Privacy Flag Project Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployments

Grant Agreement No.653426
Topic: DS-01-2014 (Privacy)
Innovation Action

Deliverable D4.2
Second year report on Technical enablers’ development

Document Number: Privacy Flag/WP4/D4.2
Contractual Date of Delivery: 01.05.2017
Editor: CTI
Work-package: WP4
Distribution / Type: Public (PU) / Report (R)
Version: 1.0
Total Number of Pages: 94

This deliverable has been written in the context of the Privacy Flag Horizon 2020 European research project, which is supported by the European Commission and the Swiss State Secretariat for Education, Research and Innovation. The opinions expressed and arguments employed do not engage the supporting parties.

Deliverable 4.2 (“Second year report on Technical enablers’ development”)
Abstract

Privacy Flag combines crowd sourcing, ICT technology and legal expertise to protect citizen privacy when visiting websites, using smart-phone applications, or living in a smart city leveraging user-friendly solutions provided as a smart phone application, a web browser add-on and a public website. It will:

1. Develop a highly scalable privacy monitoring and protection solution with:
   - Crowd sourcing mechanisms to identify, monitor and assess privacy-related risks;
   - Privacy monitoring agents to identify suspicious activities and applications;
   - Universal Privacy Risk Area Assessment Tool and methodology tailored to European norms on personal data protection;
   - Personal Data Valuation mechanism;
   - Privacy enablers against traffic monitoring and finger printing;
   - User friendly interface informing about the privacy risks when using an application or website.

2. Develop a global knowledge database of identified privacy risks, together with online services to support companies and other stakeholders in becoming privacy-friendly, including:
   - In-depth privacy risk analytical tool and services;
   - Voluntary legally binding mechanism for companies located outside of Europe to align with and abide to European standards in terms of personal data protection;
   - Services for companies interested in being privacy friendly;
   - Labelling and certification process.

3. Collaborate with standardization bodies and actively disseminate towards the public and specialized communities, such as ICT lawyers, policy makers and academics. Eleven (11) European partners, including SMEs and a large telco operator (OTE), bring their complementary technical, legal, societal and business expertise; Privacy-Flag intends to establish strong links with standardization bodies and international fora and it also intends to assess and incorporate outcomes from over 20 related research projects. It will build and ensure long term sustainability and growth.
Executive Summary

This document presents the work under the Work Package (WP) 4 – Technical Enablers. Aim of this WP is to research and develop the required technical enablers and tools for security and privacy that will provide protection mechanisms for users, contributing towards the improvement of privacy protection and risk detection through collective user activities. These will help “infuse” privacy risk awareness as well as privacy risk detection knowledge to users, in order to make them “take a more active role in handling their own privacy”.

Deliverable D4.2 describes WP4’s progress during the second year (Y2) of the project and continues the work presented in D4.1, which has described WP4’s progress during the first year (Y1).

Section 1 describes the purpose of this document and gives a general description of WP4 and of its dedicated tasks.

Section 2 provides the new architecture of the technical enablers and shows the structure of the interactions among them.

Section 3 describes the privacy and security enablers of the Privacy Flag platform. The description includes the updated Top 25 Threat Matrix and its implementation in the platform’s backend, the security and privacy components, how anonymity is handled and the recommended resources for the Privacy Flag users.

Section 4 discusses the Privacy Flag Web browser add-on and its functionality along with its integration with the UPRAAM evaluation methodology, along with critical work performed during the second year and what improvements have been achieved in the usability and User Interface component of the add-on.

Section 5 provides the details of the Smart phone application that alerts users about possible privacy and security risks of applications they use and sites they visit. The section also describes the improved User Interface, new functionalities, and the communication of the application with the backend for obtaining threat related information.

Section 6 discusses a major (and of direct practical value) component of the PrivacyFlag Early Warning System, the PrivacyFlag Observatory. This is hosted in a web site and provides an overview of the privacy and security situation in the Internet. The information it provides (in numerical, text as well as graphic format) is useful to numerous stakeholders such as common users, developers, researchers and enterprises.

Section 7 discusses, briefly, the concept of the Distributed Agents, whose task is to crowdsource information about privacy and security risks. In addition, it implements part of the Top25 Threat Matrix automatic analysis in the front end based on the risk evaluation component.

Section 8 discusses the Evaluation Component and how it deploys the Database in order to obtain information from the Database in order to make it available to other components.

Section 9 discusses the Early Warning System of Privacy Flag, which relies on the Evaluation Component and the Observatory, in order to detect privacy and security outliers (i.e., applications of web sites). This is affected through the use of statistical tools including Peirce’s criterion, Pearson’s correlation, the z-score, and Chebyshev’s Theorem.

Section 10 is focused on one of the most vital components of the Privacy Flag platform, the Database. This stores all crowdsourced and computed information and execute SQL queries sent remotely by all
other Privacy Flag components. The section discusses the Database architecture as it was implemented during the second year of the project in order to incorporate the requirements of the evolved Privacy Flag architecture.

Section 11 discusses the Website and Backend Management component. The latter one was implemented in order to facilitate the presentation to the audience of information from other Privacy Flag platform components and provide assessment result of websites and smartphone application in form of a ranking list.

Section 12 contains the bibliography.
### Version History

**Table 0.1: Version History**

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comments, changes, status</th>
<th>Authors, reviewers</th>
<th>contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>21.02.2017</td>
<td>First Draft</td>
<td>CTI</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>24.02.2017</td>
<td>Browser add-on and application update</td>
<td>Velti</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>09.03.2017</td>
<td>T4.1, Network Anonymity, building blocks, Top25 threats, etc.</td>
<td>UL</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>03.04.2017</td>
<td>Website and backend management platform</td>
<td>DNET</td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>26.04.2017</td>
<td>PF observatory, Threat Detection Mechanisms, Database and Server Implementation, WP4 architecture, Distributed Agents, Evaluation Component, Early Warning System</td>
<td>CTI</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>30.04.2017</td>
<td>Quality check before review round</td>
<td>CTI</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>08.05.2017</td>
<td>Final version based on the reviews of two PF independent reviewers</td>
<td>CTI</td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>09.05.2017</td>
<td>Additional review</td>
<td>UoA</td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>10.05.2017</td>
<td>Full final conceptual and editorial review by the project coordinator</td>
<td>OTE</td>
<td></td>
</tr>
</tbody>
</table>
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Contributors

Table 0.2: Contributors

<table>
<thead>
<tr>
<th>First name</th>
<th>Last name</th>
<th>Partner</th>
<th>E-mail</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sotiris</td>
<td>Nikoletseas</td>
<td>CTI</td>
<td><a href="mailto:nikole@cti.gr">nikole@cti.gr</a></td>
</tr>
<tr>
<td>Vasileios</td>
<td>Vlachos</td>
<td>CTI</td>
<td><a href="mailto:vsvlachos@gmail.com">vsvlachos@gmail.com</a></td>
</tr>
<tr>
<td>Stamatou</td>
<td>Ioannis</td>
<td>CTI</td>
<td><a href="mailto:stamatiu@ceid.upatras.gr">stamatiu@ceid.upatras.gr</a></td>
</tr>
<tr>
<td>Adelina</td>
<td>Madhja</td>
<td>CTI</td>
<td><a href="mailto:madia@ceid.upatras.gr">madia@ceid.upatras.gr</a></td>
</tr>
<tr>
<td>Ioannis</td>
<td>Katsidimas</td>
<td>CTI</td>
<td><a href="mailto:jkatsidima@ceid.upatras.gr">jkatsidima@ceid.upatras.gr</a></td>
</tr>
<tr>
<td>Andreas</td>
<td>Drakos</td>
<td>Velti</td>
<td><a href="mailto:adarakos@velti.com">adarakos@velti.com</a></td>
</tr>
<tr>
<td>Karim</td>
<td>Emara</td>
<td>UL</td>
<td><a href="mailto:karim.emara@uni.lu">karim.emara@uni.lu</a></td>
</tr>
<tr>
<td>Nenad</td>
<td>Gligoric</td>
<td>DNET</td>
<td><a href="mailto:nenad.gligoric@dunavnet.eu">nenad.gligoric@dunavnet.eu</a></td>
</tr>
<tr>
<td>Nancy</td>
<td>Alonistioti</td>
<td>UoA</td>
<td><a href="mailto:nancy@di.uoa.gr">nancy@di.uoa.gr</a></td>
</tr>
<tr>
<td>Alexandros</td>
<td>Kostopoulos</td>
<td>OTE</td>
<td><a href="mailto:alexkosto@oteresearch.gr">alexkosto@oteresearch.gr</a></td>
</tr>
<tr>
<td>Ioannis</td>
<td>Chochliouros</td>
<td>OTE</td>
<td><a href="mailto:ichochliouros@oteresearch.gr">ichochliouros@oteresearch.gr</a></td>
</tr>
</tbody>
</table>
## Glossary

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>2FA</td>
<td>Two-Factor Authentication</td>
</tr>
<tr>
<td>ACM</td>
<td>Association for Computing Machinery</td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>apk</td>
<td>application</td>
</tr>
<tr>
<td>app</td>
<td>application</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous System</td>
</tr>
<tr>
<td>BD</td>
<td>Blue-ray Disc</td>
</tr>
<tr>
<td>CA</td>
<td>certificate authority</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code-Division Multiple Access</td>
</tr>
<tr>
<td>CDN</td>
<td>Content Delivery Network</td>
</tr>
<tr>
<td>CLI</td>
<td>Command-Line interface</td>
</tr>
<tr>
<td>CVE</td>
<td>Common Vulnerabilities and Exposures</td>
</tr>
<tr>
<td>DA</td>
<td>Distributed Agent</td>
</tr>
<tr>
<td>DB</td>
<td>Data Base</td>
</tr>
<tr>
<td>DF</td>
<td>Diffie-Hellmann</td>
</tr>
<tr>
<td>DNS</td>
<td>Domain Name System</td>
</tr>
<tr>
<td>DNSSEC</td>
<td>Domain Name System Security Extensions</td>
</tr>
<tr>
<td>DSA</td>
<td>Digital Signature Algorithm</td>
</tr>
<tr>
<td>E2E</td>
<td>End-to-End</td>
</tr>
<tr>
<td>EC</td>
<td>European Commission</td>
</tr>
<tr>
<td>EFF</td>
<td>Electronic Frontier Foundation</td>
</tr>
<tr>
<td>EU</td>
<td>European Union</td>
</tr>
<tr>
<td>EWS</td>
<td>Early Warning System</td>
</tr>
<tr>
<td>EXP</td>
<td>Export</td>
</tr>
<tr>
<td>FA</td>
<td>Factor Authentication</td>
</tr>
<tr>
<td>FI</td>
<td>Future Internet</td>
</tr>
<tr>
<td>FIA</td>
<td>Future Internet Architecture</td>
</tr>
<tr>
<td>GA</td>
<td>Grant Agreement</td>
</tr>
<tr>
<td>GPS</td>
<td>Global Positioning System</td>
</tr>
<tr>
<td>GSM</td>
<td>Global System for Mobile Communications</td>
</tr>
<tr>
<td>HPKP</td>
<td>HTTP Public Key Pinning</td>
</tr>
<tr>
<td>HSTS</td>
<td>HTTP Strict Transport Security</td>
</tr>
<tr>
<td>HTML5</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>Hypertext Transfer Protocol Secure</td>
</tr>
<tr>
<td>ICT</td>
<td>Information and Communication Technologies</td>
</tr>
<tr>
<td>ID, id</td>
<td>Identifier</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messaging</td>
</tr>
<tr>
<td>IoT</td>
<td>Internet of Things</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>LSO</td>
<td>Local Shared Objects</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MD</td>
<td>Message Digest</td>
</tr>
<tr>
<td>MIT</td>
<td>Massachusetts Institute of Technology</td>
</tr>
<tr>
<td>MITM</td>
<td>man-in-the-middle attack</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>NMAP</td>
<td>Network Mapper</td>
</tr>
<tr>
<td>OCSP</td>
<td>Online Certificate Status Protocol</td>
</tr>
<tr>
<td>OFTE</td>
<td>on-the-fly encryption</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System</td>
</tr>
<tr>
<td>P3P</td>
<td>Privacy Preferences Project</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>PET</td>
<td>Privacy Enhancing Technology</td>
</tr>
<tr>
<td>PF</td>
<td>Privacy Flag</td>
</tr>
<tr>
<td>PGP</td>
<td>Pretty Good Privacy</td>
</tr>
<tr>
<td>PHP</td>
<td>Hypertext Preprocessor</td>
</tr>
<tr>
<td>PK</td>
<td>Public Key</td>
</tr>
<tr>
<td>PoC</td>
<td>Proof-of-Concept</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RC</td>
<td>Rivest Cipher</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio-Frequency Identification</td>
</tr>
<tr>
<td>RTC</td>
<td>Real Time Communications</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>SHA</td>
<td>Secure Hash Algorithm</td>
</tr>
<tr>
<td>SIGSAG</td>
<td>Special Interest Group on Security, Audit and Control</td>
</tr>
<tr>
<td>SME</td>
<td>Small- and Medium-sized Enterprise</td>
</tr>
<tr>
<td>SMS</td>
<td>Short Message Service</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SSL</td>
<td>Secure Sockets Layer</td>
</tr>
<tr>
<td>STD</td>
<td>standard deviation</td>
</tr>
<tr>
<td>SVG</td>
<td>Scalable Vector Graphics</td>
</tr>
<tr>
<td>Sync</td>
<td>synchronization</td>
</tr>
<tr>
<td>TBB</td>
<td>Tor Browser Bundle</td>
</tr>
<tr>
<td>TLD</td>
<td>Top-Level Domain</td>
</tr>
<tr>
<td>TLS</td>
<td>Transport Layer Security</td>
</tr>
<tr>
<td>Tor, TOR</td>
<td>The Onion Ring</td>
</tr>
<tr>
<td>UI</td>
<td>User Interface</td>
</tr>
<tr>
<td>UL</td>
<td>Uplink</td>
</tr>
<tr>
<td>UPRAAM</td>
<td>Universal Privacy Risk Area Assessment Methodology</td>
</tr>
<tr>
<td>UPRAAT</td>
<td>Universal Privacy Risk Area Assessment Tool</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WebRTC</td>
<td>Web Real Time Communications</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>Wireless Fidelity</td>
</tr>
</tbody>
</table>

Deliverable 4.2 (“Second year report on Technical enablers’ development”)
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WOT, WoT</td>
<td>Web of Trust</td>
</tr>
<tr>
<td>WP</td>
<td>Work Package</td>
</tr>
<tr>
<td>www</td>
<td>World Wide Web</td>
</tr>
<tr>
<td>XML</td>
<td>Extended Markup Language</td>
</tr>
</tbody>
</table>
Table of Contents

1. INTRODUCTION .................................................................................................................. 15
   1.1 PURPOSE AND SCOPE OF WP4 .................................................................................. 15
   1.2 PURPOSE AND SCOPE OF THE CURRENT DOCUMENT ............................................ 15

2. GENERAL ARCHITECTURE .................................................................................................... 17

3. PRIVACY FLAG ENABLERS .................................................................................................. 19
   3.1 MODERN TOP 25 THREATS – THREAT MATRIX ....................................................... 19
   3.1.1 Automatic Threat Detection Mechanisms .................................................................. 19
   3.1.1.1 Threat Detection Mechanisms implemented in the Backend ............................. 19
   3.1.1.2 Threat Detection Mechanisms implemented in the Frontend (Add-on) .......... 26
   3.2 SECURITY AND PRIVACY COMPONENTS ............................................................... 29
   3.2.1 Securing Backend Server ....................................................................................... 29
   3.2.2 Securing Communication ....................................................................................... 29
   3.2.3 Browser Add-on Code Sample ............................................................................... 30
   3.2.4 Smartphone App Code Sample ............................................................................. 31
   3.2.5 Anonymizing Exchanged Data ............................................................................... 33
   3.3 RECOMMENDED RESOURCES FOR USERS ............................................................. 34
   3.3.1 Web browsing ......................................................................................................... 34
   3.3.2 Virtual Private Network (VPN) ............................................................................... 36
   3.3.3 Email service and client ......................................................................................... 37
   3.3.4 Search Engines ...................................................................................................... 38
   3.3.5 Instant Messaging (IM) and VoIP Software ......................................................... 38
   3.3.6 Encrypted Cloud Storage ........................................................................................ 39
   3.3.7 Disk and File Encryption ....................................................................................... 39
   3.3.8 Password Management .......................................................................................... 40
   3.3.9 Two-Factor Authentication (2FA) .......................................................................... 40
   3.3.10 Additional Resources ............................................................................................ 41
   3.4 NETWORK ANONYMITY ............................................................................................... 41
   3.4.1 Overview of Privacy Enhancing Technologies ...................................................... 41
   3.4.2 Latency Enhancement ............................................................................................. 42

