MINOS: A tool to analyze HTTP requests for compliance to GDPR

MINOS: A Tool to Analyze HTTP Requests For Compliance to GDPR

Lorenzo Laudadio, Riccardo Coppola, Antonio Vetrò, Marco Torchiano, Juan Carlos De Martin
Dept. of Control and Computer Engineering
Politecnico di Torino
Turin, Italy
first.last@polito.it

***Abstract*—Many of the currently available services and commodities extensively share personal data and digital identities, raising privacy, security and ethical concerns.**

In this paper, we present a novel software, MINOS, designed to enhance users' privacy awareness while browsing the web. Our approach combines a user-friendly browser with a backend tailored to record and analyze HTTP requests directed to countries outside the European Economic Area (EEA).

As opposed to other available tools, our solution uniquely addresses the issue of data transfers to third countries.

Preliminary simulations have yielded promising results and clear directions for further research and development of the tool.

***Index Terms*—Privacy, GDPR, Personal Data Transfer, Software Development.**

I. Introduction

In an era where digital connectivity permeates every aspect of our lives, the sharing of personal data has become not only commonplace but widespread and in many cases essential. The sharing of what constitutes our digital identity – and can reveal much about our lives – is the basis for many services and commodities we use in our everyday lives, especially when demanding highly personalized services. At the same time, many companies have built their business on extracting and selling such data, with commercial exchanges of personal data having nowadays a significant impact on the economy [1].

This constant flow of information has inevitably raised critical issues regarding privacy, security, and ethical use of personal data. These concerns have been brought to the public attention by several high-profile cases in recent years, such as the Edward Snowden case in 2013, when the extensive surveillance and bulk collection of telephone metadata and internet communications by the NSA were exposed to the public [2]; the 2018 Cambridge Analytica Scandal, when it was revealed that personal data of millions of Facebook users were harvested without their consent to create targeted political advertisements during the 2016 US presidential election campaign [3]; the recent report by Amnesty International about the systematic use of biometric surveillance systems targeting civilians in the 2023 Israelo-Palestinian conflict [4].

As a result of the growing concerns about privacy and personal data protection, the GDPR has been enforced by the European Union (EU), to give European citizens more control over their data and to create a unified framework for data protection across European countries. GDPR, among other things, specifies how personal data transfers from the European Union and the European Economic Area (EEA) to third countries should be managed: it forbids personal data transfers to countries not subject to an adequacy decision, which cannot guarantee appropriate safeguards and for which special derogations cannot apply. In most cases, the internal law of these countries conflicts with the Charter of Fundamental Rights of the European Union.

The problem of personal data transfer from EEA to third countries motivated us to develop MINOS, a tool conceived to aid users of the web in analyzing the data collected during their navigation, and notify the sending of personal data to blacklisted hosts.

We describe the architecture and the main features of MINOS in Section III, after reporting on the current regulatory frame and presenting similar tools in Section II. Then, in Section IV, we report a running sample of the usage of MINOS. In Section V we discuss the result and compare MINOS with similar tools. Finally, in Section VI, we compare the tool to the state of the art and identify future work directions.

II. Background and Related Work

The GDPR (General Data Protection Regulation) is the law established by the European Union to regulate personal data processing. Its main goals are to protect the privacy and rights of individuals regarding their personal data and to coordinate data protection laws across the EU [5].

According to the GDPR, personal data is defined as *any information relating to an identified or identifiable natural person, including location and online identifiers, such as the IP address of a host.*

The regulation establishes (in Art. 45) that the European Commission (EC) can issue an *adequacy decision* in favour of a non-EEA country if such a country offers an adequate level of protection for personal data. Whenever an adequacy decision is issued for a country, data can be transferred from the EEA to that country without additional measures.

In the absence of an adequacy decision, data can be transferred to a third country if it provides additional safeguards, enforceable data subject rights and effective legal remedies for data subjects (Art46) or in presence of some additional derogations allowing for such data transfers (art. 49).

