<table>
<thead>
<tr>
<th>Section</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Getting Started</td>
<td></td>
</tr>
<tr>
<td>1.1 Glossary</td>
<td>3</td>
</tr>
<tr>
<td>1.2 Features</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Requirements</td>
<td>5</td>
</tr>
<tr>
<td>2 Setup</td>
<td></td>
</tr>
<tr>
<td>2.1 Installation Guide</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Platform Wizard</td>
<td>7</td>
</tr>
<tr>
<td>3 Security</td>
<td></td>
</tr>
<tr>
<td>3.1 Threat Model</td>
<td>13</td>
</tr>
<tr>
<td>3.2 Application Security</td>
<td>19</td>
</tr>
<tr>
<td>3.3 Encryption Protocol</td>
<td>26</td>
</tr>
<tr>
<td>3.4 Penetration Tests</td>
<td>28</td>
</tr>
<tr>
<td>4 User Documentation</td>
<td></td>
</tr>
<tr>
<td>4.1 Common to All Users</td>
<td>29</td>
</tr>
<tr>
<td>4.2 For Administrators</td>
<td>36</td>
</tr>
<tr>
<td>4.3 For Recipients</td>
<td>64</td>
</tr>
<tr>
<td>4.4 For Whistleblowers</td>
<td>67</td>
</tr>
<tr>
<td>5 Developer Documentation</td>
<td></td>
</tr>
<tr>
<td>5.1 Development Environment</td>
<td>71</td>
</tr>
<tr>
<td>5.2 Software Libraries</td>
<td>73</td>
</tr>
<tr>
<td>5.3 Release Procedure</td>
<td>73</td>
</tr>
<tr>
<td>5.4 Continuous Integration</td>
<td>74</td>
</tr>
<tr>
<td>5.5 Code Documentation</td>
<td>74</td>
</tr>
<tr>
<td>Index</td>
<td>75</td>
</tr>
</tbody>
</table>
GlobaLeaks is an open-source, free software intended to enable anyone to easily set up and maintain a secure whistle-blowing platform.

**Note:** This documentation is thank to the support of our community. Join us and contribute with your additions and suggestion. In any of the page you find a link that enables you to provide suggestions and corrections. We remind you that in case of any software issue or bug you may always report on the ticketing system.
CHAPTER ONE

GETTING STARTED

1.1 Glossary

This is a list of jargon that is used inside of the GlobaLeaks software. Here we define it in a way to be unambiguous and uniform across all of their references.

**Admin**  The user that has setup and is maintaining the Platform.

**Context**  A topic covered by a GlobaLeaks based project (ex. Corruption).

**GlobaLeaks**  Our project, a FLOSS, security focused and flexible whistleblowing framework.

**Notification**  Method used to notify a Recipient that a new Submission or an update of an existing Submission is available.

**Platform**  A system running the GlobaLeaks framework.

**Questionnaire**  A questionnaire is a set of questions that the whistleblower should fill out for completing a Submission to a Platform.

**Receipt**  An access token that allows the Whistleblower to be authenticated to the secret (it effectively acts as a password) generated by the Platform and returned to the Whistleblower when a Submission has been completed. It enables the Whistleblower to modify and update the submission eventually uploading new files, adding comments to the Tip, and reading Receivers’s comments about the submitted materials.

**Recipient**  It is the receiver of an anonymous Submission sent by the Whistleblower.

**Submission**  The action performed by the Whistleblower to submit a Tip to the Platform. This action is performed by the Whistleblower in respect to the Context requirement.

**Whistleblower**  A user that is reporting to the Platform by uploading material and filling out the submission form.

1.2 Features

1.2.1 User Features

- Multi-user system with customizable user roles (whistleblower, recipient, administrator)
- Entirely manageable from a web administration interface
- Support for more than 40 languages with support for Right-to-Left (RTL)
- Let whistleblowers decide if and when to confidentially declare their identity
- Exchange multimedia files with whistleblower
- Chat with Whistleblower to discuss the report
• Unique 16-digit receipt for the whistleblower to log back in anonymously
• Simple recipient interface for receiving and analyzing reports
• Support for the categorization of the reports with labels
• Support for the user search of reports
• Support for assigning and creating case management statuses
• Customizable look and feel (logo, colour, styles, font, text)
• Define multiple reporting channels/contexts (e.g. per-topic, per-department)
• Manage multiple whistleblowing sites from a single dashboard
• Advanced questionnaire builder
• Whistleblowing system statistics

1.2.2 Legal Features

• Ready-made questionnaires for compliance law
• Bidirectional anonymous communication (comments/messages)
• Customizable case management workflow (statuses/sub-statuses)
• Whistleblower identity conditional reporting workflow
• Manage conflict of interest in the reporting workflow
• Custodian functionality to authorize access to whistleblower identity
• GDPR privacy by design
• GDPR configurable data retention policies
• GDPR compliant subscriber module for new users of SaaS services
• No logs of IP addresses
• Integratable with existing enterprise case management platform
• Free Software OSI Approved AGPL 3.0 License

1.2.3 Security Features

• Full data encryption of whistleblower reports and recipient communication
• Digital anonymity support with Tor integration
• Built-in HTTPS support with TLS 1.3 standard (SSLabs A+ rating)
• Automatic free digital certificate enrollment (Let’s Encrypt)
• Multiple penetration tests with full public reports
• Conform to industry standards and best practices for application security (OWASP)
• Two-Factor authentication (2FA) support compliant with standard TOTP RFC 6238
• Integrated network sandboxing with iptables
• Integrated application sandboxing with AppArmor
• Complete protection against automated submissions (spam prevention)
• Subject to continuous peer-review and periodic security audits
• PGP support for encrypted email notifications
• Does not leave traces in browser cache

1.2.4 Technical Features

• Multi-site support enabling to run multiple virtual site on the same setup
• Responsive UX made with Bootstrap CSS Framework
• Built-in Accessibility Support with WAI-ARIA compliance
• Automated Software Quality Measurement and Continuous Integration Testing
• Long-Term Support plan (LTS)
• Built with lightweight framework technologies (AngularJS and Python Twisted)
• Embedded database - SQLite (optional support for other databases)
• Automatic setup of Tor Onion Services Version 3
• Integrated backup support
• Support for self-service signup for whistleblowing SaaS service setup
• Support for Linux operating system (Debian/Ubuntu)
• Deb Packaging with repository for update/upgrades
• Fully self-contained application (no web or application servers needed)
• Integrated collaborative translation/localization (Transifex)
• Easy integration of the platform with existing websites and intranets
• HTTP/2 support
• Rest API
• Provisioning API

1.3 Requirements

Make sure you understand and satisfy each of the following technical requirements.

1.3.1 Hardware Requirements

Requirements:
• CPU: Dual core 2.0GHz
• RAM: 1GB
• STORAGE: 20GB
• I/O: 10Mbit/s (shared)

Please note that GlobaLeaks is designed to run on servers even smaller than the above configuration.
The storage size should be defined based on your data retention policies and the expected use of the platform.
1.3.2 Software Requirements

GlobaLeaks is designed to run on GNU/Linux and developed and tested specifically for Debian based systems. The software lifecycle of the platform includes full support for all Debian and Ubuntu LTS versions starting from Debian 10 and Ubuntu 20.04.

On these platforms the support is guaranteed following the official long term support timelines:

- Release End of Life Timeline defined by Debian;
- Release End of Life Timeline defined by Ubuntu.

Support for more distributions is planned.
2.1 Installation Guide

The following is intended to guide you through the installation of GlobaLeaks.

Before starting the installation, make sure that your system satisfy the Requirements.

**Warning:** GlobaLeaks is built to give the best technical anonymity to the Whistleblower. In addition the software with specific configurations enables the possibility to protect the identity of the platform administrator and the server’s location but this requires advanced setup procedures not considered in this simplified installation guide. By executing the commands below your IP address and the location of your system could be tracked by the network providers and as well our systems will be receiving the same information in order to satisfy the provisioning of the software.

**Install GlobaLeaks**

In order to install GlobaLeaks Copy & Paste the following commands in your terminal:

```bash
wget https://deb.globaleaks.org/install-globaleaks.sh
chmod +x install-globaleaks.sh
./install-globaleaks.sh
```

At the end of the process a web interface will be reachable locally on port 8082 and remotely on port 80 and you will be able to proceed with the **Platform wizard**.