4. PRIVACY FLAG WEB BROWSER ADD-ON ......................................................................... 46
   4.1 MAIN ARCHITECTURE DESCRIPTION ....................................................................... 46
   4.2 BROWSER ADD-ON FUNCTIONALITY ....................................................................... 47
   4.2.1 Updated Functionality Workflow ........................................................................... 47
   4.2.2 Privacy Enablers Integration .................................................................................. 48
   4.2.3 UPRAAM Integration ............................................................................................... 48
   4.3 YEAR 2 DEVELOPMENT OVERVIEW ....................................................................... 48
   4.3.1 UI updates .............................................................................................................. 48
   4.3.2 Backend communication ........................................................................................ 50
   4.3.3 Browser add-on files and source code ................................................................... 51
5. PRIVACY FLAG SMARTPHONE APPLICATION ................................................................. 52
   5.1 ARCHITECTURE ........................................................................................................ 52
   5.2 MOBILE APPLICATION FUNCTIONALITY .................................................................. 53
      5.2.1 Updated Functionality Workflow ........................................................................ 53
      5.2.2 Privacy Enablers Integration ................................................................................ 53
      5.2.3 UPRAAM Integration ......................................................................................... 54
   5.3 YEAR 2 DEVELOPMENT OVERVIEW ........................................................................... 54
      5.3.1 UI Updates – App Version 2 ............................................................................... 54
      5.3.2 Automated Evaluation, App Permissions – App Version 2 .................................... 56
      5.3.3 Backend Communication – App Version 2 .......................................................... 56
      5.3.4 UI Updates – App Version 3 ............................................................................... 56
   5.4 SMARTPHONE APPLICATION FILES AND SOURCE CODE ........................................ 59
6. PRIVACY FLAG OBSERVATORY ....................................................................................... 60
   6.1 INTRODUCTION .......................................................................................................... 60
   6.2 DEVELOPMENT OVERVIEW ..................................................................................... 61
      6.2.1 PrivacyFlag Observatory Website – Version 1 .................................................... 61
      6.2.2 PrivacyFlag Observatory Website – Version 2 .................................................... 62
      6.2.3 PrivacyFlag Observatory Website – Version 3 .................................................... 65
      6.2.4 Categories of Charts ........................................................................................... 66
         6.2.4.1 Confidentiality ................................................................................................. 66
         6.2.4.2 Security ............................................................................................................ 68
         6.2.4.3 Privacy ............................................................................................................. 74
7. DISTRIBUTED AGENTS ................................................................................................... 77
8. EVALUATION COMPONENT ............................................................................................ 78
9. EARLY WARNING SYSTEM ............................................................................................ 80
10. DATABASE AND SERVER IMPLEMENTATION .............................................................. 82
    10.1 DATABASE SCHEMA .................................................................................................. 82
        10.1.1 Database schema for the PF add-on ............................................................... 82
        10.1.2 Database schema for the PF smartphone application ...................................... 84
    10.2 GIT SERVER .............................................................................................................. 86
    10.3 APIs .......................................................................................................................... 86
11. WEBSITE AND BACKEND MANAGEMENT PLATFORM .............................................. 87
    11.1 FRONTEND ............................................................................................................... 87
    11.2 BACKEND ............................................................................................................... 90
    11.3 ARCHITECTURE ...................................................................................................... 91
12. LIST OF REFERENCES .................................................................................................... 93
List of Tables
Table 5-1: Pilot comments on App version 2 and performed actions........................................... 56
List of Figures

Figure 2-1: Depiction of the Privacy Flag architecture 18
Figure 3-1: Browser configuration to use Tor 43
Figure 3-2: CDF description that contains all 1000 measured time values per routing method 44
Figure 3-3: Display of the loading time as the mean of the ten fetching instances per website 45
Figure 4-1: The Privacy Flag Web Browser Add-on Architecture 46
Figure 4-2: The Privacy Flag Web Browser Add-on in action 48
Figure 4-3: The Privacy Browser Add-on UI: initial screen (a); UPRAAM evaluation forms based on task T3.2 (b-f), and; submission of evaluation screen (z) 50
Figure 5-1: The Privacy Flag Smartphone Application Architecture 52
Figure 5-2: The Privacy Flag Smartphone Application version 2 intro flash screen (a); setting user preferences (b); main menu (c); viewing specific applications with existing evaluation (d-e-f), and; providing user evaluation (g-z) 55
Figure 5-3: The Privacy Flag Smartphone Application version 3 with new listing of the apps(a) showing icons in the list of permissions and specific message when android version is lower to 6 (b-c) updated information page (d) and error message (e) 58
Figure 6-1: Privacy Flag Observatory - Version 1 62
Figure 6-2: Privacy Flag Observatory - Version 2 64
Figure 6-3: Privacy Flag Observatory - Version 3 66
Figure 6-4: Percentage of websites employing SSL/TLS 67
Figure 6-5: Percentage of websites providing HSTS 67
Figure 6-6: Percentage of sites using a valid certification chain 68
Figure 6-7: Percentage of websites using certificate pinning 68
Figure 6-8: Average number of links to malicious sites 69
Figure 6-9: Percentage of websites using technologies with known security issue 69
Figure 6-10: Percentage of websites using potentially risky advanced HTML5 APIs 70
Figure 6-11: Percentage of websites using following techniques 70
Figure 6-12: Percentage of websites using Web SQL Database or HTML5 Indexed DB 71
Figure 6-13: Percentage of websites using technologies with known issues – Flash 71
Figure 6-14: Percentage of websites using potentially dangerous advanced HTML5 APIs – WEBRTC 72
Figure 6-15: Percentage of websites using potentially dangerous advanced HTML5 APIs – Web Audio API 72
Figure 6-16: Percentage of websites using technologies with known security issues – ActiveX 73
Figure 6-17: Percentage of websites using technologies with known security issues – Java 73
Figure 6-18: Percentage of websites using technologies with known security issues – Silverlight 74
Figure 6-19: Average number of HTTP Cookies over time 74
Figure 6-20: Mean number of various cookies in all websites 75
Figure 6-21: First and second interval of the change of HTTP Cookies in all websites over time 75
Figure 6-22: The total amount of information currently processed 76
Figure 10-1: Database schema for the PF Add-on 83
Figure 10-2: Database schema for the PF smartphone application 85
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Figure 11-1: Websites risk assessment list on the website 88
Figure 11-2: User engagement 89
Figure 11-3: Privacy Flag End-user Tools presentation 90
Figure 11-4: The privacy Flag Website and backend management platform architecture 92
1. Introduction

1.1 Purpose and scope of WP4

The purpose of WP4 is to design and implement the mechanisms and enablers that will ensure the privacy of prospective users as well as enabling them to monitor and report privacy threats detected on their smartphone and in their web browsers.

Main objectives of this work package are to:

- Provide privacy protection mechanisms for users.
- Contribute towards the improvement of privacy protection and risk detection through collective user activities and reporting.
- Help infuse privacy risk awareness as well as privacy risk detection knowledge to users, in order to make them take a more active role in handling their own, as well as others’, privacy.

WP4 consists of the following tasks:

- **T4.1** will research, analyze and design the required security and privacy enablers for the Privacy Flag platform and users. It will identify existing security solutions and resources to be used by the platform. Additionally, T4.1 will develop complementary building blocks to support the platform in order to fill the holes and provide a list of recommended resources for the Privacy Flag users.
- **T4.2** will design and implement an advanced Evaluation Component that will provide access to the PrivacyFlag Observatory and basic Early Warning Services. The PrivacyFlag Observatory utilizes multiple sources of input to offer timely and valid information on possible privacy threats.
- **T4.3** will design and develop a web browser add-on that will allow users to get information on potential privacy risks when browsing throughout the Internet. The add-on informs users whether a web site is considered as privacy friendly -or not friendly- based on the analysis conducted by the Privacy Flag backend system.
- **T4.4** will study and implement a smartphone application that will inform users on the potential privacy risks related to their installed applications. It will also provide the evaluation tools collection crowd sourcing inputs.
- **T4.5** will provide the underlying server infrastructure that will support the rest of the Privacy Flag tools, as well as the required database that will store the collected data from the various tools.
- **T4.6** will create the backend management platform together with the corresponding web site and the underlying database.

1.2 Purpose and scope of the current document

This document presents the progress and achievements of WP4 within the context of the second year (Y2) of the Privacy Flag project. It focuses upon the modification and enhancements made to the technical enabler components to accommodate the new Privacy Architecture and the UPRAAAM methodology as well as the current Database schema and its automatic risk evaluation capabilities, which support the Privacy Flag Early Warning System. The components and APIs described in this
document have been successfully integrated into a fully working pilot system that has already been activated in the framework of the pilot operation in the context of WP5. Major improvements over the first year (Y1) include the privacy and anonymity components, the integration of the UPRAAM methodology into the smartphone application, the development of the database, the finalization of the Top 25 Threat Matrix and the detection of outliers. Work on these and other WP4 issues is ongoing and will be described in the Deliverable D4.3.
2. General Architecture

Privacy Flag’s architecture empowers people to monitor and control their privacy, themselves, with a user-centered solution equipped with friendly interfaces.

As seen in the figure below, the Privacy Flag architecture consists, on the users’ side, of a smartphone application and a mobile phone web browser add-on through which a user may submit evaluations with respect to the privacy of the visited sites and downloaded applications.

The Distributed Agents (DAs) running in the background, produce automatically evaluations of the sites and applications.

In the center of the architecture lies the Database which contains all the collected and derived information coming from users and the distributed agents, in real time. All the PF components communicate with this database containing information about privacy breach incidents, privacy violating app, web sites and services as well as possible threat combinations which, when they coexist, impose severe privacy risk for the user.

The PF components running on the users’ smartphones continuously monitor the installed applications as well as users’ connections to websites in order to detect -using also the experts’ advice coded in the database- unwanted external access to users’ data. Upon detection of such privacy breach attempts, the central database is notified so as to update itself with a, possibly, new identified privacy threat as well as send back to the enabler advice directed to the user of how the privacy breach can be avoided.
In parallel, upon detection of security threats, the Distributed Agents activate the Early Warning System (EWS) notifying users about the threats and suggesting to them actions in order to avoid security and privacy issues.

Finally, the Privacy Flag Observatory publishes all information stored and computed in the database in a text as well graphical form in order to provide useful information to the public and the experts with respect to the latest security and privacy breach threats.

In the following sections, we describe the work performed on the architecture’s components, i.e. the Privacy Flag enablers, during the second year of the project in the context of WP4.
3. Privacy Flag enablers

3.1 Modern Top 25 Threats – Threat Matrix

Several security and privacy threats were identified in Year 1 of the project and discussed in Section 2.1 of the Deliverable D4.1. Automatic checks for these threats have been implemented either in the backend or the browser add-on. The selection is made based on the requirement of the threat and upon the condition whether the automatic checks can be implemented in the client side. If the latter not possible then the automatic checks are directly implemented in the backend, taking the advantage of the implementation flexibility and powerfulness of the server.

3.1.1 Automatic Threat Detection Mechanisms

3.1.1.1 Threat Detection Mechanisms implemented in the Backend

The following list of threats presents the ones implemented in the backend.

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website provide HSTS?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Websites attacks: protocol downgrade attacks and cookie hijacking</td>
</tr>
<tr>
<td>High-level Description</td>
<td>HTTP Strict Transport Security (HSTS) is a web security policy mechanism that allows web servers to declare that web browsers should only interact with it using secure HTTPS connections and never via the insecure HTTP protocol. Additionally, the lifetime and public key pins can be defined by the server that the client browser will cache. From now on, the browser will refuse to connect via HTTP or via HTTPS in case none of the public key pins is part of the certificate chain. This hinders Man-in-the-middle attacks.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Confidentiality of communications</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that takes the URL as input and reads the HTTP headers. If “Strict-Transport-Security” header exists and the “max-age” is well configured (i.e., &gt; 30s), then the script returns true, otherwise it returns false</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website use a trustworthy certification chain?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>The server may use a certificate that is issued by untrusted or compromised CA or the attacker tries to impersonate the genuine server.</td>
</tr>
</tbody>
</table>

1 For more details about this deliverables, see: http://privacyflag.eu/wp-content/uploads/2017/03/Privacy_Flag_Deliverable_204.1_final_v1.0.pdf
2 More information can be found, inter-alia, at: https://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security
3 A man-in-the-middle attack (MITM), sometimes called a "bucket brigade attack") is an attack where the attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other. More information can be found, for example, at: https://en.wikipedia.org/wiki/Man-in-the-middle_attack
### High-level Description

In order for an SSL certificate to be trusted, it must have been issued by a certificate authority (CA) that is trusted by the connecting client (web browser). If the certificate was not issued by a trusted known CA, the client will then check to see if the certificate of the issuing CA was issued by a known trusted CA, and so on until either a trusted CA can or cannot be found. The list of SSL certificates, from the root certificate to the website certificate, represents the SSL certificate chain.

### Threat Category

Confidentiality of communications

### Implementation Details

This threat is implemented as a backend script that takes the URL as input and analyses the server certificate. First, it obtains the server certificate and its issuer certificate and uses OpenSSL library to verify them using CA certificates obtained from a trusted source. It also verifies that all certificates are not revoked by check the specified OCSP URI in the HTTPS header.

### Return Value

True/False

---

### Name

Does the website use Certificate pinning? (HTTP public key pinning)

### Threat

Website impersonation

### High-level Description

HTTP Public Key Pinning (HPKP) is a security mechanism which allows HTTPS websites to resist impersonation by attackers using miss-issued or fraudulent certificates. For example, attackers might compromise a certificate authority (i.e., the entity that issues soft authentication certificates for websites) and then miss-issue certificates for any domain. To combat this risk, the webserver can provide a list of “pinned” public key hashes; on subsequent connections web browsers expect that server to use one or more of those public keys in its certificate chain.

### Threat Category

Confidentiality of communications

### Implementation Details

This threat is implemented as a backend script that takes the URL as input and reads the HTTP headers. If “public-key-pins” header exists, the “max-age” is well configured (i.e., > 30 s) and there is a valid SHA256 hash for the public key, then the script returns true, otherwise it returns false

### Return Value

True/False

---

4 In cryptography, a certificate authority (CA) is an entity that issues digital certificates. A digital certificate certifies the ownership of a public key by the named subject of the certificate. This allows others (relying parties) to rely upon signatures or on assertions made about the private key that corresponds to the certified public key.

5 More general information can be found, for example, at: [https://support.dnsimple.com/articles/what-is-ssl-certificate-chain/](https://support.dnsimple.com/articles/what-is-ssl-certificate-chain/)

6 OpenSSL is an open source project that provides a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. For more details see, for example: [https://www.openssl.org/](https://www.openssl.org/)

7 For more related information see, for example: [https://www.openssl.org/docs/man1.0.2/apps/x509v3_config.html](https://www.openssl.org/docs/man1.0.2/apps/x509v3_config.html)

8 More information can be found, inter-alia, at: [https://en.wikipedia.org/wiki/HTTP_Public_Key_Pinning](https://en.wikipedia.org/wiki/HTTP_Public_Key_Pinning)

9 Secure Hash Algorithms (SHA) are a family of cryptographic hash functions published by the National Institute of Standards and Technology (NIST). In particular, SHA-2 is a family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size, that is: SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. More related and generalized information can be found, for example, at: [https://en.wikipedia.org/wiki/Secure_Hash_Algorithms](https://en.wikipedia.org/wiki/Secure_Hash_Algorithms)
<table>
<thead>
<tr>
<th>Name</th>
<th>Is backward compatibility with unsecure SSL or TLS versions disabled?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>HTTPS traffic encrypted using weak ciphers is prone to be decrypted by an eavesdropper</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Transport Layer Security(^\text{10}) (TLS) and its predecessor, Secure Sockets Layer (SSL) are standard security technologies for establishing an encrypted link between a server and a client—typically a web server (website) and a browser. The strength of the protection mechanism is determined by the authentication, encryption, and hashing algorithms, collectively known as a cipher suite, chosen for the transmission of sensitive information over the TLS/SSL channel. There are some cipher suites known to be broken or weak. SSL(-\text{enabled}) servers should be configured to disable these insecure cipher suites.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Confidentiality of communications</td>
</tr>
</tbody>
</table>
| Implementation Details | This threat is implemented as a backend script that takes the URL as input and uses 3\(^\text{rd}\) party tools to analyse the cipher suites supported by the server. First, if the server supports SSL v2 or v3, then the script returns false directly because SSL is considered obsolete. Otherwise, the automatic check uses NMAP\(^\text{11}\) script that ranks the strength of each cipher suite supported by the server. This script iterates on a large list of ranked cipher suites and negotiates with the server in a handshake. Then, the minimum rank of all accepted cipher suites is obtained. If it is higher than a specific threshold, the API returns true, otherwise it returns false. The following cipher suites list is considered weak and rejected by the API:  
  - NULL ciphers suite (no encryption)  
  - Cipher suites with Export (EXP) level (can be easily broken)  
  - MD5\(^\text{12}\) (collision attacks)  
  - RC4\(^\text{13}\) (crypto-analytical attacks) |
| Return Value | True/False |

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the encrypted user traffic is prone to fingerprinting?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Even if the user navigates the web anonymously through the Tor(^\text{14}) network, the related traffic is prone to fingerprinting (i.e., revealing the website the user visits). This fingerprinting is possible due to the uniqueness of traffic features from a website to another.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Based on a large-scale experiment of website fingerprinting over Tor network, we built a generic assessment tool that gives a vulnerability score for a large list of websites. Using the most visited 1000 websites (according to Alexa(^\text{15}) ranking), we recorded 40 instances from its web traffic using 20 lab machines over several days. Using this big traffic data, we trained a generic assessment model which gives a score if the anonymous traffic to a given website is vulnerable to website fingerprinting</td>
</tr>
</tbody>
</table>