In the context of navigating the web within the EU, it might be of the user's interest to identify third-party trackers[1], protecting online privacy and checking for GDPR compliance.

Several tools have been proposed in related white and grey literature for that purpose. However, from our analyses of these tools, none of them seems to be able to detect data transfers to non-EEA domains.

Privacy Badger is a browser extension by the Electronic Frontier Foundation (EFF) that blocks third-party trackers during the navigation [6]. It identifies the third-party domains that embed images, scripts and advertising in the visited web pages and looks for tracking techniques being used. If it observes a third-party tracking behaviour, it automatically disallows content from the tracker.

Blacklight by The Markup is a web application which emulates the user interaction on a web page and tries to detect possible privacy violations [7]. It works by opening a headless browser and visiting the URL homepage as well as additional randomly selected web pages from the same website.

ImmuniWeb security test is an AI-aided online tool for security assessment [8]. It also offers a free GDPR compliance test, which includes the following checks: check for a privacy policy to be present on the website, checks on the website security, check for TLS encryption, cookie protection and cookie disclaimer.

2gdpr is a web application which allows testing for GDPR compliance [9]. 2gdpr also claims to be able to detect personal data transfers to non-adequate countries. We tested the tool on an Italian website that we knew contained requests to Russian domains (Russia is one of the countries for which an adequacy decision is currently missing), but the report did not include any reference to those requests.

webXray is an open-source Python software for analyzing web traffic and content, extracting legal policies from web pages and identifying the companies which collect user data. It exploits the Chrome DevTools Protocol. It can produce several statistics on the analyzed web pages, such as the cookies present on the web pages, the JavaScript code, the most frequently occurring third-party domains and requests, etc. At the time of writing, the official website (https://webxray.org) is made inaccessible by an authentication form, and the original GitHub repository is missing (https://github.com/timlib/webXray), however previous snapshots are available through the Wayback Machine service (https://web.archive.org/web/20240110232139/https://webxray.org/).

OpenWPM is an extensible open-source tool designed for large-scale web privacy measurements, automating data collection on tracking technologies such as cookies and fingerprinting scripts by simulating user browsing behaviour [10]. It captures detailed logs of network requests and JavaScript execution, enabling comprehensive analysis of web tracking practices.

[1]A third-party tracker is a tool or piece of code used by an external company to collect data about online activities of an user while she is visiting a certain website.

III. The MINOS Tool

In this section we describe the technological aspects of the MINOS tool. Our final purpose was to develop a software tool targeted at non-advanced users, for navigating the web while analysing and reporting HTTP requests to non-EEA domains. The software is open-source and available on Zenodo [11].

A. Requirements

Throughout the rest of the paper, the following key terms will be used:

- **Bad host/bad domain:** A host/domain which is located outside of the EEA.
- **Bad request:** A request made to a bad host.
- **Blacklist:** Refers to an internal blacklist which collects bad domains.
- **Bad group:** Refers to the domain groups within the blacklist. Each domain in the blacklist belongs to a domain group (e.g. the domain `.youtube.co` belongs to the group `youtube`).

We identified and implemented the following functional requirements for the MINOS tool:

1) **Navigation:** The user should be able to navigate the web.
   a) The user should visualize the start screen with a "Verify a website"[2] button.
   b) After clicking on the "Verify" button, a window with an input bar and the "Start" button should be displayed.
   c) The user should be able to input a URL in the input bar.
   d) By pressing the "Start" button the navigation should start. The URL cannot be changed anymore.
   e) If the host is unreachable, an error message should be displayed and the application should quit.
   f) If the host is reachable, the user should be able to navigate.
   g) By clicking on the "Analyze" button, the user should be able to continue with the analysis step.
2) **Analysis:** The user should be able to analyze the data which have been collected during the navigation, to possibly identify bad hosts. The requests made by the user should be matched against the internal blacklist of non-EEA domains[3].
   a) A dialog which allows saving the navigation log should be displayed.
   b) If no bad hosts have been detected, an info message should be displayed and the application should quit.
   c) The list of bad hosts detected should be displayed.