2.2 Platform Wizard

After installing of GlobaLeaks you can proceed with the platform wizard.

Open a browser at port 80 or port 8082 on your remote or local IP respectively.
2.2.1 Choose the Primary Language for Your Site

In the first page of the wizard you are invited to select the language of your site Default choice is English, but many others languages are available and more are expected to be available in the future.

2.2.2 Choose a name for Your Project

In the second section of the wizard configure the name of your project.
2.2.3 Configure the Account for the Administrator of Your Whistleblowing Site

In the third section of the wizard configure the account details of the administrator of your project.

Keep in mind to choose a strong password in order to protect this sensitive account; an indication of the strength of the choosen password is shown to guide you in this task.
2.2.4 Configure the Account for the First Recipient of Reports

In the forth section of the wizard configure the account details of the first recipient for the reports sent to your project.
2.2.5 Read and Accept the License

In the fifth section of the wizard you are invited to read and accept the License of GlobaLeaks.

2.2.6 Complete the Wizard

The sixth section of the wizard notifies you the completion of the wizard.
Congratulations!
You have completed the platform wizard.

Proceed
3.1 Threat Model

GlobaLeaks is an Open Whistleblowing Framework that can be used in many different usage scenarios that may require very different approaches to obtain both security and flexibility. Whistleblowing policies and procedures within a corporation for compliance purposes are reasonably different from the ones of a Media outlet or the ones for Hacktivism initiatives. Given the flexibility of uses of GlobaLeaks, the threat model considers different usage scenarios, as threats can vary.

3.1.1 Actors Matrix

As a first step we define the actors, which are the users that interact with a GlobaLeaks platform.

<table>
<thead>
<tr>
<th>Actor</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whistleblower</td>
<td>The user who submits an anonymous report through the GlobaLeaks platform. Whistleblowers are persons in a wide range of different threat models depending on the usage scenario and the nature of information being submitted.</td>
</tr>
<tr>
<td>Recipient</td>
<td>The user (person or organization) receiving the anonymous report submitted by the Whistleblower. The recipients act reasonably in good faith, e.g. if any of them were to give their credentials or private information to the attacker, that would be unreasonable.</td>
</tr>
<tr>
<td>Administrator</td>
<td>The user (person or organization) that is running the GlobaLeaks platform. Administrator may not represent the same entity running, promoting and managing the whistleblowing initiatives (e.g., hosted solutions, multiple stakeholders projects, etc). The Administrator has to be considered in all scenarios described as a trusted entity with reference to the data exchanged by actors. The Administrator in most scenarios won’t be a trusted entity with respect to the identity of actors.</td>
</tr>
</tbody>
</table>

It’s highly relevant to apply each of the security measures always in relationship to the actors using GlobaLeaks, while always considering the security and usability tradeoff.
3.1.2 Anonymity Matrix

The anonymity of different actors must be differentiated and classified depending on the context of use represented by the following definitions:

<table>
<thead>
<tr>
<th>Actor</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Anonymous</td>
<td>Actor’s identity and their location cannot be disclosed.</td>
</tr>
<tr>
<td>Confidential</td>
<td>The system is designed to remove or limit any recording of identifiable information that when registered is maintained encrypted and managed confidentially.</td>
</tr>
</tbody>
</table>

The following matrix relates the previous definition to different architectural use and implementation of GlobaLeaks software:

<table>
<thead>
<tr>
<th>Recipient</th>
<th>Anonymous</th>
<th>Confidential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Whistleblower</td>
<td>Anonymous</td>
<td>Confidential</td>
</tr>
<tr>
<td>Administrator</td>
<td>Anonymous</td>
<td>Confidential</td>
</tr>
</tbody>
</table>

Different uses of GlobaLeaks must consider the requirements for different actors in the anonymity matrix. The Anonymity level is reported in the user interface, with the aim of making the user aware of it. The Administrator can configure the Anonymity level required for each actor.

3.1.3 Communication Security Matrix

The security of communication with respect to third party transmission monitoring may have different requirements depending on its context of use.

The following matrix applies the previous definition related to different architectural implementations of GlobaLeaks software:

<table>
<thead>
<tr>
<th>Communication security matrix</th>
<th>Tor</th>
<th>HTTPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security Level</td>
<td>High security</td>
<td>Medium security</td>
</tr>
</tbody>
</table>

3.1.4 Identity Disclosure Matrix

Independently of the anonymity matrix, various actors may decide to, or be required to disclose or not disclose their identity.

<table>
<thead>
<tr>
<th>Undisclosed</th>
<th>The actor’s identity is not disclosed and its disclosure is not likely.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partially disclosed (pseudonym)</td>
<td>The actor operates under a pseudonym while interacting with the platform.</td>
</tr>
<tr>
<td>Optionally disclosed</td>
<td>The actor’s identity is by default not disclosed, but they are given the chance to disclose it on a voluntary basis (e.g., in some workflows an anonymous tip-off MAY receive a follow-up, while a formal report with identity disclosed MUST receive a follow-up).</td>
</tr>
<tr>
<td>Disclosed</td>
<td>The actor decides to, or is required to, disclose their identity to other actors.</td>
</tr>
</tbody>
</table>

Identity disclosure is a highly relevant topic, because even in an Anonymous High security environment, identity disclosure may be an valuable option for specific whistleblowing initiative workflows.
If an actor starts dealing with an Anonymity set “Anonymous” and with an “Undisclosed Identity” they can always decide, at a later stage, to disclose their identity. The opposite is not possible. This is one of the key considerations to provide actors protection around GlobaLeaks.

Voluntary identity disclosure may be required in certain whistleblowing procedures because, generally:

- A tip off MAY receive a follow-up and can be anonymous;
- Formal reports MUST receive a follow-up and in that case cannot be anonymous.

The “MAY” vs. “MUST” is with respect to the actions of recipients and is a fundamental element of the guarantee provided to whistleblowers in many initiatives (e.g., a corporate or institutional whistleblowing platform should not follow a MUST approach for Anonymous submission follow-up, considering such submissions just tip offs and not formal reports).

### 3.1.5 Usage Scenarios Matrix

In this section you will find examples that show how different anonymity levels of different actors can be mixed together depending on the context of use.

| Media outlet | A Media outlet, whose identity is disclosed, decides to start a Whistleblowing initiative. The outlet’s recipients are disclosed to Whistleblowers, so that they can trust a specific journalist rather than the outlet itself. Full anonymity must be assured to whistleblowers and their identity cannot be disclosed in connection with anonymous submissions. The whistleblower MAY choose to willingly disclose their identity (e.g. when the journalist’s source-protection record is trusted). |
|-----------------------------------------------|
| Corporate compliance | A Corporation needs to implement transparency, or anti-bribery law compliance, by promoting its initiatives to employees, consultants and providers. The recipients are partially disclosed because they are represented by different divisions of the “Internal Audit” business unit of the company. The Whistleblower is guaranteed full anonymity, but they can optionally disclose their identity (tip off vs formal report). |
| Government tax whistleblowing | A Government Authority (central or local) with its own public identity wants to promote Tax Whistleblowing with Rewards procedures for Whistleblowers (e.g. IRS). The recipients are not known because they are an internal division not exposing their names to the Whistleblower in advance. The Whistleblower MUST disclose their identity in order to be eligible for rewards. |
| Human Rights Activism Initiative | A Human Rights Group starts a Whistleblowing initiative to spot human rights violations in a dangerous place. The organization requires anonymity to avoid retaliations and takedowns, and operates under a Pseudonym. The Recipients MUST not be disclosed to the Whistleblowers, but a Partial Disclosure by pseudonym can be acceptable in order to give proper trust to “Who the whistleblower is submitting to”. The Whistleblower MUST be guaranteed anonymity and their identity cannot be disclosed. |
| Citizen media initiative | A Citizen media initiative with its own public identity wants to collect tips on a specific topic (political, environmental malpractice, corruption, etc) in a medium-low risk operational context. The recipients could be public or use Pseudonym in order to avoid complete exposure. The Whistleblower, if the topic is not life-threatening, can be allowed to submit also in a Confidential way to lower the entrance barrier. |
| Public Agency Initiative | A local public agency wants to setup a Street Hole Reporting service with it’s own public identity. The recipient can be disclosed to facilitate the CRM (Citizen relationship management) and Whistleblower identity protection is not required. |
3.1.6 GlobaLeaks Security Matrix

Below we show how different usage scenarios can require different anonymity levels, communication security requirements and identity disclosures for different actors.