\(^{10}\) More information can be found, inter-alia, at: [https://en.wikipedia.org/wiki/Transport_Layer_Security](https://en.wikipedia.org/wiki/Transport_Layer_Security) \(^{11}\) More information about Network Mapper (NMAP) can be found, inter-alia, at: [https://en.wikipedia.org/wiki/Nmap](https://en.wikipedia.org/wiki/Nmap) \(^{12}\) Also see, for example: [https://en.wikipedia.org/wiki/MD5](https://en.wikipedia.org/wiki/MD5) \(^{13}\) Also see, for example: [https://en.wikipedia.org/wiki/RC4](https://en.wikipedia.org/wiki/RC4) \(^{14}\) More information about Tor (The Onion Ring) project can be found at: [https://www.torproject.org/](https://www.torproject.org/) \(^{15}\) More information about Alexa can come by considering: [http://www.alexa.com/topsites](http://www.alexa.com/topsites)
<table>
<thead>
<tr>
<th>Threat Category</th>
<th>Privacy compromising mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Details</td>
<td>Currently the scores for 1000 websites are stored in a database and fetched for each requested URL. We are planning</td>
</tr>
<tr>
<td>Return Value</td>
<td>Score (percentage; 0-100) and Rank (1 – 3, 1 means low vulnerability while 3 mean high vulnerability)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Is DNSSEC enabled?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>DNS cached records can be manipulated by an attacker which redirects the user to malicious servers.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Domain Name System(^{16}) (DNS) is the system that converts human readable addresses (e.g., <a href="http://www.google.com">www.google.com</a>) to machine readable addresses (i.e., IP address). DNSSEC(^{17}) was designed to protect applications from using forged or manipulated DNS data, such as that created by DNS cache poisoning. All answers from DNSSEC protected zones are digitally signed. By checking the digital signature, a DNS client is able to check if the information is identical (i.e. unmodified and complete) to the information published by the zone owner and served on an authoritative DNS server.</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website comply with the P3P privacy policy?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>The website privacy policy is not</td>
</tr>
<tr>
<td>High-level Description</td>
<td>The Platform for Privacy Preferences Project(^{18}) (P3P) enables websites to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by browsers. Websites implementing such policies make their practices explicit and thus open them to public scrutiny.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Privacy compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This API uses 3rd party tool “dig” to do this check which verifies if there is a digital signature for the domain name of the given URL</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

\(^{16}\) More information can be found, \emph{inter-alia}, at: \url{https://en.wikipedia.org/wiki/Domain_Name_System}

\(^{17}\) For more details also see, \emph{inter-alia}: \url{https://en.wikipedia.org/wiki/Domain_Name_System_SecurityExtensions}

\(^{18}\) The platform for Privacy Preference Project (P3P) is a protocol allowing websites to declare their intended use of information they collect about web browser users. For more details also see: users \url{http://www.w3.org/P3P/}

\(^{19}\) The World Wide Web Consortium (W3C) is an international community where Member organizations, a full-time staff, and the public work together to develop Web standards. More relevant information about W3C’s activities can be found at: \url{http://www.w3.org/}

Deliverable 4.2 (“Second year report on Technical enablers’ development”)
<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Is the encryption method (cipher suite) negotiated between client and server considered as secure?</td>
</tr>
<tr>
<td>Threat</td>
<td>Weak or insecure cipher suite may let an eavesdropper to reveal the user information.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Old browser or misconfigured web server may support obsolete or weak cipher suites. This API should verify if the negotiated cipher suite between the browser and the server of the currently visited website is considered secure.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Confidentiality of communications</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>N/A</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>It is figured out it is not possible to gain information about the negotiated cipher suite from the browser add-on. Also, it does not make sense to perform this check from backend because we are not aware about the cipher suites supported by the user’s browser. Thus, this threat will be replaced by another and UL will develop its API.</td>
</tr>
<tr>
<td>Return Value</td>
<td>N/A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Does the website use technologies with known security issues - PDF?</td>
</tr>
<tr>
<td>Threat</td>
<td>When you open a new file that has active content (data connections, macros, and so on), it may contain viruses and other security hazards that could harm your computer or your organization’s network.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>If a malicious macro is loaded into an Office application like Word via an infected document, it can use features like “AutoExec” to automatically start with Word or “AutoOpen” to automatically run whenever you open a document. In this way, the macro virus can integrate itself into Word, infecting future documents.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Security compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If files of this kind exist then the script returns true, otherwise it returns false.</td>
</tr>
<tr>
<td>Return Value</td>
<td>Integer</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Does the website use technologies with known security issues – notably, Flash?</td>
</tr>
<tr>
<td>Threat</td>
<td>Known bugs in add-ons causing malicious agents to take control of users’ devices.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Flash (the following also hold for other similar add-ons), which is commonly used add-on in browsers and websites, runs inside the same process and memory as the web browser; but frequent bugs in this software give hackers lots of opportunities to gain access to devices’ memory. When they do that, they can cause the browser to jump to a specific memory address and take control of the device. The defense against this threat relies on the identification of its source code pattern.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Security compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If the pattern for this particular threat exists then the script returns true, otherwise it returns false.</td>
</tr>
</tbody>
</table>

Deliverable 4.2 ("Second year report on Technical enablers’ development")
<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website contain links to malicious sites (Google’s Safebrowsing API(^{20}))?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Unsafe web resources like social engineering sites (phishing and deceptive sites) and sites that host malware or unwanted software.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Hacked sites contain hidden links to malicious sites which can inject malicious code to a user’s devices upon, unwittingly, visiting them.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Security compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that calls Google’s Safe browsing API which has the URL as input. Depending on the API’s output the script returns true if the webpage is safe, otherwise it returns false. Safe Browsing is a Google service that lets client applications check URLs against Google’s constantly updated lists of unsafe web resources.</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website use potentially dangerous advanced HTML5(^{21}) APIs: Web Audio API(^{22})?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Users are exposed to unauthorized recordings.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>The Web Audio API involves handling audio operations inside an audio context, and has been designed to allow modular routing. Basic audio operations are performed with audio nodes, which are linked together to form an audio routing graph. Several sources -with different types of channel layout- are supported even within a single context. This modular design provides the flexibility to create complex audio functions with dynamic effects.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Privacy compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If the functions used for this particular technology exist then the script returns true, otherwise it returns false.</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website use potentially dangerous advanced HTML5 APIs: WebRTC?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>HTML5-WebRTC is a very effective mechanism for providing real-time communication but it is also used by hackers to leak sensitive information or de-anonymize users.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>WebRTC (Web Real-Time Communication)(^{23}) is a collection of communications protocols and application programming interfaces that enable real-time communication over peer-to-peer connections. This allows web browsers to not only request resources from backend servers, but also real-time information from browsers of other users. This enables applications such as...</td>
</tr>
</tbody>
</table>

\(^{20}\) For more details, also see: [https://developers.google.com/safe-browsing/v4/](https://developers.google.com/safe-browsing/v4/)

\(^{21}\) For more details also see, among others: [https://en.wikipedia.org/wiki/HTML5](https://en.wikipedia.org/wiki/HTML5)

\(^{22}\) Also see: [https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API)

\(^{23}\) For more details also see: [https://webrtc.org/](https://webrtc.org/)

Deliverable 4.2 (“Second year report on Technical enablers’ development”)
<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website use potentially dangerous advanced HTML5 APIs: Geolocation (GPS)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>The HTML5 Geolocation API realizes location-based services via the Web by granting web sites the geographical location information of user devices. However, the Geolocation API can violate a user’s location privacy due to its coarse-grained permission and location models.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>The W3C Geolocation API is an effort by the World Wide Web Consortium (W3C) to standardize an interface to retrieve the geographical location information for a client-side device. The most common sources of location information are IP address, Wi-Fi and Bluetooth MAC address, radio-frequency identification (RFID), Wi-Fi connection location, or device Global Positioning System (GPS) and GSM/CDMA cell IDs. The location is returned with a given accuracy depending on the best location information source available.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Privacy compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If the functions used for this particular technology exist then the script returns true, otherwise it returns false.</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website use technologies with known security issues - ActiveX?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>The usage of this particular obsolete technology indicates security and privacy issues to the user’s system. They can stop a computer from functioning correctly, collect your browsing habits and personal information without your knowledge, or can give you content, like pop-up ads that you do not want. Also, ActiveX controls might contain unintended code that allows &quot;bad&quot; websites to use them for malicious purposes.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>ActiveX is a Microsoft technology supported in older Microsoft browsers. It can be used to build complex scripts to automate many tasks. ActiveX normally operates from the web site directly to the users systems. Therefore, many security issues arise from this approach. Even Microsoft has discontinued ActiveX.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Security and privacy compromising mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If the pattern for this particular threat exists then the script returns true, otherwise it returns false.</td>
</tr>
</tbody>
</table>

---

24 For more details also see: https://www.sitepoint.com/html5-geolocation/
25 Also see, for example: https://en.wikipedia.org/wiki/ActiveX
### Return Value

<table>
<thead>
<tr>
<th>Name</th>
<th>True/False</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Security compromising mechanisms</td>
</tr>
</tbody>
</table>

### High-level Description

An applet is a small Internet-based program written in Java, a programming language for the Web, which can be downloaded by any computer. The applet is usually embedded in an HTML page on a Web site and can be executed from within a browser.

### Implementation Details

This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If a java applet or the object constructed for this particular technology exist then the script returns true, otherwise it returns false.

### Return Value

<table>
<thead>
<tr>
<th>True/False</th>
</tr>
</thead>
</table>

### Name

Does the website use technologies with known security issues - Silverlight?

### Threat

The usage of this particular obsolete technology indicates security bugs and privacy issues to the user’s system like exposing private e-mails and other data.

### High-level Description

Silverlight is a deprecated application framework for writing and running rich Internet applications, similar to Adobe Flash. While early versions of Silverlight focused on streaming media, later versions supported multimedia, graphics, and animation and gave developers support for CLI languages and development tools. Silverlight is also one of the two application development platforms for Windows Phone, but web pages that use Silverlight cannot run on the Windows Phone or Windows Mobile versions of Internet Explorer, as there is no Silverlight plugin for Internet Explorer on those platforms.

### Implementation Details

This threat is implemented as a backend script that takes the URL as input and scans the source code of the web page. If the pattern for this particular threat exists then the script returns true, otherwise it returns false.

### Return Value

<table>
<thead>
<tr>
<th>True/False</th>
</tr>
</thead>
</table>

### 3.1.1.2 Threat Detection Mechanisms implemented in the Frontend (Add-on)

For the implementation of the threat a number of JavaScript libraries are used, including Chrome APIs. Nevertheless, while the threats were identified in Year 1 of the project, the implementation of methods to detect them is also connected with the information available to and the capabilities of the browser add-on, as many times the respective developers do not allow specific code and request to be executed, due to security of the browser itself.

---

26 For more details also see: https://www.microsoft.com/silverlight/
27 For more details also see, inter-alia: https://en.wikipedia.org/wiki/Command-line_interface
28 Also see: https://www.microsoft.com/en-us/windows/windows-10-mobile-upgrade
29 Also see: https://www.microsoft.com/en-us/mobile/
The following list of threats presents the ones implemented in the add-on, and have the status established after the 2nd year of the project.

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website provide data encryption (SSL/TLS)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Easy to intercept users sensitive information at multiple points using a network sniffer(^{30}). All data are visible to the ISP as well as to the network gateway</td>
</tr>
<tr>
<td>High-level Description</td>
<td>The data encryption allows the authentication of the website and the communication between the user browser and the server. Data encryption protocols protect against man-in-the-middle attacks.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Confidentiality of Communications</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>The add-on checks the current tab url and checks if https is enabled or not.</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>What information does the website/server directly learn about a user (using forms)?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Retrieve user information without his knowledge</td>
</tr>
<tr>
<td>High-level Description</td>
<td>A common input method on websites are HTML forms where users are asked to provide information that will then be sent to the website’s server. This information can range from name and address, to passwords and credit cards. The user should be informed on the data any website has collected from him over a period of time.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Privacy Compromising Mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>N/A</td>
</tr>
<tr>
<td>Return Value</td>
<td>User submitted information through URPAAM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Which communication parties is data transferred to?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>3rd parties may retrieve user information without his knowledge.</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Some websites may send the user information (as name, address, e-mail) to third parties for numerous reasons (e.g. advertising) without the user knowledge, comprising user’s privacy.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Privacy Compromising Mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>N/A</td>
</tr>
<tr>
<td>Return Value</td>
<td>User submitted information through URPAAM</td>
</tr>
</tbody>
</table>

| Name                                                      | Does the website use HTML cookies? |

\(^{30}\) A network sniffer monitors data flowing over computer network links in real time. It can be a self-contained software program or a hardware device with the appropriate software or firmware programming. For more details also see: https://www.lifewire.com/definition-of-sniffer-817996

Deliverable 4.2 (“Second year report on Technical enablers’ development”)
<table>
<thead>
<tr>
<th>Threat</th>
<th>Misuse of cookies for authentication or store of sensitive information</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-level Description</td>
<td>Cookies are small text files composed of alphanumeric characters, which are created on user’s computer when the browser accesses a website that uses cookies. Every time the user loads the website, the browser sends the cookie back to the server, to notify the user’s previous activity. HTTP cookies are used for authentication or to store sensitive information.</td>
</tr>
<tr>
<td>Threat Category</td>
<td>Privacy Compromising Mechanisms</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>While the website is loading, a listener is used to catch the communication of cookies with the server. The cookies are then separated based on their characteristics to calculate the number of HTTP cookies.</td>
</tr>
<tr>
<td>Return Value</td>
<td>Value [0…n]</td>
</tr>
</tbody>
</table>

| Name | Does the website use third party cookies? |
| Threat | Tracking user’s browsing history |
| High-level Description | Cookies are small text files composed of alphanumeric characters, which are created on user’s computer when the browser accesses a website that uses cookies. Every time the user loads the website, the browser sends the cookie back to the server to notify the user’s previous activity. Third party cookies are used to track user’s browsing history. |
| Threat Category | Privacy Compromising Mechanisms |
| Implementation Details | While the website is loading, a listener is used to catch the communication of cookies with the server. The cookies are then separated based on their characteristics to calculate the number of third party cookies. |
| Return Value | Value [0…n] |

| Name | Does the website use HTML5 Web SQL database |
| Threat | A third-party advertiser could use a unique identifier stored in its local storage area to track a user across multiple sessions. |
| High-level Description | Sometimes websites are using this methodology to store user information as, for example, unique identifier of the user. If accessed by a third-party it could be used to track the user across multiple browsing sessions and learn more about the users browsing history. |
| Threat Category | Privacy Compromising Mechanisms |
| Implementation Details | Under evaluation the possibility of retrieving this information from the add-on. |
| Return Value | True/False |

| Name | Does the website use LSOs? |
| Threat | Detailed Tracking, allows spyware or malware to be installed |
| High-level Description | Local Shared Objects\(^{31}\) (LSO) or Flash Cookies work the same way as normal cookies and are used by the Adobe Flash Player\(^{32}\) to store information at the user’s computer. They exhibit a similar privacy risk as normal cookies, but are not as easily blocked, meaning that the option in most browsers to not accept |

---

\(^{31}\) Also see, for example: https://en.wikipedia.org/wiki/Local_shared_object

\(^{32}\) For more details also see: https://get.adobe.com/flashplayer/
cookies does not affect Flash cookies. Flash cookies are unlike HTTP cookies in a sense that they are not transferred from the client back to the server. Web browsers read and write these cookies and can track any data by web usage. HTTP cookies can't save more than 4 Kilobyte of data, while Flash cookies can save up to 100 Kilobyte by default.

<table>
<thead>
<tr>
<th>Threat Category</th>
<th>Privacy Compromising Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Details</td>
<td>Under evaluation the possibility of retrieving this information from the add-on.</td>
</tr>
<tr>
<td>Return Value</td>
<td>True/False</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Does the website use Supercookies?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat</td>
<td>Disrupts or impersonates legitimate user requests to another website that shares the same Top-Level Domain as the malicious website</td>
</tr>
<tr>
<td>High-level Description</td>
<td>Supercookies are the cookies with an origin of a Top-Level Domain (such as .com) or a Public Suffix (such as .co.uk). An attacker in control of a malicious website could set a supercookie and potentially disrupt -or impersonate- legitimate user requests to another website that shares the same Top-Level Domain (TLD) or Public Suffix as the malicious website.</td>
</tr>
<tr>
<td>Implementation Details</td>
<td>N/A</td>
</tr>
<tr>
<td>There is no way to differentiate cookies and supercookies from the add-on, as there is no specific way if cookie or not unless compared with other websites. From the add-on the list of cookies can be retrieved (see above) and passed over to the backend. In discussion if this identification can take place in the back-end.</td>
<td></td>
</tr>
<tr>
<td>Return Value</td>
<td>Value [0…n]</td>
</tr>
</tbody>
</table>

3.2 Security and Privacy Components

3.2.1 Securing Backend Server

HTTPS must be configured on the server and all traffic goes through it. Also, all other unused or unsecured ports (e.g., HTTP) are disabled. A certificate for privacy flag will be issued from "Let’s Encrypt" CA. A domain name for the privacy flag backend should be reserved and inserted in the certificate.

