can be intercepted by an adversary. We also found that the use of login forms in the landing page of hospital websites is mostly available in China (596/4324, 13.8%) and Australia (38/160, 23.7%), and some of these forms are submitted via HTTP. For example, 346/596 (58.1%) Chinese hospital websites with login forms sent login credentials in the clear — e.g., after clicking the top right button of hospital site <http://www.ahs2y.com/>, a login form is opened (<http://111.39.250.98:7001/defaultroot/login.jsp>); once the account name and password is entered and submitted, the credentials are sent over plain HTTP.

5.4.4 Third-party tracking scripts

We found 9443/19,483 (48.5%) of hospital websites included at least one known tracking script. Hospital websites in Oceania (140/227, 61.7%) and North America (2805/4666, 60.1%) had a high percentage of websites with known tracking scripts. Hospital sites in Asia (2844/7183, 39.6%) had a relatively lower proportion of sites with known tracking scripts; see Fig. 20. Top known trackers included on hospital websites (19,483) are:

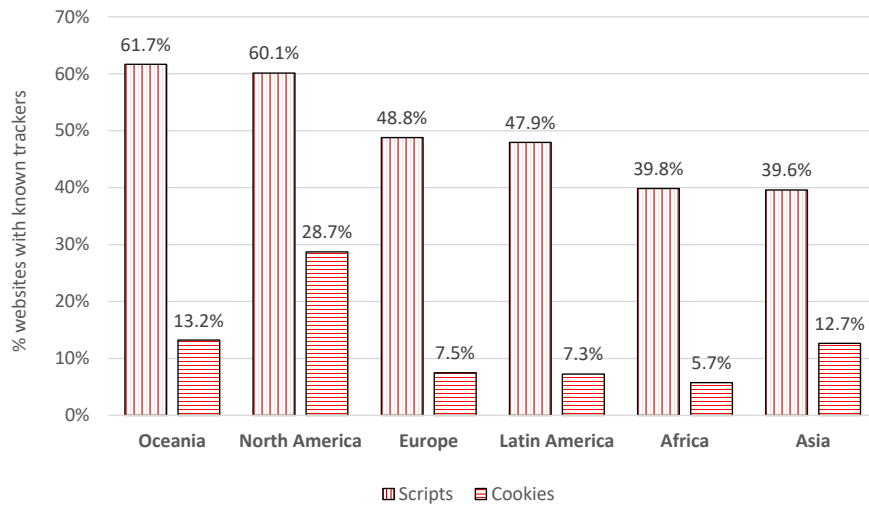


Figure 20: Percentage of hospital websites with known tracking scripts/cookies.

googleanalytics (6607, 33.9%), *googletagmanager* (4816, 24.7%), *facebook* (2552, 13.1%) and *cloudflare* (564, 2.9%); see Fig. 21. Both *googletagmanager* and *googleanalytics* are used to collect tracking/marketing data on hospital websites; *gtag.js* sent event data to Google Analytics, Google Ads and Google marketing platforms. Google Maps was included in 1591 hospital websites; YouTube videos were embedded in 1372 hospital websites; *Addthis* (*s7.addthis.com*) contained adware that redirected users to promotional websites (246/19,483, 1.3%).

There were no significant differences relating to the proportion of hospital websites with various categories of third-parties (i.e., ads, known trackers, unknown trackers) between different geographical regions; see Fig. 22. However, some countries (with more than 9 hospital websites in our dataset) in different regions had known tracking scripts in most of its hospital websites — Finland (18/23, 78.3%); Belarus (10/13, 76.9%); Norway (28/38, 73.7%); Latvia (18/25, 72.0%); Kuwait (7/9, 77.8%); Japan (702/1012, 69.4%). We also found known tracking scripts that are region specific; *bdstatic.com*, *qq.com* and *50bang.org* only tracked websites in Asia; *adsvr.org*, *rtrk.com*, *btttag.com* and *cloudfront.net* were only found on North American hospital websites; Oceania had only one regional script domain (*turbolion.io*); Africa had no regional tracking script.

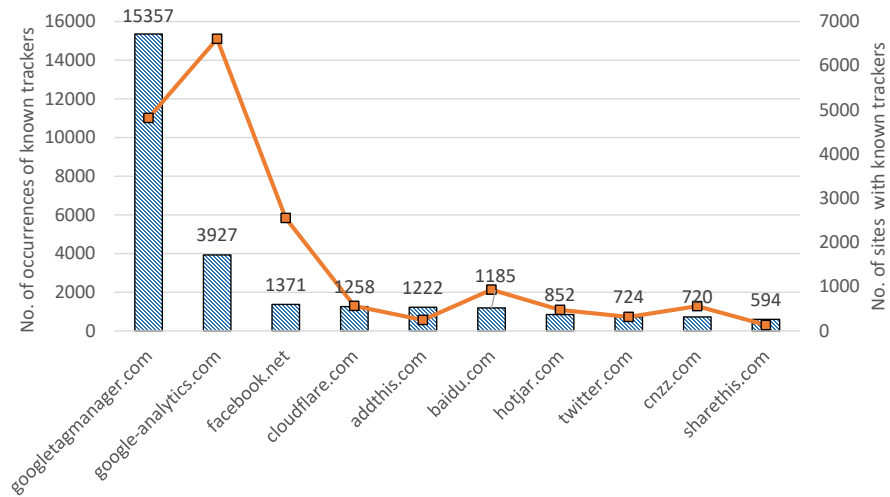


Figure 21: Top-10 known tracking scripts on hospital sites - the bars show the number of occurrences of known tracking scripts (vertical axis to the left), while the line chart shows the number of hospital websites with known tracking scripts.

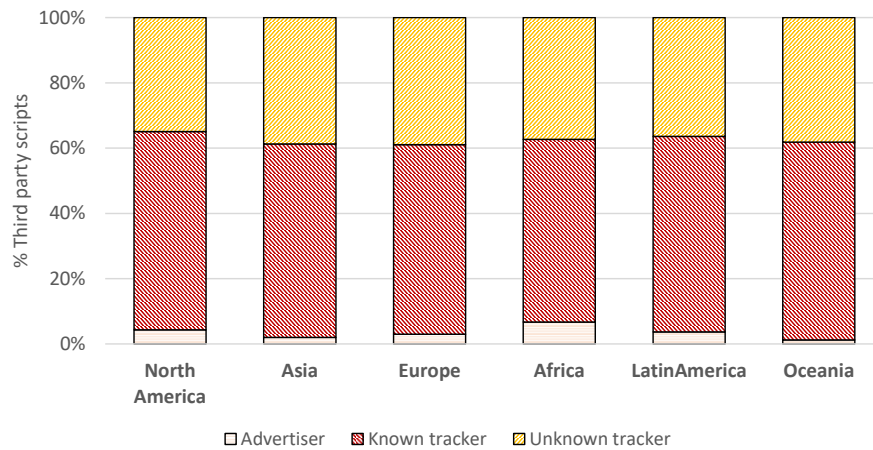


Figure 22: Proportions of third-party scripts in different categories (tracking, advertising and unknown) included on hospital websites by region.

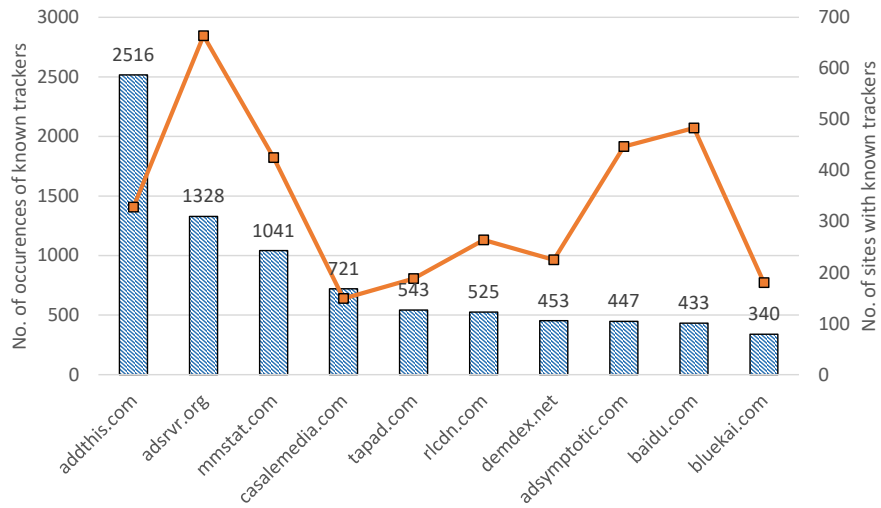


Figure 23: Top-10 known tracking cookies on hospital sites - the bars show the number of occurrences of known tracking cookies (vertical axis to the left), while the line chart shows the number of websites with such cookies.

5.4.5 Third-party tracking cookies

We found 2839/19,483 (14.6%) hospital websites from 85 countries set known tracking cookies; see Fig. 23. The top-3 regions with the highest proportion of known tracking cookies set on hospital sites were Asia (3086/8689, 35.5%), North America (8186/28,960, 28.7%) and Oceania (141/594, 23.7%); see Fig. 24. Taobao (Alibaba) that collects user behaviours for targeted advertising [16], *mmstat.com* sets third-party cookies on a large proportion of hospital websites in China (425/4324, 9.8%). Similarly, a large proportion of known tracking cookies (483/4324, 11.2%) were set by *baidu.com* on Chinese hospital sites.

We also examined the cookie validity duration by regions, and found that 1017/3264 (31.2%) known tracking cookies set on hospital websites in Asia, were valid for more than 1000 years. Known tracking cookies that expire after 5 years include *mmstat.com* (1039) and *baidu.com* (431); see Table 17.

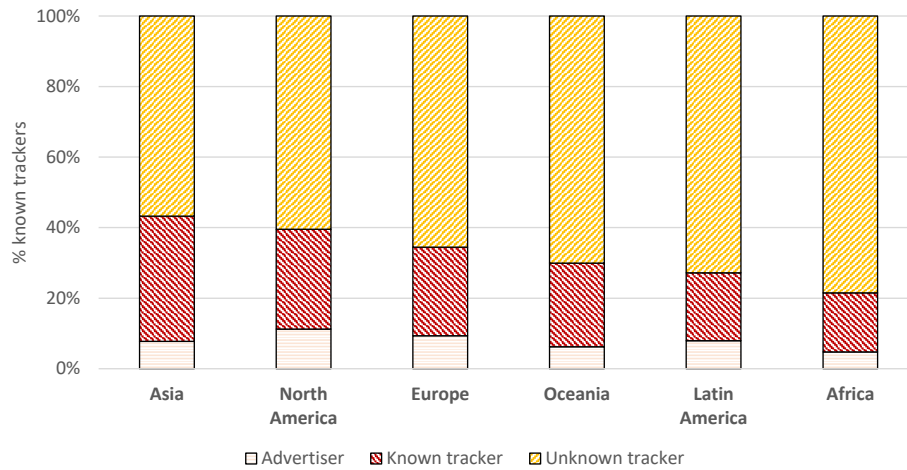


Figure 24: Proportions of third-party cookies in different categories (tracking, advertising and unknown) set on hospital websites by region.

Tracker	#Sites	Cookie Expiry Duration			
		1m-1y	1y-5y	5y-100y	> 1000y
addthis.com	2516	110	2345	-	-
adsvr.org	1328	1328	-	-	-
mmstat.com	1041	2	-	237	802
casalemedia.com	721	572	-	-	-
tapad.com	543	543	-	-	-
rlcdn.com	525	525	-	-	-
demdex.net	453	453	-	-	-
adsymptotic.com	447	447	-	-	-
baidu.com	433	-	-	418	13
bluekai.com	340	340	-	-	-

Table 17: The top-10 known tracking cookies and their expiry periods (m=month, y=year).

5.4.6 Fingerprinting APIs

We found a large number of fingerprinting APIs (total: 3,082,179, unique: 222) included in the JavaScript source files used in hospital websites. Most common fingerprinting APIs include: *window.navigator* (1,146,303), *Storage* (407,847), *CanvasRenderingContext2D* (340,164), *HTMLCanvasElement* (133,657), hardware related APIs (32,394), *window.screen* (18,888), *RTCPeerConnection* (747), *window.navigator.geolocation* (722) and *AudioContext* (291). We also found several fingerprinting APIs with acoustically relevant characteristics of the audio signal — *GainNode* (49), *AnalyserNode*(110), *OscillatorNode*(374) and *ScriptProcessorNode* (38). Combinations of multiple fingerprinting APIs can be used to identify a user with a high precision [97].

5.5 Recommendations

Based on our analysis, we suggest a few possible mitigation strategies to reduce privacy exposures to third-parties from the perspective of site developers and regulators. Developers should analyze scripts used for tracking/fingerprinting, and use only those scripts that are required for the proper functioning of the sites. Similarly, the use of session replay scripts should be avoided, or at least configured properly to reduce the risk of data exposures. Since software packages and applications are becoming a target for malware and supply chain attacks (cf. SolarWinds [215]), developers should always scan the dependent software packages/libraries to ensure that hospital websites do not inherit such vulnerabilities.

From our manual analysis, we observed that while the privacy policies of some hospitals explicitly mention that they do not share any information with third-parties, several sites still send personal information to session replay services such as *Yandex* and *Hotjar*.

For example, `sanfil.pt` (in Portugal) explicitly states in its privacy policy⁷ that the information collected from users will not be shared with third-parties, while in reality, when a user uses the online chat function available on the website, all the chat messages are sent to *Hotjar*. In addition, despite the privacy policy⁸ of *lifeclinic* (in Greece), and user agreement⁹ of `rami-spb.ru` claim that a user's personal data will not be disclosed to third-parties, personal information (e.g., username, phone, email and doctor's speciality) is leaked to *Yandex*. Therefore, regulators should invest into developing tools to detect such contradictory statements and violations to improve data privacy in the long run. We also observed that 33 of hospital websites are flagged as malicious by VirusTotal, possibly due to the use of malicious third-party resources (e.g., the use of fake and malicious *jQuery* libraries) in those sites. Therefore, developers need to be vigilant in including third-party libraries in hospital websites, and should do proper scanning before using such dependencies.

Hospital websites continue to expand its services in digital space; the COVID-19 pandemic also contributed to the recent rapid increase of online hospital services. Given such growth, and the use of sensitive information at hospital services, proper safeguards should be implemented to prevent potential privacy/security exposures. Furthermore, governments should introduce and periodically review existing privacy regulations (e.g., the US HIPPA [291]) to protect sensitive information pertaining to patient identity and health records.

⁷<https://www.sanfil.pt/cookies/>

⁸<https://www.lifeclinic.gr/privacy-policy/>

⁹<https://www.rami-spb.ru/Content/poljzovateljskoe-soglashenie-ob-ispoljzovanii-sajta/4091>

5.6 Summary

Similar to other popular commercial sites, hospital sites include commercial trackers hosted by top tech giants. We found that 10,417 (53.5%) hospital websites included such tracking scripts/cookies; 4.2% (815/19,635) of hospital websites set tracking cookies that are valid for more than 1000 years; 222 unique fingerprinting APIs were in included scripts found in hospital websites. Furthermore, sensitive user information is relayed to remote servers by including session replay scripts in hospital websites — Hotjar (469), Yandex (226), FullStory (4).

Chapter 6

Privacy analysis of religious websites and mobile apps

6.1 Introduction

With the advancement of technology, significant changes are made as to how religious practices are conducted during the last couple of decades [44]. The early online churches simply used websites with static pages (e.g., scriptorium pages of religious texts) to share information with an increased audience. Gradually, these websites started to include dynamic content hosting various interactive services (e.g., chat and messaging services, podcasts, videos of sermons, interactive worship). Also, with the proliferation of mobile devices, religious services were offered through mobile apps [46]. The recent COVID-19 pandemic has also resulted in offering religious services through online social media platforms (e.g., Facebook Live, YouTube) [210], and religious faiths in the United States have strengthened due to the pandemic [219]; 57% of the adults in the United States who attended religious services at least monthly, are now watching religious services online due to the pandemic [219]; churches supplement their revenue using virtual offering (e.g., donation) services. Unfortunately, various third parties included on religious online services

to support various functionalities, are used to track users [108], and engage in privacy violations [64] leaking sensitive information; a prayer app (*Muslim Pro*) that eases the practicing of daily rituals prescribed in Islam, has leaked user location data to a broker (*X Mode*), which in turn had sold the same information to its contractors (including US military contractors) [176]; another prayer app (*pray.com*) sold the prayers of a grieving user who suffered a tragedy [42]. Also, while the possible influences from artificial intelligence (AI) technology on religious online services is still an under-studied area, potential exposures of highly confidential conversations relating to spiritual needs of users through chatbots (included on religious online services) will impact the privacy of users. In addition, security issues in religious online services can expose sensitive information of users; the Vatican site was hacked and compromised (in 2020) [280] with the aim of stealing sensitive information.

Past studies primarily discussed the evolution of digital religious communities from traditional religious institutions. Campbell [45] studied Internet trends and their implications on religious practices (including social and cultural shifts) and challenges related to online religious networks. The author observed that studying the religious practices of Internet users leads to a more refined understanding of the complex interactions with online services. Campbell et al. [46] provided a methodological approach to study religious-oriented mobile apps available on iTunes app store. The authors reviewed 451 religious app functions and their use, and group those apps into 11 categories.

In this work, we perform a large scale web privacy measurement of religious websites and Android apps. To the best of our knowledge, this is the first measurement study on privacy/security of religious online services, performed on a global scale. For the web privacy measurements, we use 62,373 websites collected from the *URL Classification* [163] source, after filtering out false positives (i.e, non-religious sites) using VirusTotal [298]

website categorizations. Thereafter, we crawl the extracted religious websites using OpenWPM [223] web privacy measurement framework. We analyze the instrumented tracking metrics (third party scripts/cookies, fingerprinting APIs) using the instrumented data saved to the OpenWPM database. We identify religious websites that use session replay services, by inspecting the traffic sent by potential sites including session replay services with *HTTP Toolkit* [144]. In addition, we examine religious sites that send personal information to external parties using the chatbot functionality. We look for leaked personal/sensitive information (e.g., name, email address, address, prayer requests, confessions, user’s location provided for searches) from religious websites that use HTTP or configured to use session replay. To find potential TLS vulnerabilities and weaknesses, we collect and analyze TLS certificates of 45,004 religious websites. In order to find other vulnerabilities in religious websites (e.g., Cross Site Scripting, SQL Injection, Path Traversal), we scan 11,888 religious websites using the *Wapiti* scanner. We also collect religious Android apps, and leverage MobSF [196], LiteRadar [182], and mitmproxy (with Google UI/Application Exerciser Monkey), to perform static and dynamic analysis techniques (using a Pixel 6 phone). However, we limit the security evaluation of religious online services due to possible legal and ethical issues. We also use VirusTotal [298] to identify religious sites, Android APKs and included third party domains hosting scripts/cookies that are malicious.

Contributions and notable findings.

1. We develop a framework to collect religious websites and Android apps by eliminating false positives from given external source(s), and a test methodology to evaluate the privacy and security exposures from these religious websites.

2. 198/62,373 (0.3%) religious websites include session replay services — e.g., *FullStory* (*fullstory.com*), *Inspectlet* (*inspectlet.com*), *Luckyorange* (*luckyorange.com*), *Yandex* (*yandex.com*). We observed that users’ personal/sensitive information is sent from the analyzed religious websites to session replay services (*FullStory*, *Yandex*, *Inspectlet*). Such

shared sensitive information includes name, phone number, address, email address, message/comment, prayer request, location searches, login information, donation information, and keywords used in site searches.

3. 19/11,888 (0.16%) religious websites were found to be vulnerable — SQL Injection (9), Reflected Cross Site Scripting (7), Server Side Request Forgery (2), Path Traversal (1). The Path Traversal attack (on *christcc.org*) exposes several local files under */etc* directory (e.g., */etc/password*).

4. 7/1454 (0.48%) religious Android apps leaked sensitive information (e.g., user credentials, API key, phone number) from unprotected Firebase endpoints. In addition, 2 apps (*cdff.mobileapp*, *com.avrpt.teachingsofswamidayananda*) sent user credentials/device information over HTTP.

5. 17,418/62,373 (27.9%) and 3569/62,373 (5.7%) of religious sites include commercial tracking scripts and cookies, respectively. These trackers embed analytic and other third party services (e.g., social media plugins) on religious websites. Google dominates in tracking on both religious sites (32%) and apps (78%). There were tracking cookies that expire after a long period of time (including 4 tracking cookies by center.io on 4 religious sites that expire in year 9999). In addition, 1351/1454 (93%) of religious Android apps included tracking SDKs.

6. 69/62,373 (0.11%) religious websites were flagged as malicious at least by 5 security engines used by VirusTotal (e.g., *samenleesbijbel.nl*, *csiholytrinitychurch.com*). We also observed 12 malicious domains set tracking scripts/cookies on religious sites. Additionally, 29/1454 (2%) religious Android apps were flagged by VirusTotal by at least one security engine; *islamictech.sifgo* religious Android app was flagged by 10 security engines in VirusTotal.

7. 14/24 ((58.3%) religious websites that use HTTP, sent personal/sensitive information

(name, email address, phone number, address, message, prayer request, confession, date of birth, password).

We disclosed our findings on security vulnerabilities of the 10 websites and 9 Android apps to the corresponding admins/developers. We also notified Google about *islamictech.sifgo*.

6.2 Related work

Web privacy measurements. There are various privacy measurement studies that are performed in the past. Englehardt et al. [97] implemented OpenWPM, a fully automated web privacy measurement framework. Using OpenWPM, Englehardt et al. [97] performed a web privacy measurement of the top-1M Alexa popular sites (mostly commercial sites), and found Google and Facebook dominates in tracking. Samarasinghe et al. [239] measured tracking on 150,244 government websites and 1166 Android apps, and found commercial trackers on those online services (mostly Google trackers), although it was unexpected to have trackers on government sites that are funded by the taxpayers. Hoy et al. [143] studied 102 church websites in the United States and found that they collect personal identifying information. The confidential information that are entered to church guest books and prayer requests, were leaked from corresponding church websites. We studied tracking on religious websites and found a larger proportion of those sites with Google trackers (32%, 19,772 out of 62,373 websites). In addition, we found 22 websites leak sensitive information of users (e.g., name, address, email, donation amount, prayer requests) to session recording services.

Privacy analysis of mobile apps. Several past studies analyzed privacy and security issues in mobile apps. For example, Binns et al. [34] studied 959,000 apps from US and UK Google Play stores, and found that third party tracking follows a long tail distribution dominated by Google (87.75%). Nguyen et al. [199] performed a large-scale measurement

on Android apps to understand violation of General Data Protection Regulation (GDPR) explicit consent. They found 28.8% (24,838/86,163) of apps sent data to ad-related domains without explicit user consent. Several recent studies (e.g., [56]) analyzed COVID-19 tracing apps, and highlighted privacy and surveillance risks in these apps. In contrast, we study privacy and security issues of 1454 religious Android apps and found Google specific tracking SDKs in a large proportion (78%, 1132 out of 1454) of them.

Analysis of SSL/TLS certificates used in online services. Felt et al., [105] measured the HTTPS adoption on the web, and found the number of top websites (from HTTPWatch Global, Alexa top-1M, Google top-100) that use HTTPS (by default) doubled between early 2016 and 2017. Alabduljabbar et al. [9] investigated the potential vulnerabilities (SSL/TLS) in free content websites (FCW) and premium websites. The authors found 17% and 12% of free websites have invalid and expired certificates, respectively. The authors also found more FCWs (38%) use ECDSA signature algorithm compared to premium websites (20%). We analyze TLS certificates of 45,004 religious websites and found 92.9% and 7.1% of HTTPS sites use RSA and ECDSA signature algorithms, respectively.

6.3 Methodology

In this section, we provide details of our website and apps collection methodology. Then, we elaborate our privacy analysis and measurement techniques; see Figure 25 for an overview of our methodology.

6.3.1 Collecting religious websites and Android apps

Religious websites. We acquired a list of 583,784 websites (on April 26, 2022) from *URL Classification* [163] that are categorized as *Religion*; 448,646 (out of 583,784, 76.9%) are

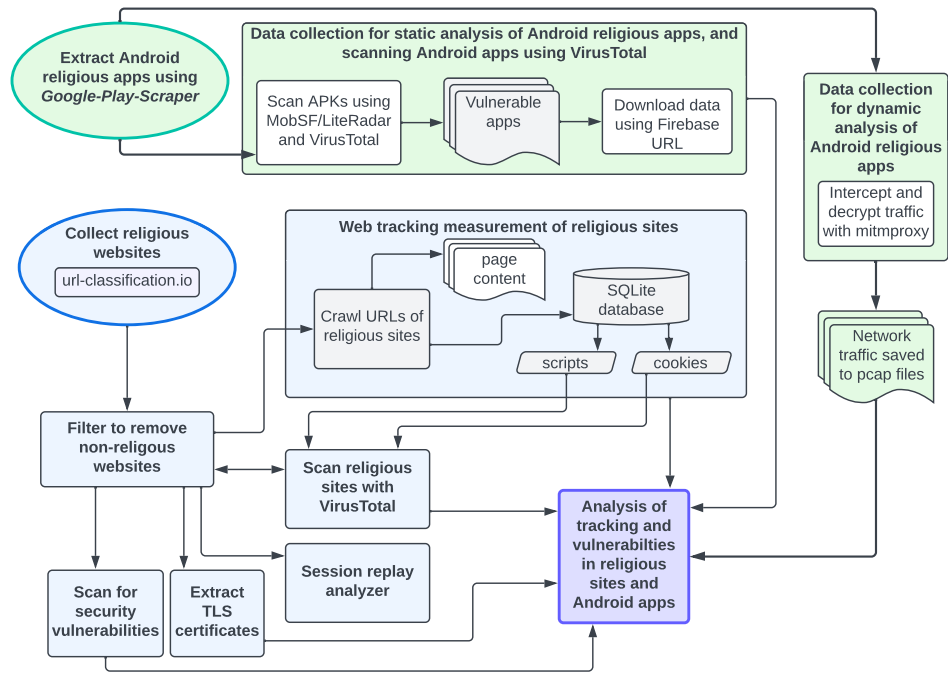


Figure 25: Overview of our methodology.

classified into multiple categories (including *Religion*). *URL Classification* provides a confidence rank for classified categories of each website, and with manual inspection, we find websites ranked 50 and above are likely religious sites; 202,968 (out of 583,784, 34.8%) websites are ranked 50 and above. To ensure, false positives are eliminated, we scan the the 202,968 websites with VirusTotal [298], and filter 62,373 (out of 583,784, 10.7%) websites that are flagged as *Religion* by at least one security engine included in VirusTotal.

Religious Android apps. We feed unique keywords related to major religions (i.e., Christianity, Islam, Hinduism, Buddhism) to *Google-Play-Scraper* [132], that crawls and extracts 2512 Android apps matching those search keywords from Google Play Store. We eliminate false positives by manual inspection, and finally select 1454 apps for our analysis.

6.3.2 Web privacy measurements

We configure OpenWPM [223] web privacy measurement framework to run with 10 parallel browser instances in headless mode. We configure OpenWPM instrumentations for HTTP requests/responses, JavaScript, cookies, DNS requests and callbacks. JavaScript instrumentation also collects passive fingerprinting APIs included in religious websites. To mimic a new request, and to avoid any influence from past browsing history, for each URL visit, we clear the browser profile after each visit to a website. We use a physical machine (connected to our university network) running Ubuntu server 20.4 LTS, 64GB RAM, 1TB SSD, AMD Ryzen Threadripper 2950X 16-Core Processor for our measurements between May 1, 2022 - May 7, 2022. A total of 62,373 religious sites were successfully crawled. We also configure OpenWPM to save the site content to a *LevelDB* [174] database. The instrumented tracking metrics extracted from OpenWPM are saved to an SQLite database for further analysis. The saved information in the database contains both stateful (i.e., scripts/cookies) and stateless (fingerprinting) forms of tracking metrics. We then extract scripts and cookies hosted on third-party domains (i.e., domains of scripts/cookies that do not match the domain of the religious site that they are included). We use *EasyPrivacy* [92] filtering rules that block third party trackers in religious sites to identify known third party tracking scripts/cookies.

6.3.3 Session replay scripts and chatbot services in religious websites

We identify a list of known session replay scripts offering session replay services [115] — *FullStory* (fs.js), *Inspectlet* (inspectlet.js), *Lucky Orange* (core/lo.js), *Yandex* (watch.js, tag.js). Then we extract the religious websites (198 out of 62,373, 0.32%) that include those scripts, from the *javascript* table of OpenWPM SQLite database. Thereafter, we inspect these 198 sites manually, to identify possible personal/sensitive information leaked during user interactions with the religious websites (e.g., while submitting messages and prayer

requests, donating to religious institutions). During the interactions with these websites, we use crafted data (e.g., name, email, date of birth, messages, amount for donations), but do not submit the form, as input information is sent to remote servers, after each keystroke during user input. Personal information is also sent during interactions with chatbots in religious websites. We manually inspect the network traffic using *HTTP Toolkit* [144] to identify information sent over the network.

6.3.4 Security issues in religious websites

Potential security issues in religious websites can cause privacy issues. In this section, we discuss security issues in the analyzed religious websites.

Malicious religious websites. In order to determine if the religious websites and included third party domains (hosting scripts/cookies) are malicious, we scan all 62,373 religious websites, and included 1906 third party tracking domains using VirusTotal. Note that, at least in some cases, VirusTotal engines¹ may misclassify or delay in updating domain categorization labels [217]. We report domains that are flagged by at least 5 security engines as malicious.