GlobaLeaks, through its user interface, will enable each actor with appropriate security awareness information, and will enforce specific requirements to specific actors by the application of clear configuration guidelines.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Actor</th>
<th>Anonymity level</th>
<th>Identity disclosure</th>
<th>Communication security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Media outlet</td>
<td>Whistle-blower</td>
<td>Anonymous</td>
<td>Undisclosed</td>
<td>High security</td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td>Corporate compliance</td>
<td>Whistle-blower</td>
<td>Anonymous</td>
<td>Optionally disclosed</td>
<td>High security</td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>No anonymity</td>
<td>Partially disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td>Government tax whistleblowing”</td>
<td>Whistle-blower</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>No anonymity</td>
<td>Undisclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td>Human Rights Activism initiative</td>
<td>Whistle-blower</td>
<td>Anonymous</td>
<td>Undisclosed</td>
<td>High security</td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>Anonymous</td>
<td>Partially disclosed</td>
<td>High security</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>Anonymous</td>
<td>Partially disclosed</td>
<td>High security</td>
</tr>
<tr>
<td>Citizen media initiative</td>
<td>Whistle-blower</td>
<td>Confidential</td>
<td>Optionally disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>Confidential</td>
<td>Confidential</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td>Public agency initiative</td>
<td>Whistle-blower</td>
<td>No anonymity</td>
<td>Optionally disclosed</td>
<td>No security</td>
</tr>
<tr>
<td></td>
<td>Recipient</td>
<td>No anonymity</td>
<td>Undisclosed</td>
<td>Medium security</td>
</tr>
<tr>
<td></td>
<td>Admin</td>
<td>No anonymity</td>
<td>Disclosed</td>
<td>Medium security</td>
</tr>
</tbody>
</table>

The previous schema gives only some examples of GlobaLeaks’s flexibility; but different anonymity, identity and security measures apply to other usage scenarios and actors.
3.1.7 Data Security Matrix

This section highlights the data that is handled by GlobaLeaks and how different protection schemes are applied to GlobaLeaks handled data.

The following data are the one involved within GlobaLeaks:

<table>
<thead>
<tr>
<th>Data</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questionnaire answers</td>
<td>The data associated with a submission such as the filled forms and selectors provided by the Whistleblower.</td>
</tr>
<tr>
<td>Submission attachments</td>
<td>The files associated with a submission.</td>
</tr>
<tr>
<td>Platform configuration</td>
<td>The data for the configuration and customization of the platform.</td>
</tr>
<tr>
<td>Software files</td>
<td>All the files that the software requires to work, including configuration defaults.</td>
</tr>
<tr>
<td>Email notifications</td>
<td>Data sent to notify recipients of a new report via email</td>
</tr>
</tbody>
</table>

Below a matrix showing different security measures applied on data.

<table>
<thead>
<tr>
<th>Data</th>
<th>Encryption</th>
<th>Metadata cleanup</th>
<th>Blacklisting</th>
<th>Sanitization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Questionnaire answers</td>
<td>Encrypted on the database with per-user / per/submissions keys</td>
<td>N/A</td>
<td>Keyword blacklist- ing</td>
<td>Antispam, Anti XSS</td>
</tr>
<tr>
<td>Submission attachments</td>
<td>Encrypted on the filesystem with per-user / per/submissions keys</td>
<td>Optional</td>
<td>Extension blocking, Antivirus</td>
<td>N/A</td>
</tr>
<tr>
<td>Platform configuration</td>
<td>Encrypted database with admin password</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Software files</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Email notifications</td>
<td>Encrypted with PGP when recipients keys are available</td>
<td>N/A</td>
<td>Antispam to prevent flooding</td>
<td>N/A</td>
</tr>
</tbody>
</table>

3.1.8 Threats to Confidentiality and Anonymity

In this section we highlight several threats that require specific explanation.

**Browser History and Cache**

GlobaLeaks tries to avoid, by using properly crafted HTTP headers and other triks, leaking information into any actor's browser history or cache. This privacy feature cannot guarantee the safety of the user against a forensics analysis of their browser cache and/or history, but it is provided as an additional safety measure.

**Metadata**

Every file can contain metadata related to the author or the whistleblower. The cleanup of metadata of submitted files is a particular topic that attempts to protect an “unaware” whistleblower from leaking information in a document that may put their anonymity at risk. In the context of GlobaLeaks, by default no automatic metadata cleanup is implemented because metadata is considered fundamental in the evidence preservation. For that reason metadata cleanup is an optional operation that could be suggested to Whistleblowers or operated by Recipients when sharing the document with other persons. A valuable software resource for this aim is the [Metadata Anonymization Toolkit](https://0xacab.org/jvoisin/mat2)
Malware and Trojans

GlobaLeaks could not prevent an attacker to use the platform maliciously trying to target recipients users with malware and trojans in general. Considering this and in order to be less vulnerable to risks of data exfiltration perpetrated with trojans, Recipients should always implement proper operation security by possibly using a laptop dedicated to reports visualization and possibly open file attachments on computers disconnected from the network and other sensible information. Wherever possible they should use operation with specialized secure operation systems like [Tails](https://tails.boum.org/) and at least run an up-to-date Anti-Virus software.

Data Stored Outside the Platform

GlobaLeaks does not provide any kind of security for data that is stored outside the GlobaLeaks system. Is responsibility of Recipients to protect the data they download from the platform on their personal computer or that they share with other persons with external usb drives. The operatin system used or the pen drive adopted should offer encryption and guarantee that in case of device loss or stealing no one could access the data therein contained.

Environmental Factors

GlobaLeaks does not protect against environmental factors related to actors’ physical locations and/or their social relationships. For example if a user has a video bug installed in their house to monitor all their activity, GlobaLeaks cannot protect them. Likewise, if a whistleblower, who is supposed to be anonymous, tells their story to friends or coworkers, GlobaLeaks cannot protect them.

Incorrect Data Retention Policies

GlobaLeaks implements by default a strict data retention policy of 90 days to enable users to operate on the report for a limited time necessary for the investigations. If the platform is configured to retain every report for a long time and Recipients do not manually delete the unnecessary reports, the value of the platform data for an attacker increases and so too does the risk.

Human Negligence

While we do provide the Administrator the ability to fine tune their security related configurations, and while we do continuously inform the actors about their security related context at every step of interactions, GlobaLeaks cannot protect against any major security threats coming from human negligence. For example, if a Whistleblower submits data that a third party (carrying on an ex-post facto investigation) can use to identify them as the unique owner or recent viewer of that data, then the Whistleblower cannot be protected by GlobaLeaks.

Advanced Traffic Analysis

An attacker monitoring HTTPS traffic, with no ability to decrypt it, can still identify the role of the intercepted users, because the Whistleblower, Recipient and Administrator interfaces generate different network traffic patterns. GlobaLeaks does not provide protection against this threat. We suggest using Tor pluggable transports or other methods that provide additional protection against this kind of attack.
3.2 Application Security

The GlobaLeaks software tries to conform with industry standard best and practices and its security is a result of applied research.

This document try to detail every aspect implemented by the application in relation to the security design.

3.2.1 Architecture

GlobaLeaks is made up of two main software components: a Backend and a Client:

- The Backend is a python backend that runs on a physical backend and exposes a REST API
- The Client is a javascript client-side Web Application that interacts with Backend only through XHR.

3.2.2 Authentication

The confidentiality of the authentication is protected by either Tor Onion Services v3 or TLS version 1.2+

This section describes the authentication methods implemented by the system.

Password

By accessing the GlobaLeaks login interface, Administrators and Recipients need to insert their respective username and password. If the password submitted is valid, the system grants access to the functionality available to that user.

Receipt

Whistleblowers access their Reports by using a Receipt, which is a randomly generated 16 digits sequence created by the Backend when the Report is first submitted. The reason of this format of 16 digits is that it resembles a standard phone number, making it easier for the whistleblower to conceal the receipt of their submission and give them plausible deniability on what is the significance of such digits.