3.2.2 Securing Communication

Secure communication between frontend and backend is done through two security levels. The first level is to let the user traffic route through the Tor network to maximize the security and privacy levels. Otherwise, if the user does not use Tor at the moment, the frontend will communicate through a direct HTTPS connection, but in this case the user anonymity is not guaranteed.

To route Privacy Flag (PF) traffic through Tor network, the user should install Tor, which differs based on the user device. More specifically, the browser add-on user has to download the "Tor Browser Bundle" from https://www.torproject.org and start it. While it is running, the add-on in Chrome will

---

33 For further information also see, inter-alia: http://searchsecurity.techtarget.com/definition/supercookie
seamlessly use the Tor proxy running in the localhost. For the Android app (see (Google 2015)), the user needs to install the “Orbot” app and start a Tor connection from Orbot\textsuperscript{34}. While it is running, the PF app will use the proxy server provided by Orbot.

To implement this 2-levels concept, the proxy settings of the frontend should be changed such that the traffic routes through the Tor proxy (i.e., localhost:9150) if the requested URL is located on the PF domain, while the traffic routes directly otherwise. Therefore, the PF domain name must be passed in the frontend initialization to differentiate the user traffic from PF traffic.

Since Privacy Flag project aims to increase the users awareness about privacy and its enablers (i.e., Tor), we suggest to show an elegant warning message if the user is not using Tor and give the user a link to a page on the PF website describing the Tor advantages. Then, we may ask the user if he/she likes to continue using PF frontend without using Tor.

Implementation details for both add-on and Android app are provided in the next two sections.

### 3.2.3 Browser Add-on Code Sample

To allow changing proxy settings from the add-on, the "proxy" permission should be added to the manifest. The code sample can be found in the js/torproxy.js file which contains `testproxy()` function that sets the proxy settings, access a webpage on a specific domain and displays results in the add-on window.

#### Set proxy configuration

Since we need to selectively use Tor proxy only with URLs hosted on the PF server, then `pac_script` proxy mode should be used that enables proxy settings based on the visited URL. Thus, on the add-on initialization, the proxy settings should be modified as follows:

```javascript
var pfdomain = 'privacyflagbackend.eu'; //To be updated by the real domain name

var configtor = {
  mode: "pac_script",
  pacScript: {
    data: "function FindProxyForURL(url, host) {
      if (dnsDomainIs(host, "" + pfdomain + ") || dnsDomainIs(host, 'www." + pfdomain + ") ||
    dnsDomainIs(host, 'torproject.org'))\n" +
      " return 'SOCKS5 localhost:9150';\n" +
      " return 'DIRECT';\n" +
    "}
  }
};

chrome.proxy.settings.set({value: configtor, scope: 'regular'}, function() {});
```

**Verify Tor proxy is running**

---

\textsuperscript{34} Orbot is an application that allows mobile phone users to access the web, instant messaging and email without being monitored or blocked by their mobile internet service provider. Orbot brings the features and functionality of Tor to the Android mobile operating system. For more details see: https://www.torproject.org/docs/android.html.en
To check if the Tor proxy is running correctly, we can access the [https://check.torproject.org](https://check.torproject.org) page which says “Congratulations. This browser is configured to use Tor.” in the title if the client connects to it through Tor network and it responds by “Sorry. You are not using Tor.” title otherwise. Alternatively, we can make it simpler by trying to access any webpage on PF domain or [torproject.org](http://torproject.org). If the connection successes, then Tor proxy is running. For example, this code snippet performs this check:

```javascript
var configsystem = {  
  mode: 'system',  
};

$.get( 'https://check.torproject.org', function() {})
  .done(function(data) {  
    // connection succeeded, then Tor proxy is working
    //document.getElementById('torcheck').innerHTML = data;
  })
  .fail(function() {  
    // connection failed, then warn the user
    if(confirm('It seems that Tor browser is currently not running. Are you sure to continue using Privacy Flag without using Tor?') == true) {  
      // User wants to continue without using Tor, then roll-back the system
      // proxy settings
      chrome.proxy.settings.set({value: configsystem, scope: 'regular'});
    } else {  
      // Otherwise, keep Tor proxy settings.
    }
  });
```

The add-on should keep the user choice in the session to avoid asking the user about his/her choice every time the server is accessed.

**Access the PF backend**

The add-on is free to use any library to send HTTPS request to the PF server. The following shows a jQuery example:

```javascript
$.get( '<https://<pf domain>>', function() {})
  .done(function(data) {  
    // data contains the HTTPS response
  })
```

### 3.2.4 Smartphone App Code Sample

The android app follows a similar concept that changes the proxy setting to route the PF traffic through the Tor proxy.

**Set proxy settings**

To let the app communicate to an external server, proper permissions should be added to the app manifest. These permissions are as follows:
UL developed a customized class “TorProxySelector” that should be included to handle routing PF traffic. In the app initialization, a new object from this class should be created passing the default system proxy settings and the PF domain name as a string. Then, the set this object as the default as follows:

```java
TorProxySelector torproxy = null;
torproxy = new TorProxySelector(ProxySelector.getDefault(), "privacyflagbackend.eu");
ProxySelector.setDefault(torproxy);
```

**Verify Tor proxy is running**

To check if the Tor proxy is running, you can call the `isTorConfigured()` function after accessing a page on the PF domain. If it returns true, then the Tor proxy is supposed to work properly. If not, the TorProxySelector class is configured to automatically roll back to the original system settings.

**Access the PF backend**

UL has developed a wrapper class “SSLConnection” for the HttpsURLConnection built-in class to simplify the asynchronous HTTPS communication. What is needed from the class that sends requests to the server is to implement the `SSLConnection.SSLConnectionAsyncResponse` interface and override the `processFinish()` function. The latter function should process HTTPS responses when they arrive.

To send a request to the server, it is required to create an SSLConnection object and call execute function with passing the requested URL in the RequestDetails object. For example:

```java
new SSLConnection(this).execute(new RequestDetails(url));
```