HTTP/HTTPS traffic and TLS certificates used in religious websites. We use *Py-OpenSSL* [226] to collect the TLS certificates (in X509 format) of the analyzed religious websites. Then we extract various information of the collected certificates — i.e., validity duration, common name, issuer information (e.g., issuer name, issuer country, issuer organization), signature algorithm, public key size (for RSA only). We identify the protocol used in each web request (i.e., HTTP, HTTPS). We also analyze the collected information, to determine whether any of the religious websites send personal/sensitive information over plain HTTP, or the associated certificates used in religious websites expose users to risks.

¹<https://tinyurl.com/2p8ynsfj> (we exclude CRDF and Quttera for their unreliable results as we observed).

Other security issues in religious websites. We randomly selected 11,888 religious websites (out of 62,373), and scanned them using the *Wapiti* [303] scanner to find other security issues (e.g., Cross Site Scripting, Server Side Request Forgery, SQL Injection). *Wapiti* crawls the web pages of a given website, and looks for scripts and forms in web pages where it can inject payloads to identify vulnerabilities. We configured *Wapiti* to use 15 seconds as *max-attack-time* and *max-scan-time*, and scan up to a depth of 5 levels from the base URL.

6.3.5 Android app analysis

Tracking SDK detection. We perform static analysis, using LiteRadar [182] by feeding APK files of each of the religious Android apps. The output from this process includes the tracking SDKs included in religious Android apps, the use of tracking SDKs, and requested permissions (including dangerous permissions such as camera, contacts, microphone, SMS, storage, and location).

Misconfigured Firebase database. Many Android apps, including religious apps, use Google Firebase [126] (a widely used data store for mobile apps) to manage their backend infrastructure. However, due to possible misconfiguration, Android apps connected to Firebase database can be vulnerable. Exposed data from Firebase vulnerabilities includes personally identifiable information (PII) and plain text passwords. We leverage MobSF [196] to extract URLs of unprotected Firebase endpoints for each APK file, which contains potential vulnerabilities; we then download the exposed data from the Firebase datastore URL² and check for apparent sensitive and PII items, including: user identifiers, passwords, email addresses, and phone numbers. However, for ethical/legal considerations, we do not validate the leaked information (e.g., login to an app using the leaked user credentials). Then we remove the downloaded datastore.

²The URL is of the form *<Firebase project name>.firebaseio.com/.json* (e.g., <https://catholic-connect-213606.firebaseio.com/.json>).

Dynamic analysis. We use a rooted Pixel 6 mobile phone with Android 12, to proxy traffic from newly installed apps via mitmproxy [194]. To avoid collecting traffic from other apps, we uninstall all other apps, except those apps required for basic functionalities (e.g., Camera, Google Play Store). A mitmproxy root certificate is installed on the phone. We also install mitmproxy on a separate desktop machine to collect and decrypt HTTPS traffic. Both the desktop machine and phone are connected to the same Wi-Fi network. We use adb [123] to automate the installation, launch, and uninstallation of the apps. We also use Monkey [124] with 5000 events (e.g., touch, slide, swipe, click) for each app; login to app UI is not supported (if prompted). The network traffic is captured and stored in pcap files. We use the captured network traffic to determine sensitive information (e.g., device identifiers sent to trackers, leaked hardcoded user/admin credentials and API keys) sent to external entities. We close mitmproxy and uninstall the installed religious app before moving to the next app.

Session replay from Android apps. We leverage the dynamic analysis to inspect third party domains included in apps, to identify those known session replay services (e.g., Yandex, Hotjar, MouseFlow, UXCam) to which apps send HTTP requests. For this exercise, we use Burp Suite [222] to identify apps that send sensitive information to corresponding session replay services.

Malicious domains and apps. We scan the APK files of 1454 religious Android apps with VirusTotal. We also scan 1539 domains included in apps (as found in the network traffic) with VirusTotal.

6.3.6 Ethical considerations and limitations

We do not use the sensitive information (e.g., user identifiers and passwords) extracted from static and dynamic analyses of Android apps for any intrusive validations that may have an impact to the privacy of users. In addition, we did not retain any data from exposed

Firestore databases. The *Wapiti* black-box scanner we use to find vulnerabilities in religious websites, limits the scope of the scan only to the web page (e.g., add/remove query parameters).

EasyPrivacy [92] filtering rules that we use are not comprehensive enough to identify all possible tracking scripts/cookies set on religious sites (especially country specific trackers). We also resorted to use manual steps in verifying false positives/negatives of religious websites and Android apps, which are not trivial to automate (e.g., inspection of sensitive information relayed from session replay services to third parties). Android apps with obfuscated code may have impacted our static analysis, but not so on our dynamic analysis. Random clicks triggered from the UI automation that use monkeyrunner, may not precisely target the specific targeted areas on the UI.

6.4 Results: Religious websites

6.4.1 Session replay and chatbot services

With session replay services that are included in websites, a user's session is replayed through the browser and sent to a remote third party; information replayed includes user interactions on a website, such as typed inputs, mouse movements, clicks, page visits, tapping and scrolling events. During this process, user's sensitive information can be exposed to third-party servers that host session replay scripts. We identified four session replay services on the analyzed religious sites (62,373): *FullStory* (4), *Inspectlet* (5), *Lucky Orange* (1), *Yandex* (187). The Lucky Orange session replay service was included only on one analyzed religious site (*discoverquran.com*), and we found session replaying on this site was disabled by the site owner. FullStory was used (e.g., in *fbckahoka.org*, *emmausdenver.com*) to replay requests for religious material and prayer requests by users. Inspectlet

was used to replay meta-information (e.g., page title, browser information, dependent resources of websites requested) of religious sites (e.g., *gbcga.com*, *afci.com.au*) browsed by users, which can be leveraged for fingerprinting. We found personal information (e.g., name, email, phone, message, address, login ID), donation details (e.g., donation amount), prayer requests and keywords used during site searches being replayed to Yandex session replay services from 19 religious sites; see Table 18.

Furthermore, AI-based chatbots are being included in religious websites to emulate personal human conversations. Exposure of these conversations to adversaries may divulge personal information of users. We observed chatbots of two religious sites shared personal conversation to third parties: *chertzumc.com* transmitted user conversations in base64 format to an external domain (*chat.amy.us*), and *immersivehistory.com* sent user conversations as is, over a websocket to a third party domain (*socket.tidio.co*).

Leakage type	Religious site	SRS	Leaked information
Personal information	glorygod.ru, aglow.org.uk, novizavet.ru, standrews.ru, slovo-istini.com, zhslovo.ru, sda-spb.ru	Yandex	Name, phone number, email, address/city, message
	nehemiah.ru	Yandex	Location entered to search for the closest church
	mbs.ru, belchurch.org	Yandex	Login ID
	solba.ru	Yandex	Email address used to subscribe for a newsletter
Request for religious material	fbckahoka.org	FullStory	Email address, sermon notes
Request for prayer	fbckahoka.org	FullStory	Full name, email, phone, prayer request
	solba.ru	Yandex	Name, message, donation amount of the prayer request for a patient (Corona and other diseases), and to succeed in studies/exams
Meta information of site requests	lifeteen.com	FullStory	links clicked by users (relating to various religious missions)
	gbcga.com	Inspectlet	Page title, URL browsed, browser information (i.e., browser type, version, webkit, user-agent).
	afci.com.au	Inspectlet	URL and dependencies (CSS, JavaScript) of the site browsed
	bengalipdfbooks.info	Yandex	Links clicked by users
Donation details	novizavet.ru	Yandex	First name, last name, donation amount
	rpconline.ru	Yandex	Donation amount, mode of payment (e.g., bank card)
Keywords uses for searches	new-church.ru, wolrus.org, sda-spb.ru, kateho.ru	Yandex	Keywords used in site searches that may include sensitive information

Table 18: Use cases for information leakage with session replay services (SRS) on religious sites.

6.4.2 Religious sites with security issues

The *Wapiti* scanner identified security issues in 19 (out of 11,888) religious websites — SQL Injection (9), Reflected Cross Site Scripting (7), Server Side Request Forgery (2), Path Traversal (1); see Table 19 for examples of security issues in religious websites. *Christcc.org* is vulnerable to the Path Traversal attack that exposes the local `/etc/passwd` file. Although, user passwords are not revealed from the `/etc/passwd` file, the content (e.g., full names, list of system users indicating software installed on the host) of it can be used for reconnaissance and social engineering efforts, which may eventually lead to reverse shells and local privilege escalations. The potential Reflected Cross Site Scripting attacks that can be launched by some websites (e.g., *abccolumbia.org*, *christcc.org*, *cogsabbath.org*), are proof of the attacker’s ability to execute much more harmful attacks (e.g., steal credentials, hijack user accounts, exfiltrate sensitive information) on users. The same applies to religious websites (e.g., *abccolumbia.org*, *aoffcc.com*, *welfarebc.com*) subjected to SQL Injection vulnerability, where the consequences from such attacks (e.g., unauthorized viewing of user data, removal of data from database tables, attacker gaining database administrative rights) are far reaching. We also scanned religious Android apps pertaining to these religious websites (for security issues) using *Wapiti*, and found *com.subsplashconsulting.s_R858KV* (CCC Camp Hill, PA App) app that corresponds to *christcc.org* religious website, contains 2 endpoints (`https://app.easytithe.com/AppAPI/api/account/churchInfo`, `https://app.easytithe.com/AppAPI/api/account/paymentList`) that are vulnerable to SQL Injection.

Security issue	Website	Details of the security issue
Reflected Cross Site Scripting (XSS)	spiritofmedjugorje.org	This vulnerability is found via injection of parameter <i>ArticleSeq</i> (e.g., https://spiritofmedjugorje.org/index.php?ArticleSeq=%3C%2Fscript%3E%3CScript%3Ealert%28%27wfj7hux5b6%27%29%3C%2Fscript%3E)
SQL Injection	abccolumbia.org	Injection of parameter <i>media_id</i> (e.g., https://abccolumbia.org/video.php?media_id=10%27%20AND%2092%3D92%20AND%20%2714%27%3D%2714). The parameter value passed to <i>media_id</i> is decoded as <i>10' AND 92=92 AND '14'='14</i>
Path Traversal	christcc.org	Linux local files disclosure vulnerability via injection of parameter <i>path</i> — exposes <i>/etc/passwd</i> , <i>/etc/group</i> , <i>/etc/hosts</i> , <i>/etc/host.conf</i> , <i>/etc/resolv.conf</i> , <i>/etc/profile</i> , <i>/etc/csh.login</i> , <i>/etc/fstab</i> , <i>/etc/networks</i> , <i>/etc/services</i> files (e.g., https://christcc.org/vcf_download.php?path=%2Fetc%2Fpasswd)
Server Side Request Forgery (SSRF)	allsaintsphoenix.org	SSRF vulnerability via injection of parameter <i>url</i> (e.g., https://allsaintsphoenix.org/s/cdn/v1.0/i/m?url=http%3A%2F%2Fexternal.url%2Fpage&methods=resize%2C500%2C5000)

Table 19: Examples of security issues in religious websites.

6.4.3 Religious sites flagged as malicious

We found 69 (out of 62,373, 0.1%) religious sites were flagged as malicious by VirusTotal (at least by 5 engines). We only considered sites that apparently were used for malicious purposes according to VirusTotal category labels and community comments, containing keywords including malware, compromised, infection, spyware, fraud, weapons, command and control, bot network and callhome. We also observed 12 malicious domains host tracking scripts/cookies on religious sites, as per VirusTotal (at least by 5 engines): *freecontent.date* (modifies files in Chrome extension folder) and *iclickcdn.com* (website redirected to malicious pages) were flagged as malicious by more than 10 engines. With *Retire.js* [231], we found JavaScript sources (i.e., *bootstrap*, *jquery*, *swfobject*) included in 3 religious sites (*wierdapark-suid.co.za*, *divyabodhanam.org* and *divyabodhanam.org*) were using legacy script versions that are vulnerable to Cross Site Scripting.

6.4.4 Analysis HTTP/HTTPS traffic from religious websites

We analyze the HTTP/HTTPS traffic and characteristic of TLS certificates used in religious websites. We were able to extract 45,004 (72.2%, out of 62,349) websites that use HTTPS; 17,345 requests failed (e.g., because of timeout).

Use of HTTP in religious websites. We found 24 religious websites (out of 62,373, 0.04%) use plain HTTP for communication. HTTP is not secure, and allow adversaries to listen to the traffic sent from these websites, and capture sensitive personal information. We found 14 out of 24 of religious websites that use HTTP, send personal/sensitive information (first/last names, email address, phone number, address, message/comment, prayer request/confession, date of birth/age, password) of users over the clear; see Table 20 for top-5 religious websites that leak personal/sensitive information over HTTP.

Validity period of TLS certificates. Popular browsers (e.g., Google Chrome) have announced in 2020, SSL/TLS certificates cannot be issued for more than 13 months (397 days) [221].

Larger validity periods make it tedious to roll out changes to cryptographic primitives of certificates (e.g., update to a stronger encryption algorithm) by certificate issuers, and to ensure the trust of an identity (i.e., website’s domain). We found 590 (out of 45,004, 1.3%) of the reli-

Website	Name	Email	Phone	Address	Message	DOB/Age	Password	PR/Confession
eliotchapel.org	✓	✓	✓	✓		✓	✓	
nbcog.net	✓	✓	✓	✓	✓			
therockchurchla.org	✓	✓	✓		✓			✓
walkatliberty.com	✓	✓			✓		✓	
catholicfamily.net	✓	✓				✓		✓

Table 20: Top-5 religious websites with most leak-ages of personal/sensitive information over HTTP — DOB = Date of Birth, PR = Prayer Request

gious websites that use HTTPS have a validity period between 24-28 months in the issued certificates; none of the certificate issuers of these certificates are free certificate authorities

— e.g., *Sectigo Limited* (398), *GoDaddy.com, Inc.* (80), *Starfield Technologies, Inc.* (61), *DigiCert Inc* (27).

Analysis of certificate issuers. We observed that the top-5 certificate authorities that issue certificates for the analyzed religious websites are *Let's Encrypt* (29,357/45,004, 65.2%), *cPanel, Inc.* (4996, 11.1%), *Cloudflare, Inc.* (2945, 6.5%), *GoDaddy.com, Inc.* (2416, 5.4%), *DigiCert Inc* (1799, 4%). We also explored the country level distribution of TLS certificate issuing organizations, and found United States (42,618/45,004, 94.7%) and United Kingdom (1724, 3.8%) dominates in the distribution.

TLS certificate signature analysis. We found 41,804 (out of 45,004, 92.9%) of HTTPS religious sites use RSA signature algorithms — i.e., *sha256 with RSA* (41,697), *sha384 with RSA* (106), *sha512 with RSA* (1); all RSA signature algorithms use a public key of at least 2048 bits. In addition, 3200 (out of 45,004, 7.1%) HTTPS religious websites use ECDSA (Elliptic Curve Digital Signature Algorithm) signature algorithm — i.e., *ecdsa with SHA256* (2966), *ecdsa with SHA384* (234). The ECDSA signature algorithm uses shorter keys for the same security level as in RSA with larger keys. Although ECDSA is a more efficient signature algorithm, recent studies found it is more vulnerable to attacks [9].

6.4.5 Third-party tracking scripts

We found 27.9% (17,418/62,373) of religious websites had at least one known tracker on their landing pages, and a total of 359 unique known trackers. We observed popular non-commercial religious websites include commercial trackers on them — e.g., *churchofjesuschrist.org* (a top ranked religious website [255]) included third party scripts from 7 unique commercial tracking domains. The most common known commercial trackers on religious websites were *google-analytics.com* (12,653, 20.3% of websites), *googletagmanager.com* (7064, 11.3%) and *wp.com* (3713, 6%); see Figure 26 for top-10 known tracking scripts. Religious sites we analyzed, are often developed using WordPress and Squarespace

website building services. The scripts included by the former are used for pixel tracking, while the latter use analytics to track users. In addition, the Facebook (*facebook.net*) social media plugin included in religious sites is used to collect information on users' browsing behaviors (e.g., websites and other apps visited), and share this information with other third parties. Furthermore, the PayPal plugin included in religious websites (for online donations) can also be used to track users.

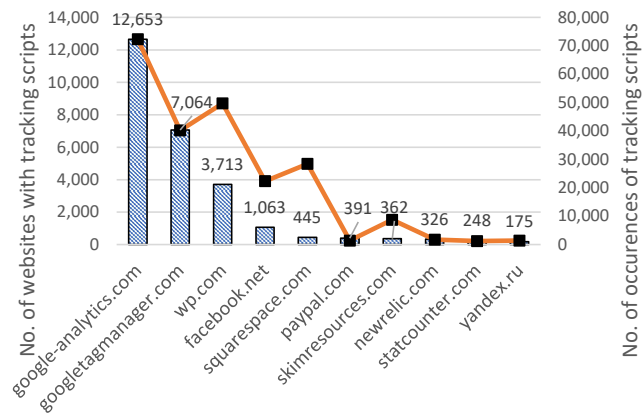


Figure 26: Top-10 known third-party tracking script sources on religious sites — the bars show the number of religious sites with trackers (vertical axis to the left), while the line chart shows the number of occurrences of trackers (vertical axis to the right).

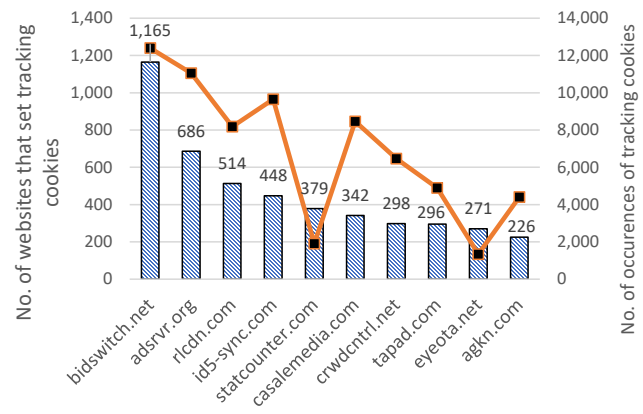


Figure 27: Top-10 known third-party tracking cookies set on religious sites — the bars show the number of religious sites with trackers (vertical axis to the left), while the line chart shows the number of occurrences of trackers (vertical axis to the right).

6.4.6 Third-party tracking cookies

We found 3569/62,373 (5.7%) websites set tracking cookies.

The most number of cookies are set by *bidswitch.net* (1165/62,373, 1.9%), *adsvr.org* (686/62,373, 1.1%) and *rlcdn.com* (514/62,373, 0.01); see Figure 27. *Biblehub.com* and *biblegateway.com* are top ranked religious websites [255] that included cookies set by 42 and 16 tracking domains, respectively; a cookie set by *cpmstar.com* (an adware) on *biblehub.com* expires after 20 years. Cookies set by *statcounter.com* (used for web analytics) expires after 5 years; see Table 21. We also found tracking cookies set by *center.io* on 4 religious websites (*zionbaptistva.com*, *lavendervines.com*, *effect900.com*, *catholicfundraiser.net*) expire in year 9999.

Tracker	#Sites	Cookie expiry		
		1m-1y	1y-5y	> 5y
bidswitch.net	1165	1	1	-
adsvr.org	686	-	690	-
rlcdn.com	517	4	513	-
id5-sync.com	454	390	-	-
demdex.net	201	402	-	-
statcounter.com	379	-	-	379
casalemedia.com	342	2	343	-
crwdcntrl.net	298	298	-	-
tapad.com	298	296	-	-
eyeota.net	271	-	3	-

Table 21: The top-10 known tracking cookies and their expiry periods (m=month, y=year).

6.5 Results: Religious Android apps

Static analysis results: Tracking SDKs and exposed Firebase databases. With LibRadar, we found a total of 7398 tracking SDKs (203 unique) on 1454 religious Android apps. We also used LibRadar to check the usage types of these SDKs (e.g., *Google Mobile Services* is used as a development aid, *Google Analytics* is used for mobile analytics). Similar to religious websites, most tracking SDKs in apps were also from Google (1132/1454,

78%) and Facebook (205/1454, 14.1%). Note that Google tracking SDKs are also used for ad and mobile analytics. Although the collection of analytics can help provide a better user experience and improve protection (e.g., fraud detection [206]), it can also be effectively used for tracking/profiling. A notable example is the *com.prayapp* app that embedded 10 tracking SDKs (including Google and Facebook). The app collects personal information (e.g., location, app usage), and apparently, the app owners also purchase data (e.g., gender, age, ethnicity, religious affiliation) from third parties for better profiling [110]; they may also share personal information to third parties (e.g., advertisers) for commercial purposes.

We found 55 (3.8%, 1454) religious Android apps exposed their Firebase databases due to unprotected endpoints; 7 of these apps leaked sensitive information—e.g., user name, password, phone number, email, profile picture, chat details, API key, device type. However, we did not verify/use/store this info (deleted immediately after checking the data types). Notable examples: Vedic Library (*com.hinbook.library*) — an app that supports individual spiritual enhancement (100K+ installs), and Catholic Connect (*com.catholicconnect*) — a social media platform to build and collaborate between Catholic communities (10K+ installs).

Dynamic analysis results. Examples from what we observed from our dynamic analysis include a Christian dating chat app (*cdff.mobileapp*, 1M+ installs), that sent login information via HTTP to a domain owned by the same owner (*christiandatingforfree.com*). We also found *cdff.mobileapp* and *com.avrpt.teachingsofswamidayananda* sent device information (device ID, device model, device manufacturer, device operating system, screen resolution) over HTTP to *christiandatingforfree.com* and *avrpt.com* domains, respectively (both the apps and corresponding domains are owned by the same party). Such device data can be used to passively track users by fingerprinting their devices.

Session replaying from apps. We found that the UXCam session replay service collected

users' location (i.e., GPS coordinates) from the *Tabella Catholic* app. Hotjar and Mouse-Flow collected fingerprinting information from *Muslim kids* (e.g., device model) and *Budhist Sangam* (e.g., mouse events) apps, respectively.

Religious apps and 3rd-party domains flagged as malicious. 29/1454 religious apps were flagged as malicious by VirusTotal: one app by 10 engines, eight apps by two engines and 20 apps flagged by one engine. *islamictech.sifgo* (50K+ installations), is flagged as malicious by 10 security engines. 8 apps included the *Android.WIN32.MobiDash.bm* [184] stealthy adware that usually displays ads when the mobile device screen is unlocked. 8 apps contained the *AdLibrary:Generisk* [1] malware that steals information (e.g., Facebook credentials). We also observed calls to two malicious 3rd-party domains by religious apps — *jainpanchang.in* and *orthodoxfacts.org* third party domains were included in *com.mosync.app_Jain_Panchang* (Jain Panchang) and *com.orthodoxfacts* (Orthodox Sayings) religious Android apps, respectively. Jain Panchang requires the *WRITE_SECURE_SETTINGS*³ Android permission, allowing the app to read/write secure systems settings, which is not supposed to be used by third-party apps.

6.6 Recommendations

To safeguard the privacy/security of users using religious online services, from tracking and privacy exposures, adherence to best practices is vital. Therefore, developers need to be vigilant in including third party scripts/libraries in religious websites, and should do proper scanning before using such dependencies. Privacy regulations require personal data used to interact with religious websites to be protected; according to GDPR [101], personal data relating to religious beliefs are deemed sensitive. However, we observed religious online services do not fully comply with these regulations. Proliferation of privacy regulations should drive faith based organization to partner with trusted service providers that comply

³<https://tinyurl.com/489ee9xu>

with industry standards/best practices. In addition, routine risk assessments, audits and inspections of the policies/procedures of religious online services should be carried out by the owners of these services.

6.7 Summary

Online religious services raise concerns about user privacy. Information with deeply personal content shared by faith-based communities over online religious services are accessed by various third parties (via tracking scripts/cookies, session replay) that include commercial entities, governments (for surveillance purposes) [176]. We observed 196 religious websites include session replay scripts pertaining to FullStory, Inspectlet, and Yandex. AI-based chatbots included on at least 2 religious websites may divulge personal information via user conversations with third parties. Security issues were detected in a few religious sites, that may lead into privacy issues; 19 religious websites were vulnerable to security issues (i.e., SQL Injection, Reflected Cross Site Scripting, Server Side Request Forgery, Path Traversal); 69 websites were flagged as malicious by VirusTotal; 14 religious websites that use HTTP, send personal/sensitive information over the clear. We observed 28% and 6% of religious websites, included tracking scripts and cookies, respectively. Google dominated in tracking on both religious websites and Android apps.

Chapter 7

Cloaking behaviors of malicious websites

7.1 Introduction

Websites are often used to launch social engineering attacks. For example, phishing websites exploit human weaknesses to steal sensitive user information; similarly, malware websites employ techniques to deceive users to download malware (e.g., ransomware) infecting user machines; cyber-criminals take advantage of ads hosted on low-tier networks using social engineering techniques [292]. Sophisticated phishing and malware websites hosted on *squatting* domains are deployed to deceive users by impersonating websites of high profile companies and organizations (the so-called *elite* phishing domains [282, 216]). The domains hosting these phishing sites are subjected to typo-squatting (e.g., foxnews.com) and combo-squatting (e.g., support-apple.com-identify.us). These phishing sites impersonate trusted brand names using fake web content and typo-squatted domain names.

Additionally, phishing and malicious sites employ evasion techniques to avoid exposing malicious content to search engine crawler as opposed to human users [261, 211, 156]. The practice of displaying different content to a crawler as opposed to a browser/user is known as *cloaking*. Cloaking helps attackers to reduce the possibility of getting their services blacklisted. To discourage such practices, search engine providers also offer guidelines for

website owners/maintainers—see e.g., Google [125].

There have been several studies on malware and phishing sites, albeit not so much on *squatting/elite* phishing and malicious domains engaged in cloaking. Past studies on cloaked malicious sites relied on specific types of websites and attacks (e.g., payment sites and phishing). Tian et al. [282] found 1175 (0.18%) phishing sites that are likely impersonating popular brands from 657,663 squatting domains (extracted from a collection of 224 million DNS records listed in ActiveDNS project [165]). They focused mainly on phishing web pages identified using specific keywords from logos, login forms, and other input fields (mostly credential phishing). Invernizzi et al. [156] studied web cloaking resulting from blackhat search engine optimizations and malicious advertising, using websites relating to luxury storefronts, health, and software. Oest et al. [203, 204, 202] used crafted PayPal-branded websites, and impersonated websites targeting a major financial service provider to study phishing. As such, the data sets used in these past studies do not cover a wide variety of malicious URLs. Our methodology also includes capturing cloaking in dynamic elements (e.g., iframes) of websites and taking semantics of web content into consideration, which were not adequately addressed in the past; e.g., Tian et al. [282] did not consider dynamically/JavaScript-generated page content due to high overhead.

We focus on understanding cloaking behaviors of a broad set of malicious sites hosted on squatting domains. These sites engage in phishing, malware distribution, and other social engineering attacks. We use DNSTwist [85] to generate already registered squatting domains that are potentially malicious. DNSTwist uses fuzzy hashing to identify malicious squatting domains by comparing its web page content with the corresponding seed domain. The squatting domains extracted from DNSTwist host content from a wide variety of possible malicious websites. To verify the ground truth of malicious squatting sites generated from DNSTwist, we adopt a semi-automated process leveraging the Symantec SiteReview [271] tool, which significantly outperformed both commercial and academic tools

(e.g., VirusTotal [298], Off-the-Hook [186]) in our manual tests; cf. Vallina et al. [294].

We compare page content between a search engine crawler and browser client to detect cloaked malicious websites. For this purpose, we develop a crawler to collect page source, links, content, screenshots, headers from websites hosted on squatting domains. To distinguish between dynamic vs. cloaked pages, we employ a set of heuristics; see Section 7.3.3. To mimic a regular user browser (Chrome) and a search engine crawler (Google), we simply rely on custom browser user-agents and referrer headers. For the remainder of this chapter, we use *GooglebotUA*, *ChromeUA*, *ChromeMobileUA* for search engine crawler, browser (desktop) and browser (mobile) user-agents interchangeably. Attackers may also leverage various evasion techniques to obfuscate the page-layout and HTML source, e.g., keywords in response headers to trick a search engine crawler [282], manipulate visual similarity between a phishing and a corresponding benign site [211]. Hence, we also examine the extent of such obfuscation in cloaked malicious websites.

Out of the 100,000 squatting domains (i.e., domain list category A in Table 22), VirusTotal flagged only 2256 (2.3%) domains as malicious—in contrast to the ground truth (74%), as verified via our semi-automated process. From the 100,000 squatting domains, we found 3880 (3.88%) as cloaked; 127 (i.e., 3.3% of 3880) of these cloaked domains are flagged by VirusTotal—in contrast to our established ground truth (80%).

On dynamic sites, we observed different types of cloaked content (e.g., technical support scams, lottery scams, malicious browser extensions, malicious links) served to users from the same domain at different times.¹ The number of cloaked sites identified from dynamic sites (861, 0.9%) is also significant, although it is certainly a lower bound as the

¹Note that serving dynamic content to GooglebotUA by a website may not necessarily be treated as cloaking. Response from a dynamic site to GooglebotUA may serve a non-dynamic version of the content that is tailored for that site (e.g., static HTML version), known as *dynamic rendering*; see: <https://developers.google.com/search/docs/guides/dynamic-rendering>. Although with dynamic rendering, a static view of a dynamic website is shown to GooglebotUA, the response content rendered to ChromeUA is dynamic. However, we consider serving significantly different content between ChromeUA and GooglebotUA as cloaking (e.g., page about cats to GooglebotUA and a page about dogs to ChromeUA).

dynamism exhibited by these sites is inconsistent between consecutive requests.

Our results may be impacted by several factors: sites disallowing requests from automated crawlers, limitation of our heuristics, dynamism of cloaking, and the use of SiteReview for establishing our ground-truth. Still, our findings uncover several cloaking behaviors of malicious sites and our methodology can also help detect these sites at scale.

Contributions.