[2]The names of the buttons have been translated to ease the readability for non-Italian speakers.

[3]The blacklist was compiled by MonitoraPA, a community of volunteers which maintains an automated distributed observatory on the Italian Public Administration [12], and is contained in the `hosts.json` file which can be found in the repository.

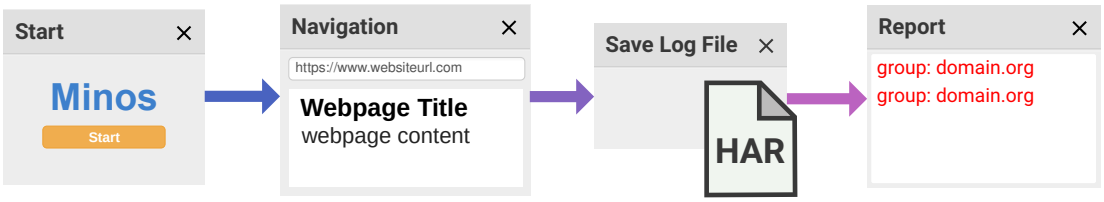

Fig. 1. Mockups of the MINOS workflow

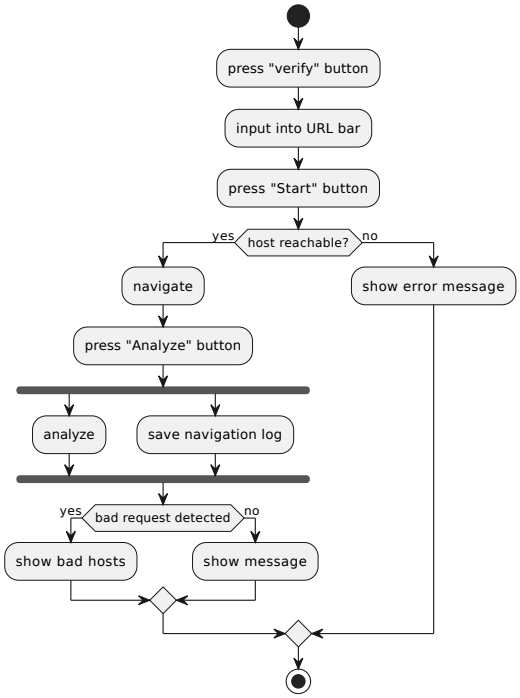

Fig. 2. Activity Diagram of the User's interaction with the MINOS tool

Figure 1 shows a mockup of the application, while Figure 2 illustrates the activity diagram. The process initiates with the user pressing the "Verify" button, followed by inputting a URL and pressing the "Start" button. After the "Start" button is pressed, MINOS tries to contact the host; if the host is unreachable, an error message is displayed. If the host is reachable, the browser navigates to the URL and the user is prompted to press the "Analyze" button. The analysis phase encompasses two primary activities: performing a detailed analysis of the web requests and saving the navigation log. A subsequent check identifies any bad requests encountered during navigation. If the software was able to detect bad requests, then a report window is shown, displaying the list of bad hosts. Otherwise, a message informing the user that no bad requests were detected is displayed.

B. Architecture

The software has been developed by using the Electron JavaScript framework, consequently, the architecture heavily depends on the Electron process model: the architectural diagram is shown in Figure 3.

An Electron app consists of two kinds of process: the *main* process and the *renderer* process. The main process runs in a `Node.js` environment (that is, a JavaScript runtime environment), and therefore it can interact with the local filesystem. The renderer process is responsible for rendering the web content (i.e. the GUI). It does not have access to the Node.js environment. It is possible to write *preload scripts*, which act like a sort of bridge between the main and renderer process, and allow them to communicate.