The SSLConnection constructor takes the object that implements SSLConnectionAsyncResponse interface which can be the caller class. The execute function takes a RequestDetails object that assign an internal Id for the request and takes the requested URL. The requestID can be used later on when processing the response. The execute function is asynchronous and returns immediately. When a response arrives the processFinish callback function is called with the response details. A definition sample of this function can be the following:

```java
public void processFinish(ResponseDetails response){
    // Here, you will receive all responses
    // You can differentiate among responses by checking the
    // RequestDetails object included in the response

    if (response.isError)
        // Handle errors here (response.ErrorMsg)
    else
        // Work with response here (response.HTTPResponse)
}```
The processFinish function will receive a ResponseDetails object that contains the following members:

- Request which contains an automatically assigned RequestID and the URL. By checking the included request, you can associate the returned response to its corresponding request.
- isError which set true when there is an error occurred
- HTTPResponse which is a string of the returned response.
- ErrorMsg which contains the error message if isError is true.

### 3.2.5 Anonymizing Exchanged Data

Even if Tor network is employed to route the PF traffic, exchanged data must be anonymous or at least anonymized. The following two tables show user data that are transferred to and stored in the server by add-on and smartphone app.

#### Browser Add-on

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Recommendations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device/User Id</td>
<td>Collecting device or user ID directly contradicts with the user anonymity.</td>
</tr>
<tr>
<td>Website name</td>
<td>The URL must be truncated to the scheme, subdomain(s) and domain names. Directory path and page name should not be sent to the server, except there is a need for that. In any case, the query string should not be sent to the server.</td>
</tr>
<tr>
<td>Website evaluation</td>
<td>There is no anonymity issue in this data item.</td>
</tr>
<tr>
<td>(UPRAAM + automated evaluation)</td>
<td></td>
</tr>
</tbody>
</table>

#### Smartphone App

| Data Item                          | Recommendations                                                                                                                                                                                                 |
|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
| Device/User Id                     | Collecting device or user ID directly contradicts with the user anonymity.                                                                                                                                       |
| List of all applications installed in the app | Application names should not be sent to the server because recent studies showed that user's habits and traits can be accurately identified using such list of installed apps. For more information, please check this study [http://privmetrics.data61.csiro.au/wp-content/uploads/2015/02/MC2R_paper_1569898227.pdf](http://privmetrics.data61.csiro.au/wp-content/uploads/2015/02/MC2R_paper_1569898227.pdf) Hashing application names may be a workaround in the sense that it hardens identifying the user’s interests directly, but user identification may be still possible. Another step can be taken is to divide sending the hashed applications list into several requests, hopping that rejoining the |
requests is not trivial (e.g., assuming that several users from the same public IP send their app list concurrently).

| Application evaluation (UPRAAM + permissions) | There is no anonymity issue in this data item. |

### 3.3 Recommended Resources for Users

The second important goal of Task 4.1 is to recommend trusted tools and resources for end-users to be able to preserve their privacy. Thus, we surveyed the cyber-activities that users commonly do with their computers and smart devices to identify possible privacy leaks and the recommended tools to mitigate these leaks. The conducted survey included a wide range of cyber-activities from surfing the Internet and sending instant messages to securely managing user credentials. Here we recommend a list of best practices and tools that are helpful for users to preserve their cyber-security and privacy.

#### 3.3.1 Web browsing

While users browse the web, they leave footprints or traces behind them, whether locally on their machines or remotely on the web servers. However, there are some tips that can mitigate this problem:

**Private Browser Window**

Most web browsers supports a private browsing mode which deletes all data related to the user session once the browser window is closed. Session data include browsing history, filled forms, cached media and first-party cookies.

**Useful Add-ons**

There are many browser add-ons that claim that they improve the user privacy. However, not all of them are really doing so, or doing it in the correct way. We list here most popular add-ons that are recommended -and/or offered- by authentic sources.

- **uBlock Origin (Firefox/Chrome):** It is an efficient wide-spectrum Ads blocker that is not memory consumer but supports so many Ads filters than other popular blockers. It is completely open source, has no monetization strategy, and does not allow so-called "acceptable ads" like AdBlock Plus. [https://github.com/gorhill/uBlock/wiki/Blocking-mode](https://github.com/gorhill/uBlock/wiki/Blocking-mode)

- **Self-Destructing Cookies (Firefox):** It automatically removes cookies when they are no longer used by open browser tabs. [https://Add-ons.mozilla.org/en-gb/firefox/Add-on/self-destructing-cookies/](https://Add-ons.mozilla.org/en-gb/firefox/Add-on/self-destructing-cookies/)

- **Click&Clean (Firefox/Chrome):** deletes browsing and download history, temporary files and cache, remove cookies including Flash LSO. [https://www.hotcleaner.com/](https://www.hotcleaner.com/)
**Disconnector** *(Firefox/Chrome)* is an open source add-on that visualizes and blocks the invisible websites that track the user. It blocks third party tracking cookies and gives the user control over all site scripts. Disconnect was named the best privacy tool by the New York Times (2016). [https://disconnect.me/disconnect](https://disconnect.me/disconnect)

**Privacy Badger** *(Firefox/Chrome)* stops advertisers and other third-party trackers from secretly tracking the user. If an advertiser seems to be tracking the user across multiple websites without his/her permission, Privacy Badger automatically blocks that advertiser from loading any more content in the browser. [https://www.eff.org/privacybadger](https://www.eff.org/privacybadger)

**HTTPS Everywhere** *(Firefox/Chrome)*: It encrypts the user communications with many major websites, making the browsing more secure. This add-on is a collaboration between The Tor Project and the Electronic Frontier Foundation (EFF). [https://www.eff.org/https-everywhere](https://www.eff.org/https-everywhere)

**Decentraleyes** *(Firefox/Chrome)*: It emulates Content Delivery Networks (CDNs) locally by intercepting requests, finding the required resource and injecting it into the environment. This all happens instantaneously, automatically, and no prior configuration is required. [https://Add-ons.mozilla.org/firefox/Add-on/decentraleyes/](https://Add-ons.mozilla.org/firefox/Add-on/decentraleyes/)

**uMatrix** *(Firefox/Chrome)*: Many websites integrate features which let other websites track you, such as Facebook Like Buttons or Google Analytics. uMatrix gives the user control over the requests that websites make to other websites. This gives greater and more fine-grained control over the information that the user leaks online. [https://Add-ons.mozilla.org/firefox/Add-on/umatrix/](https://Add-ons.mozilla.org/firefox/Add-on/umatrix/)

**NoScript** *(Firefox)* and **ScriptSafe** *(Chrome)*: is a highly customizable add-on to selectively allow Javascript, Java, and Flash to run only on websites that the user trusts. Not for casual users because it requires technical knowledge to configure.  
- [https://Add-ons.mozilla.org/firefox/Add-on/noscript/](https://Add-ons.mozilla.org/firefox/Add-on/noscript/)

**Harmful Add-ons**

On the other side, there are some popular add-ons that are supposed to protect user privacy but, in fact, they sell user data to 3rd parties:

**Ghostery** gives the user the control of the personal data that can be shared with the trackers on the visited websites. It can also block the offending trackers and significantly speed up the browsing experience. However, it was reported by an MIT review that its company helps advertising companies that want to improve their use of tracking code by selling them data collected from Ghostery users who have enabled a data-sharing feature in the tool. Source: [https://www.technologyreview.com/s/516156/a-popular-ad-blocker-also-helps-the-ad-industry/](https://www.technologyreview.com/s/516156/a-popular-ad-blocker-also-helps-the-ad-industry/)

---

35 For more details see: [https://www.eff.org/](https://www.eff.org/)
36 For more details see: [http://www.addthis.com/social-buttons/facebook-like-button](http://www.addthis.com/social-buttons/facebook-like-button)
37 For more details see: [https://www.google.com/analytics/](https://www.google.com/analytics/)
**Web of Trust (WOT)** uses crowdsourcing to rate sites based on trustworthiness and child safety. It turns out that it also collects a bunch of data about the user browsing habits. According to a report from the German television channel NDR\(^38\), WOT sells user data to various third party companies without even a proper anonymization, which means it is easy to re-identify the user. Source: [https://en.wikipedia.org/wiki/WOT_Services#Privacy_issues](https://en.wikipedia.org/wiki/WOT_Services#Privacy_issues)

**Tor Browser**

When users really care about their anonymity, they are advised to use the Tor browser. The Tor network ([https://www.torproject.org/](https://www.torproject.org/)) is a group of volunteer-operated servers that allows people to improve their privacy and security on the Internet. Tor’s users employ this network by connecting through a series of virtual secure tunnels, rather than making a direct connection to the destination. Along the same line, Tor is an effective censorship circumvention tool, allowing its users to reach otherwise blocked destinations or content. Tor can also be used as a building block for software developers to create new communication tools with built-in privacy features. The Privacy Flag add-on and smartphone app enablers uses this feature and routes the Privacy Flag traffic through Tor if the user installs and enables Tor as explained in Section 2.2.2. More detailed information about Tor can be also found in Section 2.4.

Since the Tor network only considers the anonymity of data transport, the Tor project created the Tor Browser Bundle\(^39\) (TBB) which consists of a modified Mozilla Firefox web browser, the TorButton\(^40\), TorLauncher\(^41\), NoScript\(^42\) and HTTPS Everywhere\(^43\) extensions and the Tor proxy. The Tor Browser automatically starts Tor background processes and routes traffic through the Tor network. Upon termination of a session the browser deletes privacy-sensitive data such as HTTP cookies and the browsing history.

### 3.3.2 Virtual Private Network (VPN)

Virtual Private Network (VPN) extends a private network across a public network, typically the Internet. It enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. Business users employ VPNs to connect remote datacenters, and individuals can use VPNs to get access to network resources when they’re not physically on the same local area network (LAN), or as a method for securing and encrypting their communications when they are using an untrusted public network (e.g., open WiFi). When the user connects to a VPN, he/she usually launches a VPN client on his/her machine, log-in with his/her credentials, and the user device exchanges trusted keys with a remote VPN server. Once both client and server have verified each other as authentic, all of user traffic is encrypted and secured from eavesdropping.

---

\(^{38}\) [https://www.ndr.de/](https://www.ndr.de/)

\(^{39}\) Also see: [https://www.torproject.org/projects/torbrowser.html.en](https://www.torproject.org/projects/torbrowser.html.en)

\(^{40}\) For more details see: [https://www.torproject.org/docs/torbutton/](https://www.torproject.org/docs/torbutton/)

\(^{41}\) Also see: [https://www.torproject.org/projects/torbrowser.html.en](https://www.torproject.org/projects/torbrowser.html.en)

\(^{42}\) Also see: [https://noscript.net/](https://noscript.net/)

\(^{43}\) For more details also see: [https://www.eff.org/https-everywhere](https://www.eff.org/https-everywhere)
Recently, VPN services are commonly used to browse the Internet securely and anonymously. However, not all VPN providers are completely anonymous even if they declare themselves so. Some providers do logging of the IP addresses and activities of their users. Thus, VPN should not be considered as an anonymity tool but as a security layer for the user traffic so that it is not easily monitored or censored by intermediate parties. Nevertheless, we recognize privacy-friendly VPN providers since they support OpenVPN\textsuperscript{44}, do not require personal information to register, adopt no logging policy and offer protection against DNS leaks. Some examples are TorGuard\textsuperscript{45}, NordVPN\textsuperscript{46}, EarthVPN\textsuperscript{47} and Proxy.sh\textsuperscript{48}.

It is worthy to stress on the fact that user traffic is encrypted only when and during a VPN connection is established. Traffic transmitted before establishing a VPN connection (e.g., emails sync on startup) or while the connection is disconnected is not secure nor private. There are some VPN clients and 3\textsuperscript{rd} party tools (e.g., VPNCheck Pro\textsuperscript{49}) that stop the traffic or alert the user when communication outside the VPN tunnel occurs. Furthermore, there is a serious leak caused by WebRTC. WebRTC is a new communication protocol that relies on JavaScript and allows different platforms to communicate via a common set of protocols. This technology can leak the user actual IP address even from behind a VPN. Thus, WebRTC is recommended to be disabled in the web browser when using VPN. It can be disabled in Firefox by setting "media.peerconnection.enabled" to "false" in "about:config". For Chrome, users should Install Google official extension WebRTC Network Limiter.

### 3.3.3 Email service and client

Google has clarified its email scanning practices in a “terms of service” update, informing users that incoming and outgoing emails are analyzed by automated software. The revisions explicitly state that Google’s system scans the content of emails stored on Google’s servers as well as those being sent and received by any Google email account. This happens because emails are stored in clear text in the mail servers, even if they are transmitted to the server in an encrypted form. This information leakage is similar to other free popular email service providers.

Thus, to overcome this serious privacy leak, users are advised to use secure email services such as ProtonMail (https://protonmail.com/). Messages are transmitted and stored on ProtonMail servers in encrypted format. This means that they do not have the technical ability to decrypt the user messages and, as a result, they are unable to hand the user data over to third parties. They also do not save any tracking information, such as the IP addresses used to log in. Also, it does not require any personal identifiable information to register.

In case of using corporate email service, users are advised to use Pretty Good Privacy\textsuperscript{50} (PGP) to encrypt and sign their messages. PGP can be used to send messages confidentially. For this, PGP combines symmetric-key encryption and public-key encryption. The message is encrypted by using a symmetric encryption algorithm, which requires a symmetric key. Each symmetric key is used only once and is

\textsuperscript{44} More informative details can be found at: https://openvpn.net/
\textsuperscript{45} For more details also see: https://torguard.net/
\textsuperscript{46} For more details also see: https://nordvpn.com/
\textsuperscript{47} For more details also see: http://www.earthvpn.com/
\textsuperscript{48} For more details also see: https://proxy.sh/
\textsuperscript{49} Also see: http://www.quavi.com/vpncheck_pro.php
\textsuperscript{50} For more details also see, inter-alia: https://en.wikipedia.org/wiki/Pretty_Good_Privacy
also called as a session key. The message and its session key are sent to the receiver. The session key must be sent to the receiver so they know how to decrypt the message; but to protect it during transmission it is encrypted with the receiver’s public key. Only the private key belonging to the receiver can decrypt the session key. PGP supports message authentication and integrity checking. The latter is used to detect whether a message has been altered since it was completed (the message integrity property), and the former to determine whether it was actually sent by the person or entity claimed to be the sender (a digital signature). Because the content is encrypted, any changes in the message will result in failure of the decryption with the appropriate key. The sender uses PGP to create a digital signature for the message with either the RSA\textsuperscript{51} or DSA\textsuperscript{52} algorithms. To do so, PGP computes a hash (also called a message digest) from the plaintext and then creates the digital signature from that hash using the sender’s private key. There are several add-ons that support PGP in different email clients such as Enigmail (https://www.enigmail.net) for Mozilla Thunderbird\textsuperscript{53} and SeaMonkey\textsuperscript{54}.

### 3.3.4 Search Engines

Search engines (like Google, Yahoo) use search queries issued by users to deliver more personalized search results. This maybe apparently a good feature but to be enabled, search engines store the user query, timestamp and IP address to track users and infer the user interests. This information is shared to the requested website which threatens the user privacy. Also, this feature prevents showing the results from the whole Internet, only things interested to the user.

On the bright side, there are some private search engines out there such as DuckDuckGo (https://duckduckgo.com/) which does not track users nor collect their personalized information. Also, Startpage (https://www.startpage.com/) fetches results from Google on behalf of the user without recording her IP address or log anything about its behavior. Similarly, SearX (https://searx.me/) is an open source meta-search engine, aggregates the results of other search engines while not storing information about its users.

### 3.3.5 Instant Messaging (IM) and VoIP Software

Instant Messengers (like Skype, Whatsapp, Viber, Facebook Messenger) are used by almost everyone nowadays in sending text message and place voice and video calls. Few of these popular IM have recently started to support end-to-end encryption like Whatsapp and Viber which makes conversation secret even from the servers of the provider. Facebook Messenger supports end-to-end (E2E) encryption only for mobile devices and text messages, but it is not enabled by default. Even if this feature is support by IM app, user privacy is still not fully safe. In August 2016, WhatsApp announced that it will start to share user data with its parent company Facebook in order to draw in adverts to the platform. Third party companies will be able to send targeted messages directly to WhatsApp users.

To keep user messages safe and private, users are advised to use privacy-friendly open-source IM services. Signal (https://whispersystems.org/) is a free and open source mobile app which provides end-to-end encrypted instant messaging and voice calls (Video calls are recently beta supported).

\textsuperscript{51} For more details also see, inter-alia: https://en.wikipedia.org/wiki/RSA_%28cryptosystem\textsuperscript{52} For more details also see, inter-alia: https://en.wikipedia.org/wiki/Digital_Signature_Algorithm\textsuperscript{53} For more details also see, inter-alia: https://en.wikipedia.org/wiki/Mozilla_Thunderbird\textsuperscript{54} For more details also see: https://www.seamonkey-project.org/
Encrypted group chats are also supported. **Wire** ([https://wire.com/](https://wire.com/)) supports similar features as Signal but it is supported for desktop, web and mobile platforms. For desktop or laptop machines, **Ricochet** ([https://ricochet.im/](https://ricochet.im/)) is another alternative which uses the Tor network to reach user contacts without relying on messaging servers. Ricochet starts a Tor hidden service locally on a person’s computer and can communicate only with other Ricochet users who are also running their own Ricochet-created Tor hidden services. This way, Ricochet communication never leaves the Tor network. A user screen name (e.g., ricochet:hslmfs47dmcqctb) is auto-generated upon first starting Ricochet; the first half of the screen name is the word “ricochet", with the second half being the address of the Tor hidden service. Before two Ricochet users can talk, at least one of them must privately or publicly share their unique screen name in some way.

3.3.6 Encrypted Cloud Storage

Cloud storage (like Dropbox, OneDrive and Google Drive) started to be an essential tool for everyone to share files and media among the user’s devices or contacts. Although all cloud services promise a security guarantee, they store user files unencrypted in the cloud whether on their own or shared servers. This means that the technical team or the government can have access to the personal user files without notice. Also, if the servers got compromised, hackers have full access on the user files.

There are several secure alternatives for cloud storage like **Sync** ([https://www.sync.com/](https://www.sync.com/)), **Seafile** ([https://www.seafile.com/](https://www.seafile.com/)) and **SpiderOak** ([https://spideroak.com/](https://spideroak.com/)). Sync is a zero-knowledge cloud services which means the files are encrypted on the user machine before they are synced with the cloud servers. Sync.com use a 256-bit AES encryption on files and lock them with 2048-bit RSA private keys. For maximum security, passwords are never transmitted to Sync. Furthermore, Sync does not store passwords or password hashes during account creation, or when you log in. Another model is offered by Seafile. This cloud service gives users the opportunity to host their files on their own servers. It also offers to host user files on their servers either in Germany or with Amazon Web Service in the US.

3.3.7 Disk and File Encryption

Disk encryption software protects the confidentiality of data stored on computer media (e.g., a hard disk, floppy disk, or USB device). Operating systems usually use and enforce access controls where each user account has specific rights on each file and directory. However, disk encryption passively protects data confidentiality even when the OS is not active, for example if data is read directly from the hardware or by a different OS. Disk encryption generally refers to complete encryption that operates on an entire volume mostly transparently to the user, the system, and applications. This is generally distinguished from file-level encryption which enabled by the user action on a single file or group of files. Disk encryption usually includes all aspects of the disk, including directories, so that an attack cannot determine content, name or size of any file. It is well suited to portable devices such as laptop computers and external drives, which are particularly susceptible to being lost or stolen. If used properly, someone finding a lost device cannot penetrate actual data, or even know what files might be present.

---

55 For more information also see, inter-alia: [https://en.wikipedia.org/wiki/Advanced_Encryption_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)
One example for disk encryption tools is VeraCrypt (https://veracrypt.codeplex.com/) which can create and maintain an on-the-fly-encrypted volume. On-the-fly encryption (OTFE) means that data is automatically encrypted right before it is saved and decrypted right after it is loaded, without any user intervention. No data stored on an encrypted volume can be read (decrypted) without using the correct password/keyfile(s) or correct encryption keys. The on-the-fly encryption does not mean that the whole file that is to be encrypted/decrypted must be stored in RAM before it can be encrypted/decrypted. There are no extra memory (RAM) requirements. VeraCrypt supports individual file encryption as well. For file encryption, GnuPG (https://www.gnupg.org/) is a simple option which is open-source and supported on all desktop operating systems (Windows, Linux and MacOS).