1. We measure cloaking in malicious websites between a client browser (ChromeUA) and a search engine crawler (GooglebotUA) using a broader set of malicious domains with a more comprehensive methodology compared to existing work. Our technique improves the detection of cloaked malicious sites compared to past studies (e.g., cloaking in dynamically generated web content), and detect various scams (e.g., deceptive prize notices and lottery scams) and malicious content (e.g., malicious browser extensions) rendered in cloaked web pages.
2. Our methodology can identify 80% cloaked malicious domains from our ground truth; the detection rate also remained consistent between repeated measurements. For comparison, see e.g., Oest et al. [202] (detected 23% cloaked phishing sites in their full tests), Invernizzi et al. [156] (detected 4.9% and 11.7% cloaked URLs with high-risk keywords in Google advertisements and search results respectively), and VirusTotal (3.3% with our own dataset).
3. We highlight the role of domain generation engines such as DNSTwist [85], which can quickly provide a list of highly-likely malicious domains to serve as ground-truth, especially if used along with our heuristics.

7.2 Related work

In this section, we compare previous work on detecting malicious sites, analyzing resiliency of blacklists, and the use of various heuristics to detect malicious sites. We also compare our methodology and results with past work.

Vadreu et al. [292] studied social engineering attacks delivered via malicious advertisements, and found 11,341 (16.1%) out of 70,541 publisher sites hosting malicious ads. Except for lottery/gift (18%) and fake software (15.4%), Google Safe Browsing (GSB) [133] detected only under 1.4% of other types of malicious ads (e.g., technical support). Tian et al. [282] studied elite phishing domains targeting desktop and mobile users, and found sites hosted on these domains were mostly used for credential phishing (e.g., impersonating of payment, payroll and freight systems). They found 1175 out of 657,663 squatting domains were related to phishing; as the source of their domain list, they used 224 million DNS records in ActiveDNS project [165]). However, only 100 (8.5% of 1175) domains were flagged as malicious by PhishTank, eCrimeX and VirusTotal (with 70+ blacklists). They also compared evasion techniques between a desktop and a mobile client (Chrome). We study search-engine-based cloaking (ChromeUA vs. GooglebotUA), focusing on various types of malicious websites (beyond credential phishing).

Invernizzi et al. [156] studied variations in cloaking with search and advertisement URLs. They used several cloaking detection techniques based on web page features, e.g., content, screenshot, element, request tree and topic similarities; we adopt some of these techniques. In addition to static content analysis, we also analyze dynamic content. We compare screenshots of web pages between ChromeUA and GooglebotUA using OCR to find discrepancies in visual appearance (i.e., cloaking). Some of these discrepancies are not detected by simply comparing the content, but by supplementing other methods (e.g., semantics of a web page). The differences in the meaning of a page's content between the crawler and the browser (i.e., semantic cloaking) are used to deceive a search engine

ranking algorithm [313], where search engine operators are more likely to be duped with the cloaked content. We use topic similarity evaluated using the LDA algorithm [246] to identify the semantic differences of web pages between ChromeUA and GooglebotUA.

Oest et al. [202] presented a scalable framework called *PhishFarm* for testing the resiliency of anti-phishing and browser blacklists, using 2,380 phishing sites deployed by the authors. Between mid-2017 and mid-2018, they found that the blacklisting functionality in mobile browsers was broken and cloaked phishing sites were less likely to be blacklisted compared to non-cloaked sites. The authors also mentioned blacklisting malicious websites remained low for mobile browsers compared to desktop browsers. We also observed a similar trend in our tests.

Rao et al. [229] used characteristics of a URL (i.e., hostname, full URL) to determine legitimate websites. Marchal et al. [186] used parts of a URL that are manipulated by a phisher (e.g., subdomains, web application path) to detect phishing sites. Panum et al. [211] reviewed highly influential past work to assess strategies with adversarial robustness to detect phishing. These strategies include distinguishing between phishing and benign websites using visual similarity and leveraging URL lexical features. In our study, we use DNSTwist to generate potential malicious typo-squatting domains using lexical information of seed domains.

In summary, past measurement studies [282, 164, 156, 292] are mostly focused on specific categories of malicious websites (e.g., phishing, malware, social engineering). Each of these categories of websites may participate in cloaking. Several studies have used self-crafted URLs hosting content of particular malicious categories (e.g., phishing) or brands (e.g., PayPal) [203, 202, 282]. We use a broad set of registered squatting domains—combo-squatting (HTTPS only) and typo-squatting domains, hosting different types of potentially malicious websites to study cloaking behaviors.

7.3 Methodology

In this section, we explain our methodology to study cloaking behaviors in phishing and malware websites. We generate domains that may host potential phishing/malicious sites and pass them as input to our crawler. Various features (e.g., headers, links, page source/content, screenshots) are saved, and processed by an *analyzer* to identify cloaked sites and the results are stored into a database for further evaluation; see Fig. 28 for an overview of our experimental setup.

7.3.1 Generating squatting domains

Attackers are more inclined to impersonate popular websites, both in content and domain name, by hosting malicious sites on squatting domains [316, 318, 282]. These domains can be categorized as typo-squatting or combo-squatting. The domain lists used in our work is listed in Table 22. We generate 100,000 squatting domains (see list category A) using the following methods. The squatting domains sampled from these methods are from possible malicious domains.

List label	Number of domains	Experiment type
A	100,000	Cloaking is measured between ChromeUA and GooglebotUA
B	25,000	Cloaking is measured between ChromeUA and GooglebotUA for desktop environment, and between ChromeMobileUA and GooglebotUA for mobile environment. A random subset of domains from list A is used.
C	10,000	Comparison of HTTP vs. HTTPS cloaked sites (5000 each) hosted on combo-squatting domains. We use the same user-agents as in list A to identify cloaked domains.
D	5000	Comparison of user-agent vs. referrer cloaking of sites hosted on squatting domains. For referrer cloaking, we use ChromeUA with referrer header: <code>http://www.google.com</code> .

Table 22: Squatting domain lists used in our experiments

Typo-squatting domains from DNSTwist

DNSTwist [85] takes a specific domain name as a seed, and generates a variety of potential registered phishing/malware domains. The domains generated in two consecutive runs of DNSTwist are not the same. This is because DNSTwist passes the seed domain provided to a function (`DomainFuzz`), which randomly generates many permutations of domain names similar to the seed domain, but with typographical errors. To determine domains hosting malicious content, DNSTwist use fuzzy hashes to identify sites serving similar content as their original domains (using the *ssdeep* option).

We provide top 1983 Tranco websites [170] as seeds to DNSTwist. From Mar. 22, 2019 to Mar. 27, 2019, we generate 277,075 already registered, unique typo-squatting domains; we then randomly choose 92,200 of these domains for our experiments (to save time). We choose the timings of the extraction of domains around the same time as the actual crawling of the sites, to ensure most of them are still responsive during crawling as typo-squatting domains can be recycled quickly [282].

The typo-squatting domains generated from DNSTwist are of the following types, explained using `google.com` as the seed domain. (1) Addition: A character is added at the end of the *public suffix+I²* segment of the domain (`googlea.com`). (2) Bitsquatting: Flips one bit of the domain (`foogle.com`). (3) Homoglyph: visually similar domains, although the characters are not the same as the seed domain (`g0ogle.com`). (4) Hyphenation: A hyphen is added in between the characters of the seed domain (`g-oogle.com`). (5) Insertion: A character is inserted in between characters of the seed domain (`goo9gle.com`). (6) Omission: A character in the seed domain is removed (`goole.com`). (7) Repetition: A character in the seed domain is repeated consecutively, two or more times (`ggoogle.com`). (8) Replacement: A character in the seed domain is replaced with another character (`toogle.com`). (9) Sub-domain: A period is

²A public suffix is defined as “one under which Internet users can (or historically could) directly register names” see: <https://publicsuffix.org>.

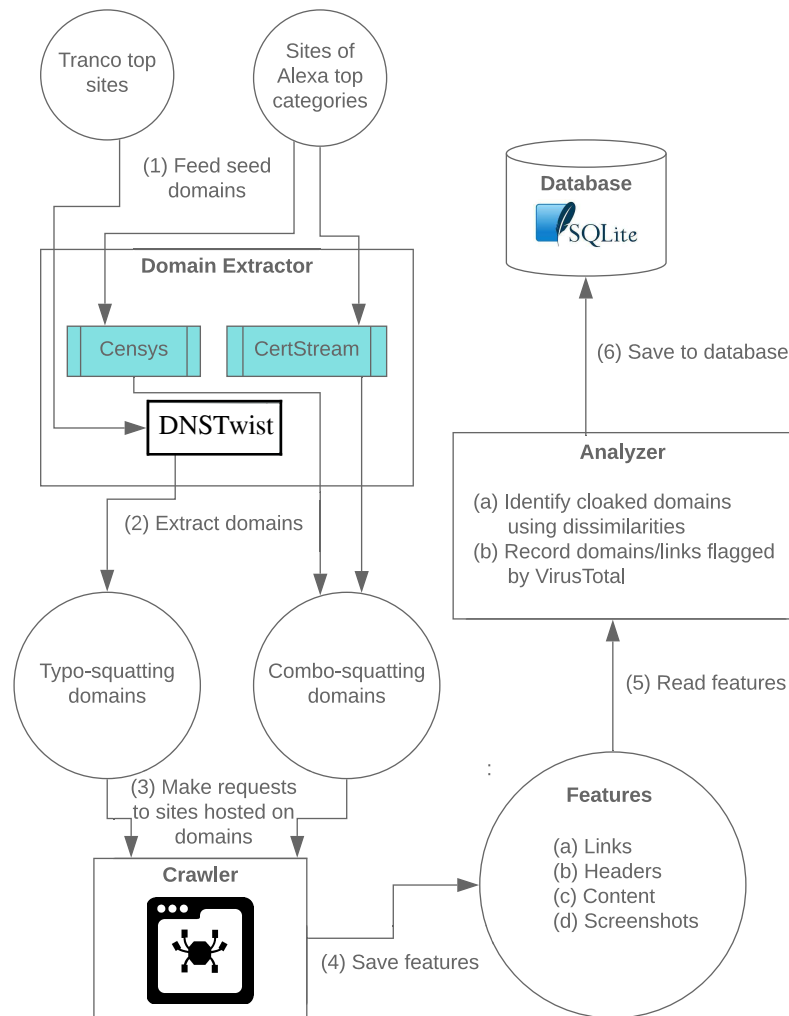


Figure 28: Our system setup.

inserted in between any two characters of the seed domain to transform it to a sub-domain (`g.oogle.com`). (10) Transposition: Position of two characters in the seed domain is swapped (`gogole.com`). (11) Vowel-swap: A vowel character is replaced with another vowel (`goagle.com`).

Combo-squatting domains

Combo-squatting domains are concatenations of the target domain with other characters or words. These domains generally do not have spelling deviations from the target, and require active involvement from the attacker (e.g., social engineering); cf. typo-squatting is

passive and relies on a user’s accidental typing of a domain name [164]. Combo-squatting domains are also used in phishing attacks [282]. Therefore, we generate 7800 combo-squatting domains as follows.

We collect top-50 sites of 16 categories (e.g., adult, business, computers, health, news, shopping, sports), and top-50 Alexa sites specific to China—a total of 850. During our preliminary manual verification, we observe a lot of phishing and malware sites are hosted in China, and thus we choose Alexa top-50 sites from China.

Then, we identify domain names that partially match any of these 850 domains from certificates that are used to host HTTPS phishing/malware sites in order to deceive legitimate users [147]. We only consider certificates issued after Jan. 1, 2019 to minimize the collection of already recycled domains. We collect combo-squatting domains that serve content over HTTPS between Apr. 4–9, 2019 using the following sources. (1) Censys: This is a search engine [88] that aggregates information of networked devices, websites and certificates deployed. We check the subject common name field of certificates against our 850 target domains. (2) CertStream: The certificate data in Certificate Transparency (CT) logs is polled over a websocket (`wss://certstream.calidog.io`) in CertStream [43]. We then check the common name field of certificates against our target domains.

To derive combo-squatting domains served via HTTP, we extract domain names from the DNS A records from Project Sonar [224]. After extracting the domains running on port 80 that return a 200 response code (i.e., non-recycled domains), we partially match them with the 850 target domains, to filter the combo-squatting domains that are derivations of the top brands.

As many combo-squatting domains are benign (e.g., `mail.google.com`), we use SquatPhish [268] to filter only those domains exploited for phishing that are derived from above sources (i.e., Censys, CertStream, Project Sonar). SquatPhish leverages a machine learning model to identify phishing pages based on the HTML source and text extracted

from images included in a web page. We use SquatPhish to filter 7800 phishing domains from 205,263 combo-squatting domains collected from the above mentioned sources. We do not consider domains that return a 4xx or 5xx response code, as those domains may already have been recycled.

7.3.2 Our crawler

To identify cloaking activity, we extract features from 100,000 web pages hosted on potential malicious squatting domains, using GooglebotUA and ChromeUA (and a subset of the same websites by ChromeMobileUA). We use GooglebotUA, ChromeUA and ChromeMobileUA for our experiments by manipulating the “user-agent” field of the request header; see 2.5 for a discussion on cloaking types.

We use Puppeteer [130] to implement our crawler. Puppeteer provides high level APIs to control the Chrome browser and can be customized to run as headless to load dynamic content before saving the web pages. Compared to other alternatives (including Selenium [251]), Puppeteer offers the flexibility of handling failed requests gracefully and is less error prone [282]. Tian et al. [282] also used a crawler based on Puppeteer. However, unlike them, our crawler renders content that is dynamically generated before saving (Tian et al. [282] chose not to consider content dynamically generated by JavaScript due to the high overhead). We believe that dynamic source files (e.g., JavaScript, Flash) may render differently based on the user-agent of a request (e.g., the list of links shown in an iframe are benign for GooglebotUA, but malicious for ChromeUA). To identify web pages with dynamic content, we request the home page of each website twice from GooglebotUA and ChromeUA. The GooglebotUA and ChromeUA are represented as C and B, and the iterations of requests from each client is 1 and 2, the sequence of requests made for a particular website is labeled as C1, B1, C2, B2.

Web servers may have heuristics to determine automated crawlers and reject their service. We incorporated few mitigation steps in our crawler to minimize these effects; e.g., we manipulate webdriver, plugins and language properties of the navigator object of the browser accordingly [155].

We crawl the sites hosted on squatting domains between April 10 to April 13, 2019, and run the crawler on 10 Amazon EC2 instances (c5.2xlarge) setup with Ubuntu 16.04 (8 vCPU, 16GB RAM). For our experiments, we do not consider sites where the differences in content between GooglebotUA and ChromeUA are minor or benign. Some of these sites redirect to non-malicious top-1M Tranco sites [170] from ChromeUA. There are also sites that throw connection errors; see Section 7.4 for an overview of issues encountered during crawling.

During crawling of each site, we gather features to identify potential cloaking activity of possible malicious domains. These features include HTTP headers, page source/content (both static and dynamic), links including those generated from dynamic content (includes those in DOM objects within iframe elements) and screenshots.

7.3.3 Analyzer

The analyzer process applies heuristics to features of websites collected during crawling, in order to identify cloaked websites. In this section, we explain the heuristics and rules applied while processing the saved features. These heuristics are only applied if the HTML page source and screenshots are successfully saved for all C1, B1, C2, B2 visits of a website. The results evaluated by the *analyzer* are saved into a SQLite database.

Skipping domains with benign content

Domain name registrars (e.g., GoDaddy, Sedo) advertise domains available for sale on their landing pages. The content of such landing pages sometimes differ slightly between

GooglebotUA and ChromeUA. Since such differences should not be attributed to cloaking, we skip those domains from processing. For non-English sites, we use Google translator to detect the language of those sites and translate the content to English prior to processing. For screenshots, we use the Tesseract-OCR [274] library to extract the textual content. If the extracted text from a screenshot is non-English, Tesseract-OCR library takes a significant amount of time to process (sometimes over 30 seconds). Therefore, we call the Tesseract-OCR library for only those sites identified as cloaked with content dissimilarities method using our heuristics described in Section 7.3.3. Some domains are redirected to top-1M Tranco sites [170]. Legitimate companies may buy squatting domains to protect users (a request to the possible squatting domain is redirected to the corresponding legitimate site [316])—see Section 7.3.3. Therefore, we do not consider these domains for our experiments.

Eliminating squatting domains owned by popular sites

Entities owning popular domains (e.g., top Tranco sites) buy squatting domains to safeguard its clients who may accidentally browse to those sites by mistyping their URLs. These squatting domains may not always redirect a user to the original popular site. To eliminate such domains from our measurements, we use the organization owning both the squatting and corresponding popular domains using the WHOIS records [192]. If both these domains are registered by the same organization, we disregard them from our analysis. Out of all cloaked domains, only one squatting domain (`expedia.com`) is owned by the same organization (Expedia, Inc: `expedia.com`). Therefore, we eliminate the particular domain from our analysis. However, the following types of squatting domains are not eliminated from the analysis, as we cannot determine if those domains are also owned by the corresponding popular domain’s organization: 8 domains with WHOIS registrant name/organization information recorded as “REDACTED FOR PRIVACY”, and 20

squatting domains registered by *Domains By Proxy* [86] where the registrar itself is listed as the WHOIS administrative contact.

Domains with exceptions

We observe that some sites do not allow automated crawlers to access them. This observation holds for both GooglebotUA and ChromeUA. Unfortunately, our automation cannot determine potential cloaking activities in some sites that are prevented from accessing with GooglebotUA. Some sites display failures such as “Too many requests“, “Page cannot be displayed. Please contact your service provider for more details” and “404 - File or directory not found” when requested from our automated crawler. We experience such failures despite the use of known techniques to avoid crawler issues with accessing websites [155]. However, upon manual inspection, we notice that some of these sites engage in cloaking.

Links flagged by blacklists

Target URLs hosting phishing or malicious content that are flagged by blacklists are of different forms.

Sites redirecting to websites flagged by blacklists. We record URLs redirected from squatting domains to websites that are also flagged by blacklists. We use VirusTotal to determine how many of the redirected sites are flagged as phishing or malicious.

Identify links in iframes flagged by blacklists. We traverse the Document Object Model (DOM) objects within iframes elements (including child iframes) of sites hosted on squatting domains (level 1 URLs) to find dynamically generated second level links of sites that are flagged by blacklists. A listing of such second level links appearing on an iframe of a site is shown in Figure 29b. However, most of these links show a set of related third level links when clicked on any one of them. These third level links will lead to actual sites described in link descriptions. We run 5000 link URLs from each of these 3 levels

through VirusTotal on a daily basis to identify if any of those are flagged as phishing or malware. The first level URLs for this exercise is selected randomly from list category A in Table 22. This help us find the rate at which link URLs hosting phishing or malware content is detected by available blacklists.

Evaluate dissimilarities of website features

We evaluate the following dissimilarities based on the website features collected during crawling. These heuristics facilitate in finding websites engaged in cloaking.

Header dissimilarities. Although *title*, *keyword* and *description* are not part of the standard HTTP response header, adversaries appear to include these fields in the HTTP response headers [313]. Therefore, we compare these fields between ChromeUA and GooglebotUA to find instances of cloaking.

Link dissimilarities. We find the links in rendered web pages from GooglebotUA that are missing from ChromeUA, and vice-versa. In addition, we also identify which of those links are malicious using VirusTotal.

Content dissimilarities. We extract text surrounding *h*, *p*, *a* and *title* tags of the HTML page source following rendering of dynamic source code (e.g., JavaScript). We also consider HTML forms along with *type*, *name*, *submit* and *placeholder* attributes. Stop words (e.g., the, a, an) are removed from the extracted content.³ Then we evaluate the SimHash [54] of the extracted page source from GooglebotUA and ChromeUA, and compute the hamming distance between them.⁴ If the hamming distance exceeds a preset threshold ($tI=20$), we assume that the page is likely to be cloaked. We set the threshold after manual verification, where we find $tI=20$ gives optimal results after removing benign differences (e.g., pages having random session identifiers or timestamps). This threshold is

³<https://pythonspot.com/nltk-stop-words/>

⁴SimHash is a FuzzyHash that is used to identify similar documents. The difference between two documents is measured using the hamming distance—larger distance implies higher dissimilarity.

also close to Tian et al. [282] (distance between 24 and 36). We set a second threshold t_2 for pages with dynamic content. The same value (20) appears to be adequate in this case too. We define a static page if the following is satisfied:

$|FH(C1) - FH(C2)| = 0$ AND $|FH(B1) - FH(B2)| = 0$; here FH represents FuzzyHash.

A static page is possibly cloaked if the following is satisfied:

$|FH(C1) - FH(B1)| > t_1$ AND $|FH(C2) - FH(B2)| > t_1$.

We also compare the semantics of a page between GooglebotUA and ChromeUA to determine if the specific page is cloaked. We identify the most prominent topic of a page (i.e., topic of the page content with highest probability) using the Latent Dirichlet Allocation (LDA) algorithm [246]. A topic in LDA is a set of related words extracted from the document with probabilities of their prominence assigned to them. If T_b and T_c are the most prominent topics corresponding to page content from GooglebotUA and ChromeUA, the static page previously identified as likely to be cloaked has a high probability of being cloaked when $T_b \neq T_c$. Similarly, a page with dynamic content is cloaked if:

$(|FH(C1) - FH(C2)| > t_2$ OR $|FH(B1) - FH(B2)| > t_2)$ AND

$(|FH(C1) - FH(B1)| > t_1$ AND $|FH(C2) - FH(B2)| > t_1)$

AND $T_c \neq T_b$.

Image dissimilarities. Using the page content at source code level to determine cloaking may not be sufficient, and it should be complemented with the visual differences of the page (i.e., screenshots). This is because, content rendered by dynamic source code (e.g., JavaScript, Flash) and advertisement displayed cannot be captured from the page source. Therefore, with this method, we follow the same procedure as for *Content dissimilarities*, except that we use ImageHash as the FuzzyHash to evaluate the differences of screenshots between GooglebotUA and ChromeUA. Very small color perturbations (between benign and malicious views) in the space of humans yield significant changes in the binary representation [211] of a web page screenshot.

7.3.4 Limitations

We exclude sites that our crawler could not reach. Also, the number of cloaked sites we identify is a lower bound due to the choice of our heuristics. According to our observations, some cloaked sites with dynamic content show distinct content at different times (cf. [202]). Therefore, our results with dynamic sites are a lower bound and is based on content rendered at the time the request is initiated from the automated crawler, where these results may differ on each request for dynamic websites.

Both academic and commercial tools available are not accurate in categorizing social engineering sites hosted on squatting domains in the wild; e.g., *Off-the-Hook* [186] gives false negatives for typo-squatting domains, SquatPhish [268] mostly detects credential phishing. However, we observe that the Symantec *SiteReview* tool detects malicious squatting domains at a comparatively higher accuracy (42.6%). SiteReview accepts the domain URL as input, but not the page content. For dynamic websites, the content viewed by our crawler may not be the same as what is analyzed by SiteReview (i.e., view of a web page may change with time due to dynamic behavior). Therefore, we have limited control in identifying the content category of a site using the SiteReview tool.

7.4 Issues during crawling

In this section, we explain the errors, disallowing of requests by web servers and failures encountered during crawling of websites. The data shown in this section are based on squatting domain list category A in Table 22.

Errors during crawling. We crawled 100,000 sites hosted on squatting domains by imitating the ChromeUA and GooglebotUA user-agents. Out of them, 9712 (9.7%) and 9899 (9.9%) requests encountered errors during crawling from ChromeUA and GooglebotUA. Requests initiated from GooglebotUA had a slightly higher number of errors. Table 23

shows the top 5 errors. Most errors were due to timeouts; ChromeUA (4423, 4.54%),

Error	ChromeUA	GooglebotUA
Navigation Timeout Exceeded: 30000ms exceeded	4423	4502
ERR_NAME_NOT_RESOLVED	1640	1594
Execution context was destroyed, most likely because of a navigation	893	584
ERR_CONNECTION_REFUSED	820	608
ERR_CERT_COMMON_NAME_INVALID	343	341

Table 23: Top 5 errors encountered during crawling

GooglebotUA (4502, 4.5%). We set a 30 seconds timeout for each request made from the crawler, as it is a reasonable time interval within which a web page can load. Setting a higher timeout value not only reduces our ability to crawl a larger number of URLs within a reasonable time period, but also increase the chance of crashing the crawler. If the timeout is increased from 30 to 60 seconds, we were able to successfully crawl more sites, although adhoc crashing of the crawling automation is experienced. However, in this case, the timeout errors observed was lower than having a 30 seconds timeout; ChromeUA (3418), GooglebotUA (3745). ERR_NAME_NOT_RESOLVED are DNS related errors that are most likely to be caused by issues related to client browser issues or firewall settings [213]; ChromeUA (1640, 1.6%), GooglebotUA (1594, 1.6%). To validate this aspect, we crawled sites that resulted in ERR_NAME_NOT_RESOLVED errors from a separate residential machine located in the same city, and found a significant proportion of them didn't show this error; ChromeUA (316, 0.3%), GooglebotUA (380, 0.4%). Some of these sites even didn't return an error from the new location; ChromeUA (219, 0.2%), GooglebotUA (263, 0.3%). ERR_CONNECTION_REFUSED errors are usually caused by DNS, proxy server or browser cache issues. Some sites threw errors due to loosing of its execution context. This can happen when a web page loses its execution context while navigating from the crawler. Therefore, running a callback relevant for a specific context that is not applicable during the current navigation can throw an error. ERR_CERT_COMMON_NAME_INVALID errors signal a problem with the SSL/TLS connection where the client cannot verify the certificate.

Failures based on user-agent. Web hosting providers often block clients with unusual

Failure	ChromeUA	GooglebotUA
Too many requests	2640	2448
Page cannot be displayed. Please contact your service provider for more details	1368	1369
404 - File or directory not found	72	472

Table 24: Failures while crawling

traffic. We observed 152 (0.2%) sites were blocked from GooglebotUA by the web hosting provider. In order to block requests from bots (e.g., GooglebotUA), web hosting services use different techniques to identify them [248]. For example, honeypots consisting of links that are only visible to bots are used to attract crawlers, to detect and have them blocked [248]. Different types of content observed in these blocked sites are in Table 25. Content of some of these sites are in Chinese (e.g., <http://diirk.com>). Also, during our crawling, we noticed some sites did not accept requests initiated from automated crawlers. These failures depend on the user-agent of the request. We show these failures in Table 24. Some sites showed “Too many requests” failures when requesting a site from both GooglebotUA and ChromeUA. This behavior was consistent between ChromeUA (2640, 2.6%) and GooglebotUA (2448, 2.4%). This failure was also observed from a real browser when the site was requested repeatedly. We found “404 - File or directory not found” errors were more than six times higher with GooglebotUA (472, 0.5%) compared to ChromeUA (72, 0.07%). The robots.txt file which is in the root directory of a website can be configured to prevent automated crawlers from requesting the site [248]. However, some of these sites may not want to block popular search engine crawlers such as Google, as otherwise it will impact their site ranking. We found 19,040 (19%) websites were disallowed according to the rules in robots.txt which is significant. Our crawler is able to scrape the content of these websites. From these sites, 4722 (4.7%) showed benign content, and the rest of them mostly contained a listing of links and phishing/malware related content. Out of the sites that are disallowed from robots.txt, a smaller fraction showed “Too many requests” failures; ChromeUA (1583, 1.5%) and GooglebotUA (1460, 1.4%).

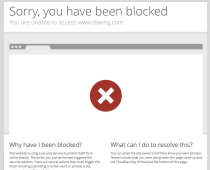


Page content	No. of sites	Example site
Sorry, you have been blocked. You are unable to access [DOMAIN] Why have I been blocked? This website is using a security service to protect itself from online attacks.	136	http://dawng.com 
Your request has an illegal parameter and has been blocked by the webmaster settings!... (Chinese translation)	12	http://diirk.com 
..Access to this page has been denied.. An action you just performed triggered a security alert and blocked your access to this page. This could be because you submitted a SQL command, a certain word or phrase, or invalid data. ...	4	https://support.bed-booking.com 

Table 25: Sites blocked from GooglebotUA

7.5 Ground truth

Some sites hosted on squatting domains are malicious and they may engage in social engineering attacks of various forms such as credential phishing, spear phishing, tech scams, and social engineering ad campaigns. However, most existing tools detect only particular types of social engineering attacks. For example, SquatPhish [268] is a machine learning model to detect phishing sites with input fields (mostly credential phishing). *Off-the-Hook* [186] is a client side browser extension capable of detecting most forms of phishing pages but does not support the detection of sites hosted on squatting domains. We find Symantec’s SiteReview online tool is very effective in correctly categorizing most social engineering sites compared to other tools. Although the accuracy of the ground truth determined from SiteReview may not be perfect, from our manual analysis, we found it to be reliable. However, it does not offer any API to automate the malicious domain detection. Note that SiteReview appears to use the RIPE Network Coordination Center (NCC) [233] to categorize websites.⁵

⁵Bluecoat, the original developer of SiteReview (acquired by Symantec) is a member of RIPE NCC, see: <https://www.ripe.net/membership/indices/data/eu.blue-coat-systems.html>

We hosted a web page on a Microsoft Azure cloud domain that closely resembles content of a malicious site, and submitted the page to SiteReview which categorized the site as *Suspicious* within 24 hours. Our web page is not shared with anyone or have any backlinks that are used for search engine optimizations (SEO). During this time, we notice requests *only* from IP addresses assigned to RIPE NCC every hour. A *Chrome* user-agent is used by all these access requests to our page.

We use SiteReview to identify categories of 3880 cloaked and 3880 non-cloaked domains. The cloaked domains are identified using the content dissimilarities method in Section 7.6.3. Some of these cloaked and non-cloaked domains are flagged as malicious by SiteReview. 171 cloaked and 187 non-cloaked domains were unreachable during our tests. The number of cloaked malicious domains flagged by SiteReview (1636, 44.11%) is significantly higher compared to that of non-cloaked malicious domains (1022, 27.67%); see Table 26.