3.2.3 Password Security

The following password security measures implemented by the system.

Password Storage

Password are never stored in plaintext but the system maintain at rest only an hash. This apply to every authentication secret included whistleblower receipts.

The platform stores Users’ passwords hashed with a random 128 bit salt, unique for each user.

Passwords are hashed using Argon2, a key derivation function that was selected as the winner of the Password Hashing Competition in July 2015.

The hash involves a per-user salt for each user and a per-system salt for each whistleblower.
Password Complexity

The system enforces the usage of complex password by implementing a custom algorithm necessary for ensuring a reasonable entropy of each authentication secret.

Password are scored in three levels: strong, acceptable, insecure. A strong password should be formed by capital letters, lowercase letters, numbers and a symbols, be at least 12 characters long and include a variety of at least 10 different inputs. An acceptable password should be formed by at least 3 different inputs over capital letters, lowercase letters, numbers and a symbols, be at least 10 characters and include a variety of at least 7 different inputs.

Two Factor Authentication (2FA)

Users are enabled to enroll for Two Factor Authentication via their own preferences. The system implements Two Factor Authentication (2FA) based on TOTP as for RFC 6238.

Password Change on First Login

The system enforces users to change their own password at their first login. Administrators could as well enforce password change for users at their next login.

Periodic Password Change

By default the system enforces users to change their own password at least every year. This period is configurable by administrators.

Proof of Work on Login and Submissions

The system implements an automatic proof of work on every login that requires every client to request a token, solve a computational problem before being able to perform a login or file a submission.

Slowdown on Failed Login Attempts

The system identifies multiple failed login attempts and implement a slowdown procedure where an authenticating client should wait up to 42 seconds to complete an authentication. This feature is intended to slow down possible attacks requiring more resources to users in terms of time, computation and memory.

Password Recovery

In case of password loss users could request a password reset via the web login interface clicking on a Forgot password? button present on the login interface. When this button is clicked, users are invited to enter their username or an email. If the provided username or the email correspond to an existing user, the system will provide a reset link to the configured email. By clicking the link received by email the user is then invited to configure a new email different from the previous.

In case encryption is enabled on the system, a user clicking on the reset link would have first to insert their Account Recovery Key and only in case of correct insertion the user will be enabled to set a new password.

In case 2FA is enabled on the system, a user clicking on the reset link would have first to insert an authentication code taken from the authentication API.
3.2.4 Web Application Security

This section describes the Web Application Security implemented by the software in adherence with the OWASP Security Guidelines.

Session management

The session implementation follows the OWASP Session Management Cheat Sheet security guidelines. The system assigns a Session to each authenticated user. The Session ID is 256 bits long secret generated randomly by the backend. Each session expire accordingly to a timeout of 5 minutes. Session IDs are exchanged by the client with the backend by means of an header (X-Session) and do expire as soon that users close their browser or the tab running GlobalLeaks. Users could explicitly log out via a logout button or implicitly by closing the browser.

Cookies and XSRF Prevention

Cookies are not used intentionally to minimize XSRF attacks and any possible attack based on them. Instead than using Cookies authentication is based on a custom HTTP Session Header sent by the client on authenticated requests.

HTTP Headers

The system implements a large set of HTTP headers specifically configured to improve the software security and achieves score A+ by Security Headers and score A+ by Mozilla Observatory.

Strict-Transport-Security

The system implements strict transport security by default.

```
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
```

The preload feature is left optional to users and following the best practices is left disabled as default.

Content-Security-Policy

The backend implements the following Content Security Policy (CSP):

```
Content-Security-Policy: default-src 'none'; script-src 'self'; connect-src 'self';
    style-src 'self'; img-src 'self' data:; font-src 'self' data:; media-src 'self';;
    form-action 'self'; frame-ancestors 'none'; block-all-mixed-content
```

Permissions-Policy

The backend implements the following Permissions-Policy header configuration to limit the possible de-anonimization of the user by disabling dangerous browser features:

```
Permissions-Policy: camera=('none') display-capture=('none') document-domain=('none')
    fullscreen=('none') geolocation=('none') microphone=('none') speaker=('none')
```
X-Frame-Options

The backend configure the X-Frame-Options header to prevent inclusion by means of Iframes in any site:

```
X-Frame-Options', b'deny'
```

Referrer-Policy

Web-browsers usually attach referrers in their http headers as they browse links. The platform enforce a referrer policy to avoid this behaviour.

```
Referrer-Policy: no-referrer
```

X-Content-Type-Options

When setting up Content-Type for the specific output, we avoid the automatic mime detection logic of the browser by setting up the following header:

```
X-Content-Type-Options: nosniff
```

X-XSS-Protection

In addition in order to explicitly instruct browsers to enable XSS protections the Backend inject the following header:

```
X-XSS-Protection: 1; mode=block
```

Cache-Control

The backend by default sends the following headers to instruct client’s browsers to not store resources in their cache. As by section 3. Storing Responses in Caches of RFC 7234 the platform uses the Cache-control HTTP header with the configuration no-store not instruct clients to store any entry to be used for caching; this settings make it not necessary to use any other headers like Pragma and Expires.

```
Cache-Control: no-store
```

Crawlers Policy

For security reasons the backend instructs crawlers to avoid any caching and indexing of the application and uses the Robots.txt file to enable crawling only of the home page; indexing of the home page is in fact considered best practice in order to be able to widespread the information about the existance of the platform and ease access to possible whistleblowers.

The configuration implemented is the following:

```
User-agent: *
Allow: /$
Disallow: *
```

As well the platform instruct crawlers to not keep any cache by injecting the following HTTP header:
For high sensitive projects where the platform is intended to remain hidden and communicated to possible whistleblowers directly the platform could be as well configured to disable indexing completely.

The following is the HTTP header injected in this case:

Anchor Tags and External URLs

In addition to the protection offered by the header `Referrer-Policy: no-referrer` that prevents to pass the referrer while visiting the application sets the rel attribute `noopener` to each of the external links. This protects from execution of malicious content within the context of the application.

Input Validation (Server)

The system adopts a whitelist based input validation approach. Each client request is checked against a set of regular expressions and only requests matching the expression are then processed.

As well a set of rules are applied to each request type to limit possible attacks. For example any request is limited to a payload of 1MB.

Input Validation (Client)

The client implement strict validation of the rendered content by using the angular component `ngSanitize.$sanitize`

Form Autocomplete OFF

Form implemented by the platform make use of the HTML5 form attribute in order to instruct the browser to do not keep caching of the user data in order to predict and autocomplete forms on subsequent submissions.

This is achieved by setting `autocomplete="false"` on the relevant forms or attributes.

3.2.5 Network Security

Connection Encryption

The software adopts Tor as default, preferred and recommended connection encryption protocol for its security and each GlobaLeaks server by default implement an Onion Service v3. The use of Tor is recommended over HTTPS for its advanced properties of resistance to selective interception and censorship that would make it difficult for a third party to selectively capture or block access to the site to specific whistleblower or company department.

The software enables as well easy setup of HTTPS offering both automatic setup via Let'sEncrypt and manual setup.

The configuration enables only TLS1.2+ and is fine tuned and hardened to achieve SSL Labs grade A+.

In particular only following ciphers are enabled:

3.2. Application Security
Network Sandboxing

The GlobaLeaks backend integrates **iptables** by default and implements strict firewall rules that only allow inbound and outbound connections from 127.0.0.1 (where Tor is running with Tor Onion Service).

As well it automatically applies network sandboxing to all outbound communications that get automatically `torrified` (sent through Tor), being outbound TCP connections or DNS-query for name resolution.

### 3.2.6 Data Encryption

The data, files, messages and metadata exchanged between whistleblowers and recipients is encrypted using the GlobaLeaks **Encryption Protocol**. In addition to this GlobaLeaks implements many other encryption components and the following is the set of the main libraries and their main usage:

- **Python-NaCL**: is used for implementing data encryption
- **PyOpenSSL**: is used for implementing HTTPS
- **Python-Cryptography**: is used for implementing authentication
- **Python-GnuPG**: is used for encrypting email notifications

### 3.2.7 Application Sandboxing

The GlobaLeaks backend integrates **AppArmor** by default and implements a strict sandboxing profile enabling the application to access only the strictly required files. As well the application does run under a dedicated user and group “globaleaks” with reduced privileges.