3.3.8 Password Management

Good security practices require users to use different passwords for each website and service. For accounts of any significance, those also need to be strong passwords of one form or another. But if we combine those two requirements (one password per site and strong passwords) then it turns to a very complex task. To handle this complexity, there are many password managers but only few of them are really secure. Master Password (https://ssl.masterpasswordapp.com/) is based on a password generation algorithm that guarantees the passwords can never be lost. Its passwords are not stored, but they are generated on-demand from the user name, the site and the master password. Therefore, no syncing, backups or Internet access needed. KeePass (http://keepass.info/) is a free open source traditional password manager, which helps the user manage passwords in a secure way. All passwords are stored in one database, which is locked with one master key or a key file. The databases are encrypted using the best and most secure encryption algorithms currently known: AES and Twofish⁵⁶.

3.3.9 Two-Factor Authentication (2FA)

After the numerous account breaches occurred last year even from giant service providers (such as LinkedIn, Dropbox, Yahoo, Myspace, etc.), many providers started to support a two-factor authentication to protect their users from being affected by this kind of attacks. Two-factor or two-step authentication adds a second level of authentication to an account log-in. When the user have to enter only the username and password to login to a service, that has considered a single-factor authentication. 2FA requires the user to enter an additional secret (or verification code) that the user (should) possess or knows. This additional secret is transferred to the user through the user’s phone or can be a biometric feature like a fingerprint. For example, the verification code can be sent in a SMS or generated by a dedicated mobile app that stores a secret (such as Google Authenticator, Duo Mobile, the Facebook app). Therefore, 2FA offers the user greater account security by authenticating the identity by more than one method. This means that, even if someone were to get hold of the user’s primary password, they could not access the account unless they also had the mobile phone, or another secondary means of authentication.

The 2FA authentication has variant terms from service to another. Facebook calls the process “login approvals,” Twitter calls it “login verification,” and Google calls it “2-step verification.” An extensive list of services supporting 2FA is available at https://twofactorauth.org/.

Also see, inter-alia: https://en.wikipedia.org/wiki/Twofish

⁵⁶ Also see, inter-alia: https://en.wikipedia.org/wiki/Twofish
3.3.10 Additional Resources

Since security and privacy threats and their countermeasures are continuously evolve, we provide here links to trusted online guides that provide up-to-date tips, tools and best practices for safer and more private communication.

- https://ssd.eff.org/ is a surveillance self-defense guide provided by The Electronic Frontier Foundation (EFF) that provides best practices and step-by-step tutorials that help users of different levels of profession to defend themselves from surveillance by using secure technology.
- https://www.privacytools.io is a socially motivated website that provides information about tools for protecting data security and privacy. If the website is down or compromised, users can access it on GitHub https://privacytoolsio.github.io/privacytools.io/
- https://www.bestvpn.com/the-ultimate-privacy-guide/ is an excellent privacy guide written by the creators of the bestVPN.com website.
- https://guardianproject.info The guardian project provides open-source mobile security software to help end users to communicate freely and protect themselves from intrusion and monitoring.
- https://prism-break.org/ is a crowdsourced portal for privacy-aware, generally open source, alternatives to the most popular applications.
- http://www.ghacks.net/2015/12/28/the-ultimate-online-privacy-test-resource-list/ is a diverse and comprehensive list of online privacy tests that help users find out what kind of information programs and services reveal about users and their devices.

3.4 Network Anonymity

3.4.1 Overview of Privacy Enhancing Technologies

In addition to the already mentioned Privacy Enhancing Technologies (PETs) in D4.1, this section aim to present the current research activities performed during the 2nd year of the project. Although the Tor network is the identified anonymization solution for Privacy Flag, we will assess here the newly proposed systems:

Hornet:

Hornet by Chen et al. was published in the Proceedings of the 22nd ACM SIGSAC57 Conference on Computer and Communications Security 2015. Hornet was heavily misunderstood by media and promoted as a faster Tor, with speeds up to 96GBit/s. The important part that media missed is that Hornet was designed as an onion routing scheme for a Future Internet Architecture (FIA) like e.g., Scion58. But if it comes to the point that a new Internet architecture will be deployed, it would be very beneficial to have onion routing deployed at the autonomous system (AS) level routers to provide privacy in the network by default. The design of systems like this should be closely observed, but for the rest of the project, we will stick to Tor since we design solutions for the current internet.

Riffle:

---

57 Also see: https://www.sigc.org/
58 For more specific details also see: https://www.scion-architecture.net/
Riffle by Kwon et al. was published in the Proceedings of Privacy Enhancing Technologies 2016 and multiple news websites announced it as a new rival to Tor. Unfortunately, media exaggerated its capabilities since Riffle describes a proof of concept (PoC) of an anonymity system that also builds on onion routing and mixnets. But in contrast to what Tor offers, Riffle does not aim at providing a low latency anonymity system for web browsing. In the paper, authors analysed two use cases for Riffle, namely a microblogging service (similar to Twitter) and a file sharing application. Riffle works by providing a relatively small set of anonymity servers that serve a large number of clients. In its setup, anonymity is ensured as long as at least one honest server is in the chain of the used nodes. Anonymity guarantees offered by Riffle are achieved by two different privacy primitives: verifiable shuffle for upstream communication and private information retrieval for downstream communication.

3.4.2 Latency Enhancement

We worked on enhancing the performance of Tor by fine-tuning the relays selection process. Currently, Tor clients select relays for a circuit position randomly with probability roughly proportional to the product of each relay’s weight retrieved from the Directory Authorities. The relay weight is updated hourly and proportional to the relay’s self-reported and measured bandwidth. This selection mechanism may allow the client to select relays from different continents in a single circuit which results in unavoidable latency.

We modified this mechanism by limiting the relay selection over a single continent (i.e., EU) hoping in reducing the latency experienced by Tor users.

Installation

To install this enhanced version on Ubuntu (see Canonical 2016) – Windows is not yet supported), the user must do the following:

1. Install the following packages:
   ```
   sudo apt-get install tor python python-pip
   sudo pip install stem python-geoip-geolite2
   ```
2. Download Tor Browser Bundle from www.torproject.com and uncompress it in the home directory (“~/tor-browser_en-US/”)
3. Copy PF_Tor folder (in OneDrive) to the home directory (“~/PF_Tor/”)
4. Run the following commands in separate terminals:
   a. `tor -f ~/PF_Tor/torrc`
      The directory “~/PF_Tor/data” should now contain 6 files and the terminal says “[notice] Bootstrapped 100%: Done”. Keep this terminal open.
   b. In a new terminal, run python “~/PF_Tor/go.py” – This should show some updates with the last line says “Circuit initialized, start attaching everything”
   c. In a new terminal, start Tor Browser Bundle by running “~/tor-browser_en-US/Browser/start-tor-browser”
5. In the Tor Browser, navigate to any website (e.g., https://check.torproject.org) and click on the green onion icon beside the URL bar. You can see information about the Tor circuit where all nodes should be within Europe as shown in the below screenshot.
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Evaluation

We evaluated the modified Tor with the EU-routing by fetching the top 100 websites ranked by Alexa, ten times per website. Then, we measure the time of loading the website in the standard Tor (i.e., vanilla Tor) and in the modified Tor (i.e., EU-routing Tor). In both variants, the Tor circuit was discarded and a new one was created after 10 of 1000 fetching processes. The evaluation results can be found in the following two graphs. Figure 3-2 describes a CDF (Cumulative Distribution Function) that contains all 1000 measured time values per routing method (Vanilla Tor vs. EU-routing Tor). It describes which percentage of the fetched websites (x-axis) was finished loading in which timespan (y-axis). We notice that more fetching instances in the EU-routing Tor have been loaded faster than in the Vanilla Tor. In contrast, Figure 3-3 displays the loading time as the mean of the ten fetching instances per website. The error bars describe the standard deviation (STD). Mean and STD are displayed for each of the 100 websites for both relay selection methods (Vanilla Tor vs. EU-routing Tor). It is clearly visible that the EU-routing approach incorporates less deviation during multiple fetching processes. The modified Tor seems to be more stable, which will probably enhance the user experience.

Figure 3-1: Browser configuration to use Tor
Figure 3-2: CDF description that contains all 1000 measured time values per routing method
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Figure 3-3: Display of the loading time as the mean of the ten fetching instances per website
4. Privacy Flag Web Browser Add-on

The Privacy Flag web browser add-on is a tool that allows users to get information about potential privacy risks when browsing throughout the Internet. The add-on informs users whether a web site is considered safe or not, based on the analysis conducted by the Privacy Flag backend system; an analysis which includes both input gathered by technical enables and by exploiting the power of crowdsourcing data from end-users incorporating the UPRAAM methodology.

The Privacy Flag web browser add-on is one of the main points of interaction between end-users and the Privacy Flag project.

The following section presents -in brief- the architecture and the functionality scenarios for the browser add-on (as defined in WP1) as well as the two levels of evaluation, the automated evaluation (originated by WP4) and the integration of UPRAAM (defined in WP2 and specified in WP3).

Overall, an update on the status and the development of the browser add-on during the second year of the project is presented.

4.1 Main Architecture Description

Figure 4-1 presents the architecture of the browser web add-on as this has been defined in WP1 and more specifically, D1.259 (“Privacy Flag Initial Architecture Design”) and updated in D1.3 (“Updated architecture design”).

The defined architecture shows where the web browser add-on is positioned within the whole Privacy Flag architecture and what the main interaction with the rest of the components is.

![Figure 4-1: The Privacy Flag Web Browser Add-on Architecture](image)

Summarizing the work presented in D1.2, the browser add-on communicates with the Privacy Flag backend through a web service to exchange information as, for example, the site a user is visiting or

---

the evaluation he/she is providing. The backend tackles the evaluation of the web site (through both the UPRAAM assessment and the automatic assessment) and feeds back the web browser with the information on the risks involved with a specific website. The whole communication with the backend needs to take into account the anonymization of the user, which is handled by the work presented in Section 3.2.

4.2 Browser Add-on Functionality

Main functionality of the Privacy Flag Web Browser add-on is to enable users to be informed on the level of privacy risk when accessing a website through a browser while in parallel allow users to provide their own assessment for the specific website.

In Privacy Flag Deliverable D4.1 ("First Year Report on Technical Enablers development") we presented the main concepts of the browser add-on, under the persona and scenario methodology, and a first demo version of the add-on.

Here, we focus on the development updates that took place during the second year of the project.

4.2.1 Updated Functionality Workflow

Following the needs of the project, the functionality workflow of the browser add-on was updated to reflect the following:

**Step 1:** While the user browses through the sites, the browser add-on informs the user of the current evaluation of the site, by changing the color of its icon. The evaluation of the site comes from the existed information on the Privacy Flag backend and contains both the automated evaluation (through the implementation of the Threat List Matrix) and the UPRAAM evaluation. If no evaluation exists, the add-on informs the user accordingly. In parallel, the browser add-on starts calculating automatically possible threats (e.g. http and 3rd party cookies) and stores the information locally.

**Step 2:** When the user opens the pop-up menu he/she gets a visualisation of the site classification (Privacy Friendly/Not Friendly) and has the option to provide his/her own evaluation by answering the UPRAAM defined questions.

**Step 3:** When the user submits his/her evaluation, a JSON file is created that contains the user answers, the automated calculated threats on the browser add-on and a unique identifier (see the example of JSON file as below).

**Step 4:** (independent - optional): Crowdsourcing Evaluation Tool experts evaluate manually the web site and submit their evaluation to the database, the Local Crowdsourcing Evaluation Tool Score.

**Step 5:** (independent - out of order): The backend performs various calculations based on the threat matrix and, in combination with forecasting epidemiology models, calculates the browser add-on Local Threat Level Score.

**Step 6:** PF back-end decides based on

- The browser add-on Local Threat Level Score,
- The Local Crowdsourcing Evaluation Tool Score,
- Mean Threat Level Score,
- Mean Crowdsourcing Evaluation Tool Score.

---

4.2.2 Privacy Enablers Integration

According to the analysis provided, Privacy Flag enables to be integrated are towards anonymity (as in task T4.1) and threat analysis (as in task T4.2).

The first direction is related with the communication of the browser add-on with the Privacy Flag backend, as presented in Section 7. When a user submits his/her data, the system should not be able to identify the specific user and protect his/her anonymity. The browser add-on will use the provided APIs for the communication with the backend, respecting end-users’ privacy, as specified in Section 3.2.

On the other hand, the browser add-on integrates a set of automated analysis that is passed to the Privacy Evaluation Component in order to get an evaluation of a specific site. This type of analysis is related with the Top 25 threats that were presented earlier in this document. For the successful integration of the threat analysis a number of tests are to take place either in the add-on (e.g. 3rd party cookies) or by a backend scraping mechanism (e.g. PDF, Flash, etc.).

4.2.3 UPRAAM Integration

One of the key elements of the Privacy Flag project is the users own evaluation through the UPRAAM evaluation. As defined in WP2, UPRAAM includes a set of questions to be made to end-users in order to capture their understanding of privacy. UPRAAM includes a scoring framework in order to create an evaluation metric for the privacy evaluation.

For the Privacy Flag browser add-on, UPRAAM is adjusted to be able to make the necessary questions to capture evaluation related to websites based on the input of task T3.2.

4.3 Year 2 Development Overview

During the second year of the PF project the browser add-on was updated with numerous updates in relation to UI, threat identification and communication with backend.
4.3.1 UI updates

During the second year of the project, the UI of the add-on was updated to “reflect” the changes and the project decisions. To this end, the following changes occurred:

- The icon of the add-on changes based on the current evaluation of the site, allowing users to see the evaluation even without opening the plugin.
- Only one total evaluation of the web site is used (during the first demo, the evaluation was separated in automated and UPRAAM).
- Updated the evaluation form of UPRAAM integrating all comments received from task T3.2.

At this point, we have to note that the implementation of the questionnaire in the browser add-on is not a trivial procedure, and a fully customized solution needed to take place for the Privacy Flag. Moreover, looking at existing solutions in the Chrome extension store, the possibility of having surveys included in add-on is something not existed and Velti S.A. will investigate the possibility of future exploitation even towards this direction.

(a)

(b)
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Figure 4-3: The Privacy Browser Add-on UI: initial screen (a); UPRAAM evaluation forms based on task T3.2 (b-f), and; submission of evaluation screen (z)

4.3.2 Backend communication

During the second year of the project, the backend communication was also specified and implemented. The backend communication is used by the add-on to:
Retrieving current evaluation of a website.
Send the user evaluation of a website (both threats and UPRAAM evaluation).
Retrieve the latest UPRAAM questionnaire.

All communication between the add-on and the backend takes place over HTTP REST request.

### 4.3.3 Browser add-on files and source code

For the moment, the project has not yet gone public, regarding its tools. Nonetheless, the execution file and the source code are available in Privacy Flag servers.

**Browser add-on installation instructions**

1. Download and unzip the file.
2. In Chrome, open the extension menu.
3. From the menu select the developer mode (top right button).
4. Select Load unpacked extensions.
5. From the menu select the previous unzipped folder.
5. Privacy Flag Smartphone Application

The Privacy Flag smartphone application allows users to get information on potential privacy risks from installed applications in their Android-powered mobile phones and tablets. The application informs users whether installed software is considered as privacy friendly - or not friendly - based on the analysis conducted by the Privacy Flag backend system; that is, an analysis which includes both input gathered by technical enablers and by exploiting the power of crowdsourcing data from end-users incorporating the UPRAAM methodology. Combined with the Privacy Flag web browser add-on, the smartphone application is one of the main points of interaction between end-users and the Privacy Flag project.

The following chapter presents in brief the architecture and the functionality scenarios for the smartphone application (as defined in WP1) as well as the two levels of evaluation, the automated evaluation (originating from WP4) and the integration of UPRAAM (as defined in WP3).

Overall an update on the status and the development of the browser add-on during the second year of the project is presented.

5.1 Architecture

Figure 5-1: The Privacy Flag Smartphone Application Architecture

Figure 5-1 presents the architecture of the smartphone application as this has been defined in WP1 and more specifically, D1.2 (“Privacy Flag Initial Architecture Design”) and updated in D1.3 (“Updated architecture design”). The defined architecture shows where the app is positioned within the whole Privacy Flag architecture, and what the main interaction with the rest of the components is.

Summarizing the work presented in D1.2, the application communicates with the Privacy Flag backend through a web service to exchange information related to the already installed applications. The backend tackles the evaluation of the apps (through both the UPRAAM assessment and the automatic assessment) and feeds back the application with the information on the risks involved with a specific application. The whole communication with the backend needs to take into account the anonymization of the user, which is handled by the work presented in Section 3.4.
5.2 Mobile Application Functionality

Main functionality of the Privacy Flag Web Smartphone Application is to inform users on privacy risks originating from applications installed in their mobile phones.

In Privacy Flag Deliverable D4.1 (“First Year Report on Technical Enablers development”) we presented the main concepts of the Application, under the persona and scenario methodology, and a first demo version of the app.

Here we focus on the development updates that took place during the second year of the project.

5.2.1 Updated Functionality Workflow

Following the needs of the project, the functionality of the application was updated as follows:

**Step 1:** When first installed and opened, the application asks users to submit for a first and only time their preferences in regards to privacy and the user permission. The user’s answers and main information are passed to the backend through a JSON format file, in order to include this in the following calculations.

**Step 2:** The smartphone application reads the user’s installed applications (and checks for new or updated ones if this isn’t the first time the user uses the PF application) and sends the list of applications to the backend in order to retrieve their evaluation (privacy friendly, not friendly or not evaluated).

**Step 3:** The users browse among the installed applications and can view its evaluation and the user set permissions that have been given to the app.

**Step 4:** (independent-optional). The smartphone evaluations’ contributors manually evaluate applications and submit their evaluation to the database. This is called as the Crowdsourcing Evaluation Tool score. In addition, the user set permissions of the application are also transmitted to the backend.

**Step 5:** (independent - out of order). The Privacy Flag backend performs various calculations based on Machine Learning algorithms for outlier detection. It also employs advanced statistical models and epidemiological metrics to detect outliers (applications with vastly different Threat Level Scores) which indicate possible data leakage. The outputs of the database calculations are the Mean Threat Level Score and the Mean Crowdsourcing Evaluation Tool Score.

**Step 5:** The PF backend decides based on:

- The smartphone application’s Local Threat Level Score.
- Users evaluation based on the UPRAAM Score.
- Mean Threat Level Score.
- Mean UPRAAM Score.

5.2.2 Privacy Enablers Integration

Similarly to the browser add-on, the Privacy flag mobile application integrates privacy enablers related to anonymity (as in task T4.1) and threat analysis (as in task T4.2).

The first direction is related with the communication of the smartphone application with the Privacy Flag backend, as presented in Section 11. When a user submits his/her data, the system should not be able to identify the specific user and protect his/her anonymity. The smartphone application will use
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

the provided APIs for the communication with the backend, respecting end-users’ privacy, as specified in Section 3.2.

In regards to the automated analysis, this has to do with specific elements the application can read from the Android system and then pass to the backend, in order to have an evaluation for the installed applications. This part of the evaluation is quite tricky, since it is also related with the operating system permissions. As presented in year one, this analysis includes the user set permissions.

5.2.3 UPRAAM Integration

One of the key elements of the Privacy Flag project is the users’ own evaluation through the UPRAAM evaluation. As defined in WP2, UPRAAM includes a set of questions to be made to end-users in order to “capture” their understanding of privacy. UPRAAM includes a scoring framework in order to create an evaluation metric for the privacy evaluation.