Category	Cloaked domains	Non-cloaked domains
Suspicious	1550 (41.79%)	920 (24.91%)
Malicious Sources/Malnets	56 (1.51%)	71 (1.92%)
Scam/Questionable Legality	11 (0.30%)	12 (0.32%)
Phishing	13 (0.35%)	9 (0.24%)
Spam	4 (0.11%)	4 (0.11%)
Potentially Unwanted Software	1 (0.03%)	3 (0.08%)
Malicious Outbound Data/Botnets	1 (0.03%)	3 (0.08%)
Total active domains	3709	3693

Table 26: SiteReview categorization of malicious squatting domains - cloaked vs. non-cloaked

We classify (**1024**) active cloaked domains (as of Oct. 15, 2019) using a semi-automated process with SiteReview to identify how many of them are malicious. During this process, we reclassify sites that SiteReview failed to classify or misclassified. This semi-automated process is used to determine the ground truth as described below.

- We found **413** sites serving content related to social engineering attacks (SEA); 383

suspicious sites with content that poses an elevated security or privacy risk; 23 malicious sites; 5 phishing sites; and 2 sites with potential unwanted programs. Some sites are classified into more than one of the mentioned categories.

- SiteReview was unable to classify 361 sites, labeled as “not yet rated (NYR)”. With manual inspection, we observed that some NYR sites show content similar to social engineering attack (SEA) sites. Therefore, for each of the NYR sites, we compute the SimHash [54] of the page source, and then compare the SimHash value with all SEA sites. We classify a NYR site as SEA, if the hamming distance between the SimHashes of the NYR and SEA sites is under 20, and the hamming distance is the lowest between the NYR site and any one of the SEA sites. For example, assume that the NYR site xyz shows similar content as sites in SEA categories A and B with hamming distances of 8 and 5, respectively; then we label xyz as of category B. With this approach, we could correctly classify **306** NYR sites as SEA (out of 361).
- SiteReview classified 250 sites into benign categories. With manual inspection, we found **102** false positives in this categorization (i.e., malicious sites classified as benign); 2 Chinese sites, 1 deceptive site flagged by Google Safe Browsing [133], 80 sites with iframes that include links to malicious targets, 17 sites with promotional contests (e.g., online casino), 1 shopping site and 1 site showing that the operating system (Windows 10) is infected.

From the above mentioned observations, we found a total of 821 malicious sites (413+306+102) in different social engineering categories from the 1024 cloaked sites. Therefore, the percentage of malicious sites from those that are cloaked is 80.2%. This value may change due to the dynamicity of the content rendered from these cloaked sites (i.e., some sites alternatively show benign and malicious content during successive requests and at different times). We emphasize that SiteReview is only used to validate our ground truth, and our methodology is not dependent on SiteReview.

We also apply the ground truth analysis to sites hosted on 1500 randomly selected squatting domains generated from DNSTwist (from list category A in Table 22) and found 74% (1110 of them are malicious. These squatting domains contain both cloaked and uncloaked sites.

7.6 Dissimilarities

Sites with content discrepancies between GooglebotUA and ChromeUA may be cloaked, assuming differences are due to evasion techniques adopted by adversaries. In this section, we delve into such differences using the domain list category A in Table 22.

7.6.1 Link dissimilarities

We evaluate the number of links in web pages that appear with ChromeUA, but not with GooglebotUA, and vice-versa. We found that 21,616 distinct links appeared in ChromeUA (1557 sites), compared to 10,355 links in GooglebotUA (1235 sites); i.e., ChromeUA observed over twice the number of links compared to GooglebotUA.

Dynamic pages rendered from both ChromeUA and GooglebotUA show listings of advertisements links. These links changed on successive refreshing of the page from the same client or with different clients (e.g., ChromeUA and GooglebotUA).

7.6.2 Header dissimilarities

We inspect the *title*, *description* and *keywords* header fields to find the sites where the header fields are different between GooglebotUA and ChromeUA.

Apart for the title header field, description/keywords fields in headers had significant discrepancies with GooglebotUA. Upon manual inspection, we observed that the dissimilarities in title & description header fields were benign as they mostly contained the domain

Header	# diff	# only with GooglebotUA	# only with ChromeUA
Title	2644	2190	3388
Description	3530	4839	1375
Keywords	265	716	408

Table 27: Header dissimilarities—the last two columns show the number of the specific header type that exists only from one user-agent (empty in the other)

name or content that relate to sale of the domain. According to Table 27, 716 sites had the keywords header field injected only with GooglebotUA (e.g., health, wellness, surgery) and its use may had an impact in improving the rank of those websites. Many keywords added to HTTP headers were sent to the crawler to perform semantic cloaking [313].

7.6.3 Content dissimilarities

We compare pages rendered between ChromeUA and GooglebotUA using syntactical and semantic heuristics as defined in Section 7.3.3. Sites that show benign content (e.g., website under construction) are excluded. While cloaking is prevalent in static pages, we also observed cloaking in pages with dynamic content. In the case of the latter, a significant number of sites showed cloaking behaviors at random when they were requested repeatedly.

Failure	# Content dissimilarity	# Image dissimilarity
HTTP 404 Not Found	398	0
HTTP 403 Forbidden	349	302
“Coming soon”	244	64
HTTP 500 Server Error	14	0

Table 28: Failures from GooglebotUA

With our automated process, we found 2183 (2.2%) sites with static content and 83 (0.08%) sites rendering dynamic content were cloaked by examining the page source/content using heuristics; see Table 30. Out of them, 1763 (1.8%) and 42 (0.04%) sites serving static and dynamic content were redirected to other URLs respectively. The top 5 target URLs where these sites were redirected (for both static and dynamic sites) were

plt2t.com (27), yourbigprofit1.com (24), www.bate.tv (10), yvxi.com (8) and www.netradioplayer.com (7). Out of these target domains, plt2t.com redirected to another website that showed “your computer was locked” scam message occasionally, with the aim of getting the victim to call a fake tech support number.

Protocol	Content type	With content dissimilarities		With image dissimilarities	
		Cloaked sites	Redirects	Cloaked sites	Redirects
HTTP	static	192	166	142	118
	dynamic	21	7	22	7
HTTPS	static	52	36	37	27
	dynamic	3	2	1	1

Table 29: Combo-squatting domains served via HTTP/HTTPS

Most cloaked sites (361) from the squatting domain list category *A* in Table 22 had a content length difference of 1-10 KB between ChromeUA and GooglebotUA, compared to 121 cloaked domains that had a content length difference greater than 10 KB. Although this implies that in most cloaked sites, the content length difference between ChromeUA and GooglebotUA is minimal, the difference in presented content may be significant due to the use of dynamic rendering technologies (e.g., AngularJS, Puppeteer).

Phishing sites often adopt HTTPS to give a false sense of security to the victim users (see e.g., [202]). In Table 29, we compare cloaked vs. non-cloaked sites served via HTTP and HTTPS (using combo-squatting domains, category *C* in Table 22); cloaking is less apparent in HTTPS sites, where majority of the certificates (55) are issued by the free certificate provider Let’s Encrypt.

We observed the following major content differences between ChromeUA and GooglebotUA:

- Out of 100,000 squatting domains in list category *A* of Table 22, 2337 sites appeared to be dynamic only from GooglebotUA, and 2183 from ChromeUA. No overlap in domains was observed between GooglebotUA and ChromeUA. We were unable to

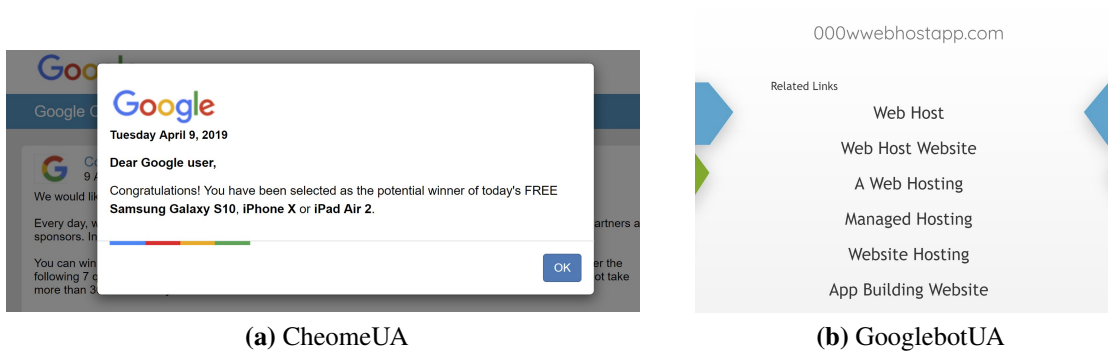


Figure 29: Cloaking differences for site: 000webhostapp

differentiate the content of these sites between GooglebotUA and ChromeUA, as when checked manually, the most probable topic of the page content as determined by Latent Dirichlet Allocation (LDA) algorithm [246] differed drastically on each request due to dynamic nature of the sites. Among these sites, there were also sites displaying dynamically populated links within iframe elements from ChromeUA, while such iframes appeared to be empty from GooglebotUA. These links related to various areas of businesses (e.g., Car Insurance, Credit Cards).

- The failures with content dissimilarities in Table 28 were observed from GooglebotUA, while with ChromeUA a different view of the content was displayed. For examples, the websites that showed “Coming soon” page content from GooglebotUA, showed the actual page content when requested from ChromeUA. Malicious sites also returned error codes when they detected the visitor was not a potential victim [202] (e.g., a search engine crawler).

Figures 29 to 31 are examples of instances where cloaking was used for phishing/malware purposes.

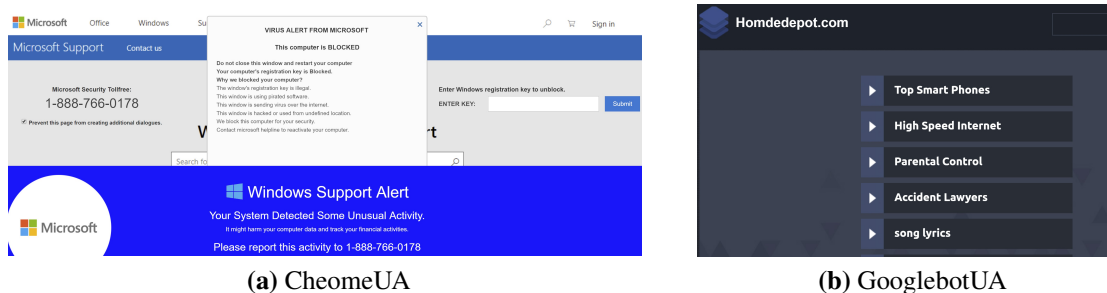


Figure 30: Cloaking differences for site: homdedepot.com

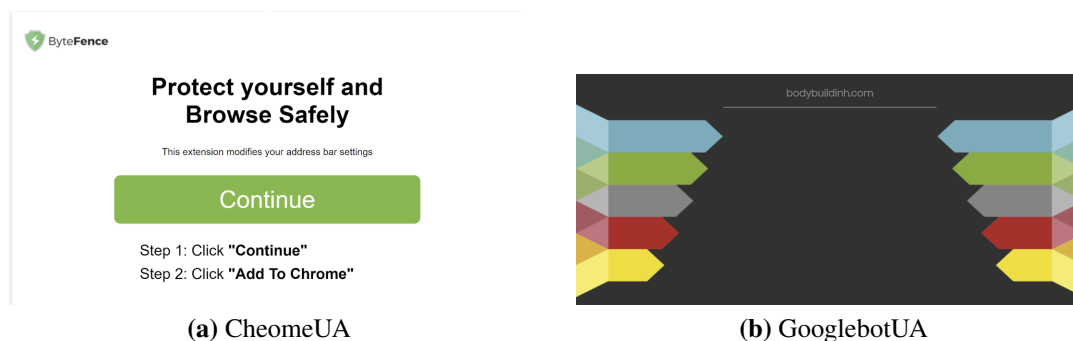


Figure 31: Cloaking differences for site: bodybuildinh.com

7.6.4 Image dissimilarities

We also determine cloaking by comparing the differences of screenshots of web pages between ChromeUA and GooglebotUA using image dissimilarity techniques. The number of sites with static content subjected to cloaking was 1710 (1.7%), while those with dynamic content was 784 (0.8%). We observed 960 (1%) and 490 (0.5%) of these sites with static and dynamic content, respectively, were redirected to other websites. In contrast to content dissimilarity method, with image dissimilarity, we found more cloaked sites that were also dynamic.

Page content alone is insufficient to detect cloaking due to technologies used in websites (e.g., Flash) that render dynamic content. Visual identity of a benign website can be shared by a malicious website with undetectable perturbations to humans, although their binary representations are completely distinct [211]. In addition, advertisements on web

pages can be more tailored to a specific client, and may be hidden from GooglebotUA. The failures as identified from image dissimilarity technique in Table 28 were only observed from GooglebotUA. Although with image dissimilarity technique, the detection of cloaking was better, the text extracted from screenshots using the Tesseract [274] OCR library was sometimes inaccurate. For example, Tesseract reads “Coming soon” as “Coming scan”. Despite our manual efforts to minimize the impact of these inaccuracies, the inaccuracies of Tesseract may have affected the accuracy of the results in Table 28.

Cloaking of domains served via HTTPS giving a false sense of security to users were a fraction when compared to those domains using HTTP; static content (37, 26%), dynamic content (1, 5%). There were 38 (0.04%) cloaked sites running on combo-squatting domains with valid TLS certificates as shown in Table 29.

7.6.5 Comparison of results of cloaking detection techniques

The dissimilarities techniques we use to identify cloaked sites focus on different structural elements of a web page. The results of content and image dissimilarities converge to some extent as they are applied on the syntactical and visual perspectives of the page content.

With link dissimilarities technique, we observed links are more prevalent with ChromeUA as opposed to GooglebotUA (for both static and dynamic content). The links shown in web pages hosted on domains were 209% and 140% for static and dynamic pages from ChromeUA compared to GooglebotUA. However, the links appeared in dynamic content were 6x and 9x when compared to static content with ChromeUA and GooglebotUA, respectively. This may mean that phishing/malware domains suppress links from GooglebotUA to avoid detection.

We also observed keywords in headers from GooglebotUA that were not seen from ChromeUA. These keywords that were exclusive to GooglebotUA may influence the search engine ranking algorithms for corresponding sites. With header similarities technique, the

keyword header fields related to specific categories of content appeared only with GooglebotUA. Therefore, these keyword header fields may possibly have been leveraged to manipulate the rankings of websites.

With content and image dissimilarities methods, we find cloaked websites from both static and dynamic websites with potential malicious content. With both content and image dissimilarity methods, we found a very small fraction (3880, 3.9%) of domains participate in cloaking. There were 880 cloaked sites that overlap between content and image dissimilarities. Out of 3880 sites 127 (3.3%) were flagged by VirusTotal. However, according to our ground truth (see Section 7.5), 80% of the cloaked sites were malicious. The low detection rate of malicious sites by VirusTotal highlights that blacklists are not effective in identifying a large proportion of social engineering sites. With image dissimilarities, a larger number of cloaked sites were found with dynamic content compared to content dissimilarities. Conversely, a large number of cloaked websites were identified using content dissimilarities with static content compared to image dissimilarities. Identifying dynamic content is more effective by analyzing the screenshots of web pages, as dynamic content may not be captured from the page source. Some of the cloaked sites that are dynamic, rendered different content on each refresh of the page. In some sites, benign and cloaked content were rendered alternatively when the page is refreshed multiple times. Since the dynamicity of sites depends on the time accessed, our results are a lower bound.

Manually inspecting 100 cloaked sites (from list category A in Table 22), we found 22 (22%) of them had differences in content. Few examples of differences in site content between ChromeUA and GooglebotUA are shown in Figures 29 (deceptive prize notice), 30 (technical support scam), 31 (prompting to install a malicious browser extension). The browser extension in Figure 31 (*ByteFence Secure Browsing*⁶) is a known malicious

⁶<https://botcrawl.com/bytedefence-secure-browsing/>

browser extension detected by reputable antivirus engines due to suspicious data collection habits and browser redirects. Most of these sites served content that changed between subsequent requests and at times alternated between malicious and non-malicious content.

7.7 Discussion

We discuss below observations from our analysis in Section 7.6.

7.7.1 Dynamicity in squatting sites

We found few squatting domains (644, 0.6%) showed dynamicity in rendered content that changed between two consecutive requests with ChromeUA. Since, dynamic sites can serve different content only after multiple requests or change between static/dynamic content alternatively, our results are a lower bound. Therefore, detection of dynamic sites with cloaked content is difficult compared to that of static sites. There were 83 cloaked sites identified using content dissimilarities in Section 7.6.3 out of the 644 dynamic sites. These cloaked dynamic sites changed between consecutive requests to show various forms of malicious content (e.g., technical support/lottery scams, malicious browser extensions).

7.7.2 Malicious squatting domains generated from DNSTwist

DNSTwist [85] uses fuzzy hashes,⁷ to identify malicious sites, by comparing the fuzzy hashes between web page content of a seed domain and the corresponding typo-squatting domain. For a 100% match, the typo-squatting web page content is similar to content hosted on the corresponding seed domain (includes situations where typo-squatting domain redirects to seed domain). When the comparison returns a match of 0, the web page

⁷ssdeep: <https://ssdeep-project.github.io/ssdeep/index.html>.

of the typo-squatting domain is most likely malicious. Out of 119,476 typo-squatting domains generated from DNSTwist, 76,178 (63.76%) returned a match of 0. We randomly selected 500 typo-squatting domains from list category *A* in Table 22, and found 187 malicious domains (37.4%) using SiteReview. Therefore, a significant proportion of DNSTwist generated typo-squatting domains are indeed malicious.

7.7.3 Relevance of seed domains

We find that the number of seed domains of cloaked squatting domains with a single permutation (345, 0.3%) is considerably high compared to those with multiple permutations. There were only 229 seed domains with 2-7 permutations of cloaked squatting domains. The 7 seed domains in Figure 32 generated 8-13 permutations of squatting domains. The categories of services offered by these seed domains include government (`service.gov.uk`), gaming (`epicgames.com`), search engine (`google.com.ph`), health (`health.com`) and news sites (`cnbc.com`). We also show the number of seed domains of the generated cloaked squatting domains as a comparison in Table 30. With both content and image dissimilarities, we find the proportion of squatting domains to seed domains is higher with static content (1.89%-2.18%) compared to that of dynamic content (1.08%-1.42%).

7.7.4 Detection of cloaked sites by blacklists

To study evasion of blacklists by cloaked squatting domains, we randomly selected 5000 squatting domains that are cloaked from domain list category *A* in Table 22, and ran them daily through VirusTotal between May 2, 2019 – June 5, 2019. At the end of this period (June 5, 2019), 92 (1.84%) were flagged by VirusTotal; phishing: 40, malicious: 41, malware: 22. Since our ground truth showed 80% of squatting domains were malicious (see Section 7.5), it appears that most phishing/malware squatting domains are not blacklisted.

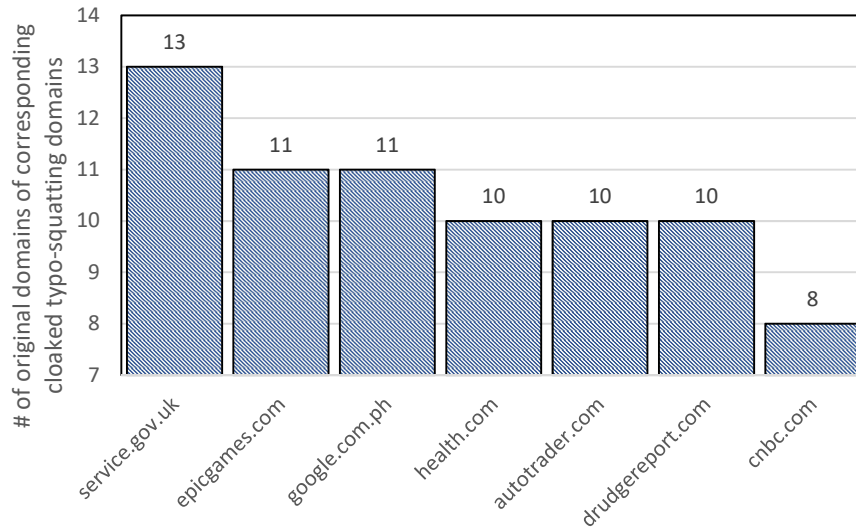


Figure 32: Top 7 seed domains of the corresponding cloaked domains with 8-13 permutations.

Type of domain	Nature of content	With content dissimilarities			With image dissimilarities		
		Cloaked sites	Redirects	Target URLs flagged by VirusTotal	Cloaked sites	Redirects	Target URLs flagged by VirusTotal
Squatting	static	2183	1763	27	1710	960	6
	dynamic	83	42	0	784	490	3
Seed	static	1153	1012	20	985	693	6
	dynamic	77	38	0	552	382	3

Table 30: Projecting results of cloaked squatting domains to corresponding seed domains (i.e., squatting domain vs. seed)

After approximately 3 months from the time of this experiment (on Aug. 26, 2019), we observed that URLs blacklisted by VirusTotal have not changed significantly (87, down from 92). Further, on Sep. 4, 2019, we applied our methodology described in Section 7.3 to 2268 cloaked domains previously identified, and found 1038 (45.78%) of them were still showing cloaked content. These cloaked domains may contain malicious content although they were not flagged by blacklists. The remaining domains (1230) were either recycled or showed exceptions described in Section 7.3.3. Therefore, it appears that the rate at which these cloaked sites were detected by blacklists is extremely slow.

Device	Content type	With content dissimilarities			With image dissimilarities		
		Cloaked sites	Redirects	Target URLs flagged by Virus-Total	Cloaked sites	Redirects	Target URLs flagged by Virus-Total
Desktop	static	607	498	7	484	289	3
	dynamic	20	9	0	230	135	1
Mobile	static	797	689	2	660	364	1
	dynamic	44	30	0	206	174	0

Table 31: Variation in cloaking between device types

Typo-squatting domains hosting malicious content may get recycled more frequently. This behavior may cause delays in blocking new websites or slow reactions to domain take-downs that host malicious content [217]. We found 2256 out of 100,000 squatting domains (cloaked and uncloaked) as malicious in Apr. 2019. However, 2048 of these domains remained active as of Nov. 15, 2019, and out of those domains, 67 of them were no longer flagged by VirusTotal. These websites showed benign content that is different from when it was previously flagged by VirusTotal.

7.7.5 Variations of cloaking in different device types

A significant proportion of web traffic comes from mobile devices and mobile users are more vulnerable to phishing attacks [202]. We identified cloaked websites using the heuris-

Type	Nature of content	With content dissimilarities		With image dissimilarities	
		Cloaked sites	Redirects	Cloaked sites	Redirects
Referrer	static	9	5	4	3
	dynamic	3	2	18	15
User-agent	static	99	80	59	36
	dynamic	4	1	46	31

Table 32: Variation between user-agent vs. referrer cloaking

tics defined in Section 7.3.3 for 25,000 sites (category *B* in Table 22) hosted on squatting

domains from both desktop and mobile browsers (Chrome); see Table 31. Cloaked sites with static content in mobile environment are more apparent compared to desktop environment. Similarly, redirections of sites hosted on squatting domains to target URLs are comparatively high in mobile environments. A significant number of cloaked sites overlap between desktop and mobile browsers as identified by content (326) and image (119) dissimilarity methods. The differences of the overlapping sites between desktop and mobile environments were mostly related to its layout. Tian et al. [282] found more phishing pages with mobile web browsers compared to desktop environment, and we observed a similar pattern for cloaked sites. The number of target URLs of redirections blacklisted by VirusTotal was low with mobile browsers compared to that of desktops. Oest et al. [202] observe mobile browsers (including Chrome) failed to show blacklist warnings between mid-2017 and late-2018. Although they claim that following their disclosure the protection level is comparable between mobile and desktop browsers, we noticed sites flagged by VirusTotal for mobile browsers were less than that of desktops.

7.7.6 User-agent vs. referrer cloaking

We compare websites identified as cloaked between user-agent and referrer cloaking. For both types of cloaking, we use the same sites in domain list category *D* in Table 22 that are hosted on typo-squatting/combo-squatting domains. As with our previous experiments, user-agent cloaking is measured between GooglebotUA and ChromeUA. For referrer cloaking, we use ChromeUA, but to mimic clicks initiated through search engine results, we set the referrer header to `http://www.google.com/`. As shown in Table 32, for sites with static content, cloaked sites identified from user-agent cloaking were 11x-16x higher than that of referrer cloaking (from both content and image dissimilarities methods in Section 7.3.3).

7.7.7 Relevance of type of squatting domains for cloaking

Most cloaked sites are hosted on combo-squatting domains as shown in Figure 33. This may mean that combo-squatting domains are more effective in cloaking phishing and malware site content. Panagiotis et al. [164] find most combo-squatting domains are not remediated for a long period of time (sometimes up to 1000 days). Therefore, many occurrences of abuse happen before they are detected by blacklists.

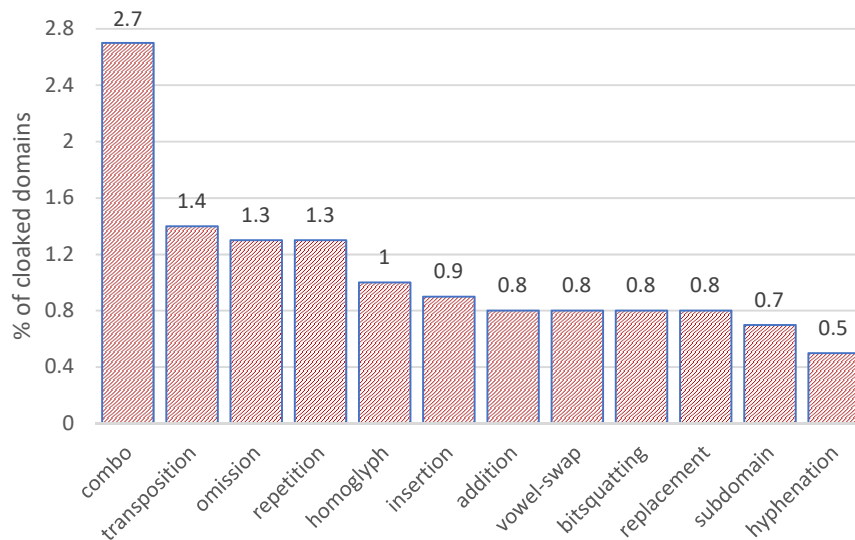


Figure 33: Cloaking by type of squatting domain.

7.7.8 Relevance of cloaking by other factors

The top 10 countries hosting the largest number of squatting domains with cloaked content were United States (1508), Germany (145), Netherlands (53), Australia (53), Seychelles (41), Canada (34), Switzerland (26), Japan (17), France (16) and British Virgin Islands (15). Therefore, most of these cloaked sites were hosted in the United States and Germany. Tian et al. [282] observed a similar pattern where most phishing sites are spread in these countries. The top 5 registrars of squatting domains hosting cloaked content were GoDaddy (477), Sea Wasp (225), Xinnet Technology Corporation (115), Tucows, Inc. (84), Enom, Inc. (82). GoDaddy had registered the most number of cloaked domains.

7.8 Recommendations

Majority of Internet traffic is originated from mobile users, and mobile browsers are prone to phishing attacks [203]. However, anti-phishing protection in mobile browsers trail behind that of desktop browsers. We observed cloaking of websites (with static content) that are potentially malicious in mobile browser (Chrome) is comparatively higher to desktop browser (Chrome). Bandwidth restrictions imposed by carriers in mobile devices is a barrier to desktop-level blacklist protection [203]. Therefore, at least over a Wi-Fi connection, the full blacklist should be checked by mobile browsers.

Since some major search engine crawlers are also owned by companies who develop browsers (e.g., Google, Microsoft), these companies can complement their existing detection techniques by comparing the views of a web page between a browser and crawler infrastructure, to tackle website cloaking. Some solutions in this aspect are already proposed in past studies [156]. Another countermeasure is to have domain registrars add extra checks in their fraud detection systems to detect domains that are permutations of popular trademarks having a higher entropy. This will facilitate registrars to request more information, if a domain registered is suspicious in carrying out malicious activities under the disguise of cloaking. A similar practice can be adopted by certificate authorities prior to issuing certificates for suspicious domains.

7.9 Summary

Cloaked malicious sites deliver phishing, malware and social engineering content to victimize users. We found 22% of cloaked domains show malicious content (technical support scams, lottery scams, malicious browser extensions, malicious links), with significant differences between ChromeUA and GooglebotUA. In addition, we also found cloaking

behaviors in a considerable number of squatting domains hosting dynamic content at irregular time intervals. This type of cloaking in dynamic sites is harder to detect, and may go unnoticed by the detection algorithms. Some squatting domains redirect a website through multiple intermediary domains to its final destination. [316]. We found 1.8% (1805 domains in list category A in Table 22) cloaked squatting domains engaged in redirections with content dissimilarities.

We found 716 cloaked domains included benign keywords (e.g., health, wellness, surgery) in request headers, from the perspective of GooglebotUA, possibly to influence the ranking of these sites. Also, empty iframes and error pages are observed from GooglebotUA, when the view from ChromeUA for the same was malicious. We also observed a larger number of malicious links from ChromeUA, while benign links were viewed for the same from GooglebotUA. In addition, a relatively larger number of cloaked sites with dynamic content were identified with Image dissimilarities method.

We used DNSTwist to generate typo-squatting domains. The domain generation algorithms used in DNSTwist are highly successful in generating malicious domains. According to SiteReview along with our heuristics, 74% of these typo-squatting domains were malicious. Although, some of these malicious domains are short-lived, the attackers may cause harm to users during the domain life time due to slow reaction to blocking such domains.