### 3.2.8 DoS Resiliency

To avoid applicative and database denial of service, GlobaLeaks apply the following measures:

- It tries to limit the possibility of automating any operation by requiring human interaction (e.g. with the implementation of proof of work)
- It is written to limit the possibility of triggering CPU intensive routines by an external user (e.g. by implementing limits on queries and jobs execution time)
- It implements monitoring of each activity trying to implement detection of attacks and implement proactively security measures to prevent DoS (e.g. implementing slowdown on fast-operations)
3.2.9 Other Measures

Encryption of Temporary Files

Files being uploaded and temporarily stored on the disk during the upload process are encrypted with a temporary, symmetric AES-key in order to avoid writing any part of an unencrypted file’s data chunk to disk. The encryption is done in “streaming” by using AES 128bit in CTR mode. The key files are stored in memory and are unique for each file being uploaded.

Secure File Delete

Every file deleted by the application is overwritten before releasing the file space on the disk.

The overwrite routine is performed by a periodic scheduler and acts as following:

- A first overwrite writes 0 on the whole file;
- A second overwrite writes 1 on the whole file;
- A third overwrite writes random bytes on the whole file.

Secure Deletion of Database Entries

The platform enables the SQLite capability for secure deletion that automatically makes the database overwrite the data upon each delete query:

```
PRAGMA secure_delete = ON
PRAGMA auto_vacuum = FULL
```

Exception Logging and Redaction

In order to quickly diagnose potential problems in the software when exceptions in clients are generated, they are automatically reported to the backend. The backend backend temporarily caches these exceptions and sends them to the backend administrator via email.

In order to prevent inadvertent information leaks the logs are run through filters that redact email addresses and uuids.

Entropy Sources

The main source of entropy for the platform is /dev/urandom.

In order to increase the entropy available on the system the system integrates the usage of the Haveged daemon.

UUIDv4 Randomness

Resources in the system like submissions and files are identified by a UUIDv4 in order to not be guessable by an external user and limit possible attacks.
**TLS for SMTP Notification**

All of the notifications are sent through SMTP over TLS encrypted channel by using SMPT/TLS or SMTPS, depending on the configuration.

### 3.3 Encryption Protocol

GlobaLeaks implements an encryption protocol designed to implement a trade off between security and usability necessary to enable easy access and anonymous reporting by whistleblowers that is requirement for the most common whistleblowing scenarios.

The protocol is intended as well to provide reasonable security from attackers seizing the backend and attempting bruteforce decryption.

Encryption is implemented for each submission protecting questionnaire’s answers, comments, attachments and involved metadata. The keys involved in the encryption are per-user and per-submission and only users to which the data was destined could access the data. This mechanism guarantees that only the user could access the data. Users that would forget their password would lose access to data that won’t be accessible anymore. To handle with this condition the system implements as well **Key recovery** and **Key Escrow** mechanisms.

#### 3.3.1 Encryption’s Workflow

- Users chooses a personal secure password at first login;
- The system creates a personal user keypair and stores it asymmetrically encrypted with a secret derived from the personal user password;
- The whistleblower files a report;
- The system assigns personal access credentials to the whistleblower;
- The system generates a symmetric key for the encryption of the report, the attached files and comments and the involved metadata and starts encrypting the data;
- The system generates an asymmetric keypair and store it symmetrically encrypted using a secret derived from the whistleblower access credentials;
- The system grants every involved recipient and the whistleblower access to the symmetric encryption key of the report by assigning each of the user an asymmetrically encrypted copy of the key;
- Users furtherly proceed exchanging information on the report by using their personal access credentials and unlocking their own personal asymmetric keys and symmetric keys of the accessed report.

#### 3.3.2 Encryption’s Details

**Algorithms**

<table>
<thead>
<tr>
<th>Type</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asymmetric encryption</td>
<td><strong>Libsodium SealedBoxes</strong>, an encryption implementation that combines Curve25519, XSalsa20 and Poly1305 algorithms.</td>
</tr>
<tr>
<td>Symmetric encryption</td>
<td><strong>Libsodium SecretBoxes</strong>, an encryption implementation that combines XSalsa20 and Poly1305.</td>
</tr>
</tbody>
</table>
Users’ Credentials

The system used two different type of credentials depending on the user role:

<table>
<thead>
<tr>
<th>Credentials type</th>
<th>User role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Passwords</td>
<td>For authenticated users identified by a username</td>
</tr>
<tr>
<td>16-digits random receipts</td>
<td>For anonymous Whistleblowers</td>
</tr>
</tbody>
</table>

Assumptions:
- The system enforces authenticated users password quality based on strong password complexity rules.
- The system enforces expiration of receipts low the number of active receipts.

Users’ Keys

<table>
<thead>
<tr>
<th>Type</th>
<th>Generation</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC Curve</td>
<td>Generated by the backend at first user login for authenticated users and on submission for whistleblowers</td>
<td>Keys are stored on the backend encrypted using symmetric encryption. The symmetric key used for encrypting users’ keys is derived from the users’ credentials using the KDF function Argon2ID. The parameters for Argon2ID used for KDF are chosen stronger than the parameters one used for authentication of related user which the hash is stored. The parameters are chosen to require 128MB per Login and 1 second of computation.</td>
</tr>
<tr>
<td>Key-pair</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data Encryption’s Keys

<table>
<thead>
<tr>
<th>Type</th>
<th>Generation</th>
<th>Storage</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 bit keys</td>
<td>Generated by the backend for each report</td>
<td>Keys and stored on backend filesystem encrypted using asymmetric encryption by means of Users’ and Whistleblower’s keys respectively</td>
</tr>
</tbody>
</table>

3.3.3 Key Recovery

The system implements a key recovery system by means of a recovery key and symmetric encryption.

Upon generation of a user key, the private key is symmetrically encrypted with a randomly generated recovery key.

For usability reasons, this recovery is kept as well encrypted on the backend making it possible for logged users in possession of their password to retrieve and print their own account recovery key.

3.3.4 Key Escrow

The system implements an optional key escrow mechanism in order to limit data loss in case of users’ password loss.

Key escrow can be enabled during the initial application wizard or alternatively could be enabled in the advanced settings of the software.

We advise enabling this option if you would like to protect whistleblowers’ submissions from being lost in the situation where recipients lose their passwords. On the other hand, we would not advise using this feature if you want to setup a system where only recipients are able to access submissions.
When the option is enabled the system will generate and assign an escrow key and assign it to the administrator that has enabled the feature; the key will be furtherly used by the system to encrypt every system key preserving a copy that could be unlocked by any administrator in the availability of the escrow key.

Administrators with access to the escrow key will be able to support any internal user in case of password loss and issue password reset. As well they will be able to grant this same privilege to other administrators or disable the feature completely.

### 3.4 Penetration Tests

GlobaLeaks is periodically subject to independent security audit and penetration tests in order to verify and improve the security of the system.

We try to get it audited at least every 2 years thanks to funding opportunities. Each user and adopter as well sometimes is able to fund additional audits.

This page lists the currently publicly available reports.

If you have carried or have the possibility to sponsor a security audit please email us at info@globaleaks.org. This would be particularly important for the general software security. When asking a company to audit the software please always remember to ask for the possibility to publish the report before this is performed; many auditors in fact may not agree with publishing afterwards and this happened many times with waste of project resources.

We additionally invite independent security researchers to apply to our Bug Bounty initiative, which it’s hosted on HackerOne.

<table>
<thead>
<tr>
<th>Date</th>
<th>Auditor</th>
<th>Goal</th>
<th>Report</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013 Q1</td>
<td>iSecPartners</td>
<td>Architecture Audit</td>
<td>Report</td>
</tr>
<tr>
<td>2013 Q2</td>
<td>Cure53</td>
<td>Web Security Audit</td>
<td>Report</td>
</tr>
<tr>
<td>2013 Q4</td>
<td>VeraCode</td>
<td>Overall Audit</td>
<td>Report</td>
</tr>
<tr>
<td>2014</td>
<td>LeastAuthority</td>
<td>Source Code Audit</td>
<td>Report</td>
</tr>
<tr>
<td>2018</td>
<td>SubGraph</td>
<td>Overall Audit</td>
<td>Report</td>
</tr>
<tr>
<td>2019</td>
<td>RadicallyOpenSecurity</td>
<td>Crypto Audit, Multi-tenancy Audit, Overall Audit</td>
<td>Report</td>
</tr>
</tbody>
</table>
4.1 Common to All Users

4.1.1 Login

Users could login by accessing the /#/login page.