The main process involves the execution of the following tasks:

- **Windows and views management:** Create the application window and the views (i.e. frames within the window). Our application has two views: one which displays local content (i.e. the top bar, the buttons, etc.) and the other which displays remote content (i.e. web pages).
- **Navigation:** Navigate to a specific web page.
- **Capture of the network traffic:** Since Electron internally relies on a Chromium engine to render the web pages, we can exploit the *Chrome DevTools Protocol* (CDP) API, which is integrated into all the Chromium-based web browsers. Electron offers a convenient API to exploit the CDP: the *Chrome Debugger API*.
- **Analysis of the network traffic:** The URLs of the HTTP requests coming from the application are matched against an internal blacklist of domains. A longest-matching strategy is used: if the URL of a request matches against more bad hosts only the longest one is taken.
- **Logging:** The log of the navigation is stored in a HAR file. HAR is a JSON-based format which was proposed as a draft by the W3C Web Performance Working Group but never standardized. Even if this standard format is no longer maintained, all the major web browsers use the HAR format for logging web navigation sessions, therefore we chose this format mainly for compatibility reasons. To create the HAR object in memory, we relied upon a slightly modified version of the Chrome HAR Capturer, an open-source project which allows to translate HTTP requests to HAR format [13].

The renderer process has the following components:

- **LocalView:** it displays the application GUI, i.e. the URL bar and the buttons.
- **WebView:** it displays the remote web pages which are fetched by the Main process.

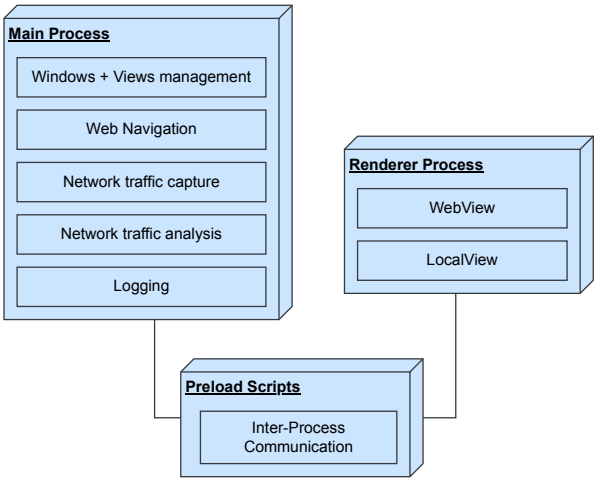

Fig. 3. Architectural Diagram

IV. Preliminary Evaluation

We performed a preliminary evaluation of the MINOS tool, to verify the core functionalities and to assess the compliance of our software to the functional requirements detailed in Subsection III-A. To conduct the evaluation, we selected a representative sample consisting of 10 Italian Public Administration websites (4 municipalities, 1 national order, 1 art teaching institution, 1 university, 1 museum, 1 science institute, 1 tourism organization), namely:

- www.accademiacarrara.it,
- www.comune.peio.tn.it,
- www.comune.rivadelgarda.tn.it,
- www.viaggiareinpuglia.it,
- www.unical.it,
- www.museosanmichele.it,
- www.gssi.it,
- www.comunediroccagiovine.it,
- www.comune.correggio.re.it,
- www.architettibelluno.it.

This is a random selection from a collection of Italian Public Administration websites known to generate requests to third countries in the past. To These websites were taken from the publicly available database OpenDataIPA [14].

The assessment process is the following:

1) We run MINOS on the websites and save the list of detected bad hosts (if any).
2) We run the Chromium web browser on the same websites and export the HAR log of the navigation.
3) We use a script to check that all the bad hosts present in the Chromium-HAR were effectively present in the MINOS report.

To make sure that the Chromium navigation was as similar as possible to the MINOS one, we navigated only the homepages of the websites. Figures 4, 5 show two sample results of the preliminary evaluation of MINOS on two different websites. The tool detected requests to both US (e.g. *Google, Facebook*) and Russian domains (e.g. *Yandex*).