In past studies, URLs used for crawling mostly include crafted websites or those belonging to specific malicious categories (phishing, social engineering ad campaigns). In contrast, the squatting domains we used host potential malicious content mimicking a variety of popular sites. The URLs of cloaked malicious websites we found may eventually get flagged by various blacklisting entities (e.g., VirusTotal). We observed more squatting domains and dynamically generated links identified from iframe elements are getting flagged as phishing or malicious by VirusTotal over time. The cloaked sites blacklisted

by VirusTotal is a fraction (3.3%), which implies that a larger number of cloaked sites go undetected. Our ground truth showed that nearly 80% of the cloaked sites were malicious, which means nearly 77% of the malicious squatting domains were not detected by VirusTotal. Therefore, the undetected portion of cloaked malicious sites is significant. Our detection rate of cloaked malicious sites is significantly higher compared to past studies [202, 156]. Therefore, our heuristics can be used to compare the detection accuracy of cloaking of malicious sites that mimic popular sites, with that of other studies.

Cloaking delays and slows down blacklisting [202]. We found 46% of cloaked squatting domains with potential malicious content (from a sample of 2268 domains in list category *A* in Table 22), continue to cloak content even after 3 months, reaffirming that the techniques used by blacklisting entities are not effective for cloaked sites.

Chapter 8

Longitudinal study of the TLS ecosystems in networked devices

8.1 Introduction

Beyond user-level computing devices and back-end servers, there are many other Internet-connected devices that serve important roles in everyday IT operations. Such devices include routers, modems, printers, cameras, SCADA (supervisory control and data acquisition) controllers, DVR (digital video recorders), HVAC (heating, ventilating and air conditioning technology), CPS (cyber physical systems), and NAS (network-attached storage) devices. Several past studies have identified critical security issues in these devices, including authentication bypass, hard-coded passwords and keys, misconfiguration, serious flaws in their firmware and web interfaces; example studies include: [236, 73, 72, 67, 68, 208]. The massive DDoS attack on DynDNS as attributed to the Mirai botnet (e.g., [22]), populated by DVRs, IP cameras and other IoT devices, shows the clear danger of security flaws and weaknesses in these devices. Antonakakis et al. [22] argue that the absence of sound security practices in the IoT space leads to a fragile state of its environment impacted by vulnerabilities in devices. The Reaper [310] IoT botnet appears to be more severe than

Mirai, as Reaper is capable of exploiting numerous device vulnerabilities, as opposed to Mirai's rather simple albeit effective exploitations of default credentials; see also [272].

Over the years, manufacturers of networked devices have implemented some security mechanisms, notably, the adoption of SSL/TLS for communicating with other devices. With the help of the ZMap [91] high-speed IPv4 scanner, some recent projects analyzed the TLS ecosystem for web, email and SSH servers, and identified and measured significant security issues in TLS deployments in the wild; see e.g., [90, 140, 89, 6].

Heninger et al. [138] highlighted faulty random number generators in networked devices (see also the recent follow-up work [137]). Chung et al. [59] analyzed over 80 million invalid TLS certificates, and attribute most of them to network devices, including modems/home routers, VPNs, NAS, firewalls, IP cameras and IPTVs. In Oct. 2016, we studied the state of the TLS ecosystem for networked devices [240] and found many devices using cryptographic primitives that are phased out from modern browsers and web servers.

The types and number of devices available in Censys have increased since 2016, with significantly more devices supporting TLS (73.7%) compared to 2016 (29.4%). However, still some devices continue to support weak crypto primitives, while in few device types, the use of such primitives has increased. In this work, we evaluate the progress of securing the TLS ecosystem for devices by performing a similar measurement study in a more comprehensive form and compare the results with our previous study. We extracted certificates of devices and Alexa sites, and process the raw data following the same methodology as in our previous study. There are few new device types added to Censys since 2016. The number of Alexa sites is now restricted to Top-1M in Censys.

We analyze certificates and TLS parameters of 6,319,951 devices (out of 8,570,047), collected from Censys (<http://www.censys.io>) on May 6, 2018. Unsurprisingly, many devices still continue to use cryptographic primitives that are currently being phased

out from modern browsers and web servers. The state of the TLS ecosystem doesn't appear to have gained any significant progress. Specifically, we found a significant number of devices using unsafe RSA 512-bit keys (3760 certificates) and 768-bit keys (8338 certificates), slightly lower than our findings in Oct. 2016. The vulnerable/deprecated RC4 stream cipher is still widely used in devices (302,038). A large number of devices (167,900) also use (deprecated) SSLv3. No traces of SSLv2 are found in the snapshot taken in May 2018. We also compare TLS security parameters between devices and Alexa Top-1M sites, which clearly highlights the differences in these two domains. In all security aspects that we consider (SSL/TLS version, signature, encryption and hashing algorithms, and RSA key length), devices on average are more vulnerable than Alexa sites.

Similar to our previous study, we communicated our findings to top manufacturers of vulnerable devices. Interestingly, as in our previous study, Cisco appears to have the highest number of vulnerable devices. Furthermore, the information of devices (e.g., model/serial numbers) in Censys with weaker cipher suites is limited, inhibiting us from providing manufacturers concrete identifying information of these devices. We refrained from carrying out intrusive testing to find more specific information of these devices to avoid jeopardizing systems in production. Overall, we hope our results will serve as a catalyst to quick fixing of TLS issues in devices, so that these devices do not remain less secure than the HTTPS/web ecosystem in the long run.

Contributions.

- We carried out a measurement study to assess the vulnerabilities in devices based on their TLS certificates and protocol parameters. Our current study is more comprehensive (cf. [240], conducted in Oct. 2016) as new device types and more data relating to devices are added to Censys since 2016. Although the rate of adoption of TLS is remarkable for devices between 2016 and 2018, the use of weak primitives haven't reduced significantly. Ironically, the use of weak primitives has increased in

some devices and vice-versa with strong primitives.

- We find an increase of devices with ICS protocols (notably in S7 and Modbus) compared to a study performed by Mirian et al. [193] in 2016. These protocols were originally designed to operate within closed networks without explicit security measures. Although, Mirian et al. found a similar behavior as ours, we report the rate of increase of devices supporting these protocols (except for DNP3) is higher than what they observed in Mar. 2016.
- From our follow-ups with the leading manufacturers of vulnerable devices, apparently, security patches from vendors remain unadopted by many device owners. Beyond adopting secure updates in a timely manner, we also briefly discuss a few counter-measures to improve the security of these devices.

The remainder of this chapter is organized as follows. We discuss related work pertaining to TLS deployments in Section 8.2. We elaborate our methodology and the devices in focus for our study in Section 8.3. In Section 8.4, we provide the details of our analysis and results in terms of: the prevalence of weak security practices, and changes (between 2016 and 2018) in the use of weak and strong cryptographic primitives for devices; we also compare the overall results of devices with Alexa-1M HTTPS websites. In Section 8.5, we present our disclosure procedure and responses from manufacturers of devices with most weaknesses. We list limitations of our experiments and future improvements in Section 8.6. We suggest a few recommendations to improve the state of device security in Section 8.7, and finally, conclude in Section 8.8.

8.2 Related work

We briefly discuss measurement studies on real-world TLS deployments.

To allow researchers to analyze SSL certificates, the EFF SSL Observatory project [95] offered the first large-scale, open certificate repository containing SSL certificates for the IPv4 address space in 2010. Later, in 2013, Durumeric et al. [90] analyzed the ZMap collected data of web applications (HTTPS) over a period of 14 months to uncover all public certificate authorities (CAs) and the certificates they issued. Censys [88] is a search engine used to query information relating to hosts and networks stored in daily ZMap scans. As an example application for Censys, the prevalence of the unauthenticated Modbus protocol among SCADA systems has been studied. Numerous such systems have been found across the globe. However, non-SCADA devices, specifically, the TLS ecosystem for those devices have not been studied. We extend existing work to understand the TLS ecosystem for networked devices, mostly used at home, enterprise, and industrial environments, and physical/network infrastructures.

Heninger et al. [138] reported in 2012 that RSA/DSA algorithms as used specifically in embedded network devices are vulnerable due to faulty random number generators. They found that 0.75% of TLS certificates share keys, and RSA private keys can be easily calculated for 0.50% of TLS hosts (also reported similar results for RSA/DSA keys as used in the SSH protocol). However, other TLS/certificate parameters were not analyzed in this study.

Pa et al. [208] propose the IoT honeypot (IoTPOT) to analyze malware attacks against devices such as home routers, smart fridges, and other IoT devices. Their honeypot data also shows significant increase in Telnet-based attacks, including DDoS, against IoT devices. Costin et al. [67] devise a platform to find possible reuse of fingerprints of SSL certificates, public/private keys of devices in ZMap datasets; many devices were found with reused keys.

Industrial Control Systems (ICS) are becoming popular facilitating the remote and electronic control of physical equipment and sensors. Although these devices with no in-built security are originally designed to work in closed environments, in recent years they are

connected to build smart grids. Mirian et al. [193], studied the Internet-connected vulnerable devices, and found an increase of devices supporting BACnet, DNP3, Modbus, Fox and S7.

Shodan.io is a search engine similar to Censys, targeted towards IoT devices (full access requires paid subscriptions). In addition to IPv4 devices, Shodan claimed to have scanned millions of IPv6 addresses, reportedly by exploiting a loophole in the NTP Pool Project [24]. Arnaert et al. [23] highlight challenges in aggregating search results from Shodan and Censys, and propose an ontology to make them more usable and effective for finding vulnerable IoT devices.

There have several large-scale measurement studies of vulnerable IoT/CPS devices in the recent years, including potentially malicious scanning activities. Galluscio et al. [117] used an algorithm with data from the darknet to infer compromised unsolicited IoT devices. They found 11,000 such devices, most of which are embedded into active CPS infrastructures, and can be recruited into botnets. Leveraging a network telescope (consisting of unused, new IP ranges), Fachkha et al. [103] studied the probing of CPS devices supporting 20 common CPS protocols. They analyzed and correlated 50GB darknet data for this purpose (from one-month period), and extracted the probing events after an inferring process. They found more than 9000 such orchestrated events, attributed to unsolicited and malicious campaigns. After cross-matching these events with threat repositories, the authors found Modbus, ICCP, Niagara Fox and DNP3 are the top abused TCP CPS. Torabi et al. [286] performed a similar analysis to infer compromised IoT devices by finding those devices from the Shodan service, and identifying which of them are malicious using a threat repository/malware database. Xu et al. [314] carried out a comprehensive study of vulnerabilities in IP cameras available at <http://www.insecam.org>. In addition to cameras without password protection, the authors found open ports, network traffic rate, live video feeds streamed without owner's knowledge, and outdated/vulnerable software programs.

Note that, unlike these studies, we focus on the weaknesses specific to TLS deployment of networked devices.

Benson et al. [32] argue the fragility of the device ecosystem is attributed to unpatchable/insecure devices, insecure default passwords/misconfigurations, and the lack of suitable user interface, regulation, and cooperation between IoT manufacturers, network providers, content providers and end-users. The authors propose a *Security Monitor* to observe the aggregate view of network activity, as the low volume of attack traffic from an individual device is most likely undetectable. In addition, they propose a *Security Manager* to police the behavior of IoT devices at levels of different granularity (e.g., IP and service levels).

To improve the manual annotation process in Censys (the ZTag device tagging module), Feng et al. [107] develop an Acquisitional Rule-based Engine (ARE) capable of discovering and annotating devices automatically. ARE relies on application-layer responses from devices that run an Internet-accessible server, in conjunction with product information collected through web search. However, ARE will miss devices behind a NAT or the ones that cannot be queried from outside (e.g., no web server). Mi et al. [191] scan residential networks behind NAT to discover IP proxy machines including home IoT devices; access to residential machines is purchased from residential proxy providers such as Luminati¹ and Geosurf.² This approach is however ethically questionable at best (no consent from the device owners). Also, some proxy providers, such as Luminati disallows scanning the local network.

8.3 Methodology and device info

We rely on the Censys [88] search engine for our analysis. In this section, we provide a brief overview of Censys, and detail our methodology.

¹<https://luminati.io>

²<https://www.geosurf.com>

Censys³ enables querying data from the Internet-wide scan repository, a data repository hosting the periodic scan results as collected by the ZMap scanner [91]. Censys tags the collected data with security-related properties and device types, allowing easy but powerful search queries through its online search interface and REST API. Censys also tags TLS and certificate data of Alexa Top-1M web sites. Tagging is done by annotating the raw scan data with additional metadata, e.g., type and manufacturer for devices, and Alexa ranking for sites. The output from the application scanners is used to identify device-specific metadata. The annotation process involves ZTag (paired with ZMap and ZGrab), allowing researchers to add logic to define metadata for currently untagged devices [88]. Although Censys is now commercialized and a matured product, search capabilities in Censys are still improving (not all device metadata is defined in ZTag, although ZTag can be extended). Thus, TLS/certificate data and tag information for all device types are still not comprehensively reflected in Censys.

Table 33 lists available device types extracted from Censys, divided by their TLS support, for our datasets collected in Oct. 2016 and May 2018. Results discussed here refer to our May 2018 dataset, unless otherwise specified. We further group some device types from Censys for easier presentation as follows: modem (cable/DSL), printer (all printer models, print servers), network (generic network devices, network analyzers), SCADA (scada controller, router, gateway, server, frontend), media (set-top box, digital video recorders, VoIP, cinema), CPS (PLC, HVAC, IPMI, alarm system, environment monitor, fire alarm, industrial control system, water flow controller, light controller, power distribution unit, power monitor, power controller, solar panel). Certain device types (e.g., USB) appear to be small in numbers (9). This may be due to the fact that the tagging process in Censys is not very comprehensive. We do not consider devices that are very low in number or does not fall into our device categorizations (e.g., KVM, TV tuner, USB devices). The devices appear

³<http://www.scans.io>

to come from all around the world (78 countries with >1000 devices); the top 10 countries host about 84% of all devices compared to 56% reported in our 2016 study. Top-3 countries hosting these devices in 2018 are USA 43.5%, Mexico 15.8%, Spain 6.3% (in 2016: Germany 17.9%, USA 15.0%, India 4.9%).

Device type	Oct. 2016				May 2018			
	Non-TLS Count	Non-TLS %	TLS Count	TLS %	Non-TLS Count	Non-TLS %	TLS Count	TLS %
Infra. router	237,540	66.8	118,259	33.2	381,379	69.1	170,320	30.9
Modem	158,558	86	25,724	14	108,021	2.1	4,959,267	97.9
Camera	143,721	95.5	6809	4.5	116,691	92.2	9932	7.8
NAS	71,997	56.5	55,503	43.5	186,222	33.6	368,480	66.4
Home/office router	51,347	66.7	25,667	33.3	211,851	43.9	270,195	56.1
Network	3	0	39,857	100	1,053,091	79.9	265,715	20.1
Printer	10,148	31.3	22,296	68.7	153,147	76.7	46,463	23.3
Scada	24,909	86.8	3773	13.2	23,509	85.9	3860	14.1
CPS	12,820	93.7	868	6.3	11,423	12.3	81,572	87.7
Media	8000	87.9	1102	12.1	3647	2.5	142,293	97.5
Total	719,043	70.6	299,858	29.4	2,248,981	26.3	6,318,097	73.7

Table 33: Type-wise device distribution

For comparison, we chose the Alexa Top-1M sites. Data extracted from Censys is transformed to an intermediary format that requires a resource-intensive post-processing phase. Search queries can be executed on Censys in two ways: a RESTful web API or an SQL interface engine.⁴ We used the latter option, as it is more efficient for large-scale search results. After the TLS parameters and certificates are extracted for devices and Alexa-1M sites, we first analyze our selected security parameters and algorithms in devices. We then compare the security parameters from devices with those from Alexa-1M sites, to highlight any important differences between them. Similar to past work (e.g., [90, 172]), we choose the following certificate/TLS parameters: cipher suite (algorithms used for hashing, key encryption, key exchange and authentication, signature), SSL/TLS protocol version,

⁴Accessed via Google BigQuery interface: <https://bigquery.cloud.google.com>

and RSA key length.

8.4 Analysis and results

On May 6, 2018, we used Censys [88] to extract certificates and TLS parameters from 6,319,951 TLS-supporting devices (out of a total of 8,570,047 devices), and from 735,638 HTTPS sites in Alexa Top-1M. The number of total devices in Censys supporting TLS have increased by 21 fold since our last measurement study. Furthermore, new types of devices have been added to Censys, including: network (switch) and CPS (alarm system, environment monitor, fire alarm, IPMI, power controller, solar panel). We also noticed a new type of router: SOHO (Small Office / Home Office) appearing in the latest Censys snapshot, which we categorize as *home/office router*. Only *home routers* were found in our previous dataset. Home routers are normally used for personal use where users prefer accessing the Internet with wifi connections for ease of accessibility. In contrast SOHO routers are intended to support enterprise systems, mostly through wired Ethernet. The count of devices supporting TLS has increased significantly in May 2018 (6,318,097, 73.7%) compared to Oct. 2016 (299,858, 29.4%); the increase of modems is also extraordinary (i.e., from 25,724 to 4,959,267). In contrast, the percentages of some devices (infrastructure router, printer, network) supporting TLS have decreased from that in May 2016. This may be attributed to the variation of the proportion in which devices of different types are added to Censys. In this section, we provide the results of our analysis and compare the use of TLS/certificate parameters.

8.4.1 Prevalence of weak security practices

For each cryptographic primitive in a device certificate and TLS/SSL protocol banner, we compute the percentage to compare the parameters between devices; see Figures 34–38

for a comparison of the weak cryptographic primitives (for exact data, see Table 34). We also compare average values from devices with Alexa sites (the last two bars). For brevity, we first highlight results for algorithms and parameters that are most vulnerable. We also analyze certificate reuse in both devices and Alexa sites.

Hash functions in message authentication. Some devices still use MD5 although in small fractions. The use of MD5 in home/office routers (60,835, 22.5%) and CPS (14,665, 18%) devices are significant. In Alexa-1M sites, the MD5 usage is negligible as a percentage (1834, 0.2%) compared to our findings in 2016 (6588, 1.1%). Media (141,905, 99.7%) devices and infrastructure routers (152,601, 89.6%) mostly use SHA1; see Figure 34. MD5 is broken for more than a decade now [302]. SHA1 collision attacks are now feasible [264] (see also [265]; being phased out as of writing).

Hash functions in signature schemes. The MD5-RSA signature scheme is mostly used in printers (16,749, 36.1%), while SHA1-RSA is predominant in media (141,882, 99.7%), network (185,607, 69.9%) devices, infrastructure routers (152,601, 29.7%) and modems (3,699,856, 74.6%); see Figure 35. Devices using MD5-RSA are vulnerable to certificate collision attacks, where attackers create certificates that collide with arbitrary prefixes/suffixes [266]. Out of all the modems, the usage of SHA1-RSA is the highest in wireless modems (27,747, 75.2%). Some devices (164,847) use “unknown” algorithms; according to a Censys author (email correspondence), these algorithms are not parseable.

RSA key lengths. The use of factorable 512-bit RSA keys is a serious security issue, enabling efficient FREAK attacks (e.g., via [293]). These keys are mostly observed in infrastructure routers (3111, 1.9%), cameras (434, 4.4%) and Scada (22, 0.6%) devices. We also noticed 512-bit RSA keys in an industrial control system and two solar panels. The industrial control system with the factorable key appears to be located in Spain, and manufactured by *Opto22* [207]. Certificates with 1024-bit RSA keys are deemed to be insecure as of early 2016; see NIST SP 800-131A (at least 2048 bits should be used). However,

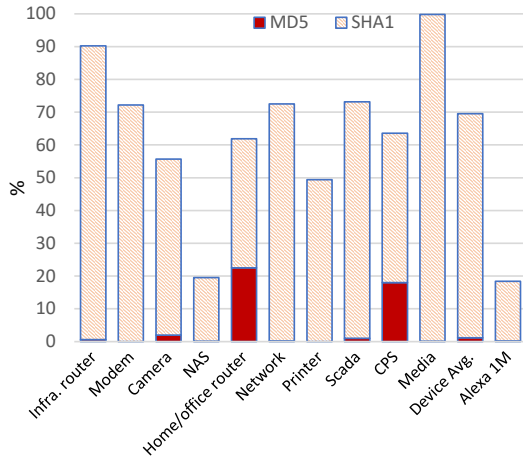


Figure 34: Hashing algorithms

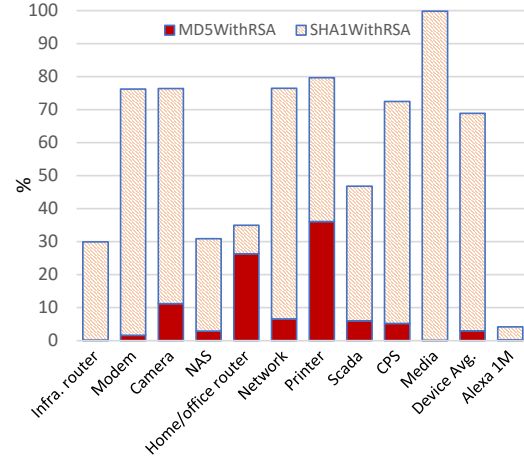


Figure 35: Signature algorithms

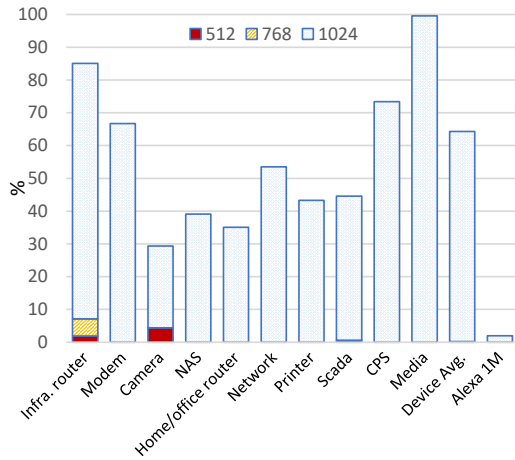


Figure 36: Key lengths (RSA)

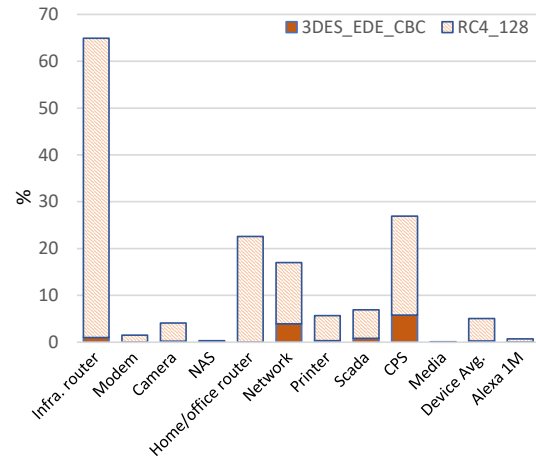


Figure 37: Encryption algorithms

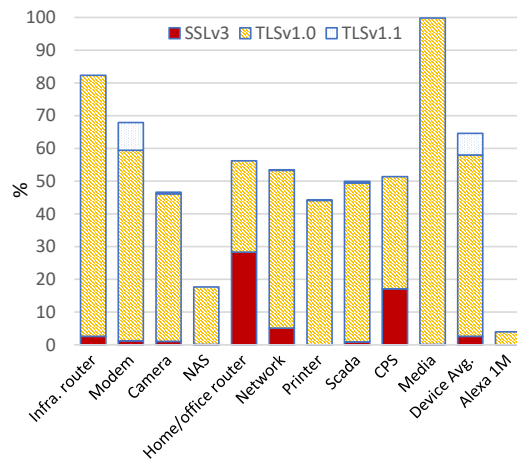


Figure 38: SSL/TLS protocol versions

	Hashing alg.		Signature alg.		RSA keylen			Enc. alg.			Protocol	
	MD5	SHA1	MD5-RSA	SHA1-RSA	512	768	1024	3DES	RC4	SSLv3	TLS 1.0	TLS 1.1
Infra. router	0.6	89.6	0.2	29.7	1.9	5.2	78	1	63.9	2.6	79.7	0
Modem	0	72.2	1.6	74.6	0	0	66.7	0	1.5	1.2	58.2	8.5
Camera	2	53.7	11.2	65.2	4.4	0	25	0.2	3.9	1.1	45	0.5
NAS	0.1	19.4	2.9	28	0	0	39.1	0	0.3	0.1	17.6	0
HO router	22.5	39.4	26.3	8.7	0	0	35.1	0	22.6	28.3	27.9	0
Network	0.2	72.3	6.6	69.9	0	0	53.5	3.9	13.1	5.1	48.2	0.2
Printer	0	49.4	36.1	43.6	0	0	43.3	0.3	5.4	0	44.1	0.1
Scada	1	72.2	6	40.8	0.6	0	44	0.8	6.1	0.9	48.5	0.5
CPS	18	45.6	5.2	67.3	0	0	73.4	5.8	21.1	17.1	34.3	0
Media	0.1	99.7	0.1	99.7	0	0	99.6	0	0.1	0.1	99.7	0
Device avg.	1.2	68.3	3	65.9	0.1	0.1	64.1	0.27	4.78	2.6	55.4	6.6
Alexa-IM	0.2	18.2	0.3	3.9	0	0	2	0.06	0.66	0	4	0

Table 34: Percentages of weak cryptographic primitives in devices (as of May 6, 2018); under Enc. alg., 3DES and RC4 represent 3DES-EDE-CBC and RC4-128, respectively. “HO router” in the first column is “home/office router”.

many devices still use 1024-bit keys (Figure 36); the use of 1024-bit keys is high in infrastructure routers (124,918, 78%) and media (141,771, 99.6%) devices. A few Alexa-1M sites (12,974, 2%) still use 1024-bit RSA keys in certificates.

Encryption algorithms. We check the use of vulnerable ciphers such as RC4 (see e.g., [118], RFC 7465), and 3DES (the Sweet32 attack [33]). Note that the ZGrab application scanner as used with ZMap includes RC4 as a supported cipher (in addition to ciphers included in the Chrome browser), to allow communication with older TLS servers. RC4 is mostly used in infrastructure routers (108,834, 63.9%), while its use is minimum in media (85, 0.1%) devices; see Figure 37. Alexa-1M sites still use RC4 at a smaller scale (4828, 0.66%). The use of 3DES cipher is limited except in CPS (4734, 5.8%) and network (10,392, 3.9%) devices. 3DES is more prevalent in firewalls (8412, 25.8%). The use of ChaCha20-Poly1305 (currently being standardized, RFC 7905) as a replacement of RC4 is still negligible in devices as an average (550, 0.09%) compared to Alexa-1M sites (15,225, 2.07%).

TLS/SSL version. SSLv3 usage (vulnerable to the POODLE attack [197]) is considerable in home/office routers (76,338, 28.3%) and CPS (13,928, 17.1%) devices. TLS 1.0 is vulnerable to the BEAST attack [87]. Media (141,861, 99.7%) and infrastructure routers (170,311, 79.7%) have a high use of TLS 1.0. However, in Alexa-1M sites (31, 4%), TLS 1.0 use is low. In our study in Oct. 2016, we found devices supporting SSLv2 (deprecated in 2011, see RFC 6176). Version rollback attacks downgrade SSLv3 to SSLv2 [10]. With the DROWN attack [27], an attacker can even break a strong RSA key, if the server shares the RSA key with an SSLv2 server. Most of these devices were of type NAS (5517) and network (2006). However, none of the current snapshots in ZMap or Censys appear to have devices using SSLv2.

Certificate issuers. Most device certificates are self-signed (68% and 71% in Oct. 2016 and May 2018, respectively), potentially making them vulnerable to man-in-the-middle (MITM) attacks. The remaining certificates are CA-signed; see Table 35 (total CAs: 1335

and 4923 in Oct. 2016 and May 2018, respectively). Some CA organizations are device manufacturers, others are browser trusted. Certificate data in Censys contains a flag indicating the browser trusted status (based on Mozilla NSS). According to the Top-10 issuer organizations data taken from 2016 and 2018 snapshots, a major change is the adoption of free certificates from *Let's Encrypt* (21,006; no certificates from traditional CAs in top 10). We could not find more details of the “Bitbug.net Network Services” certificate issuing organization. The *Issuer DN* field of certificates issued by “hw” contains email addresses from Huawei (e.g., HW@huawei.com). When contacted, Huawei confirmed the issuance of those certificates. Although “trendchip”⁵ was acquired by another company in 2010, certificates issued are still in use under its former name. Certificates of both trendchip and Bitbug.net are expired.

Oct. 2016				May 2018			
Issuer org.	Count	%	Trusted?	Issuer org.	Count	%	Trusted?
Western Digital	6846	0.67	×	Synology Inc.	143,336	2.27	×
Synology Inc.	6461	0.63	×	hw	138,154	2.19	×
ZyXEL	4220	0.41	×	Huawei	125,009	1.98	×
GoDaddy.com	1412	0.14	1213	trendchip	37,161	0.59	×
hw	1101	0.11	×	ZTE Corporation	30,841	0.49	×
TELMEX	1038	0.10	×	Let's Encrypt	22,815	0.36	21,006
TAIWAN-CA	818	0.08	818	LANCOM Systems	15,041	0.24	×
COMODO	811	0.08	630	Bitbug.net Network Services	11,376	0.18	×
StartCom Ltd.	628	0.06	399	SANGFOR	9986	0.16	×
GeoTrust Inc.	622	0.06	538	Cisco Systems	9543	0.15	×

Table 35: Top-10 organizations issuing device certificates (the “Trusted?” column represents browser trustworthiness)

Certificate reuse. Some devices often come with the same default certificate, which remains unchanged afterwards. We group certificates according to their SHA256 fingerprints for reuse detection.⁶ Many devices reuse certificates, out of which DSL and cable modems are the highest (4,763,389, 75.4%). These devices may be vulnerable to MITM attacks (cf. SSH attacks [53]). Certificates reuse in Alexa sites has reduced slightly (33% of certificates are reused in May 2018 vs. 38% in Oct. 2016, mostly due to CDN, similar to past

⁵<https://www.bloomberg.com/research/stocks/private/snapshot.asp?privcapId=28942714>

⁶Certificates with the same public key may differ in other fields, resulting in different fingerprints. We did not analyze public key reuse in certificates; the dataset we use from ZMap/Censys does not contain actual public key values.