![Login Form]

Powered by Globaleaks
4.1.2 Access the User Preferences

After login Users could access their preferences by clicking the Preferences link present in the login status bar.

4.1.3 Change Your Password

Users could change their own password by accessing the Password tab present in the Preferences page.
4.1.4 Reset Your Password

Users could request a password reset via the `/#/login` page by clicking the `Forgot password?` button. After clicking the button users are requested to type their own username or email address.
GLOBALEAKS - Password reset

Enter your account’s username or your email address to request a password reset.

Submit

GLOBALEAKS - Password reset

Password reset requested. Please check your inbox for further instructions.

Powered by Globaleaks
4.1.5 Enable Two-Factor-Authentication (2FA)

Users could enable Two-Factor-Authentication by clicking the Enable two factor authentication option inside the Preferences page.

To enable the feature the user requires to have a phone with installed a common Authenticator APP implementing the TOTP standard as by RFC 6238

4.1.6 Access and Save Your Account Recovery Key

Users could access their own Account Recovery key by clicking the Account Recovery Key button present in the Preferences page.

This is a fundamental step that any user should do at their first login after activating their own account in order to backup their own account recovery key and get sure to not incur in data loss due to password loss.
4.1.7 Load a Public PGP Key

GlobaLeaks implements by default an advanced and automatic *Encryption Protocol* that uses online keys.

Users have the optional possibility to load a personal *Public PGP Key* by uploading or pasting the key on their *Preferences* page.

When a PGP key is loaded onto the platform any file uploaded on the platform and email notification sent by the platform is encrypted with it. This is an optional additional measure of security that should be evaluated and adopted in relation the threat model of the project.
4.1. Common to All Users
4.2 For Administrators

4.2.1 User Interface

This section offers you a summary of the user interface offered to Admin users.

Anytime you log in as administrator via the Login page the application takes you to your personal administrative Homepage: This page includes some documentation about GlobaLeaks that is intended to clarify you all the up-to-date documentation in matter of software security, best practices and community support.

From this Home page you may access all the common user facilities already described in the general User Documentation.

A menu on the right offers you links to the different administrative sections:

1. Site settings
2. Users
3. Questionnaires
4. Contexts
5. Case management
6. Notification settings
7. Network settings
8. Advanced settings
9. System overview
Site Settings

This is the section that offers you all the main customization possibilities necessary for implementing a basic and functional whistleblowing site.

This section is furtherly divided in:

1. Main configuration
2. Theme customization
3. Files
4. Languages
5. Text customization

Main Configuration

In this section is configurable the logo and all the texts of the main user interfaces.
Theme Customization

In this section could be loaded CSS and Javascript files necessary to customize the interface.
Files

In this section could be loaded any file that need to be served via the platform to whistleblowers (e.g. a policy in PDF) or that are required by your CSS and Javascript customization.
Languages

In this section you could enable all the languages required by your project and configure the default language.

**Note:** Thanks to the Localization Lab and our great volunteer community, the software is already available and continuously made available in a lot of languages. This aspect of internationalization is crucial in many projects. In case you are starting a project and the required languages are not available we strongly invite you to register on our web translation platform offered by Transifex and support yourself the translation. Internationalization and Localization is in fact are crucial for the success of a whistleblowing project. Thank you!

Text Customization

Here could be configured overrides for any of the texts of the platform and of their translation.
Users

This section is where users could be created and managed. The system with the basic configuration completed with the initial Platform wizard is configured with an Administrator and a Recipient.

Depending on your project needs here you could create users with different roles and manage their respective privileges.
User Roles

The software offers the possibility to create users with the following roles:

1. Administrators
2. Recipients

Administrators

TODO

Recipients

TODO

Questionnaires

This section is where whistleblowing questionnaires could be created and managed.

By default the software implements a Default Questionnaire with a single Step and the following three questions:

- One question “Short description” of type “Multi-line text input” enabling whistleblower to provide a short summary of the fact reported;
- One question “Full description” of type “Multi-line text input” enabling whistleblowers to describe the fact reported in detail;
- One question “Attachments” of type “Attachment” enabling users to load one or more attachments.
The system with its basic configuration completed with the initial Platform wizard is also already configured with this Default Questionnaire pre-associated to the Default Context.

Depending on your project needs you could create specific questionnaire for each of your different reports’ contexts.

Depending on your project needs you may evaluate defining some questions once as Question Templates and reuse the same question in multiple questionnaires.
Steps

The software enables to organise questionnaire in one or multiple steps. For example the default questionnaire is organized with a single step including all the questions.
Questions Types

The software enables you to create questions of the following types:

1. Single-line text input
2. Multi-line text input
3. Selection box
4. Multiple choice input
5. Checkbox
6. Attachment
7. Terms of service
8. Date
9. Date range
10. Map
11. Question group

General Question Properties

Each of the software question types make it possible to configure the following properties:

- Question: The text of the question
- Hint: A hint that will be shown via a popover and a question mark near the question.
- Description: A description text that will be shown below the question
- Required: Set this field if you want this question to be mandatory
- Preview: Set this field if you want the answers to this question to appear in the preview section of the list

Question Properties by Question Type

Single and Multi Line Text Input

TODO

Selection Box, Multiple Choice Input, Checkbox

TODO
Question Groups

TODO

Conditional Questions

TODO

Contexts

This section is where whistleblowing contexts (channels) could be created and managed.

A whistleblowing channel is typically defined by the following main characteristics:

- **Name**: the name of the channel
- **Image**: an image to identify the channel
- **Description**: a description of the channel
- **Recipients**: the set of recipients that will receive reports sent to this channel
- **Questionnaire**: the questionnaire that will be proposed to whistleblowers selecting this channel
- **Submission expiration**: the data retention policy for the channel

The system with the basic configuration completed with the initial platform wizard is configured with a single Context called Default, on which is associated a recipient and the default questionnaire.

Depending on your project needs here you could create additional Contexts and configure their respective recipients and properties.

![Globaleaks - Contexts](image-url)
Submissions Expiration

The software enables to configure a data retention policy for each channel. This is a fundamental property of the whistleblowing channel that makes it possible to configure automatic secure deletion of reports after a certain period of time. This setting should be configured in relation to the risk of the channel in order to limit undesired exposure of the reports received therein.

By default a context is configured with a report expiration of 30 days.

Case Management

This section is intended to host all the main case management feature that will be offered by the software. Currently it hosts the possibility to define reports statuses and substastues intended to be used by Recipients while working on the reports.

By default the system includes the following report statuses:

1. New
2. Open
3. Closed

Within this section you may add additional Statuses between the State Open and Closed and you can furtherly define Substatuses for the Closed status (e.g. Archived / Spam)
Notification Settings

This is the section where are configured all the aspects related to the mail notifications sent by the software.

**The section is furtherly divided in:**

1. Main configuration
2. Notification templates

**Main configuration**

Here are configured the technical details about SMTP.

**Note:** By default GlobaLeaks comes with a working configuration that is based on systems offered by the GlobaLeaks developers to the community of users and testers; even though this configuration is designed by their owners with special care in relation to security and privacy you are invited to consider using alternative systems for your production enviroment.
Notification Templates

In this section are configured the notification templates.

By default globaleaks includes text and translations for each of the templates that are provided to be fully functional and studied with particular care in relation to security and privacy. Depending on your project needs you may override the default text with your customized texts.

Network Settings

In this section are configured the network settings.

The section is furtherly divided in:

1. HTTPS
2. Tor
3. IP Access control

HTTPS

Here you can configure all the aspects related to the access of the platform via the HTTPS Protocol.
In particular here are configured:

1. The domain name used by your project
2. The HTTPS key and certificates

To ease the deployment and the maintainance and reduce the costs of your project, consider using the software includes support for the Let’sEncrypt HTTPS certificates.