For the Privacy Flag mobile application UPRAAM is adjusted to be able to make the necessary questions to capture evaluation related to web-sites based on the input of task T3.2.

5.3 Year 2 Development Overview

During the second year of the project, the Privacy Flag application was updated in relation to UI, threat identification and communication with backend. More specifically, two versions were provided, one prior to the pilots and the second one integrating feedback received from them. The later was mainly focused on UI elements that were updated accordingly.

5.3.1 UI Updates – App Version 2

One of the major updates in the application in the second year of the project was the complete. The new UI includes cosmetic changes with colored enabled notifications, a new evaluation form and an additional user story for setting user preferences when the application opens for the first time.
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Figure 5-2: The Privacy Flag Smartphone Application version 2 intro flash screen (a); setting user preferences (b); main menu (c); viewing specific applications with existing evaluation (d-e-f), and; providing user evaluation (g-z)
5.3.2 Automated Evaluation, App Permissions – App Version 2

As described in detail in D4.1 ("First Year Report on Technical Enablers development"), due to the restrictions paused in the smartphone applications from the operating system, the automated evaluation is only related to the user set permissions of an applications. Those are:

- Body Sensors
- Calendar
- Camera
- Contacts
- Location
- Microphone
- Phone
- SMS
- Storage

Each time an application evaluation is submitted, the permissions that are given to this application are also passed on to the backend in order to be included in the evaluation process. The full xml file with the permissions mapping used in the Privacy Flag is provided in Annex A.

5.3.3 Backend Communication – App Version 2

During the second year of the project, the backend communication also was specified and implemented. The back-end communication is used by the add-on to:

- Submit user privacy preferences (only once per application install).
- Retrieve current evaluation of a list of applications.
- Submit user evaluation of an (both permissions and UPRAAM evaluation).
- Retrieve the latest UPRAAM questionnaire.

All communication between the add-on and the backend takes place over HTTP REST request.

5.3.4 UI Updates – App Version 3

As presented earlier, during the second year of the project, user evaluation pilots took place. Using a uniform way to report back to WP4, the pilots reported a number of issues, mainly of user interface, that could be resolved or enhanced. The following list presents the reported feedback.

<table>
<thead>
<tr>
<th>#</th>
<th>Comments</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add the same level of importance to more than one permissions (if not requirement) (home screen)</td>
<td>This in contrast to the design</td>
</tr>
<tr>
<td>2</td>
<td>Tell specifically that they have to order the permissions (home screen)</td>
<td>Fixed - updated text</td>
</tr>
<tr>
<td>3</td>
<td>Confusing that 5 is less and 1 is max in the scale (maybe highlight) (home screen)</td>
<td>Fixed - updated text</td>
</tr>
<tr>
<td>4</td>
<td>If the application which are shown to them are ordered by a certain criterion or randomly (overview of apps)</td>
<td>Fixed - ordered alphabetically</td>
</tr>
</tbody>
</table>

Table 5.1: Pilot comments on App version 2 and performed actions
5 | The selection of friendly, unfriendly and uncategorized lists to be more visible so that they can understand which list they are currently on. (overview of apps) | Fixed - updated UI and background colour

6 | A brief note explaining what its category is would be preferable (overview of apps) | Updated text - should further be discussed

7 | A description of what each permission entails (app evaluation menu) | Updated text - should further be discussed

From the received feedback, some of the recommendations were addressed, while others are to be discussed for future enhancements. Recommendations 2-5 were all related with enhancements of the UI and were all addressed by ordering the applications list alphabetically, updating related text screens and updating the UI background colour and the magnification of the icon of the page selected. For the last two comments, the related text was updated, but these are something that can be discussed in future. Finally, for the first comment, this was against the way the Privacy Flag server collects and stores this information in order to include it to the scoring of the apps.

In addition to the above, the following items were also addressed in the third version of the app:

- In version 2 of the application, the list of the evaluated applications was not refreshing unless the user closed and reopened the app. In version 3, scrolling with the figure down in the main screen will force the list update.
- In version 2, the list of the apps was consuming the whole row of a column, leaving a lot of unused space. In version 3 the list is shown with two columns (Figure 5-3-a).
- When viewing an application, instead of simply showing the permissions, the icons of the selected permissions are also shown (Figure 5-3-b).
- If the version of the Android phone is lower than 6, a specific message appears when viewing the app permissions (Figure 5-3-c).
- The information page of the app has been updated (Figure 5-3-d).
- An error message is shown when the Privacy Flag server is not reachable (Figure 5-3-e).
- The background of each application in the application list is updated.
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Deliverable 4.2 (“Second year report on Technical enablers’ development”)

Figure 5-3: The Privacy Flag Smartphone Application version 3 with new listing of the apps(a) showing icons in the list of permissions and specific message when android version is lower to 6 (b-c) updated information page (d) and error message (e)
5.4 Smartphone Application files and source code

For the moment, the project has not gone public regarding its tools. Nonetheless, the execution file and the source code are available in Privacy Flag servers and they can be installed in Android devices as separate apk files. In the future (Year 3 of the project) the installation will be through Android official store, Google Play.

Application installation instructions (Android) – developer version

- Download the apk file and store it in your mobile phone.
- In the smartphone navigate to Settings -> Security and make user the Unknown sources option is enabled.
- Open the apk file you have downloaded and the application will install.
6. PrivacyFlag Observatory

6.1 Introduction

Privacy threats are an evolving landscape. More sophisticated threats are emerging as new protective mechanisms are enabled, resulting in an arms race between malicious entities and security professionals. Theoretical studies as well as empirical data suggest that the major sources of privacy insecurity are two-fold: either obsolete and insecure technologies that cannot meet today’s privacy standards or new not fully tested powerful protocols and APIs that have not been proven resilient against complex and elaborated attacks.

PrivacyFlag was designed to provide the best available privacy protections and recommendations. The initial focus of task T4.2 was to develop a PrivacyFlag Early Warning System for privacy threats so as to inform PrivacyFlag users on imminent threats. Due to the latest developments on the World Wide Web standards, significant changes have been implemented in the latest generation browsers, mobile operating systems and programming languages. Thus, primitive and easily to intercept privacy attacks are not possible to be monitored. Most of the successful attacks are a combination of various techniques applied in a multilayered approach. In other words, a small weakness in an encryption protocol in a combination with a new powerful API could possible lead to an elaborated fingerprinting attack which can de-anonymize users. Therefore, it is very difficult to automate Early Warnings for most privacy attacks. In addition to that, the most well-known and feasible ways to detect simplistic privacy attacks have been implemented in the latest generation web browsers and mobile operating systems.

On the other hand, the most threatening situation arises not from the well-known vulnerabilities or the most basic privacy attacks, but because of the existence of the future rich platforms that are under specific circumstances can be exploited to attack users’ privacy. These technologies are not harmful per se, but require too much effort to be properly secured. Additionally, most cryptographic mechanisms provide encryption, but not with sufficient strength and robustness under all possible deployments. During the last years as privacy becomes more important, a new level of possible dangers became apparent. Artificial Intelligence (AI), Deep Machine Learning and Big Data are capable of solving the most challenging problems. Despite the beneficial use of these scientific achievements, possible harmful consequences are probable. The combination of vast amount of data among with the deep learning techniques can undermine the resilience of various security technologies.

Therefore, the Privacy Flag project was expanded to address not only currently existing, but most importantly also future privacy threats. To achieve that, the PrivacyFlag Early Warning System was extended to include a PrivacyFlag Observatory. The PrivacyFlag Observatory is focused to provide a holistic overview of the privacy landscape in the modern Internet. The basic idea is to inform users,

---

61 Also see the interesting conceptual discussion presented within: https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/
developers, stakeholders and researchers on the level of adoption of best practices as well as how prevalent are insecure, obsolete and deprecated technologies. Furthermore, interested parties can observe the rate of commitment in privacy related technologies for the most important web sites, since PrivacyFlag is based on crowdsourcing.

6.2 Development Overview

6.2.1 PrivacyFlag Observatory Website – Version 1

The initial PrivacyFlag Observatory included a small list of graphs with an explanatory title.

The depicted charts present the following information items (see Figure 6-1):

- Average number of HTTP Cookies in all websites over time. This shows how the average number of HTTP Cookies varies in time, for the checked sites.
- Percentage of websites that provides data encryption (SSL/TLS). This shows the fraction of the checked sites that employ SSL/TLS in their connections.
- Mean number of Various Cookies in all Websites. This shows the average number of Cookies, of any type, in the checked websites.
- Average number of links to malicious sites in all websites over time. This shows how the average number of malicious links varies, for the checked sites, as time progresses.
- HTTP Cookies Threat Score Interval over time. This shows the variations in the HTTP Cookies threat score as time progresses.

The link for this version of the observatory is the following:

http://150.140.193.133:2080/privacy/addon/metrics.php
6.2.2 PrivacyFlag Observatory Website – Version 2

However, due to the significant importance on the information that can be depicted on the Privacy Flag Observatory, CTI enhanced it by making the following modifications:

- New charts were designed in order to display more information items. More specifically, 17 new graphs were implemented in Version 2 (see Figure 6-2). The full list of charts is the following, with explanations only where the description is not evident from the list items (only part of them is shown in the figure but by visiting the link given below, after the links, one can see all the charts):
  - Percentage of websites that provide data encryption (SSL/TLS) (as in Version 1).
  - Average number of HTTP Cookies in all websites over time (as in Version 1).
  - Mean number of Various Cookies in all Websites (as in Version 1).
  - Average number of links to malicious sites in all websites over time (as in Version 1).
  - First and Second Interval of the change of HTTP Cookies in all websites over time.
  - Tables’ size in KB (the sizes of the relevant database tables).
  - Number of tables’ rows (the number of rows in the relevant database tables).
Percentage of websites that use technologies with known security issues.
Percentage of websites that use potentially dangerous advanced HTML5 APIs.
Percentage of websites that use user following techniques.
Percentage of websites that provide HSTS.
Percentage of websites that the encryption method (cipher suite) negotiated between client and server is considered as secure.
Percentage of websites that use a trustworthy certification chain.
Percentage of websites that use Certificate pinning.
Percentage of websites that use HTML5 Web SQL database.
Percentage of websites that use technologies with known security issues - Flash.
Percentage of websites that use potentially dangerous advanced HTML5 APIs - Web Audio API.
Percentage of websites that use potentially dangerous advanced HTML5 APIs - WebRTC.
Percentage of websites that use technologies with known security issues - ActiveX.
Percentage of websites that use technologies with known security issues - Java.
Percentage of websites that use technologies with known security issues - Silverlight.

A new library was selected in order to generate the charts. More specifically, the new library is the Google Charts (https://developers.google.com/chart/). There are many reasons behind our decision to adopt the specific library. The most important of them are the following:

- There is a wider variety of available charts, each providing different information content.
- The quality of generated graphs is higher (e.g. in terms of resolution and colours).
- Data is now retrieved in real time based on the BD using PHP or JavaScript.
- Charts are rendered using HTML5/SVG technology to provide cross-browser compatibility (including VML for older IE versions) and cross platform portability to iPhones, iPads and Android.
- No plugins are needed.
- There is an extensive set of options to perfectly configure the charts.
- The library is provided free of charge.

Moreover, using this library more detailed information can be displayed for each observatory chart when the mouse is hovering above it.

The new version of the Privacy Flag Observatory is available via the following link: http://150.140.193.133:2080/privacy/addon/new_metrics.php

---

62 For more details also see: https://www.w3schools.com/php/
63 For more details also see: https://www.w3schools.com/graphics/svg_intro.asp
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Privacy Flag
Threat Observatory / Early Warning System
(1) Percentage of websites that provide data encryption (SSL/TLS).

(2) Average number of HTTP Cookies in all websites over time.

(3) Mean number ofVersion Cookies in all Websites.

(4) Average number of link transitions in all websites over time.

(5) First and Second Interval of the change of HTTP Cookies in all websites over time.

(6) Table data in LB

Figure 6-2: Privacy Flag Observatory - Version 2

Deliverable 4.2 ("Second year report on Technical enablers’ development")
6.2.3 PrivacyFlag Observatory Website – Version 3

According to the report from the pilots that took place, there were the following three recommendations regarding the Privacy Flag Observatory:

- More detailed and in depth information about each chart is provided.
- Have some more information and guidelines about what each risk entails.
- Split the charts into information categories.

Therefore, a new version of the observatory was implemented in order to fulfil the reported recommendations. In Figure 6-3, one can see part of the new information items. Let us consider, for instance, the chart shown in this figure. The new version not only gives the percentages of the websites that employ or do not employ SSL/TLS, but also gives explanations about the involved threats and risks, enhancing the observatory’s value to the professional as well as informed audience.

The latest version of the PrivacyFlag Observatory is available via the following link, where all the new information items can be seen along with detailed explanations about their function:

http://150.140.193.133:2080/privacy/addon/observatory.php
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

**Privacy Flag**

**Threat Observatory / Early Warning System**

**Categories**

The PrivacyFlag Observatory is organized in three distinct categories: Confidentiality, Security and Privacy of Data. All of them are related to the Privacy of your Data in direct or indirect way. Find why:

<table>
<thead>
<tr>
<th>Confidentiality</th>
<th>Security</th>
<th>Privacy</th>
</tr>
</thead>
</table>

**Confidentiality**

Data encryption is the basic mechanism to protect the confidentiality of your information to remain private. It is absolutely necessary to encrypt sensitive data as passwords, credit card number etc but it is even better to encrypt everything. Modern web sites provide various encryption mechanisms. In PrivacyFlag we check whether a web site respects users privacy by encrypting his/her data. Furthermore, PrivacyFlag also analyzes the robustness and strength of the implemented encryption algorithms. Bear in mind that obsolete, weak or poorly implemented encryption algorithms offer little or no protection at all against skilled adversaries.

**Percentage of websites that provide data encryption (SSL/TLS).**

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) are standard security technologies for establishing an encrypted link between a server and a client—typically a web server (website) and a browser. The strength of the protection mechanism is determined by the authentication, encryption, and hashing algorithms, collectively known as a cipher suite, chosen for the transmission of sensitive information over the TLS/SSL channel. There are some cipher suites known to be broken or weak. SSL-enabled services should be configured to disable these insecure cipher suites.

![Figure 6-3: Privacy Flag Observatory - Version 3](image)

**6.2.4 Categories of Charts**

As shown in the top of Figure 6-3, the latest version of the PrivacyFlag Observatory is organized in three distinct categories, that is: Confidentiality, Security and Privacy of Data. All of them are related to the Privacy of your Data theme in a direct or indirect way.

**6.2.4.1 Confidentiality**

Data encryption is the basic mechanism to protect the confidentiality of your information to remain private. It is absolutely necessary to encrypt sensitive data as passwords, credit card number, etc., but it is even better to encrypt everything. Modern web sites provide various encryption mechanisms. In Privacy Flag we check whether a web site respects users’ privacy by encrypting his/her data. Furthermore, Privacy Flag also analyzes the robustness and strength of the implemented encryption algorithms. Bear in mind that obsolete, weak or poorly implemented encryption algorithms offer little or no protection at all against skilled adversaries.
The charts in this category are the following:

- Percentage of websites that provide data encryption (SSL/TLS) (Figure 6-4).
- Percentage of websites that provide HSTS (Figure 6-5).
- Percentage of websites that use a trustworthy certification chain (Figure 6-6).
- Percentage of websites that use Certificate pinning (Figure 6-7).

**Figure 6-4: Percentage of websites employing SSL/TLS**

**Figure 6-5: Percentage of websites providing HSTS**
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

6.2.4.2 Security

In Internet nothing can be 100% secure. On the other hand, there are some technologies that are less secure than others. Usually, more prone to security defects are either obsolete and deprecated solutions that are no longer up to modern standards or new untested solutions that despite good design intentions do not meet always all the requirements. Nonetheless, some of these technologies are quite prevalent but they should be used with caution.

The charts in this category are the following:

- Average number of links to malicious sites in all websites over time (Figure 6-8).
Percentage of websites that use technologies with known security issues (Figure 6-9).

Percentage of websites that use potentially dangerous advanced HTML5 APIs (Figure 6-10).

Percentage of websites that use following techniques (Figure 6-11).

Percentage of websites that use Web SQL Database or HTML5 IndexedDB (Figure 6-12).

Percentage of websites that use technologies with known security issues – Flash (Figure 6-13).

Percentage of websites that use potentially dangerous advanced HTML5 APIs – Web Audio API (Figure 6-14).

Percentage of websites that use potentially dangerous advanced HTML5 APIs – WebRTC (Figure 6-15).

Percentage of websites that use technologies with known security issues – ActiveX (Figure 6-16).

Percentage of websites that use technologies with known security issues – Java (Figure 6-17).

Percentage of websites that use technologies with known security issues – Silverlight (Figure 6-18).

**Figure 6-8: Average number of links to malicious sites**

**Figure 6-9: Percentage of websites using technologies with known security issue**
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Figure 6-10: Percentage of websites using potentially risky advanced HTML5 APIs

Figure 6-11: Percentage of websites using following techniques

HTML5 is the new Web standard; it is definitely the future of the WWW and is here to stay. As it happens with any new powerful technology, it provides a set of powerful features. On the other hand, some of these might have some security and privacy issues. PrivacyFlag have identified some potential problematic, from privacy perspective, technologies. Unless there is a good reason for a web site to use this functionality, it is better not enable them yet.

A summary of the newest, powerful yet not fully tested new technologies. As times goes by they will become more mature and robust.
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

---

**Figure 6-12: Percentage of websites using Web SQL Database or HTML5 IndexedDB**

Percentage of websites that use technologies with known security issues - Flash.

---

Flash was once the king of multimedia content and was used thoroughly on the Web. Most web sites delivered multimedia video almost exclusively in Flash. Unfortunately, Flash protocol was also ranked high as the major source of vulnerabilities and other security risks. Therefore, most modern web sites tend to abandon the Flash protocol in favor of a newer multimedia codec. So, though not always possible to avoid Flash, at all, try to use web sites with HTML5 video native players to enjoy your web video experience safely.

---

**Figure 6-13: Percentage of websites using technologies with known issues – Flash**
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

**Figure 6-14: Percentage of websites using potentially dangerous advanced HTML5 APIs – WEBRTC**

**Figure 6-15: Percentage of websites using potentially dangerous advanced HTML5 APIs – Web Audio API**
Deliverable 4.2 ("Second year report on Technical enablers’ development")
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

6.2.4.3 Privacy

All web sites use some mechanisms to store information regarding users’ preferences. This information should be related only to viewing and browsing preferences and not to store personal identifiable data of the users or their overall browsing activity. PF analyzes various tracking mechanisms and informs web users about the intrusiveness of some methods.

The charts in this category are the following:

- Average number of HTTP Cookies in all websites over time (Figure 6-19).
- Mean number of Various Cookies in all Websites (Figure 6-20).
- First and Second Interval of the change of HTTP Cookies in all websites over time (Figure 6-21).
- PrivacyFlag DB size (Figure 6-22).

![Percentage of websites that use technologies with known security issues - Silverlight.](image)

Silverlight is a new Microsoft technology based on the .NET framework. It is used for the development of highly interactive applications to enrich user experience. .NET as every middleware with direct access to your PC may be a security risk. Better avoid it, if not absolutely necessary.

**Figure 6-18:** Percentage of websites using technologies with known security issues – Silverlight

![Average number of HTTP Cookies over time](image)

It is useful to have a good estimation of the average number of cookies per site. An unusual high number of cookies in a web site is an indication of privacy issues. Watch out for outliers, web sites with too many HTTP Cookies!

**Figure 6-19:** Average number of HTTP Cookies over time
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

![Mean number of Various Cookies in all Websites.](image)