Considering all requests generated from the websites, we observed that MINOS successfully detected and reported 30 bad hosts out of the 40 present in the Chromium logs, with an accuracy of 75%.

Tool	non-EEA data transfers checking	Anti-tracking	Serverless	Open-source
MINOS	✓	✗	✓	✓
Privacy Badger	✗	✓	✓	✓
Blacklight	✗	✓	✗	✓
ImmuniWeb	✗	✗	✗	✗
2gdpr	✓	✗	✗	✗
webXray	✗	✓	✓	✓
OpenWPM	✗	✓	✓	✓

Tab. I
Comparison between MINOS and other software

V. Discussion and Comparison with Similar Tools

The initial results are promising, showing that MINOS can to detect the requests to bad hosts – as specified in the functional requirements – for a large majority of cases. An analysis of the missing bad hosts (28%) determined that they can be primarily attributed to a limitation within the current detection algorithm. Specifically, the algorithm performs checks on the bad requests as soon as they are made. The Chrome DevTools Protocol event which is associated with a request is `Network.requestWillBeSent`, therefore, whenever this Network event is triggered, we analyze the request. However, we found that there are some other events which can also result in a request, such as `Fetch.requestPaused`. The requests generated by these events are successfully captured and logged to the HAR file, but not reported in the MINOS report because they are triggered by a different kind of event. A simple but effective solution would be to move the analysis step after the generation of the HAR object and to directly analyze the entries of the HAR object.

When compared to similar privacy software, we observed that MINOS is the only one that can detect third-countries data transfers: Table I shows a comparison between MINOS and other software. As can be seen, other software addresses other problems, such as the detection of third-party trackers (Privacy Badger, Blacklight, webXray, OpenWPM). It is worth mentioning that even though 2gdpr claims to be able to identify third-countries data transfers, in our tests, it failed to detect them (this is reported in orange in Table I). Some applications adopt a server-based approach (Blacklight, ImmuniWeb, 2gdpr), differently from MINOS (which is serverless).

Finally, some of the considered applications are closed source (ImmuniWeb, 2gdpr), therefore we are not able to compare the approaches of these apps, but only the results.

VI. Conclusion and Future Work

In this paper, we introduced a novel software designed to increase users' awareness of their privacy while navigating the