**Tor**

Here you can configure all the aspects related to the access of the platform via the Tor Protocol.
IP Access Control

Here you can configure IP based Access Control.
Suggested configurations are:

1. Prevent Whistleblowers to report from within their respective work space.
2. Restrict Recipients access to their intranet.

**Url Redirects**

Here you can configure URL Redirects.
4.2. For Administrators

**Advanced Settings**

TODO
Audit Log

TODO
### GLOBELEAKS - Audit log

<table>
<thead>
<tr>
<th>ID</th>
<th>Username</th>
<th>Role</th>
<th>Name</th>
<th>2FA</th>
<th>Creation date</th>
<th>Last access</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1f7a44a-e5fc-46f9-a685-c5f82270ee611</td>
<td>admin</td>
<td>admin</td>
<td>Admin</td>
<td>✗</td>
<td>02-04-2021 11:26</td>
<td>02-04-2021 11:28</td>
</tr>
<tr>
<td>b10f5ee9-88c6-46f1-8bca-58b773c6f0c5</td>
<td>Admin2</td>
<td>admin</td>
<td>Admin2</td>
<td>✗</td>
<td>02-04-2021 11:26</td>
<td>02-04-2021 11:26</td>
</tr>
<tr>
<td>4c9e4249-e9be-4b56-89b7-b9f9d101d468</td>
<td>Custodian</td>
<td>custodian</td>
<td>Custodian</td>
<td>✗</td>
<td>02-04-2021 11:26</td>
<td>02-04-2021 11:26</td>
</tr>
<tr>
<td>9e00b0b0-c63a-4e6d-9d8b-202235a9b9b0</td>
<td>Recipient</td>
<td>receiver</td>
<td>Recipient</td>
<td>✗</td>
<td>02-04-2021 11:26</td>
<td>02-04-2021 11:28</td>
</tr>
<tr>
<td>05c9d8c9-19cc-4c85-9b20-72e730f4694b</td>
<td>Recipient2</td>
<td>receiver</td>
<td>Recipient2</td>
<td>✗</td>
<td>02-04-2021 11:26</td>
<td>02-04-2021 11:26</td>
</tr>
<tr>
<td>a7275d40-4c90-4e80-9d9d-59191fda035c5</td>
<td>Recipient3</td>
<td>receiver</td>
<td>Recipient3</td>
<td>✗</td>
<td>02-04-2021 11:26</td>
<td>01-01-1970 00:00</td>
</tr>
</tbody>
</table>

Powered by **Globaleaks**
### Audit Log

#### Anomalies

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001s</td>
<td>0.001s</td>
<td>0s</td>
<td>0.077s</td>
<td>11:25:54</td>
<td>11:26:54</td>
<td>11:27:54</td>
<td>11:28:54</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
</tr>
</tbody>
</table>

#### AuditLogJob

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.012s</td>
<td>0.036s</td>
<td>0.034s</td>
<td>0.031s</td>
<td>0.054s</td>
<td>0.019s</td>
<td>1.058s</td>
<td>0.156s</td>
<td>0.04s</td>
<td>0.037s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
</tr>
</tbody>
</table>

#### CertificateCheck

#### Cleaning

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.021s</td>
<td>0.01s</td>
<td>0.017s</td>
<td>0.007s</td>
<td>0.025s</td>
<td>0.012s</td>
<td>0.018s</td>
<td>0.01s</td>
<td>0.019s</td>
<td>0.048s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
</tr>
</tbody>
</table>

#### ExtNodesRefresh

#### Notification

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.029s</td>
<td>0.022s</td>
<td>1.549s</td>
<td>0.025s</td>
<td>7.03s</td>
<td>0.074s</td>
<td>0.075s</td>
<td>0.068s</td>
<td>0.062s</td>
<td>0.063s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
</tr>
</tbody>
</table>

#### POPCheck

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SessionsManagement</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
<td>0s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>#8</th>
<th>#9</th>
<th>#10</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:25:54</td>
<td>11:26:54</td>
<td>11:27:54</td>
<td>11:28:54</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
</tr>
</tbody>
</table>

#### UpdateCheck

**Powered by GlobalLeaks**
4.2.2 Common Configurations

Configure the Logo

The first thing you want to give to your whistleblowing site is a branding identity; this could be done by loading a logo in section Site settings / Main configuration.

Scroll down along the page to reach the “Save” button, click on it and have your logo and favicon applied.
Enable Languages

You may want your GlobaLeaks installation served on more than one language.
To do so, in the section “Site settings / Languages” select the languages you would like and add them.
Note that in the same interface you can mark the default application language.

Configure Notification Settings

GlobaLeaks sends out notifications of different events to different receivers and to admins. In order to have this working, you have to select “Notification Settings” in the “Administration Interface - General Settings” page and set up email account and related server parameters.

We suggest you to setup an email account dedicated to sending out notifications from your initiative.
Enter the followings:

- **SMTP name**: the name of your GlobaLeaks project or something that equally descriptive
- **SMTP email address**: the email address used to send notifications
- **Username**: the username corresponding to the just inserted “SMTP email address”; this is needed to authenticate to the SMTP server and send emails
- **Password**: Password of the above corresponding “SMTP email address”
- **SMTP Server Address**: it is the hostname of the SMTP server you are using to send notification emails
- **SMTP Server Port**: Port used to send outgoing emails. It is usually 465 or 587 (SMTP with TLS is at TCP port 587; SMTP with SSL is at 465)
- **Transport Security**: from the drop down menu select the opportune security level

It is better to leave untouched the pre-defined settings pertaining the notification to admins and to recipients, but in the case you want to disable them, it is possible to check the corresponding checkboxes.

You can then set the value for the time at which the notification alert of expiring submission; this value is set at 72 hours to give time to the recipient(s) to check and manage the pending submissions.

It is possible to tweak the maximum number of emails allowed in an hour, before email will be suspended in order to avoid flooding the system. It is advised to keep the pre-defined value, and eventually change it accordingly with mail server capabilities.
Once configured all the parameters for notifications, it is possible to test them by just clicking on the “Test the configuration” button.

If all is working as expected, click on the “Save” button to keep the configured parameters.

**Configure Recipients**

The Recipient is the person that will receive and process the data that whistleblowers input in the platform. You can have one or multiple Recipients per Context, and also have one Recipient that can access to multiple Contexts. The platform is very flexible on this and allows you to define in very detail your whistleblowing system and procedure.

**Configure Questionnaires**

TODO

**Configure Contexts**

TODO

**Customize the Graphic Layout**

**Example 1: Custom Background**

This CSS example shows how to customize the Background Color of the application.

```css
body {
  background-color: red;
}
```

**Example 2: Custom Font**

This CSS example shows how to customize the font of the application.

```css
@font-face {
  font-family: 'Antani';
  src: url('s/antani.ttf') format('truetype');
  font-weight: normal;
  font-style: normal;
}
body {
  font-family: 'Antani', Helvetica, Arial, Sans;
  font-size: 16px;
}
```
4.2.3 Upgrade Guide

Regular Update

To safely upgrade a GlobaLeaks installation please proceed with a backup of your setup by following the Backup and restore guide.

This is necessary so that if something goes wrong and you need to rollback, you will be able to just uninstall the current package, then install the same version of globaleaks that was previously installed and working.

In order to update GlobaLeaks perform the following commands:

```
apt-get update && apt-get install globaleaks
```

Upgrade of the Distribution Version

For security and stability reasons it is recommended to not perform a distribution upgrade.

GlobaLeaks could be instead easily migrated to a new up-to-date Debian system with the following recommended instructions:

- create an archive backup of /var/globaleaks
- instantiate the latest Debian available
- log on the new server and extract the backup in /var/globaleaks
- follow the Installation Guide; GlobaLeaks while installing will recognize the presence of an existing data directory and will use it

In Case of Errors

The above commands should allow you to perform regularly updates. On some conditions due to special updates it could be possible that those commands result in a failure. Consult this page for knowing specific FAQs on precise failures.

In case you do not find any specific documented solution for your failure, you could run the GlobaLeaks install script. The installation script in fact is designed to allow the update of GlobaLeaks and it includes fixes for the most common issue.