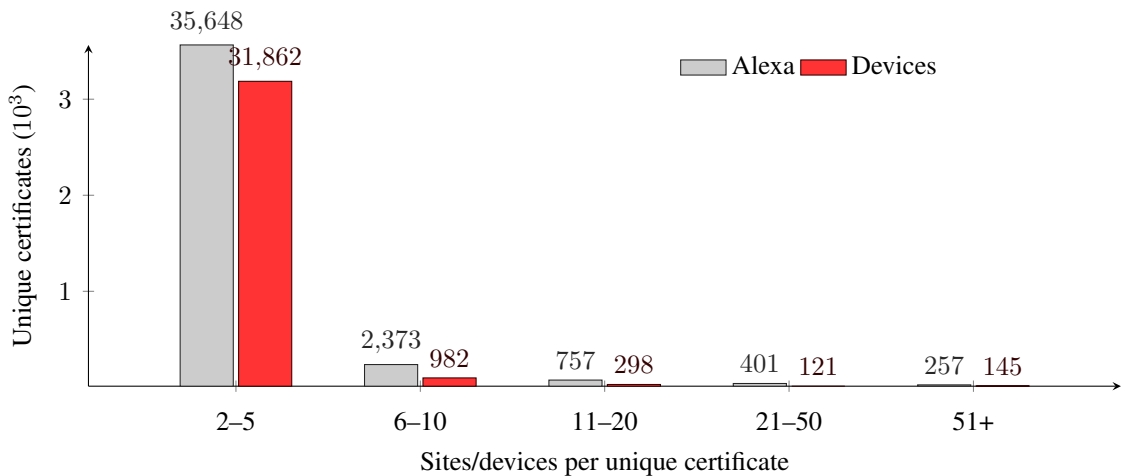


Figure 39: Unique certificates: Alexa-1M (total certs: 735,638) vs. devices (6,319,951) as of May 2018

studies, e.g., [299]); see Figure 39. Certificate reused by groups of 5+ Alexa sites/devices are relatively low.

The Common Name (CN) in most reused certificates contain non-routable IP addresses, e.g., 192.168.1.1 (274,824, 4.35%), generic identification labels, e.g., zxserver (138,135), BMS (1,345,520), or domain names, e.g., *.alarmesomfy.net (14,004).

DH prime number reuse. Many devices supporting Diffie-Hellman (DH) Key Exchange reuse prime numbers. Such reuse can be exploited via the Logjam attack, enabling a MITM attacker to downgrade connections to export grade Diffie-Hellman [6]. Alias et al. [15] reported that a timing side-channel attack is possible with DHKE used in an embedded system which can decrease the key search area, reducing the time to solve the Discrete Log Hard Problem (DLHP). Such an attack can lead to the extraction of private keys from devices. There are (308,139, 4.87%) reused primes in devices, including infrastructure routers (27,187, 0.43%), NAS (5479, 0.54%), modems (97,753, 1.55%), and network (63,443, 1%). In Censys, there are 735,638 Alexa domains supporting TLS, out of which only 3.6% (26,310) support DHKE reused prime numbers. In Oct. 2016, 0.2% of all Alexa sites reused DH prime numbers, while with Alexa-1M sites, the same reused percentage is

	Hashing alg.			Signature alg.		RSA keylen				Enc. alg.			Protocol	
	MD5	SHA1	SHA1-RSA	MD5-RSA	SHA1-RSA	512	768	1024	3DES	RC4	SSLv3	TLS 1.0	TLS 1.1	
Infra. router	0.4	-9.8	-10.6	-54.7	-18.8	-1.3	-2.2	-5.2	1	-17.6	-52.4	35.7	-0.2	
Modem	-0.4	38.8	-18.8	0.9	-6.1	-0.1	0	59.3	0	-18.4	1	25.8	8.5	
Camera	-10	-21.2	-6.1	-1.3	-19.7	3.3	-0.1	-26.5	-0.2	-17.7	-2.6	-21	-11.6	
NAS	-1	-6.1	-19.7	-7.4	-18.8	-0.2	0	5.2	-0.2	-2.3	-0.7	-4.2	-0.1	
HO router	22.3	-32.5	-18.8	26	-17.8	0	0	6.9	0	22.3	28.2	2.8	0	
Network	-0.1	-17.8	-25	4.8	-34.8	0	-0.4	-36.5	3.6	6.9	4.2	-36.2	0.1	
Printer	0	-34.8	30.2	-38.8	-11.8	0	0	-28.2	-0.6	-18.9	0	-34.2	0.2	
Scada	-2.7	-11.8	-14.6	-5.6	-13.7	-1.4	0	-5.1	-1.8	-8.4	-3.2	-19.1	-0.6	
CPS	17.7	0.9	-13.7	1.6	33.5	-0.7	0	60.5	-13.9	18.3	16.1	14.6	-19.4	
Media	-15.9	33.5	41.8	-12.2	-15.3	-0.3	-0.1	48.2	-0.5	-18.9	-15.8	40.1	-0.6	
Device avg.	-2.6	0.9	7.6	-15.3	7.6	-1	-1.9	16.2	-3.23	-14.5	-6.5	5.1	1.8	
Alexa-IM	-0.9	-13	-7.9	-0.2	-7.9	0	0	-3.8	-0.2	-1.1	0	-11.6	-0.1	

Table 36: Changes in weak cryptographic primitives in devices/Alexa-IM sites between Oct. 18, 2016 - May 6, 2018; under Enc. alg., 3DES and RC4 represent 3DES-EDE-CBC and RC4-128, respectively. Negative and positive percentages, show a reduction and increase of the respective weak TLS parameters, for the given Internet-connected devices/Alexa-IM sites (compared to 2016), respectively.

9.9% (50,292). Therefore, it appears that DHKE prime number reuse is significantly high in Alexa-1M sites compared to all of the Alexa sites.

8.4.2 Changes in the use of weak cryptographic primitives

New devices added to Censys consist of cryptographic primitives at varying proportions. These cryptographic primitives exhibit positive and negative fluctuations at the level of our device groupings, or when taken as an average. Table 36 shows changes in percentages of weak primitives. A negative value represents a reduction of the primitive compared to our previous study and vice-versa. Alexa-1M sites supporting HTTPS have increased in Censys (from 598,888 to 735,638) since 2016. The numbers for device average of SHA1 hashing algorithm (0.9%), SHA1-RSA signature algorithm (7.6%), RSA key lengths of 1024-bit (16.2%), TLS 1.0 (5.1%) and TLS 1.1 (1.8%) have increased.

It is important to note that even when the average of a device category for a weak primitive is reduced, it is still possible to observe an increase of the same primitive for a specific device in the same grouping. For example, the use of MD5 on average has reduced (-2.6%), but its use in home/office router (22.3%) and CPS (17.7%) devices has increased significantly. SHA1 use has increased in modems (38.8%) and media (33.5%) devices, while a sharp drop is noticed in home/office routers (-32.5%) and printers (-34.8%). MD5-RSA use has dropped in infrastructure routers (-54.7%) and printers (-38.8%). The 1024-bit RSA keys increased in modem (59.3%), CPS (60.5%) and media (48.2%) devices. SSLv3 usage has dropped in infrastructure routers (-52.4%). No change is observed for Alexa-1M sites (SSLv3 is not used).

Mirian et al. [193] found devices with ICS (Industrial Control Systems) protocols show vulnerabilities in equipments installed in plants. The number of vulnerable devices for specific ICS protocols (in Mar. 2016) and the percentage increase between Dec. 2015 – Mar. 2016 is shown in columns 2 and 4 of Table 37. All devices supporting ICS protocols

are tagged in Censys with the specific protocol name (e.g., BACnet, DNP3, Modbus, Fox, S7), and we use these tags to differentiate when counting devices supporting each protocol. We extracted the number of devices supporting the specific protocols from the May 2018 snapshot in Censys and calculated the percentage change from Oct. 2016. The number of devices using S7 (41.6%) and Modbus (24.9%) protocols have increased significantly. However, devices using DNP3 (1.2%) haven't increased much.

Protocol	Number of devices (Mar. 2016 [193])	Number of devices (May 2018)	Increase from Dec'15 to Mar'16 [193]	Increase from Mar'16 to May'18
BACnet	16,813	17,178	0.4%	2.1%
DNP3	429	434	2.3%	1.2%
Modbus	23,120	30,771	7.1%	24.9%
Fox	26,535	28,261	0.9%	6.1%
S7	2798	4791	18.7%	41.6%

Table 37: Changes in vulnerability – an increase in devices supporting vulnerable ICS protocols is apparent with time (specifically for Modbus and S7)

Reusable private keys. It appears that a substantial number of manufacturers include shared private keys into firmware of devices being sold [281]. These keys are mostly used to provide SSH and HTTPS access to devices. It is possible to extract these private keys after buying such devices or from a downloadable firmware. Censys tags these reused private keys, but it is not an exhaustive source to find all devices that are impacted. This is also because not all devices are persistently connected to the Internet. Viehböck et al. [296] published a list of fingerprints of devices with known private keys. Censys identifies these devices with private keys using a non-intrusive approach leveraging the fingerprints of certificates from its Internet-wide scans [249]. If a reused private key is exposed, a large number of devices may become vulnerable to impersonation, man-in-the-middle and passive decryption attacks [249]. Top-10 countries with devices including known private keys are shown in Table 38. Thailand (14.14%), United States (13.09%) and Brazil (10.06%)

are the top 3 countries that include known private keys in devices. According to a previous study [249] carried out in 2015, top 3 countries having devices with known private keys are United States (26.27%), Mexico(16.52%) and Brazil (8.10%). While the situation have improved in some countries, in some countries devices with known private keys have increased, e.g., United Kingdom (3.62%), Brazil (1.96%), Colombia (0.04%).

Country	Count	Percentage
Thailand	193,805	14.14%
United States	179,435	13.09%
Brazil	137,803	10.06%
Dominican Republic	132,787	9.69%
Mexico	86,825	6.34%
United Kingdom	80,610	5.88%
Colombia	60,291	4.4%
Spain	59,068	4.31%
Canada	35,254	2.57%
Tunisia	24,298	1.77%

Table 38: Top-10 countries with known private keys included in devices

We summarize the numbers and percentages of devices with reusable keys in Table 39. Modems, home/office routers, network and NAS devices appear to reuse a considerable number of these private keys. According to Table 40, Huawei, DrayTek and Multitech are manufacturing most of these devices. To mitigate this risk, vendors should consider assigning a random private key to each of the devices manufactured. On the other hand, users should change the default passwords and certificates (self-signed) pertaining to devices whenever possible as appropriate. However, this is not always a pragmatic approach due to lack of permissions, controls and knowledge to adopt such security measures by clients.

8.4.3 Changes in the use of strong cryptographic primitives

The use of strong cryptographic primitives appears to have reduced for certain devices between Oct. 2016 – May 2018; see Table 41. The SHA256 usage in modems (-38.4%),

Device grouping	Count	Percentage
Modem	535,530	6.2%
Home/office router	160,892	1.9%
Network	61,375	0.7%
NAS	45,632	0.5%
Camera	253	-
Infra. router	183	-
Media	121	-
Scada	90	-
Printer	85	-
CPS	11	-

Table 39: Devices groupings with a known private key as tagged in Censys

Manufacturer	Count	Percentage
Huawei	503,364	5.9%
DrayTek	151,049	1.8%
Multitech	73,173	0.9%
Ubiquiti Networks	30,030	0.4%
Telrad	27,747	0.3%
Seagate	27,617	0.3%
NetGear	10,809	0.1%
Linksys	9541	0.1%
Adtran	7379	0.1%
Allegro Software	6964	0.1%

Table 40: Top-10 manufactures of devices with a known private key as tagged in Censys.

CPS (-18.6%) and media (-17.7%) devices has dropped significantly. The use of SHA256-RSA and SHA512-RSA has significantly reduced in media (-29.7%) and Scada (-8.9%) devices, respectively. Although, the device average of SHA512-RSA has decreased slightly (-0.5%), no change is observed in Alexa-1M sites. Even though, the SHA256-ECDSA use in device grouping under consideration or device average has not reduced, the use of same signature algorithm has reduced slightly in Alexa-1M sites (-0.4%). The device average for 2048-bit (-13.3%) and 4096-bit (-0.3%) RSA keys has reduced, but the corresponding change in Alexa-1M is an increase (12%, 3.1%). The device average for AES-128-CBC (-2.8%) has reduced, but the stronger AES-256-CBC (17.55%) and AES-128-GCM (1.21%)

primitive use have increased. In contrast, in Alexa-1M sites, only the use of AES-128-CBC (-1.8%) and AES-256-CBC (-9.39%) have reduced. The device average of TLS 1.2 protocol is slightly reduced (-2.8%) as opposed to the considerable increase of the same in Alexa-1M sites (11.6%). Also, TLS 1.2 use in modems (-35.1%) has reduced while it is the opposite for cameras (35.2%).

Overall, apart from encryption algorithms, there is an increase in weak TLS primitives with the growth of devices supporting TLS. It is likely that the legacy devices accumulated over time may not get proper attention to have their firmware upgraded to latest versions to eliminate possible vulnerabilities (due to e.g., lack of oversight [290]).

8.5 Disclosure

The vulnerable devices we found in our study are manufactured by hundreds of different companies. The Top-5 manufactures of vulnerable devices are show in Table 42. We have contacted the ones with many vulnerable devices, where we could locate contact email addresses of vulnerability management support teams of these manufacturing companies from the web, explaining our findings. We have got responses from Cisco, DrayTek, Synology, Huawei and Ubiquiti Networks. According to Cisco, they allow users to import certificates of their choice, who may be using certificates with weak ciphers due to lack of awareness. As is in our previous study, Cisco appears to be the top manufacturer with vulnerable devices. Interestingly, the devices manufactured by Somfy Systems have the same number (13,897) of ciphersuites with vulnerable MD5, RC4, SSLv3 and RSA1024 cryptographic primitives. All these devices appear to be using the same TLS_RSA_WITH_RC4_128_MD5 cipher suite for negotiation during the SSL/TLS handshake.

As we found, Vigor routers produced by DrayTek are vulnerable. DreyTek informed us that the vulnerable devices are of older units where the owners haven't updated their

	Hash alg.		Signature alg.		RSA Keylen		Encryption alg.			Protocol
	SHA256	SHA256-RSA	SHA512-RSA	SHA256-ECDSA	2048	4096	AES-128-CBC	AES-256-CBC	AES-128-GCM	
Infra. router	9.4	-1.2	18.6	1.6	8.8	0.3	1.9	5.4	9.3	16.9
Modem	-38.4	18.1	-0.1	0	-58.4	-0.4	-4.7	61.5	-38.5	-35.1
Camera	31.2	15.4	0.1	0	32.1	-0.1	-2	-11.3	31.2	35.2
NAS	7.1	27.2	-0.1	0	-4.5	-0.3	-2.8	-1.8	7	5
HO router	10.2	-7.1	0	0	-6.8	-0.1	26	-58.5	10.2	-31
Network	17.9	12.4	0.1	7.9	36.6	-0.1	11	-39.4	17.8	31.84
Printer	34.8	8.6	0	0	28.2	0	0.2	-15.5	34.8	38.3
Scada	14.63	19	0.7	0.8	6.9	0.3	-6.7	2.3	14.5	22.9
CPS	-18.6	21	-8.9	0	-59.3	-0.2	-4.5	18.6	-18.6	-11.3
Media	-17.7	-29.7	0	0	-46.8	-0.1	-14.3	51.4	-17.7	-23.7
Device avg.	1.3	4.1	-0.5	0.3	-13.3	-0.3	-2.8	17.55	1.21	-2.8
Alexa-IM	13.9	8.4	0	-0.4	12	3.1	-1.8	-9.39	11.9	11.6

Table 41: Changes in strong cryptographic primitives in devices/Alexa-IM sites between Oct. 18, 2016 – May 6, 2018. Negative and positive percentages, show a reduction and increase of the respective strong TLS parameters, for the given Internet-connected devices/Alexa-IM sites (compared to 2016), respectively.

firmware. Some of these devices support the weak SSLv3 protocol. According to DrayTek: “SSLv3 is, of course, deprecated and users should use TLS1.2 which is supported by all of our current and most recent products”. Unfortunately, companies of larger scale will take more time to improve security of devices with their prevailing change management practices, where the focus on stability takes precedence over security. They claim most of their users update the units, but it is challenging to acquire 100% success due to lack of adherence by users in turning off older protocols. In May 2018, more than 800,000 DrayTek routers were found to be exploitable by a DNS reprogramming attack [273], which can eventually hijack web traffic to reveal personal information.

Huawei claims that they deny access to WAN ports by default, but some users appear to have customized their devices by opening the WAN ports, allowing possible external attacks. They plan to communicate with their customers and have the SSH/HTTPS ports of WAN devices closed, to reduce the risk of devices with known private keys. Dell claims that the reported devices appear to run very old firmware, not properly configured or already out of support. With the latest firmware, they only use TLSv1.0, TLSv1.1 or TLSv1.2 protocols, SHA256 hashing algorithm, longer key lengths (2048 bits), and no RC4 ciphers.

Synology informed us that users may be using outdated settings to host the services provided by their product(s). They were very appreciative of our efforts and plans to publish techniques in enhancing the security of their Data Security Manager (DSM) with different settings to address the problem. Ubiquiti Networks informs us that their airMAX devices used static SSL/TLS certificates until the end of 2015, at which point they fixed the problem by generating a self-signed certificate on the first boot. It appears that users are still

Manufacturer	MD5	RC4	SSLv3	<RSA1024	Device types
Cisco	1340	126,125	50,268	176,478	Infrastructure router, camera, switch, network, SOHO router, firewall, SCADA controller
DrayTek	60,775	60,877	7293	70,801	SOHO router, camera, infra. router
Synology	242	445	211	81,035	DVR, camera, SOHO router, NAS
Somfy Systems	13,897	13,897	13,897	13,897	Alarm system
Dell	760	2541	22	28,592	IPMI, laser printer

Table 42: Top-5 manufactures with vulnerable devices (in May 2018)

using Ubiquiti devices with old firmware.

8.6 Limitations

Certain statistics as extracted from Censys appear to be unusual. For example, there is only one infrastructure router from certain manufacturers, e.g., Apple, DrayTek and Huawei. We communicated such observations to a Censys author, who attributed them to be possible limitations of the current Censys logic, or device misconfiguration. Data in Censys can be queried using the Google BigQuery SQL interface. This interface allows querying data using standard SQL and facilitates downloading results in CSV and JSON formats that are easy to parse and machine process. However, Google BigQuery is not free after one year of use.

According to a Censys author, it is possible that some devices provide conflicting information on different ports, likely due to port forwarding from specific devices to device types that are tagged incorrectly. This appears to be a known issue due to fingerprinting devices at protocol-level rather than at host-level. Censys plans to work on a more advanced fingerprinting technique to address this problem in the future.

Although Censys allows users to search and analyze all types of connected devices via Google BigQuery, Censys do not have information of devices that cannot be reached via

ZMap (e.g., private/non-routable/firewalled addresses, opt-out from ZMap scanning). Furthermore, ZMap do not scan devices in their blacklist [320] or those network prefixes that fall outside in its whitelist. Therefore, to evaluate the completeness of results, correlation with alternative sources may be considered [254, 107]. Newer IoT devices are increasingly adopting IPv6 [76], which also cannot be measured by the IPv4-based ZMap scanner.

Censys requires manual effort in defining annotation rules to tag device meta-data (e.g., type, manufacturer), which is not ideal in discovering new devices at large scale. Therefore, more collective effort is also needed to improve device tagging/annotating in Censys [88].

We found thousands of vulnerable devices from many manufacturers, and contacted the top-10 of those with most vulnerable devices via email (using appropriate addresses as found in their websites). This is a manual process and is not scalable. Stock et al. [267] explore several forms of scalable/automated communication channels (e.g., email, domain WHOIS information, phone, social media) for more effective vulnerability notification.

8.7 Recommendations

Based on our analysis, we suggest a few possible way-out from the current status quo in device security. Note that these recommendations are preliminary, listed here to stimulate future work in solving TLS security issues in non-computer devices.

1. The obvious one would to enable automatic security updates to devices, instead of relying on pro-active user actions. However, for certain devices (especially the ones possibly maintained by professional administrators), care must be taken to avoid unplanned downtimes of production systems. For this purpose, vendors should perform thorough testing before releasing patches to its users [26]. In Mar. 2008, a nuclear plant was accidentally rebooted following a software upgrade [49, 96] causing an unnecessary alarm of a drop of cooling. We strongly suggest that updating should

be used as the last resort for fixing a security issue; it is far better to avoid possible security issues in the design than fixing them on-the-go. Also, updates will almost never reach to 100% of all devices. Better understanding the consequences of attacks and designing new attack detection/resilient algorithms to prevent them at the inception is vital [49, 96]. As CPS employ autonomous and real time decision making algorithms, the authors suggest to have automatic recovery built-in during the design phase.

2. As many devices may not be reachable, or not readily update-able due to operational constraints, unlike desktop/mobile/server computers, we recommend to adopt strong security measures from the beginning, including, the use of latest TLS versions, most secure cipher-suites (given the computational capabilities of a device). We argue against gradual/step-wise increase of security levels (e.g., from RSA-512 to RSA-1024) for devices, as they are difficult to update and may remain operational for years. ICS devices originally developed to operate on isolated environments decades ago, still continue to operate, which are now connected to the public Internet allowing more exposure to possible vulnerabilities [193].
3. Avoid all known pitfalls in TLS security [262, 149], e.g., the use of fixed private keys, vulnerable or soon-to-be obsolete ciphers (e.g., RC4 and RSA-1024) [152], and self-signed certificates (can be easily avoided by using free certificates from Let's Encrypt).
4. Although manufacturers may block access to remote management interfaces of devices over SSH/HTTPS, users may still customize to allow remote access to devices. Therefore, it is also prudent for ISPs to ensure remote access to customer-provided equipment (CPE) is disallowed [249].
5. Allowing insecure device settings (e.g., fixed private key), or protocols (as in many

ICS devices) with the assumption that these devices would remain only in isolated networks must be avoided. Traditionally isolated devices are often being connected to the Internet, e.g., for remote management. Failure to address the vulnerabilities of interconnected devices in smart grids will hinder modernization of such systems [175].

6. Consider system hardening to tighten system security by shutting down unnecessary applications and ports [69].

8.8 Summary

As apparent from several studies on the real-world deployment of web servers (e.g., [172, 90]), TLS can provide tangible security benefit, only when it is configured properly. Partly due to several recent high-profile measurement studies (e.g., [89, 6]), TLS security for user-facing servers is improving. However, we found many networked devices are still using weaker/broken crypto primitives in TLS, compared to Alexa sites. Based on our measurement studies carried out in Oct. 2016 and May 2018, although the number of devices supporting TLS has sharply increased, still a large number of devices supporting weaker cryptographic primitives remain vulnerable. Some manufacturers (e.g., Lenovo, Seagate) appear to have produced a larger number devices with RC4, MD5, SSLv3 and key lengths of 1024-bit (RSA) and below. We also found a considerable number of known private keys in devices, which make them vulnerable. This is more apparent in modems (6.2%) and home/office routers (1.9%). Upon reaching out to them, we were told that the primary reason for the status quo is the inaction of users in applying latest firmware upgrades. However, the reality is such that no action is taken by most manufacturers to mitigate the vulnerabilities of devices where their users are not proactive in applying security patches. Blaming users who haven't updated their devices with security patches, which

may sometimes happen due to lack of knowledge, will not solve the issue.

Note that some vulnerabilities may have no effect if the services are accessed within a local network (e.g., inside a private home network), or via a modern browser—e.g., no current browser would accept the RC4 cipher or SSLv2, even if offered by a server. As these devices are varied (unlike regular web servers), actual exploitation of their weaknesses will depend on how they are used/accessed. These seemingly obsolete attack vectors can also be revived in the presence of a vulnerable TLS proxy between a modern browser and the vulnerable server, such as an anti-virus proxy [77].

We hope our findings to raise awareness of this issue and positively influence the manufacturers to push appropriate firmware upgrades (possibly with auto-updates).

Chapter 9

Conclusion and future work

Since the creation of the web, it had since evolved, and transitioned from static content, to serve more richer and dynamic content. To enhance the user experience and the commercialization of the web (e.g., to use the web as an e-commerce platform), new technologies (e.g., cookies) were introduced, that opened opportunities for advanced tracking mechanisms. Web tracking is rooted to the commercialization of the web, that has evolved with time. Although, it is argued that tracking browsing behaviours of users is necessary, to learn for providing a better user experience [109], tracking can also expose sensitive information of users. As various tracking techniques evolved, past studies have focused on the detection [157], measurement [97, 4] and provision of counter-measures [37] against tracking (e.g., privacy enhancing browser extensions).

We observed prevalence of tracking residential users of popular websites varies across the globe. The variation of trackers on first-party sites between countries was significant; countries that enjoy a greater freedom of expression and information flow show a stronger presence of trackers; Google dominates in tracking; countries with highest and lowest tracking prominence, were UK and Ethiopia, respectively.

Our primary focus in this thesis relates to privacy measurements of online services,

including that of essential services (e.g., websites and Android apps of governments, hospitals and religions). In contrast to commercial online services, users of these essential online services interact with information that are deeply personal and sensitive in nature. Therefore, these users do not expect to have their personal information exposed to third parties (including advertisers and trackers), as otherwise they may be subjected to adverse consequences (e.g., discrimination, social stigma, physical harm). Besides, users of these essential online services do not have the luxury of moving to an alternate provider to evade from tracking. In contrary to the expectations of users, we observed that the analyzed essential online services are tracked by commercial trackers; commercial trackers from market leaders (e.g., Google, Facebook) dominates in tracking on online services. Commercial trackers that are on the analyzed essential online services include both stateful (e.g., third party scripts and cookies) and stateless (e.g., fingerprinting) forms of tracking. In addition, trackers also employ other advanced forms of tracking techniques (e.g., session replay). The information collected by these various forms of tracking include sensitive information of users. Trackers can co-relate information collected from both commercial and essential online services to better profile users. Besides, security issues introduced due to the use of malicious third party libraries, and vulnerabilities (e.g., as a result of using poor coding standards) in online services, can lead to privacy issues (e.g., exposure of sensitive information).

Furthermore, 80% of sites that mimic popular sites, that evade from search engine crawlers, are malicious (including phishing websites). This results in legitimate users fall prey to these cloaked sites, who intend to use corresponding popular websites. We identified dissimilarities in these malicious websites (i.e., in request headers, links on the web page, response content, image of the webpage), between a legitimate browser client (i.e., Chrome) and Google search engine crawler, that can be used as heuristics to identify the presence of cloaking in malicious websites.

We also observed that the vulnerabilities resulting from the use of weak TLS certificates in Internet-connected devices (that support running online services) are significant compared to that of Top-1M popular websites. These vulnerabilities can eventually cause privacy exposures of user data, that are used to interact with online services. We also noticed an improvement in the adoption of TLS in the TLS ecosystem for devices from 2016 (29%) to 2018 (74%).

Governments in various jurisdictions have enacted privacy regulations to set guidelines for the collection and processing of personal information by providers of third party resources of online services — the EU General Data Protection Regulation (GDPR) [100], California Consumer Privacy Act (CCPA) [263], Virginia Consumer Data Protection Act (CDPA) [297], Personal Data Protection Guidelines for Africa [153], Canadian Personal Information Protection and Electronic Documents Act (PIPEDA) [135]. However, our work along with past studies have found exposure of personal information of users from online services, despite the availability of provisions in specific privacy regulations to protect users from such exposures of PII [78, 212, 199]. In addition, although tracking giants may claim that they are taking proactive steps for moving away from tracking, they are instead finding alternate methods of tracking — e.g., although Google announced a plan to block tracking cookies (in 2021) from Chrome browser, they are instead replacing third party trackers with a mechanism known as *Topics* [278], which does not fully address privacy concerns [39]; with *Facebook pixel*, first-party cookies are set on domains of outgoing URLs (e.g., from Facebook to advertisement URL) by attaching tags (FBCLID), to track users without the use of third-party cookies [30].

Our research points to a direction where more work is required to identify the ever evolving techniques on tracking/information disclosure, minimize the gap between tracking and corresponding mitigation strategies, and close scrutiny of the compliance of privacy

regulations. As such, possible extensions to our work include but not limited to the following.

Extensions to work on tracking. Tracking is not limited to a particular desktop or a device, and can cross the boundary between multiple devices [319]. Effects of geolocation in cross device tracking could be an interesting future direction. As Luminati does not proxy some Google domains, future tracking studies done from a global perspective, should consider similar alternative residential proxy services (if available). Currently, OpenWPM supports only the Firefox browser. Instead of user-agent manipulation to simulate different browsers for privacy measurements, use of real browsers may provide a more comprehensive view (but may require significant engineering effort). For example, we could then easily compare tracking prevalence between the Tor and other browsers. Future work may also use the OpenWPM WebExtension tool [311] for privacy measurements in a cross-browser environment using the WebExtension API (supported by all common browsers).