Despite the mostly innocent HTTP Cookies, other types of cookies such as SuperCookies, ZombieCookies, EverCookies, LSO etc are really persistent and very difficult to remove. Here you can find an estimation of how common are to most websites.

**Figure 6-20:** Mean number of various cookies in all websites

![First and Second Interval of the change of HTTP Cookies in all websites over time.](image)

PrivacyFlag uses also some simple but well known and effective epidemiological models to monitor the increase or decrease of privacy threats. The first curve is the rate of new incidents and the second one is whether PrivacyFlag observes an increase or decrease of new privacy issues related to cookies. This is an idea borrowed from the epidemiology, utilizing a well-known and important metric, the epidemic curve.

**Figure 6-21:** First and second interval of the change of HTTP Cookies in all websites over time

Deliverable 4.2 (“Second year report on Technical enablers’ development”)
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Figure 6-22: The total amount of information currently processed
7. Distributed Agents

The PrivacyFlag Distributed Agents are a virtual subcomponent responsible for the utilization of the crowdsourcing intelligence of the PrivacyFlag platform. In technical terms, the PrivacyFlag Distributed Agents are the frontend functionality of the PrivacyFlag Evaluation Component that performs part of the automatic analysis in the browser and communicates the results back to the PrivacyFlag DataBase. More to that, it also submits the results of the UPRAAM score and receives from the PrivacyFlag Evaluation Component the final recommendation after the combined evaluation of the UPRAAM score, the automatic analysis of the Top25 Threat Matrix and the PrivacyFlag Early Warning System input, as described in more detail in the next sections.

The primary tasks of the PrivacyFlag Distributed Agents are to enable the crowdsourcing aspect of the PrivacyFlag platform, as well as to perform part of the Top25 Threat Matrix automatic analysis in the frontend. The basic responsibilities of the PrivacyFlag Distributed Agents are to collect the required URLs to analyse. The PrivacyFlag project aims to be more accurate on other similar platforms, because it is based on the crowdsourcing intelligence of its users to identify the web sites that should be analysed instead of using automated web crawlers or static lists. Thus, only web sites that have been visited by actual users and, therefore, are important are tested, having the results of the analysis stored in the PrivacyFlag DataBase and used as a reference for any newer evaluation. To implement this functionality, the PrivacyFlag Distributed Agents are developed for a number of tasks. First, they acquire each new URL that the user has been visited and submit them to the PrivacyFlag Server for the main part of the automatic analysis for privacy threats. For scalability and performance reasons most of the Top25 Threat Matrix is checked on the PrivacyFlag Evaluation Component. Nonetheless is also necessary to inspect some of the threats in the PrivacyFlag WebAdd-on as some elements (e.g. cookies) are specific to each user and therefore is not possible to have them tested in the backend. Consequently, for technical reasons some part of the automatic analysis should be performed in the frontend. Thus, it was necessary to split some of the process between the PrivacyFlag frontend and the backend of the PrivacyFlag Evaluation Component. In general, the PrivacyFlag Distributed Agents are a remote part of the PrivacyFlag Evaluation Component that will be discussed in the next section.
8. Evaluation Component

The PrivacyFlag Evaluation Component (please see the PF General Architecture figure in Section 2) is a virtual super-component, which includes the PrivacyFlag DataBase and the PrivacyFlag Distributed Agents. It is operating on the PrivacyFlag Server on top of the PrivacyFlag DataBase. In technical terms, it is a complex middleware which is responsible for the all necessary mathematical calculations to support the required analysis for the extraction of the PrivacyFlag Observatory statistical data, the outlier detection mechanisms for the PrivacyFlag Early Warning System, the evaluation of the web sites and the smart phone applications. In other words, every algorithmic aspect of the PrivacyFlag automatic analysis is part of the PrivacyFlag Evaluation Component.

The PrivacyFlag Evaluation Component, at its lowest level, utilizes complex SQL queries to extract information from the PrivacyFlag DataBase. To make this information available to other components of the PrivacyFlag platform, server side JavaScript is employed to provide communication APIs in order to obtain and pass information from the PrivacyFlag DataBase to PrivacyFlag Distributed Agents. The main part of the Top25 Threat Matrix is also evaluated in the PrivacyFlag Evaluation Component. The codebase for that purpose has been implement in Python64.

The results of all tests are stored in the PrivacyFlag Database, but these also “passed” to all other subcomponents, the PrivacyFlag Observatory, the PrivacyFlag Early Warning System and the PrivacyFlag Distributed Agents. Periodically, the PrivacyFlag Evaluation Component updates the statistics, the trends and the graphs of the PrivacyFlag Observatory. In addition to that, every time a new URL is received from a PrivacyFlag Distributed Agent the Top25 Threats tests should be executed in the PrivacyFlag Evaluation Component and, together with the results of the checks run locally on the front-side by the PrivacyFlag Distributed Agents, the UPRAAM score and the PrivacyFlag Early Warning System input, a decision is made by the Evaluation Component whether the website is privacy friendly or not. Similarly, the privacy score of the Smart Phone Applications is calculated in the PrivacyFlag Evaluation Component after the inspection of the list of the installed application and their respective permissions.

Finally, during the check of each new URL and SmartPhoneApp (only one evaluation is active at each time), but also during the scheduled periodical updates of the PrivacyFlag Observatory, extended outlier detection procedures are executed both in local and global perspective. If a web site or smart phone application scores very differently from the expected averages, a flag is raised which affects heavily the evaluation assigned to them. On the other hand, if some rapid -but constant- change appears in the general trends of the metrics presented in the PrivacyFlag Observatory, which seem inconsistent with the PrivacyFlag epidemiological models, a more generic Early Warning is issued to the PrivacyFlag Distributed Agents.

Due to its extended multi-role functionality, the PrivacyFlag Evaluation Component acts as the hub of all the other subsystems in order to facilitate the passing of information between the different

---

64 For more details see: https://www.python.org/
subcomponents. More to that, the PrivacyFlag Evaluation Component is the central analytical engine for the PrivacyFlag platform. It incorporates many different technologies and frameworks that work together to support all the required functionalities of the PrivacyFlag project. The PrivacyFlag is distributed in the sense that also includes the PrivacyFlag Distributed Agents, but the overwhelming majority of the code and functionality is implemented as a middleware in the PrivacyFlag Server.
9. Early Warning System

Privacy Flag initially emphasized in task T4.2 the PrivacyFlag Early Warning System. Due to the volatile and evolving nature of the web privacy landscape, it is not possible to predict imminent complex attacks. Therefore, the PrivacyFlag Early Warning System was initially targeting malware attacks that could affect also users’ privacy. It is also become evident that most modern mobile operating systems as well as new browsers allow limited integration with stand-alone client applications that could effectively detect malware or even interact with existing antivirus solutions, even if it were part of the operating system (i.e. Windows Defender). These developments limited significantly the available malware detection options. As a fall-back approach, the malware identification subsystem has been completely integrated to the PrivacyFlag WebAdd-on in a different form. The PrivacyFlag Early Warning System is partially implemented on the PrivacyFlag WebAdd-on, but the main functionality lies in the PrivacyFlag Server in the PrivacyFlag Evaluation Component. The part that operates in the PrivacyFlag WebAdd-on submits the URL under question to PrivacyFlag Evaluation Component, which utilizes the Google Safe Browsing API\(^{65}\) to detect websites that host malware. This approach can provide some basic malware protection, but one must consider that the size of Google’s blacklist with malware infected sites does not exceed more than a few hundred thousand sites in the vast amount of the hundreds of millions of existing web sites. As a consequence, only on very rare occasions this type of alert is activated. What is important though for the Privacy Flag project is that due to crowdsourcing intelligence, the PrivacyFlag Early Warning System gets only information and analyzes web sites that actual users visit, and is not based on static lists or automatic crawlers. Therefore, the PrivacyFlag platform has more realistic input than other automatic or semi-automatic approaches. The PrivacyFlag Early Warning system applies well-known epidemiological models to this information (i.e. the number of infected sites) and is able to calculate the epidemic curve and the rate of new infections. Nonetheless, due to the paramount asymmetry between infected and existing web-sites, the results are almost always “flat”, despite short periods of a malware epidemic in which an activity should be monitored and alerts will be issued to the PrivacyFlag Distributed Agents.

Based on the available data of the PrivacyFlag Evaluation Component and the PrivacyFlag Observatory, the PrivacyFlag Early Warning System has been transformed to an effective outlier detection mechanism. The algorithmic approach to this functionality is to check all non-Boolean Top25 Threat Matrix entries for abnormal values. There are numerous rigid mathematical methods to perform this analysis. Currently in the PrivacyFlag Evaluation Component the following have been implemented and tested or are under evaluation:

- Peirce’s criterion\(^{66}\),
- Pearson’s correlation\(^{67}\).

---

\(^{65}\) For more details see: [https://developers.google.com/safe-browsing/](https://developers.google.com/safe-browsing/)

\(^{66}\) For more informative details also see, for example: [https://en.wikipedia.org/wiki/Peirce%27s_criterion](https://en.wikipedia.org/wiki/Peirce%27s_criterion)

\(^{67}\) For more informative details also see, for example: [https://en.wikipedia.org/wiki/Pearson_correlation_coefficient](https://en.wikipedia.org/wiki/Pearson_correlation_coefficient)
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Deliverable 4.2 (“Second year report on Technical enablers’ development”)

z-score\(^68\), Chebyshev’s theorem.\(^69\)

Each time an outlier is detected, the site is marked suspicious and alarm is raised. The outliers are both used in the Evaluation Component as an extra source of input in Boolean form (true/false), but they also are passed to the PrivacyFlag Early Warning System which is responsible to push this information back to the PrivacyFlag WebAdd-on and the PrivacyFlag SmartPhoneApp to notify users. The integration with the Privacy Flag WebAdd-on and the PrivacyFlag SmartPhoneApp is expected to be delivered in the second version of PrivacyFlag platform in June 2017, which will be rolled out with enabled Early Warning Services.

A different approach is followed on the general trends of the PrivacyFlag Observatory. These data are being treated as time series. Therefore, is possible to apply basic epidemiological models on them and extracted the corresponding epidemic curves. If the predefined thresholds are surpassed, Early Warnings will be sent to Privacy Flag users.

\(^{68}\) For more informative details also see, for example: https://en.wikipedia.org/wiki/Standard_score

\(^{69}\) For more informative details also see, inter-alia: https://en.wikipedia.org/wiki/Chebyshev%27s_theorem
10. Database and Server Implementation

In this section, the current state of development for the database and server infrastructure for the Privacy Flag platform is presented. The core component of the Privacy Flag platform is the main server which is comprised of the following main components: a web server that makes available all the necessary web services and a database where all data is stored.

10.1 Database Schema

During the 2nd year of the development, two different databases were designed and developed. The first one stores the data for the Privacy Flag add-on; the second one stores the data regarding the Privacy Flag Smartphone application.

10.1.1 Database schema for the PF add-on

The database schema for the add-on can be seen in Figure 10-1. Its role is to organize the information about evaluated sites submitted by users through the PF add-on on their devices using the UPRAAM questionnaires.

The schema’s goal is to store in the database tables the evaluated websites, the scores of each threat for the visited website, the questions and the submitted answers to the UPRAAM questionnaire, the usernames of users (or device IDs), etc. A submission timestamp is also stored in order to extract useful data. Moreover, epidemiological models are used in order to evaluate the threats in the “threatAnalysis” table. Finally, outlier detection mechanisms are used in order to identify if an automatic analysis score is a normal value or not. This will be used by the Early Warning System as well in order to inform the users about the visited website. All the aforementioned information is stored to the database and thus, the add-on is able to provide an automatic evaluation of a website.
Figure 10.1: Database schema for the PF Add-on
10.1.2 Database schema for the PF smartphone application

The database schema for the smartphone application is shown in Figure 10-2. Its role is complementary to that of the add-on schema shown before and it is used to organize the information sent by the smartphone application about evaluated installed applications on their devices.

More specifically, the smartphone application database schema stores the evaluated applications that users have installed to their devices (and which application belongs to each user), the permissions that each application has, the permission group that each permission belongs to, the categories that users have to order according to their preferences, the UPRAAM questionnaire and its answers. An automatic evaluation of each application is performed when requested by the user and it uses both the UPRAAM score and the users’ ordering of the permission categories using the Borda scoring rule (Saari 2001). Figure 10-2 depicts the database schema in detail.
Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

### Database schema for the PF smartphone application

#### Submissions
- **id**
- **user id**
- **app id**
- **submission time**

#### Questions
- **id**
- **question**

#### Answers
- **id**
- **question id**
- **answer id**
- **answer**
- **answers score**

#### Upraam Answers
- **submission id**
- **answer id**

#### Permissions
- **perm id**
- **perm name**
- **group id**

#### Categories
- **id**
- **category**
- **ranking**

#### Applications
- **id**
- **app name**
- **package**
- **crowd evaluation**
- **UPRAAM score**
- **total evaluation**

#### Users
- **id**
- **username**
- **device score by crowd**

#### Perm_per_app
- **app id**
- **perm id**

#### Apps per User
- **user id**
- **app id**

#### Groups
- **group id**
- **group name**
- **category id**

#### Permissions
- **user id**
- **category id**
- **level**

*Figure 10-2: Database schema for the PF smartphone application*
10.2 Git server

During the 2nd year of development, the git server presented in the deliverable D4.1 is still running and all information provided to D4.1 are still valid, except the server’s IP address which has changed.

10.3 APIs

Several APIs were developed in the 2nd year of the project in order to provide various functionalities for both the Add-on and the Smartphone application. The API is used as an end-point communication between the services and the database. More specifically, regarding the Add-on, there are developed the following functionalities:

1. Get the existing evaluation of a website.
2. Get the latest questionnaire.
3. Send user’s answers to the questionnaire and some of the evaluated website threat scores.

Regarding the Smartphone application, the API functionalities are the following:

- Get the existing evaluation of an application.
- Get the latest questionnaire.
- Send user’s answers to the questionnaire and evaluated application’s permission information.
- Send user preferences, that is user’s ranking of the categories.

The above API calls have various requirements regarding the JSON format of the exchanged data between the Add-on (and Smartphone) and the API. However, this technical information (links for the HTTP calls, JSON format, etc.) is out of the scope of this deliverable and should not be available for security purposes.
11. Website and backend management platform

The website frontend and backend are developed to support effort from different work packages to be presented to the audience (Smartphone application and browser Add-on), to provide features of different enablers to be used over web UI for the single administrator user (i.e., change questions for browser Add-on and Smartphone application); to provide a mean for the WP6 to report their activities, as well as to provide assessment result of websites and smartphone application in form of a ranking list.

The Voluntary commitment tool (Privacy Pact) and the In-depth evaluation tool (Privacy certificate) are developed as separate units, isolated from the rest of the website. The following section presents the website frontend and backend, describing the key features and the architecture.

11.1 Frontend

The website front end is composed of several units with a focus on presenting Privacy Flag tools to the end-user, user engagement and result of assessment of websites and phone application. The list of assessed websites is given with limited list of criteria, thus just a fragment of real assessment mechanism (criteria) is presented that might be of interested to the non-technical users.

The assessment list as it appears in Figure 11-1 provides better insight into different websites over the following criteria:

1. If website certificate is secure or not.
2. If website uses trustworthy certification chain.
3. If HTTPS is used or not.
4. If it’s possible to fingerprint using different users’ patterns or not.

Before showing the list, when a user accesses this website page, the popup indicating the purpose of the list and disclaimer for assessed websites is showed. The number of criteria in the table is not final and it will be extended and depicted with attention from large set of possible criteria that project will support. This process will be done by consortium legal experts, technical partners and partners with expertise on communication with the end-user.
The assesment table for the Smartphone application is an ongoing work that will provide a means for the users to be more familiar with the privacy of installed as well as for widely used Smartphone application.

The assessment Smartphone app list will provide better insight into privacy risk that the application may impose. Before showing the list, when a user accesses the website page, the popup is showed indicating the purpose of the list and disclaimer for assessed applications. The number of criteria in the table is still under discussion and will be depicted with attention from a set of possible criteria for the Android app that the PF project supports. This process will be done by the PF consortium legal experts, technical partners and partners with expertise on communication(s) with the end-user.
The user engagement (see Figure 11-2) as a quality of the user experience that emphasizes the positive aspects of interaction – in particular the fact of user being captivated by the technology (Mounia Lalmas 2013) is ensured with Facebook and Twitter social networks widgets embedded in the website to attain visitors that come to the webpage through social media.

*A main goal of user’s retention is to ensure as much users as possible for the final version of the Privacy Flag Tools, presented at the landing page. The tools are presented as showed in the image below: Short*
description with a link to the full description and embedded video that explains what is the Privacy Flag doing for the end-users, what users could achieve and what the benefits are of these tools.

Privacy Flag Tools

- **Tool 1**: Privacy Flag browser add-on [Read more](#)
- **Tool 2**: Privacy Flag smartphone application
- **Tool 3**: Privacy Software as a Service (PSaaS)

![Privacy Flag tools](image)

*Figure 11-3: Privacy Flag End-user Tools presentation*

### 11.2 Backend

The aim of the Privacy Flag project is to combine the potential of crowdsourcing, ICT technologies and legal expertise to protect citizens’ privacy when visiting websites or using smartphone applications. To fulfil the legal requirements when using technology, even when there are anonymizations mechanisms provided by the WP2, one of the approaches was to avoid keeping records of users’ data by not including creation of user accounts functionality as a feature. In addition, citizens as end-users can use tools and the website, without creating any accounts at all. There are two backends: one for consortium users that can modify and generate content based on open source Wordpress platform [www.wordpress.org](http://www.wordpress.org) and a second one developed independently for administrator user(s) to enable complementary features for the tools and enablers as follows:

- **Question moderation module** enables administrator to login and change questions that appears for the smartphone application and the browser add-on.
- The list of questions is extracted over the API from the database. User can change existing question and can add additional questions. Synchronization when questions changed is done from the browser Add-on/ smartphone app itself.
- **Website/Smartphone application privacy risk list** is a module that shows a list of assessed website/app and to enables administrator to add/remove specific website/application that is being showed at the frontend. Administrator could also change the disclaimer popup and review potential requests for removal of website/smartphone application if certain conditions are satisfied.
Privacy Flag Threat Observatory/ Early Warning System developed in WP4, enables administrator user to evaluate different metrics and to manually monitor following analytics from the Privacy Flag system:

- Percentage of websites that provides data encryption (SSL/TLS)
- Average number of HTTP Cookies in all websites over time.
- Mean number of Various Cookies in all Websites.
- Average number of links to malicious sites in all websites over time.
- First and Second Interval of the change of HTTP Cookies in all websites over time.
- Percentage of websites that use technologies with known security issues.
- Percentage of websites that use potentially dangerous advanced HTML5 APIs.
- Percentage of websites that use a trustworthy certification chain.
- Percentage of websites that use Certificate pinning.
- Percentage of websites that use HTML5 Web SQL database.
- Percentage of websites that use technologies with known security issues – Flash.
- Percentage of websites that use potentially dangerous advanced HTML5 APIs - Web Audio API, WebRTC.
- Percentage of websites that use technologies with known security issues - ActiveX, Java, Silverlight.

11.3 Architecture

The architecture of the frontend and the backend management platform in the PF Deliverable D1.3, which is depicted in Figure 11-4, is based on the php/MySQL Apache technology (McCool 1995). The website backend is composed of two different sub-backends. One of these sub-backends is used for the generation of the frontend content deployed in Wordpress CMS, which is also separated from the
rest of the platform (database and APIs) due to some well-known security issues (CVE 2017). The other sub-backend was custom-made and was developed from scratch, based on html/css/javascript/php.

Figure 11-4: The privacy Flag Website and backend management platform architecture
12. List of References


Enabling Crowd-sourcing based privacy protection for smartphone applications, websites and Internet of Things deployment (Privacy Flag) GRANT AGREEMENT No.653426

Deliverable 4.2 (“Second year report on Technical enablers’ development”)