To run the install script for updating globaleaks perform the following commands:

```
wget https://deb.globaleaks.org/install-globaleaks.sh
chmod +x install-globaleaks.sh
./install-globaleaks.sh
```

4.2.4 Backup and Restore

The following bash script could be used in order to perform a backup manually:

```bash
#!/bin/sh
set -e

if [ ! -d "/var/globaleaks" ]; then
  timestamp=$(date +%s)
  version=""dpkg -s globaleaks | grep '^Version:' | cut -d ' ' -f2"
```
After the completion of the command you will find a tar.gz archive within the /var/globaleaks/backups. The file will have the format: globaleaks-$version-$timestamp.tar.gz

GlobaLeaks does automatically perform a backup at each platform update and the backup is kept under data retention policy and is deleted 15 days after the update.

To restore an existing backup:

- be sure globaleaks is not running; globaleaks can be shut down with “service globaleaks stop”;
- identify the version of globaleaks required for restoring globaleaks. the version is written in the backup filename;
- extract the content of the archive in /var/globaleaks with the command tar -zxvf backup.tar.gz
- install the required version of globaleaks with: apt-get install globaleaks=version (e.g. globaleaks=3)

### 4.2.5 Troubleshooting

#### Issues and Bug Reporting

If you encounter any issue and you are not able to to run GlobaLeaks:

- Be sure to strictly follow the Installation Guide for installation
- Be sure to satisfy the Technical Requirements for hardware and operating system version.
- Search on the support forum to check if a user has already encountered your issue: [https://forum.globaleaks.org](https://forum.globaleaks.org)
- Report the issue on the official software ticking system: [https://github.com/globaleaks/GlobaLeaks/issues](https://github.com/globaleaks/GlobaLeaks/issues)

#### Useful Debugging Commands

Depending on your setup. There are a few things that are usually the first things to check to see if GlobaLeaks is working.

- Is the service running?

  service globaleaks status

- Is the service responding on the loopback interface?

  curl -vvv localhost:8082

- Is the service listening on external interfaces?

  netstat -tap

- Are exceptions being generated?
Log Files

There are a few useful logs and corresponding log files when GlobaLeaks is installed.

**GlobaLeaks process:**

```
_less /var/globleaks/logs/globleaks.log

```

The verbosity is configurable via the web interface of the software inside Advanced Settings.

**Tor:**

```

```

### 4.3 For Recipients

#### 4.3.1 Access the List of the Existing Reports

The lists of the existing reports can be accessed via the link **Reports** on the sidebar of the Recipient’s Homepage.
4.3.2 Access a Submission

A submission can be accessed in two ways:

- By clicking on it like on a mailbox system from the Reports page
- By clicking on the link received on a mail notification and entering own credentials
<table>
<thead>
<tr>
<th>Context</th>
<th>Date</th>
<th>Last update</th>
<th>Expiration date</th>
<th>Connection</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>02-04-2021</td>
<td>02-04-2021</td>
<td>02-07-2021</td>
<td>HTTPS</td>
<td>Opened</td>
</tr>
</tbody>
</table>

Questionnaire answers

- **Title**: title
- **Description**: description

Attachments

<table>
<thead>
<tr>
<th>Filename</th>
<th>Download</th>
<th>Upload date</th>
<th>Type</th>
<th>File size</th>
</tr>
</thead>
<tbody>
<tr>
<td>antani.txt</td>
<td>Download</td>
<td>02-04-2021</td>
<td>text/plain</td>
<td>53 B</td>
</tr>
<tr>
<td>unknown.filetype</td>
<td>Download</td>
<td>02-04-2021</td>
<td>application/octet-stream</td>
<td>4 B</td>
</tr>
</tbody>
</table>

Files attached by recipients

- **Upload a file**
  - Description
  - Upload

Comments

- **Whistleblower comment reply**: 02-04-2021 11:28
- **Recipient comment**: 02-04-2021 11:28

Private messages

- **Recipient message**: 02-04-2021 11:28

Powered by **Globaleaks**
4.4 For Whistleblowers

4.4.1 File a New Report

A new report can be filed by accessing the homepage of the platform and clicking the **Blow the Whistle** button.

![Globaleaks](image-url)

Have you already filed a report? Enter your receipt.

XXXX XXXX XXXX XXXX [Log in]
After filing a new report the systems provides to the user a 16-digit receipt.
4.4.2 Access an Existing Report

An existing report can be accessed by entering the 16-digit receipt obtained at the end of the submission on the login interface present on the home page of the platform.
5.1 Development Environment

This guide describe how to set up an environment in order to contribute to the development of GlobaLeaks.

5.1.1 Requirements

The guide assumes you run a Debian based system and that the following software is installed on your system:

- debhelper
- devscripts
- dh-apparmor
- dh-python
- git
- grunt-cli
- node
- npm
- python3
- python3-dev
- python3-pip
- python3-setuptools
- python3-sphinx
- python3-virtualenv
5.1.2 Setup

The repository could be cloned with:

```bash
git clone https://github.com/globaleaks/GlobaLeaks.git
```

Client dependencies could be installed by issuing:

```bash
cd GlobaLeaks/client
npm install -d
grunt copy:sources
```

Backend dependencies could be installed by issuing:

```bash
cd GlobaLeaks/backend
python3 -mvenv env
source ./env/bin/activate
pip3 install --requirement requirements.txt
```

This will create for you a python virtualenv in the directory env containing all the required python dependencies.

Then, anytime you will want to activate the environment to run globaleaks you will just need to issue the command:

```bash
cd GlobaLeaks/backend && source ./env/bin/activate
```

Setup the client:

```bash
cd GlobaLeaks/client
npm install -d
grunt build
```

Setup the backend and its dependencies:

```bash
cd GlobaLeaks/backend
python3 -menv glenv
source glenv/bin/activate
pip3 install --requirement requirements.txt
```

5.1.3 Run

To run globaleaks from sources within the development environment you should issue:

```bash
cd GlobaLeaks/backend
source ./env/bin/activate
./bin/globaleaks -z -n
```

GlobaLeaks will start and be reachable at the following address http://127.0.0.1:8082
5.2 Software Libraries

The software libraries used by GlobaLeaks are listed in the following files:

- Backend: backend/requirements.txt
- Client: client/package.json
- Packaging: debian/control

5.3 Release Procedure

This is the procedure followed to publish a new GlobaLeaks release.

A release is represented by:

- A version bump;
- An updated CHANGELOG;
- A commit titled “Bump to version $number”;
- A tag commit $version signed by a core developer with its own key;
- An updated package on deb.globaleaks.org;
- A signed repository.

5.3.1 Release Tagging

The release is represented by a tag commit on Github performed via:

```bash
export DEBFULLNAME="GlobaLeaks software signing key"
export DEBEMAIL="info@globaleaks.org"
dch -i
git commit -a -m "commit before new tag message"
git push origin
git tag -s v0.1 -m 'GlobaLeaks version 0.1'
git push origin --tags
```

5.3.2 Release Packaging

The package is built by means of the official build script by issuing:

```bash
cd GlobaLeaks && ./script/build -d all
```

This command builds a package for each supported distribution and version.
5.3.3 Package Publishing

The package is published on deb.globalekas.org by issuing:

```bash
dput globaleaks ../globaleaks_${version}_all.changes
```

5.3.4 Repository Signing

The release is then signed by a core developer by using the official project key via:

```bash
gpg --detach-sign --digest-algo SHA512 -o Release.gpg Release
```

5.4 Continuous Integration

The globaleaks codebase is continuously tested for bug within a complete continuous integration lifecycle implemented. Tests are performed at every commit by:

- performing static and dynamic testing on TravisCI;
- performing end2end tests on SauceLabs to ensure compatibility with common browsers;
- tracking tests coverage and code quality with Codacy.

5.4.1 Backend Tests

The backend tests are implemented by means of python-twisted and trial Tests can be runned manually by issuing:

```bash
cd GlobaLeaks/backend && trial globaleaks
```

5.4.2 E2E Tests

End2end tests are implemented by means of the Protractor Angular JS library. Tests can be runned manually by issuing:

```bash
cd GlobaLeaks/client && ./node_modules/protractor/bin/protractor tests/protractor.config.js
```

5.5 Code Documentation

- modindex
INDEX

A
Admin, 3

C
Context, 3

G
GlobaLeaks, 3

N
Notification, 3

P
Platform, 3

Q
Questionnaire, 3

R
Receipt, 3
Recipient, 3

S
Submission, 3

W
Whistleblower, 3