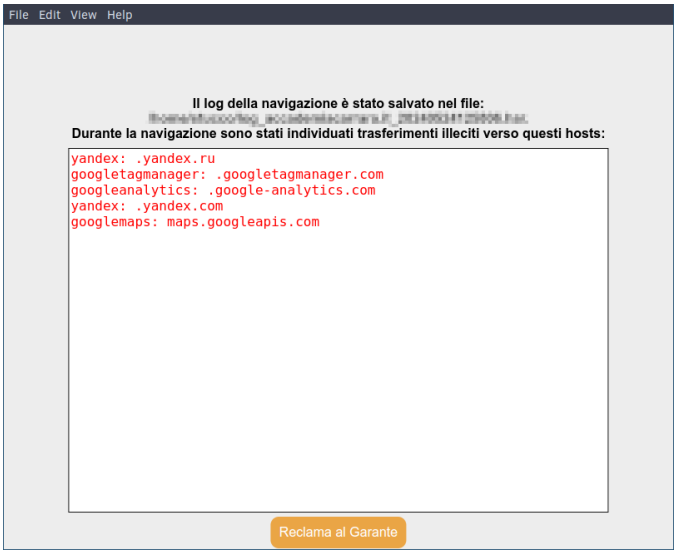

Fig. 4. Report of non-EEA requests to www.accademiacarrara.it

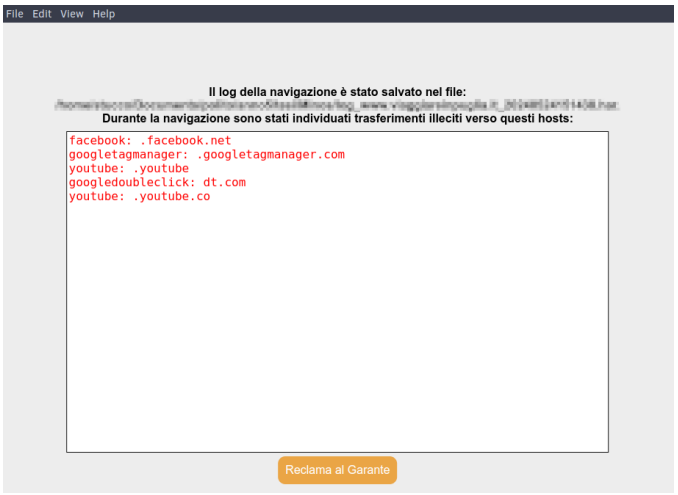

Fig. 5. Report of non-EEA requests to www.viaggiareinpuglia.it

web. Our solution integrates a user-friendly web browser with an analysis-targeted backend. The tool is capable of recording and analysing HTTP requests made to countries which are located outside of the European Economic Area. Compared to similar privacy software, our solution is the only one that focuses on the problem of third-countries data transfers.

The preliminary evaluation of our software through simulations has shown promising results in terms of effectiveness, showing that MINOS can detect the requests to bad hosts as specified in the functional requirements. The positive initial findings demonstrate the potential of MINOS as an analysis tool targeted to non-expert users, and justify the necessity for additional development and comprehensive simulations on a wider range of cases. The conduction of additional evaluations is also necessary to enhance the external validity of the present study, by taking into consideration additional types of PA websites in other European countries.

There are several possible improvements in terms of further research and development. Firstly, the capability to detect requests sent by several events. Another improvement of the detection algorithm would be to inspect also the origin of the cookies while navigating the websites. It could also be possible to dynamically update the blacklist, even though this would require a server-based approach. Regarding user interaction, future work could be to increase users' engagement with the adoption of gamification practices. In such an area, another possible improvement would be to integrate MINOS into other existing web browsers as a browser extension, to increase its spread through the public and make it more accessible.

Acknowledgment

The authors want to express their sincere gratitude to Giacomo Tesio and Massimo Maria Ghisalberti, who have followed the development of MINOS from the start to finish, and Marco Ciurcina, whose contribution has been crucial in understanding the legal concepts presented.

References

[1] S. Zuboff, *The Age of Surveillance Capitalism: The Fight for a Human Future at the New Frontier of Power*. 1st ed., 2018.
[2] T. Guardian, "Edward snowden: the whistleblower behind the nsa surveillance revelations." https://www.theguardian.com/world/2013/jun/09/edward-snowden-nsa-whistleblower-surveillance, 2013.
[3] T. Guardian, "Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach." https://www.theguardian.com/news/2018/mar/17/cambridge-analytica-facebook-influence-us-election, 2018.
[4] A. International, "Israel and occupied Palestinian territories: Automated apartheid: How facial recognition fragments, segregates and controls palestinians in the opt." https://www.amnesty.org/en/documents/mde15/6701/2023/en/, 2023.
[5] T. E. Parliament and the Council of the European Union, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," 2016.
[6] "Privacy badger." https://privacybadger.org/, 2024.
[7] "The markup - blacklight." https://themarkup.org/blacklight, 2024.
[8] "Immuniweb security test." https://www.immuniweb.com/websec/, 2024.
[9] "2gdpr." https://2gdpr.com/, 2024.
[10] "Openwpm." https://github.com/openwpm/OpenWPM, 2024.
[11] "Minos." https://doi.org/10.5281/zenodo.11384690, 2024.
[12] MonitoraPA, "Monitora PA - Osservatorio Automatico Distribuito sulla PA." https://monitora-pa.it/, 2024.
[13] "Chrome har capturer." https://github.com/cyrus-and/chrome-har-capturer, 2023.
[14] "Opendata ipa." https://indicepa.gov.it/ipa-dati/, 2024.