Improvements to tracking block lists. As with other past studies [97], we relied on EasyList/EasyPrivacy [92] filtering rules to identify advertisers and trackers from third party domains included on websites. However, these filtering rules do not cover all regional trackers in different countries. In addition, advertising/tracking domains are being removed from these filtering rules due to requests to comply with *Digital Millennium Copyright Act* (DMCA) [5]. Therefore, filtering rules are not a comprehensive technique to identify all possible trackers on websites. Also, filtering rules do not cover all forms of tracking (e.g., fingerprinting). Therefore, future work should focus on improving filtering rules to be more comprehensive or find an alternative to serve its purpose.

Explore novel forms of tracking. Web technology has evolved to offer immersive browsing experiences using *WebVR* (or *WebXR*). Unlike in traditional web sites where advertisements are typically sandboxed (e.g., using iframes), with *WebVR*, there is no practical

mechanism to sandbox ad-serving JavaScripts [171]. In addition, one of the biggest concerns of augmented reality is privacy [161]. This is because, augmented reality technology can sense user's actions, and collects information of a user, that may be shared with third parties; tracking data with *WebVR* may also include data of highly personal nature (i.e. bio-metric data such as iris/retina scans, face geometry, voiceprints). Therefore, extending privacy measurements to include *WebVR* is an interesting future direction.

Broader look into session replay. While entering textual information to input fields in web pages, session replay services may receive large payloads containing keywords of search queries [162] (based on how session replay services are configured by website admins). Users may type-in information that are highly personal, and are of various aspects (e.g., religious preferences, medical conditions, racial identity) in search input boxes, that can be used by third party session replaying services to better profile users. Future studies should have a broad look into information exposure from session replay services and other similar techniques, in privacy measurement studies.

Wider look into web cloaking from a global perspective. Cloaking techniques used in malicious sites may differ based on the geolocation [202] of the user or the language of web content. Also, cloaking behaviors may be different for various search engine crawlers (e.g., Bingbot, Yahoo, Baidu, Yandex) and browsers (e.g., Edge, Internet Explorer, Firefox). We leave such studies as future work.

Bibliography

- [1] 2-viruses.com. FlyTrap – Android apps that steal Facebook accounts, 2021. <https://www.2-viruses.com/flytrap-android-apps-stealing-facebook-accounts>.
- [2] G. Acar, S. Englehardt, and A. Narayanan. No boundaries: data exfiltration by third parties embedded on web pages. *Proceedings on Privacy Enhancing Technologies*, 2020(4):220–238, 2020.
- [3] G. Acar, C. Eubank, S. Englehardt, M. Juarez, A. Narayanan, and C. Diaz. The web never forgets: Persistent tracking mechanisms in the wild. In *ACM CCS'14*, Scottsdale, Arizona, USA, Nov. 2014.
- [4] G. Acar, M. Juarez, N. Nikiforakis, C. Diaz, S. Gürses, F. Piessens, and B. Preneel. Fpdetective: dusting the web for fingerprints. In *ACM Conference on Computer and Communications Security (CCS'13)*, Berlin, Germany, Nov. 2013.
- [5] Adguard. Ad blocking is under attack, 2017. <https://adguard.com/en/blog/ad-blocking-is-under-attack.html>.
- [6] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. VanderSloot, E. Wustrow, S. Zanella-Béguelink, and P. Zimmermann. Imperfect forward secrecy: How Diffie-Hellman fails in practice. In *ACM CCS'15*, Denver, CO, USA, Oct. 2015.
- [7] Aha.org. A high-level guide for hospital and health system senior leaders, 2022. <https://www.aha.org/center/cybersecurity-and-risk-advisory-services/importance-cybersecurity-protecting-patient-safety>.
- [8] Akamai. Internet connection speeds and adoption rates by geography, 2017. <https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/state-of-the-internet-connectivity-visualization.jsp>.
- [9] A. Alabduljabbar, R. Ma, S. Choi, R. Jang, S. Chen, and D. Mohaisen. Understanding the security of free content websites by analyzing their ssl certificates: A

- comparative study. In *Workshop on Cybersecurity and Social Sciences*, pages 19–25, 2022.
- [10] E. S. Alashwali and K. Rasmussen. What’s in a downgrade? A taxonomy of downgrade attacks in the TLS protocol and application protocols using TLS. In *I8*, Singapore, Aug. 2018.
- [11] E. S. Alashwali, P. Szalachowski, and A. Martin. Exploring HTTPS security inconsistencies: A cross-regional perspective. *Computers & Security*, 97, 2020. Article number 101975.
- [12] Alexa. The top 500 sites on the web, 2022. <https://www.alexa.com/topsites/countries>.
- [13] Alexa.com. Alexa top sites, 2021. <https://aws.amazon.com/alexa-top-sites/>.
- [14] A. A. Ali and M. Z. Murah. Security assessment of Libyan government websites. In *Cyber Resilience Conference (CRC’18)*, Putrajaya, Malaysia, Nov. 2018.
- [15] Y. F. B. Alias, M. A. M. Isa, and H. Hashim. Sieving technique to solve the discrete log hard problem in Diffie-Hellman key exchange. In *I5*, Langkawi, Malaysia, Apr. 2015.
- [16] Alibaba. The best ecommerce advertising strategies for 2021, 2021. <https://seller.alibaba.com/businessblogs/px001sb26-the-best-ecommerce-advertising-strategies-for-2021>.
- [17] O. Alrawi, C. Zuo, R. Duan, R. P. Kasturi, Z. Lin, and B. Saltaformaggio. The betrayal at cloud city: An empirical analysis of cloud-based mobile backends. In *USENIX Security Symposium’19*, Santa Clara, CA, USA, Aug. 2019.
- [18] Amnesty International. Cuba’s internet paradox: How controlled and censored internet risks cuba’s achievements in education, 2017. <https://www.amnesty.org/en/latest/news/2017/08/cubas-internet-paradox-how-controlled-and-censored-internet-risks-cubas-achievements-in-education/>.
- [19] Amplitude.com. Session replay: What it is & how to Use it effectively, 2022. <https://amplitude.com/blog/session-replay>.
- [20] Analytics Help. Understanding PII in Google’s contracts and policies, 2019. <https://support.google.com/analytics/answer/7686480?hl=en>.
- [21] Android. Permissions on Android, 2021. <https://developer.android.com/guide/topics/permissions/overview>.

- [22] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou. Understanding the Mirai botnet. In *USENIX Security Symposium'17*, Vancouver, BC, Canada, Aug. 2017.
- [23] M. Arnaert, Y. Bertrand, and K. Boudaoud. Modeling vulnerable Internet of Things on SHODAN and CENSYS: An ontology for cyber security. In *SECUREWARE'16*, Nice, France, July 2016.
- [24] ArsTechnica.com. Using IPv6 with Linux? you've likely been visited by Shodan and other scanners, 2016. <http://arstechnica.com/security/2016/02/using-ipv6-with-linux-youve-likely-been-visited-by-shodan-and-other-scanners/>.
- [25] S. Aryan, H. Aryan, and A. J. Halderman. Internet censorship in Iran: A first look. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI'13)*, Washington DC, USA, Aug. 2013.
- [26] Australian Government, Department of Defense. Assessing security vulnerabilities and applying patches, 2018. https://www.asd.gov.au/publications/protect/assessing_security_vulnerabilities_and_patches.htm.
- [27] N. Aviram, S. Schinzel, J. Somorovsky, N. Heninger, M. Dankel, J. Steube, L. Valenta, D. Adrian, J. Halderman, V. Dukhovni, and E. K'asper. DROWN: Breaking TLS Using SSLv2. In *USENIX Security Symposium'16*, Vancouver, BC, Canada, Aug. 2016.
- [28] M. A. Bashir, S. Arshad, W. K. Robertson, and C. Wilson. Tracing information flows between ad exchanges using retargeted ads. In *USENIX Security Symposium'16*, Austin, TX, USA, Aug. 2016.
- [29] BBC. Tracking tools found on EU government and health websites, 2019. <https://www.bbc.com/news/technology-47624206>.
- [30] P. Bekos, P. Papadopoulos, E. P. Markatos, and N. Kourtellis. The hitchhiker's guide to facebook web tracking with invisible pixels and click ids. *arXiv preprint arXiv:2208.00710*, 2022.
- [31] Bell Canada. Online advertising program, 2019. <https://www.bell.ca/online-marketing>.
- [32] T. Benson and B. Chandrasekaran. Sounding the bell for improving Internet (of things) security. In *17*, Dallas, TX, USA, Nov. 2017.
- [33] K. Bhargavan and G. Leurent. On the practical (in-)security of 64-bit block ciphers: Collision attacks on HTTP over TLS and OpenVPN. In *ACM CCS'16*, Oct. 2016.

- [34] R. Binns, U. Lyngs, M. Van Kleek, J. Zhao, T. Libert, and N. Shadbolt. Third party tracking in the mobile ecosystem. In *ACM WebSci'18*, Amsterdam, Netherlands, May 2018.
- [35] Blacklight. Blacklight, 2020. <https://themarkup.org/blacklight>.
- [36] Blog.mozilla.org. Cross-site tracking: Let's unpack that, 2018. <https://blog.mozilla.org/products/firefox/cross-site-tracking-lets-unpack-that/>.
- [37] W. Boumans and I. E. Poll. Web tracking and current countermeasures. 2017.
- [38] S. Braghin, A. Coen-Porisini, P. Colombo, S. Sicari, and A. Trombetta. Introducing privacy in a hospital information system. In *Software engineering for secure systems (SESS'08)*, Leipzig, Germany, May 2008.
- [39] Brave. Google's Topics API: Rebranding FLoC Without addressing key privacy issues, 2022. <https://brave.com/web-standards-at-brave/7-goo-gles-topics-api/>.
- [40] businesswire. 62% of enterprises exposed to sensitive data loss via Firebase vulnerability, 2018. <https://www.businesswire.com/news/home/20180619005540/en/62-of-Enterprises-Exposed-to-Sensitive-Data-Loss-via-Firebase-Vulnerability>.
- [41] BusinessWire.com. 62% of enterprises exposed to sensitive data loss via Firebase vulnerability, 2018. <https://www.businesswire.com/news/home/20180619005540/en/62-of-Enterprises-Exposed-to-Sensitive-Data-Loss-via-Firebase-Vulnerability>.
- [42] BuzzFeed News. Nothing sacred: These apps reserve the right to sell your prayers, 2022. <https://tinyurl.com/3z6jz7wh>.
- [43] Cali Dog Security. Introducing certstream, 2017. <https://medium.com/cali-dog-security/introducing-certstream-3fc13bb98067>.
- [44] H. Campbell. Introduction: The rise of the study of digital religion. *Digital religion*, pages 1–22, 2013.
- [45] H. A. Campbell. Religion and the internet: A microcosm for studying internet trends and implications. *New Media & Society*, 15(5):680–694, 2013.
- [46] H. A. Campbell, B. Altenhofen, W. Bellar, and K. J. Cho. There's a religious app for that! A framework for studying religious mobile applications. *Mobile Media & Communication*, 2(2):154–172, 2014.

- [47] Canadian medical association journal. Open access the commercialization of patient data in Canada: Ethics, privacy and policy, 2022. <https://www.cmaj.ca/content/194/3/E95>.
- [48] Canadian Radio-television and Telecommunications Commission (CRTC). The CRTC collaborates with international partners to fight illegitimate online marketing activities, 2018. <https://www.canada.ca/en/radio-television-telecommunications/news/2018/03/the-crtc-collaborates-with-international-partners-to-fight-illegitimate-online-marketing-activities.html>.
- [49] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry. Challenges for securing cyber physical systems. In *09*, Newark, NJ, USA, July 2009.
- [50] CDN Planet. Content Delivery Networks per country, 2021. <https://www.cdnplanet.com/geo/>.
- [51] Censys. Censys, 2022. <https://censys.io/>.
- [52] Centre for International governance innovation. What you need to know about internet censorship in iran, 2018. <https://www.cigionline.org/articles/what-you-need-know-about-internet-censorship-iran/>.
- [53] CERT. Vulnerability note 566724, 2015. <https://www.kb.cert.org/vuls/id/566724>.
- [54] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *ACM Symposium on Theory of Computing (STOC'02)*, Montreal, QC, Canada, May 2002.
- [55] K. L. Chiew, K. S. C. Yong, and C. L. Tan. A survey of phishing attacks: Their types, vectors and technical approaches. *Expert Systems with Applications*, 106:1–20, 2018.
- [56] H. Cho, D. Ippolito, and Y. W. Yu. Contact tracing mobile apps for COVID-19: Privacy considerations and related trade-offs. *arXiv preprint arXiv:2003.11511*, 2020.
- [57] Chrome DevTools. Chrome devtools, 2022. <https://developer.chrome.com/docs/devtools/>.
- [58] T. Chung, D. Choffnes, and A. Mislove. Tunneling for transparency: A large-scale analysis of end-to-end violations in the Internet. In *IMC'16*, Santa Monica, CA, USA, Nov. 2016.
- [59] T. Chung, Y. Liu, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. Measuring and applying invalid SSL certificates: The silent majority. In *IMC'16*, Santa Monica, CA, USA, Nov. 2016.

- [60] T. Chung, R. van Rijswijk-Deij, B. Chandrasekaran, D. Choffnes, D. Levin, B. M. Maggs, A. Mislove, and C. Wilson. A longitudinal, end-to-end view of the DNSSEC ecosystem. In *USENIX Security Symposium'17*, Vancouver, British Columbia, Canada, Aug. 2017.
- [61] Cisco. Cisco Umbrella 1 Million, 2020. <https://umbrella.cisco.com/blog/cisco-umbrella-1-million>.
- [62] Citizen Lab. URL testing lists intended for discovering website censorship, 2019. <https://github.com/citizenlab/test-lists/>.
- [63] Clym. How The CCPA affects the cookie policy, 2021. <https://www.clym.io/how-the-ccpa-affects-the-cookie-policy/>.
- [64] CNET. Religious apps with sinful permissions requests are more common than you think, 2019. <https://www.cnet.com/tech/services-and-software/why-so-many-android-christian-apps-have-unholy-privacy-policies/>.
- [65] CNET. These watchdogs track secret online censorship across the globe, 2019. <https://www.cnet.com/tech/services-and-software/features/the-watchdogs-tracking-secret-online-censorship-across-the-globe-ooni/>.
- [66] Cookiebot. Ad tech surveillance on the public sector web, 2019. <https://www.cookiebot.com/media/1121/cookiebot-report-2019-medium-size.pdf>.
- [67] A. Costin, J. Zaddach, A. Francillon, and D. Balzarotti. A large-scale analysis of the security of embedded firmwares. In *USENIX Security Symposium'14*, Aug. 2014.
- [68] A. Costin, A. Zarras, and A. Francillon. Automated dynamic firmware analysis at scale: A case study on embedded web interfaces. In *ASIACCS'16*, 2016.
- [69] A. Creery and E. J. Byres. Industrial cybersecurity for power system and SCADA networks. In *05*, Denver, CO, USA, Sept. 2005.
- [70] B. Csontos and I. Heckl. Accessibility, usability, and security evaluation of Hungarian government websites. *Universal Access in the Information Society*, 20(1):139–156, 2021.
- [71] Csoonline.com. Researchers notice massive increase in malicious jquery libraries, 2014. <https://www.csoonline.com/article/2136992/researchers-notice-massive-increase-in-malicious-jquery-libraries.html>.
- [72] A. Cui, M. Costello, and S. J. Stolfo. When firmware modifications attack: A case study of embedded exploitation. In *NDSS'13*, San Diego, CA, USA, Feb. 2013.

- [73] A. Cui and S. J. Stolfo. A quantitative analysis of the insecurity of embedded network devices: Results of a wide-area scan. In *ACSAC'10*, Austin, TX, USA, Dec. 2010.
- [74] Cybermetrics lab. Ranking web of hospitals, 2015. <https://hospitals.webometrics.info/>.
- [75] Cyble. Android trojan malware disguised as Syrian e-Gov Android app, 2021. <https://blog.cyble.com/2021/05/27/android-trojan-malware-disguised-as-syrian-e-gov-android-app/>.
- [76] J. Czyz, M. Luckie, M. Allman, and M. Bailey. Don't forget to lock the back door! A characterization of IPv6 network security policy. In *NDSS'16*, San Diego, CA, USA, Feb. 2016.
- [77] X. de Carnavalet and M. Mannan. Killed by proxy: Analyzing client-end TLS interception software. In *NDSS'16*, San Diego, CA, USA, Feb. 2016.
- [78] M. Degeling, C. Utz, C. Lentzsch, H. Hosseini, F. Schaub, and T. Holz. We value your privacy... Now take some cookies: Measuring the GDPR's impact on web privacy. In *NDSS'19*, San Diego, CA, USA, Feb. 2019.
- [79] Developer.mozilla.org. Referer, 2022. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referer>.
- [80] Developer.mozilla.org. Using HTTP cookies, 2022. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>.
- [81] die.net. nslookup, 2010. <https://linux.die.net/man/1/nslookup>.
- [82] die.net. curl, 2021. <https://linux.die.net/man/1/curl>.
- [83] Digital.gov. GSA govt-urls, 2021. <https://github.com/GSA/govt-urls>.
- [84] DLA piper. Data protection laws of the world, 2019. <https://www.dlapiperdataprotection.com/>.
- [85] DNSTwist. Dnstwist, 2020. <https://github.com/elceef/dnstwist>.
- [86] Domains by Proxy. Your identity is nobody's business but ours, 2019. <https://www.domainsbyproxy.com/>.
- [87] T. Duong and J. Rizzo. Here come the \oplus ninjas, 2011. <http://www.hpcc.ecs.soton.ac.uk/~dan/talks/bullrun/Beast.pdf>.
- [88] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. Halderman. A search engine backed by Internet-wide scanning. In *ACM Conference on Computer and Communications Security (CCS'15)*, Denver, CO, USA, Oct. 2015.

- [89] Z. Durumeric, J. Kasten, D. Adrian, J. A. Halderman, M. Bailey, F. Li, N. Weaver, J. Amann, J. Beekman, M. Payer, and V. Paxson. The matter of Heartbleed. In *IMC'14*, Vancouver, BC, Canada, Nov. 2014.
- [90] Z. Durumeric, J. Kasten, and M. Bailey. Analysis of the HTTPS certificate ecosystem. In *IMC'13*, Barcelona, Spain, Oct. 2013.
- [91] Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast internet-wide scanning and its security applications. In *USENIX Security Symposium'13*, Washington, D.C., USA, Aug. 2013.
- [92] EasyList. EasyList, 2020. <https://easylist.to/>.
- [93] EasyList. Other supplementary filter lists and easylist variants, 2021. <https://easylist.to/pages/other-supplementary-filter-lists-and-easylist-variants.html>.
- [94] P. Eckersley. How unique is your web browser? In *Privacy Enhancing Technologies Symposium (PETS'10)*, Berlin, Germany, July 2010.
- [95] Electronic Frontier Foundation. The EFF SSL observatory, 2010. <https://www.eff.org/observatory>.
- [96] J. Eloff and M. B. Bella. Software failures: An overview. In *Software Failure Investigation*, pages 7–24. Springer, 2018.
- [97] S. Englehardt and A. Narayanan. Online tracking: A 1-million-site measurement and analysis. In *ACM Conference on Computer and Communications Security (CCS'16)*, Vienna, Austria, Oct. 2016.
- [98] S. Englehardt, D. Reisman, C. Eubank, P. Zimmerman, J. Mayer, A. Narayanan, and E. W. Felten. Cookies that give you away: The surveillance implications of web tracking. In *TheWebConf'15*, Florence, Italy, May 2015.
- [99] K. R. Eschenfelder, J. C. Beachboard, C. R. McClure, and S. K. Wyman. Assessing US federal government websites. *Government Information Quarterly*, 14(2):173–189, 1997.
- [100] Europa.eu. EU GDPR, 2016. <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN>.
- [101] European Commission. How is data on my religious beliefs/sexual orientation/health/political views protected, 2022. <https://tinyurl.com/5cj2fmpt>.
- [102] Evgen Verzun. How tech giants benefit from data collection?, 2020. <https://evgenverzun.com/how-tech-giants-benefit-from-data-collection/>.

- [103] C. Fachkha, E. Bou-Harb, A. Keliris, N. Memon, and M. Ahamad. Internet-scale probing of CPS: Inference, characterization and orchestration analysis. In *NDSS'17*, San Diego, CA, USA, Feb. 2017.
- [104] M. Falahrastegar, H. Haddadi, S. Uhlig, and R. Mortier. The rise of panopticons: Examining region-specific third-party web tracking. In *Traffic Monitoring and Analysis (TMA'14)*, London UK, Apr. 2014.
- [105] A. P. Felt, R. Barnes, A. King, C. Palmer, C. Bentzel, and P. Tabriz. Measuring HTTPS adoption on the web. In *USENIX Security Symposium (USENIX Security'17)*, Vancouver, BC, Canada, Aug. 2017.
- [106] E. Felten and J. Mayer. How the NSA piggy-backs on third-party trackers, 2013. http://www.slate.com/blogs/future_tense/2013/12/13/nsa_surveillance_and_third_party_trackers_how_cookies_help_government_spies.html.
- [107] X. Feng, Q. Li, H. Wang, and L. Sun. Acquisitional rule-based engine for discovering Internet-of-Things devices. In *USENIX Security Symposium'18*, Baltimore, MD, USA, July 2018.
- [108] Forbes. God is not the only one watching over your church's website, 2014. <https://www.forbes.com/sites/adamtanner/2014/07/28/god-may-not-be-the-only-one-watching-over-your-churchs-website/?sh=4f32c91576b6>.
- [109] Forbes. Four tips for using website visitor tracking effectively, 2020. <https://www.forbes.com/sites/forbesbusinesscouncil/2020/05/28/four-tips-for-using-website-visitor-tracking-effectively/?sh=9d19c6b56c23>.
- [110] Foundation.mozilla.org. Pray.com, 2022. <https://tinyurl.com/2p8v5bep>.
- [111] Freedom House. Freedom of the press 2017, 2017. https://freedomhouse.org/sites/default/files/FOTP_2017_booklet_FINAL_April28.pdf.
- [112] Freedom House. Freedom on the net 2017 - Egypt, 2017. <https://freedomhouse.org/report/freedom-net/2017/egypt>.
- [113] N. Fruchter, H. Miao, S. Stevenson, and R. Balebako. Variations in tracking in relation to geographic location. In *Web 2.0 Security and Privacy (W2SP'15)*, San Jose, CA, USA, May 2015.
- [114] FullStory. How does FullStory recording work to recreate my users' experience?, 2021. <https://help.fullstory.com/hc/en-us/articles/>

360032975773-How-does-FullStory-recording-work-to-recreate-my-users-experience-.

- [115] G. Acar. Script URL substrings used to detect the embeddings from the companies offering session replay services, 2017. https://gist.github.com/gunesacar/0c67b94ad415841cf3be6761714147ca?permalink_comment_id=2271390.
- [116] G. Anzinger. Worldwide governments on the WWW, 2002. <http://www.gksoft.com/govt/en/world.html>.
- [117] M. Galluscio, N. Neshenko, E. Bou-Harb, Y. Huang, N. Ghaniy, J. Crichignoz, and G. Kaddoumx. A first empirical look on Internet-scale exploitations of IoT devices. In *17*, Montreal, QC, Canada, Oct. 2017.
- [118] C. Garman, K. G. Paterson, and T. Van der Merwe. Attacks only get better: Password recovery attacks against RC4 in TLS. In *USENIX Security Symposium'15*, Washington, DC, USA, Aug. 2015.
- [119] A. Gervais, A. Filios, V. Lenders, and S. Capkun. Quantifying web adblocker privacy. In *European Symposium on Research in Computer Security'17*, Oslo, Norway, Sept. 2017.
- [120] M. Gharaibeh, A. Shah, B. Huffaker, H. Zhang, R. Ensafi, and C. Papadopoulos. A look at router geolocation in public and commercial databases. In *ACM Internet measurement conference (IMC'17)*, London, United Kingdom, Nov. 2017.
- [121] Ghost Proxies. The difference between residential and datacenter proxies. Blog article (2019). <http://ghostproxies.com/blog/2016/06/residential-datacenter/>.
- [122] A. Gómez-Boix, P. Laperdrix, and B. Baudry. Hiding in the crowd: An analysis of the effectiveness of browser fingerprinting at large scale. In *TheWebConf'18*, Lyon, France, Apr. 2018.
- [123] Google. Android Debug Bridge (adb), 2020. <https://developer.android.com/studio/command-line/adb>.
- [124] Google. monkeyrunner, 2020. <https://developer.android.com/studio/test/monkeyrunner>.
- [125] Google. Webmaster guidelines, 2020. <https://support.google.com/webmasters/answer/35769>.
- [126] Google. Firebase, 2021. <https://firebase.google.com/>.
- [127] Google. SDKs, 2022. <https://support.google.com/analytics/answer/7373305?hl=en>.

- [128] Google Ad Manager. How exchange bidding works, 2019. <https://support.google.com/admanager/answer/7128958?hl=en>.
- [129] Google Ads. Understanding Google Ads and AdWords express country restrictions, 2019. <https://support.google.com/google-ads/answer/6163740?hl=en>.
- [130] Google Chrome. Puppeteer, 2019. <https://github.com/GoogleChrome/puppeteer>.
- [131] Google Play. ProxyDroid, 2021. https://play.google.com/store/apps/details?id=org.proxydroid&hl=en_CA&gl=US.
- [132] Google-Play-Scraper. Google-Play-Scraper, 2022. <https://pypi.org/project/google-play-scraper/>.
- [133] Google Safe Browsing. Google safe browsing, 2020. <https://safebrowsing.google.com/>.
- [134] Government of Canada. Bill C-11: An act to enact the Consumer Privacy Protection Act and the Personal Information and Data Protection Tribunal Act and to make related and consequential amendments to other acts, 2020. Proposed legislation: 2020; <https://www.justice.gc.ca/eng/csjsjc/pl/charter-charte/c11.html>.
- [135] Government of Canada. Personal information protection and electronic documents act, 2020. Enacted: 2000, last amended: 2019; <https://laws-lois.justice.gc.ca/ENG/ACTS/P-8.6/index.html>.
- [136] Haodf.com. Chinese official names of hospitals in China, 2021. <https://www.haodf.com/hospital/list-11.html/>.
- [137] M. Hastings, J. Fried, and N. Heninger. Weak keys remain widespread in network devices. In *IMC'16*, Santa Monica, CA, USA, Nov. 2016.
- [138] N. Heninger, Z. Durumeric, E. Wustrow, and J. Halderman. Mining your Ps and Qs: Detection of widespread weak keys in network devices. In *USENIX Security Symposium'12*, Bellevue, WA, USA, Aug. 2012.
- [139] Hola. Hola VPN, 2019. <http://hola.org/>.
- [140] R. Holz, J. Amann, O. Mehani, M. Wachs, and M. A. Kaafar. TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication. In *NDSS'16*, San Diego, CA, USA, Feb. 2016.
- [141] Hotjar. How to install your hotjar tracking code, 2022. <https://help.hotjar.com/hc/en-us/articles/115009336727-How-to-Install-your-Hotjar-Tracking-Code>.

- [142] howtoremove.guide. Trojan.Malware.300983.susgen, 2020. <https://howtoremove.guide/trojan-malware-300983-susgen/>.
- [143] M. G. Hoy and J. Phelps. Consumer privacy and security protection on church web sites: Reasons for concern. *Journal of Public Policy & Marketing*, 22(1):58–70, 2003.
- [144] Http toolkit.tech. Http toolkit, 2022. <https://httptoolkit.tech/>.
- [145] X. Hu, G. S. de Tangil, and N. Sastry. Multi-country study of third party trackers from real browser histories. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 70–86. IEEE, 2020.
- [146] S. Huang, F. Cuadrado, and S. Uhlig. Middleboxes in the Internet: a HTTP perspective. In *TMA'16*, Paris, France, Aug. 2017.
- [147] Hudak, Patrik. Finding phishing: tools and techniques, 2019. <https://0xpatrik.com/phishing-domains/>.
- [148] Iab.com. OpenRTB, 2022. <https://www.iab.com/guidelines/openrtb/>.
- [149] IEEE Internet initiative. Internet of things (IoT) security best practices - Feb. 2017, 2017. https://internetinitiative.ieee.org/images/files/resources/white_papers/internet_of_things_feb2017.pdf.
- [150] India Times. Hackers mined a fortune from Indian websites, 2018. <https://economictimes.indiatimes.com/small-biz/startups/newsbuzz/hackers-mined-a-fortune-from-indian-websites/articleshow/65836088.cms>.
- [151] Information and Communication Technology for Development (ICTD) Lab. HTTPS adoption measurement in governments worldwide, 2020. <https://github.com/uw-ictd/GovHTTPS-Data>.
- [152] Internet Engineering Task Force (IETF). Summarizing known attacks on Transport Layer Security (TLS) and datagram TLS (DTLS), 2015. RFC 7457 (<http://buildbot.tools.ietf.org/html/rfc7457>).
- [153] Internet Society. Personal Data Protection Guidelines for Africa, 2018. <https://www.internetsociety.org/resources/doc/2018/personal-data-protection-guidelines-for-africa/>.
- [154] Internet world stats. World country list, 2021. <https://www.internetworldstats.com/list1.htm>.
- [155] Intoli. It is *not* possible to detect and block Chrome headless, 2018. <https://intoli.com/blog/not-possible-to-block-chrome-headless/>.

- [156] L. Invernizzi, K. Thomas, A. Kapravelos, O. Comanescu, J.-M. Picod, and E. Bursztein. Cloak of visibility: Detecting when machines browse a different web. In *IEEE Symposium on Security and Privacy (SP'16)*, San Jose, CA, USA, May 2016.
- [157] U. Iqbal, S. Englehardt, and Z. Shafiq. Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors. In *IEEE Symposium on Security and Privacy (SP'21)*, Online, May 2021.
- [158] M. Jiang. The business and politics of search engines: A comparative study of Baidu and Google's search results of Internet events in China. *New Media & Society*, 16(2):212–233, 2014.
- [159] Joshua Roesslein. Tweepy, 2019. Online article (Nov 30, 2018). <https://github.com/tweepy/tweepy>.
- [160] A. Karaj, S. Macbeth, R. Berson, and J. M. Pujol. Whotracks.me: Shedding light on the opaque world of online tracking. *arXiv preprint arXiv:1804.08959*, 2018.
- [161] Kaspersky. What are the security and privacy risks of VR and AR, 2022. <https://www.kaspersky.com/resource-center/threats/security-and-privacy-risks-of-ar-and-vr>.
- [162] D. Kats, D. L. Silva, and J. Roturier. Who knows I like Jelly Beans? An investigation into search privacy. *Proceedings on Privacy Enhancing Technologies*, 2022(2):426–446, 2022.
- [163] Keywords Standings Ltd. URL Classification, 2020. <https://url-classification.io/>.
- [164] P. Kintis, N. Miramirkhani, C. Lever, Y. Chen, R. Romero-Gómez, N. Pitropakis, N. Nikiforakis, and M. Antonakakis. Hiding in plain sight: A longitudinal study of combosquatting abuse. In *ACM Conference on Computer and Communications Security (CCS'17)*, Dallas, TX, USA, Oct. 2017.
- [165] A. Kountouras, P. Kintis, C. Lever, Y. Chen, Y. Nadji, D. Dagon, M. Antonakakis, and R. Joffe. Enabling network security through active DNS datasets. In *International Symposium on Research in Attacks, Intrusions, and Defenses (RAID'16)*, Evry, France, Sept. 2016.
- [166] B. Krishnamurthy, K. Naryshkin, and C. Wills. Privacy leakage vs. protection measures: the growing disconnect. In *WTSP'11*, May 2011.
- [167] L. Stephens. Hakrawler, 2020. <https://github.com/hakluke/hakrawler>.

- [168] P. Laperdrix, W. Rudametkin, and B. Baudry. Beauty and the beast: Diverting modern web browsers to build unique browser fingerprints. In *IEEE Symposium on Security and Privacy (SP'16)*, San Jose, CA, USA, May 2016.
- [169] C. Latulipe, S. F. Mazumder, R. K. Wilson, J. W. Talton, A. G. Bertoni, S. A. Quandt, T. A. Arcury, and D. P. Miller. Security and privacy risks associated with adult patient portal accounts in us hospitals. *JAMA internal medicine*, 180(6):845–849, 2020.
- [170] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen. Tranco: A research-oriented top sites ranking hardened against manipulation. In *Network and Distributed System Security Symposium (NDSS'19)*, San Diego, CA, USA, Feb. 2019.
- [171] H. Lee, J. Lee, D. Kim, S. Jana, I. Shin, and S. Son. AdCube: WebVR ad fraud and practical confinement of third-party ads. In *USENIX Security Symposium (USENIX Security'21)*, Online, Aug. 2021.
- [172] H. Lee, T. Malkin, and E. Nahum. Cryptographic strength of SSL/TLS servers. In *IMC'07*, San Diego, CA, USA, Oct. 2007.
- [173] A. Lerner, A. K. Simpson, T. Kohno, and F. Roesner. Internet Jones and the raiders of the lost trackers: An archaeological study of web tracking from 1996 to 2016. In *USENIX Security Symposium (USENIX Security'16)*, Austin, TX, USA, Aug. 2016.
- [174] LevelDB. LevelDB, 2022. <https://github.com/google/leveldb>.
- [175] J. Liu, Y. Xiao, S. Li, W. Liang, and C. P. Chen. Cyber security and privacy issues in smart grids. *IEEE Communications surveys & tutorials*, 14(4):981–997, Jan. 2012.
- [176] Los Angeles Times. Muslims reel over a prayer app that sold user data, 2020. <https://www.latimes.com/business/technology/story/2020-11-23/muslim-pro-data-location-sales-military-contractors>.
- [177] Luke Leal. Malicious javascript used in wp site/home url redirects, 2020. <https://securityboulevard.com/2020/01/malicious-javascript-used-in-wp-site-home-url-redirects/>.
- [178] Luminati. *X-Forwarded-For* # issue 70, 2017. <https://github.com/luminati-io/luminati-proxy/issues/70>.
- [179] Luminati. Luminati proxy network, 2018. <http://luminati.io/>.
- [180] Luminati. Monetization SDK, 2019. <https://luminati.io/sdk>.
- [181] M. Richt. German government domains, 2020. <https://github.com/robbi5/german-gov-domains/>.

- [182] M. Zi'ang. LiteRadar, 2020. <https://github.com/pkumza/LiteRadar>.
- [183] M. Maass, P. Wichmann, H. Pridöhl, and D. Herrmann. Privacyscore: Improving privacy and security via crowd-sourced benchmarks of websites. In *Annual Privacy Forum (APF'17)*, Vienna, Austria, June 2017.
- [184] Malwarebytes Labs. Android/Adware.MobiDash, 2022. <https://blog.malwarebytes.com/detections/android-adware-mobidash/>.
- [185] Maple. Online doctors, virtual health & prescriptions in Canada, 2022. <https://www.getmaple.ca/>.
- [186] S. Marchal, G. Armano, T. Gröndahl, K. Saari, N. Singh, and N. Asokan. Off-the-hook: An efficient and usable client-side phishing prevention application. *IEEE Transactions on Computers*, 66(10):1717–1733, 2017.
- [187] J. R. Mayer and J. C. Mitchel. Third-party web tracking: Policy and technology. In *IEEE S&P'12*, San Francisco, CA, USA, May 2012.
- [188] MDN web docs. *X-Forwarded-For*, 2018. <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Forwarded-For>.
- [189] MedCalc. Values of the Chi-squared distribution, 2019. <https://www.medcalc.org/manual/chi-square-table.php>.
- [190] G. Merzdovnik, M. Huber, D. Buhov, N. Nikiforakis, S. Neuner, M. Schmiedecker, and E. Weippl. Block me if you can: A large-scale study of tracker-blocking tools. In *IEEE EuroS&P'17*, Paris, France, Apr. 2017.
- [191] X. Mi, Y. Liu, X. Feng, X. Liao, B. Liu, X. Wang, F. Qian, Z. Li, S. Alrwais, and L. Sun. Resident Evil: Understanding residential ip proxy as a dark service. In *IEEE S&P'19*, San Fansisco, CA, USA, May 2019.
- [192] Michael Carter. pywhois, 2010. <https://pypi.python.org/pypi/pywhois/0.1>.
- [193] A. Mirian, Z. Ma, D. Adrian, M. Tischer, T. Chuenchujit, T. Yardley, R. Berthier, J. Mason, Z. Durumeric, J. A. Halderman, and M. Bailey. An Internet-wide view of ICS devices. In *16*, Auckland, New Zealand, Dec. 2016.
- [194] mitmproxy. mitmproxy, 2021. <https://mitmproxy.org/>.
- [195] mitre. Cobalt strike, 2021. <https://attack.mitre.org/software/S0154/>.
- [196] MobSF. Mobile Security Framework (MobSF), 2020. <https://github.com/MobSF/Mobile-Security-Framework-MobSF>.

- [197] B. Möller, T. Duong, and K. Kotowicz. This POODLE bites: Exploiting the SSL 3.0 fallback, 2014. <https://www.openssl.org/~bodo/ssl-poodle.pdf>.
- [198] NBC news. Major hospital system hit with cyberattack, potentially largest in U.S. history, 2020. <https://www.nbcnews.com/tech/security/cyberattack-hits-major-u-s-hospital-system-n1241254>.
- [199] T. T. Nguyen, M. Backes, N. Marnau, and B. Stock. Share first, ask later (or never?) studying violations of GDPR’s explicit consent in Android apps. In *USENIX Security Symposium (USENIX Security’21)*, Online, Aug. 2021.
- [200] J. D. Niforatos, A. R. Zheutlin, and J. B. Sussman. Prevalence of third-party data tracking by us hospital websites. *JAMA Network Open*, 4(9):e2126121–e2126121, 2021.
- [201] OECD.org. Classification of the Functions of Government (COFOG), 2011. <https://www.oecd.org/gov/48250728.pdf>.
- [202] A. Oest, Y. Safaei, A. Doupé, G.-J. Ahn, B. Wardman, and K. Tyers. PhishFarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *IEEE Symposium on Security and Privacy (SP’19)*, San Francisco, CA, USA, May 2019.
- [203] A. Oest, Y. Safaei, P. Zhang, B. Wardman, K. Tyers, Y. Shoshitaishvili, and A. Doupé. Phishtime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists. In *USENIX Security Symposium (USENIX Security’20)*, Online, Aug. 2020.
- [204] A. Oest, P. Zhang, B. Wardman, E. Nunes, J. Burgis, A. Zand, K. Thomas, A. Doupé, and G.-J. Ahn. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *USENIX Security Symposium (USENIX Security’20)*, Online, Aug. 2020.
- [205] Office of the auditor general western Australia. Local government general computer controls, 2021. https://audit.wa.gov.au/wp-content/uploads/2021/05/Report-23_Local-Government-General-Computer-Controls.pdf.
- [206] OneSpan. Fraud analytics, 2021. <https://www.onespan.com/topics/fraud-analytics>.
- [207] Opto 22. Opto 22 products, 2018. <http://www.opto22.com/site/products.aspx>.
- [208] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow. IoT-POT: Analysing the rise of IoT compromises. In *USENIX Security Symposium’15*, Washington, D.C., USA, Aug. 2015.

- [209] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle. Website fingerprinting at Internet scale. In *NDSS'16*, San Diego, CA, USA, Feb. 2016.
- [210] Pandemic Religion. Social media use during COVID-19, 2020. <https://pandemicreligion.org/s/contributions/page/social-media-use>.
- [211] T. K. Panum, K. Hageman, R. R. Hansen, and J. M. Pedersen. Towards adversarial phishing detection. In *USENIX Security Symposium (USENIX Security'20)*, Online, Aug. 2020.
- [212] E. Papadogiannakis, P. Papadopoulos, N. Kourtellis, and E. P. Markatos. User tracking in the post-cookie era: How websites bypass GDPR consent to track users. In *TheWebConf'21*, Ljubljana, Slovenia, Apr. 2021.
- [213] PCrisk. ERR_NAME_NOT_RESOLVED - How to fix?, 2019. <https://blog.pcrisk.com/windows/12819-err-name-not-resolved>.
- [214] P. Pearce, B. Jones, F. Li, R. Ensafi, N. Feamster, N. Weaver, and V. Paxson. Global measurement of DNS manipulation. In *USENIX Security Symposium'17*, Vancouver, British Colombia, Canada, Aug. 2017.
- [215] S. Peisert, B. Schneier, H. Okhravi, F. Massacci, T. Benzel, C. Landwehr, M. Manan, J. Mirkovic, A. Prakash, and J. Michael. Perspectives on the SolarWinds incident. *IEEE Security & Privacy*, 19(02):7–13, mar 2021.
- [216] P. Peng, C. Xu, L. Quinn, H. Hu, B. Viswanath, and G. Wang. What happens after you leak your password: Understanding credential sharing on phishing sites. In *ACM Asia Conference on Computer and Communications Security (AsiaCCS'19)*, Auckland, New Zealand, July 2019.
- [217] P. Peng, L. Yang, L. Song, and G. Wang. Opening the blackbox of VirusTotal: Analyzing online phishing scan engines. In *IMC'19*, 2019.
- [218] Pew Research center. The rise of the e-citizen: How people use government agencies' web sites, 2002. <https://www.pewresearch.org/internet/2002/04/03/the-rise-of-the-e-citizen-how-people-use-government-agencies-web-sites/>.
- [219] Pew Research Center. Few Americans say their house of worship is open, 2020. <https://www.pewresearch.org/fact-tank/2020/04/30/few-americans-say-their-house-of-worship-is-open-but-a-quarter-say-their-religious-faith-has-grown-amid-pandemic/>.

- [220] Pierluigi Paganini. US Government website was hosting a JavaScript downloader delivering Cerber ransomware, 2017. <https://securityaffairs.co/wordpress/62629/hacking/us-government-website-malware.html>.
- [221] PKI Consortium. One year certs, 2020. <https://pkic.org/2020/07/09/one-year-certs/>.
- [222] PortSwigger. Burp Suite, 2022. <https://portswigger.net/burp>.
- [223] Princeton University. OpenWPM, 2020. <https://github.com/citp/OpenWPM>.
- [224] Project Sonar. Forward DNS (FDNS), 2020. https://opendata.rapid7.com/sonar.fdns_v2/.
- [225] G. Pugliese, C. Riess, F. Gassmann, and Z. Benenson. Long-term observation on browser fingerprinting: Users' trackability and perspective. *PoPETs*, 2020(2):558–577, 2020.
- [226] PyOpenSSL. PyOpenSSL, 2022. <https://pypi.org/project/pyOpenSSL/>.
- [227] Pypi.org. selenium-wire, 2022. <https://pypi.org/project/selenium-wire/>.
- [228] R. Alam. gplaydl, 2020. <https://github.com/rehmatworks/gplaydl>.
- [229] R. S. Rao, T. Vaishnavi, and A. R. Pais. Catchphish: Detection of phishing websites by inspecting urls. *Journal of Ambient Intelligence and Humanized Computing*, 11(2):813–825, 2020.
- [230] Recode.net. Google leads the world in digital and mobile ad revenue, 2017. <https://www.recode.net/2017/7/24/16020330/google-digital-mobile-ad-revenue-world-leader-facebook-growth>.
- [231] Retire.js. Retire.js, 2022. <https://retirejs.github.io/retire.js/>.
- [232] Reuters. France embraces Google, Microsoft in quest to safeguard sensitive data, 2021. <https://www.reuters.com/technology/france-embraces-google-microsoft-quest-safeguard-sensitive-data-2021-05-17/>.
- [233] RIPE NCC. RIPE NCC, 2020. <https://www.ripe.net/>.
- [234] R. Robinson. Prevalence of web trackers on hospital websites in Illinois. *arXiv preprint arXiv:1805.01392*, 2018.

- [235] F. Roesner, T. Kohno, and D. Wetherall. Detecting and defending against Third-Party tracking on the web. In *USENIX Security Symposium (USENIX Security'12)*, Boston, MA, USA, Aug. 2012.
- [236] E. Ronen, C. O'Flynn, A. Shamir, and A.-O. Weingarten. IoT goes nuclear: Creating a ZigBee chain reaction. Cryptology ePrint Archive, Report 2016/1047. <https://eprint.iacr.org/2016/1047>, 2016.
- [237] F. Rowe. Contact tracing apps and values dilemmas: A privacy paradox in a neo-liberal world. *International Journal of Information Management*, 55:102178, 2020.
- [238] S. Sahni. Firebase scanner, 2019. <https://github.com/shivsahni/FireBaseScanner>.
- [239] N. Samarasinghe, A. Adhikari, M. Mannan, and A. Youssef. Et tu, brute? privacy analysis of government websites and mobile apps. In *TheWebConf'22*, Online, Apr. 2022.
- [240] N. Samarasinghe and M. Mannan. Short paper: TLS ecosystems in networked devices vs. web servers. In *Financial Cryptography and Data Security'17*, Malta, Apr. 2017.
- [241] N. Samarasinghe and M. Mannan. Towards a global perspective on web tracking. *Computers & Security*, 87:101569, 2019.
- [242] I. Sanchez-Rola, M. Dell'Amico, D. Balzarotti, P.-A. Vervier, and L. Bilge. Journey to the center of the cookie ecosystem: Unraveling actors' roles and relationships. In *IEEE Symposium on Security and Privacy (SP'21)*, Online, May 2021.
- [243] I. Sanchez-Rola and I. Santos. Knockin'on trackers' door: Large-scale automatic analysis of web tracking. In *I8*, Saclay, France, June 2018.
- [244] S. Schelter and J. Kunegis. Tracking the trackers: A large-scale analysis of embedded web trackers. In *ICWSM'16*, Cologne, Germany, May 2016.
- [245] School of Psychology University of New England. Z-scores, 2019. https://webstat.une.edu.au/unit_materials/c4_descriptive_statistics/z_scores.htm.
- [246] T. D. Science. Topic modeling and latent dirichlet allocation (LDA) in Python, 2018. <https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24>.
- [247] ScoreCard Research. ScoreCard Research - Privacy policy, 2017. <http://www.scorecardresearch.com/privacy.aspx?newlanguage=1>.

- [248] ScrapeHero. How to prevent getting blacklisted while scraping, 2020. <https://www.scrapehero.com/how-to-prevent-getting-blacklisted-while-scraping/>.
- [249] SEC-Consult.com. House of keys: Industry-wide HTTPS certificates and SSH key reuse endangers millions of devices worldwide, 2015. <https://www.sec-consult.com/en/blog/2015/11/house-of-keys-industry-wide-https/>.
- [250] SecureIca. Exploring Google hacking techniques using dork, 2020. <https://medium.com/nassec-cybersecurity-writeups/exploring-google-hacking-techniques-using-google-dork-6df5d79796cf>.
- [251] Selenium. SeleniumHQ browser automation, 2019. <http://www.seleniumhq.org/>.
- [252] SerpApi. Baidu organic results API, 2022. <https://serpapi.com/baidu-organic-results>.
- [253] Shivam Agarwal. BlockListParser, 2016. <https://github.com/shivamagarwal-iitb/BlockListParser>.
- [254] Shodan. Shodan search engine, 2018. <https://www.shodan.io/>.
- [255] Similarweb. Top Websites Ranking for Faith and Beliefs in the world. Online article (2022). <https://tinyurl.com/2p9d43jk>.
- [256] S. Singanamalla, E. H. B. Jang, R. Anderson, T. Kohno, and K. Heimerl. Accept the risk and continue: Measuring the long tail of government HTTPS adoption. In *ACM Internet measurement conference (IMC'20)*, Online, Oct. 2020.
- [257] Softpedia news. Hacked Turkish Government website used to distribute malware, 2013. <https://news.softpedia.com/news/Hacked-Turkish-Government-Website-Used-to-Distribute-Malware-389937.shtml>.
- [258] K. Solomos, J. Kristoff, C. Kanich, and J. Polakis. Tales of favicons and caches: Persistent tracking in modern browsers. In *NDSS'21*, Online, Feb. 2021.
- [259] A. Soltani, S. Canty, Q. Mayo, L. Thomas, and C. J. Hoofnagle. Flash cookies and privacy. In *Association for the Advancement of Artificial Intelligence Spring Symposium Series (AAAI'10)*, Palo Alto, CA, USA, Mar. 2010.
- [260] O. Sørensen. Zombie-cookies: Case studies and mitigation. In *13*), Dec. 2013.

- [261] J. Spooren, T. Vissers, P. Janssen, W. Joosen, and L. Desmet. Premadoma: An operational solution for DNS registries to prevent malicious domain registrations. In *Annual Computer Security Applications Conference (ACSAC'19)*, San Juan, Puerto Rico, USA, Dec. 2019.
- [262] SSL Labs. SSL and TLS deployment best practices, 2017. <https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices>.
- [263] State of California Department of Justice. California Consumer Privacy Act (CCPA), 2021. <https://oag.ca.gov/privacy/ccpa>.
- [264] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov. The first collision for full SHA-1. In *Crypto'17*, Santa Barbara, CA, USA, Aug. 2017.
- [265] M. Stevens, P. Karpman, and T. Peyrin. Freestart collision for full SHA-1. In *Eurocrypt'16*, Vienna, Austria, May 2016.
- [266] M. Stevens, A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. Osvik, and B. de Weger. Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In *CRYPTO'09*, Santa Barbara, CA, USA, Aug. 2009.
- [267] B. Stock, G. Pellegrino, F. Li, M. Backes, and C. Rossow. Didn't you hear me? - Towards more successful web vulnerability notifications. In *NDSS'18*, San Diego, CA, USA, Feb. 2018.
- [268] SuatPhish. Squatting-domain-identification, 2018. <https://github.com/SquatPhish/1-Squatting-Domain-Identification>.
- [269] Sucuri blog. jquery.min.php malware affects thousands of websites, 2015. <https://blog.sucuri.net/2015/11/jquery-min-php-malware-affects-thousands-of-websites.html>.
- [270] P. P. Swire and K. Ahmad. *Foundations of information privacy and data protection: A survey of global concepts, laws and practices*. International Association of Privacy Professionals, 2012.
- [271] Symantec. Webpulse site review request, 2020. <https://sitereview.bluecoat.com/#/>.
- [272] TechRepublic.com. Mirai variant botnet launches IoT DDoS attacks on financial sector. News article (Apr. 5, 2018). <https://www.techrepublic.com/article/mirai-variant-botnet-launches-iot-ddos-attacks-on-financial-sector/>.
- [273] TechRepublic.com. More than 800K DrayTek routers vulnerable to DNS reprogramming attack, 2018. <https://www.techrepublic.com/>

article/more-than-800k-draytek-routers-vulnerable-to-dns-reprogramming-attack/.

- [274] Tesseract. Tesseract-OCR, 2019. <https://github.com/tesseract-ocr/tesseract>.
- [275] The Economic Times. Data Breach of Indian Patients. <https://economictimes.indiatimes.com/tech/internet/german-firm-finds-one-million-files-of-indian-patients-leaked/articleshow/73921423.cms?from=mdr>.
- [276] The EU Internet Handbook. Cookies, 2019. Online article (Dec 10, 2018). http://ec.europa.eu/ipg/basics/legal/cookies/index_en.htm.
- [277] The Guardian. Government websites hit by cryptocurrency mining malware, 2018. <https://www.theguardian.com/technology/2018/feb/11/government-websites-hit-by-cryptocurrency-mining-malware>.
- [278] The New York Times. Google introduces a new system for tracking Chrome browser users, 2022. <https://www.nytimes.com/2022/01/25/business/google-topics-chrome-tracking.html>.
- [279] The Reboot. Internet evolution: A timeline history of the network, 2020. <https://thereboot.com/internet-evolution-a-timeline-history-of-the-network/>.
- [280] The Washington Post. Chinese state-backed hackers infiltrated vatican, 2020. <https://tinyurl.com/mpttxmc>.
- [281] TheRegister.co.uk. Internet of Sins: Million more devices sharing known private keys for HTTPS, SSH admin, 2016. https://www.theregister.co.uk/2016/09/07/bad_key_security_holes_getting_worse/.
- [282] K. Tian, S. T. Jan, H. Hu, D. Yao, and G. Wang. Needle in a haystack: Tracking down elite phishing domains in the wild. In *ACM Internet measurement conference (IMC'18)*, Boston, MA, USA, Oct. 2018.
- [283] C. J. Tolbert and K. Mossberger. The effects of e-government on trust and confidence in government. *Public administration review*, 66(3):354–369, 2006.
- [284] A. Tolley and D. Mundy. Towards workable privacy for UK e-government on the web. *International Journal of Electronic Governance*, 2(1):74–88, 2009.
- [285] Tor. Tor project, 2019. <https://www.torproject.org/>.

- [286] S. Torabi, E. Bou-Harb, C. Assi, M. Galluscio, A. Boukhtouta, and M. Debbabi. Inferring, characterizing, and investigating Internet-scale malicious IoT device activities: A network telescope perspective. In *18*, Luxembourg, June 2018.
- [287] M. Trevisan, S. Traverso, E. Bassi, and M. Mellia. 4 years of EU cookie law: Results and lessons learned. *Proceedings on Privacy Enhancing Technologies*, 2019(2):126–145, 2019.
- [288] G. Tyson, S. Huang, F. Cuadrado, I. Castro, V. Perta, A. Sathiaseelan, and S. Uhlig. Exploring HTTP header manipulation in-the-wild. In *TheWebConf'17*, Perth, Australia, Apr. 2017.
- [289] weareprivacy.com. Policy highlights, 2021. <https://github.com/weareprivacy/policy-highlights>.
- [290] US-CERT. Alert (ta16-250a) - The increasing threat to network infrastructure devices and recommended mitigations, 2016. <https://www.us-cert.gov/ncas/alerts/TA16-250A>.
- [291] U.S. Government. Health insurance profitability and accountability act (HIPPA), 1996. <https://www.govinfo.gov/content/pkg/PLAW-104publ1191/html/PLAW-104publ1191.htm>.
- [292] P. Vadrevu and R. Perdisci. What you see is not what you get: Discovering and tracking social engineering attack campaigns. In *ACM Internet measurement conference (IMC'19)*, Amsterdam, Netherlands, Oct. 2019.
- [293] L. Valenta, S. Cohny, A. Liao, S. Fried, Joshua Bodduluri, and N. Heninge. Factoring as a service. In *16*, Barbados, Feb. 2016.
- [294] P. Vallina, V. Le Pochat, Á. Feal, M. Paraschiv, J. Gamba, T. Burke, O. Hohlfeld, J. Tapiador, and N. Vallina-Rodriguez. Mis-shapes, mistakes, misfits: An analysis of domain classification services. In *ACM Internet measurement conference (IMC'20)*, Online, Oct. 2020.
- [295] Vice.com. Hackers turned Virginia government websites into elaborate eBooks scam pages, 2020. <https://www.vice.com/en/article/88947x/hackers-virginia-government-websites-ebooks-scam>.
- [296] Viehböck, Stefan and Durumeric, Zakir. Fingerprints for certificates with know private keys, 2016. <https://github.com/zmap/ztag/blob/master/ztag/annotations/tlskeyknown.py>.
- [297] Virginia.gov. SB 1392 Consumer Data Protection Act; establishes a framework for controlling and processing personal data, 2021. <https://lis.virginia.gov/cgi-bin/legp604.exe?211+sum+SB1392>.
- [298] VirusTotal. VirusTotal, 2021. <https://www.virustotal.com>.

- [299] N. Vratonjic, J. Freudiger, V. Bindschaedler, and J.-P. Hubaux. The inconvenient truth about web certificates. In *WEIS'11*, Fairfax, VA, USA, June 2011.
- [300] W3s.org. Same origin policy, 2022. https://www.w3.org/Security/wiki/Same_Origin_Policy.
- [301] D. Y. Wang, S. Savage, and G. M. Voelker. Cloak and dagger: Dynamics of web search cloaking. In *ACM Conference on Computer and Communications Security (CCS'11)*, Chicago, Illinois, USA, Oct. 2011.
- [302] X. Wang and H. Yu. How to break MD5 and other hash functions. In *Eurocrypt'05*, Aarhus, Denmark, May 2005.
- [303] Wapiti. Wapiti - The web application vulnerability scanner, 2022. <https://wapiti-scanner.github.io/>.
- [304] WashingtonPost.com. NSA uses Google cookies to pinpoint targets for hacking, 2013. <https://www.washingtonpost.com/news/the-switch/wp/2013/12/10/nsa-uses-google-cookies-to-pinpoint-targets-for-hacking/>.
- [305] V. Wesselkamp, I. Fouad, C. Santos, Y. Boussad, N. Bielova, and A. Legout. In-depth technical and legal analysis of tracking on health related websites with ernie extension. In *WPES'21*, Online, Nov. 2021.
- [306] Wikipedia. .gov, 2021. <https://en.wikipedia.org/wiki/.gov>.
- [307] Wikipedia. List of sovereign states, 2021. https://en.wikipedia.org/wiki/List_of_sovereign_states.
- [308] Wikipedia. List of hospitals in Canada, 2022. https://en.wikipedia.org/wiki/List_of_hospitals_in_Canada.
- [309] Wired. How Cambridge Analytica sparked the great privacy awakening, 2019. <https://www.wired.com/story/cambridge-analytica-facebook-privacy-awakening/>.
- [310] Wired.com. The Reaper IoT botnet has already infected a million networks, 2017. <https://www.wired.com/story/reaper-iot-botnet-infected-million-networks/>.
- [311] F. Wollsn. OpenWPM WebExtension experiment / API, 2018. <https://github.com/mozilla/OpenWPM-WebExtension-Experiment>.
- [312] World mail & express americas conference. Cookie policy, 2021. <https://www.wmxamericas.com/cookie-policy/>.

- [313] B. Wu and B. D. Davison. Detecting semantic cloaking on the web. In *International World Wide Web Conference (WWW'06)*, Edinburgh, Scotland, UK, May 2006.
- [314] H. Xu, F. Xuy, and B. Chenz. Internet protocol cameras with no password protection: An empirical investigation. In *I8*, Berlin, Germany, Mar. 2018.
- [315] Z. Wang. googler, 2020. <https://github.com/jarun/googler>.
- [316] Y. Zeng, T. Zang, Y. Zhang, X. Chen, and Y. Wang. A comprehensive measurement study of domain-squatting abuse. In *IEEE International Conference on Communications (ICC'19)*, Shanghai, China, May 2019.
- [317] A. R. Zheutlin, J. D. Niforatos, and J. B. Sussman. Data-tracking among digital pharmacies. *Annals of Pharmacotherapy*, 2022.
- [318] Z. Zhou, L. Yu, Q. Liu, Y. Liu, and B. Luo. Tear off your disguise: Phishing website detection using visual and network identities. In *International Conference on Information and Communications Security (ICICS'19)*, Beijing, China, Dec. 2019.
- [319] S. Zimmeck, J. S. Li, H. Kim, S. M. Bellovin, and T. Jebara. A privacy analysis of cross-device tracking. In *USENIX Security Symposium'17*, Vancouver, British Columbia, Canada, Aug. 2017.
- [320] ZMap. Blacklisting, 2017. <https://github.com/zmap/zmap/wiki/Blacklisting>.
- [321] C. Zuo, Z. Lin, and Y. Zhang. Why does your data leak? uncovering the data leakage in cloud from mobile apps. In *IEEE Symposium on Security and Privacy (SP'19)*, San Fransisco, CA, USA, May 2019.