# GlobaLeaks - Open Whistleblowing Framework

## Design and Arcitecture Review

iSECpartners
part of nccgroup

**Prepared for:**

GLOBALEAKS

**Prepared by:**

Aaron Grattafiori — Security Engineer

Alex Garbutt — Security Engineer

Justin Engler — Security Engineer

Gabe Pike – Security Engineer

| Document Change Log | | |
|---|---|---|
| **Version** | **Date** | **Change** |
| 0.1 | 2013-02-07 | DRAFT document created |
| 0.1 | 2013-02-07 | DRAFT document sent PGP encrypted to Fabio Pietrosanti. |
| 0.2 | 2013-02-08 | DRAFT document ready for review. |
| 0.5 | 2013-01-08 | Peer reviewed by Tim Newsham and Loic Simon. |
| 1.0 | 2013-01-08 | Ready for GlobaLeaks |
| 1.0 | 2013-01-08 | Final document sent PGP encrypted to Fabio Pietrosanti. |

# Table of Contents

# 1 Engagement Structure

## 1.1 Internal and External Teams

The iSEC team has the following primary members:

- Aaron Grattafiori — iSEC Technical Lead
  aaron@isecpartners.com, (707) 484–4223
- Alex Garbutt — iSEC Security Engineer
  aegarbutt@isecpartners.com, (415) 305–1598
- Justin Engler — iSEC Security Engineer
  jengler@isecpartners.com
- Gabe Pike — iSEC Security Engineer
  gpike@isecpartners.com
- Tom Ritter — iSEC Account Contact
  tritter@isecpartners.com, (908) 838–7002
- Michael Reisinger — iSEC Project Manager
  mreisinger@isecpartners.com, (818) 624–5344

The GlobaLeaks team has the following primary members:

- Fabio Pietrosanti — GlobaLeaks
  fabio.pietrosanti@logioshermes.org
- Arturo Filastò — GlobaLeaks
  art@globaleaks.org
- Claudio Agosti — GlobaLeaks
  vecna@globaleaks.org
- Giovanni Pellerano — GlobaLeaks
  evilaliv3@globaleaks.org
- Yvette Agostini — GlobaLeaks
  yvette.agostini@gmail.com

## 1.2 Project Goals and Scope

The goal of this engagement was to review the design of Globaleaks 2.0. This included reviewing the provided documents (listed below) for security risks, privacy leaks and potential design flaws such as:

- Improper use of Tor
- Anonymity leaks by various technologies used by GlobaLeaks
- Exposure of confidential data to unauthorized actors
- Weaknesses in protections from common web application vulnerabilities
- Unidentified threats
- Unidentified risks of design

iSEC was provided with the following documents during the project:

- "GlobaLeaks Application Security Design and Details," Release 0.1
- "GlobaLeaks Threat Model and Security Design," Release 0.1
- A basic architecture diagram titled "image0001.png."
- Responses to questions midweek: "iSec GlobaLeaks Questions 1."

# 2 Executive Summary


| Application Summary | |
|---|---|
| Application Name | Open Whistleblowing Framework |
| Application Version | 0.2 |
| Application Intent | Open-Source Whistleblowing Platform for media organizations, activist groups, corporations and public agencies. |

| Engagement Summary | |
|---|---|
| Dates | February 4, 2013 – February 8, 2013 |
| Consultants Engaged | 2 |
| Total Engagement Effort | Two person weeks |
| Engagement Type | Design and Arcitecture Review |
| Testing Methodology | White Box |

Sections included within this document:

| Title, Section and Contents |
|---|
| Architecture Overview within section 4 contains a description of Actors, Assets and Components. |
| Scenario Diagrams within section 5 include diagrams for the Scenarios and Actors. |
| Design Concerns within section 6 identify Risks, Concerns and Recommendations. |
| Design Recommendations within section 6 contains recommendation for the design. |
| Attack Trees within section 8 for DoS attacks, Anonymity and other Components. |

# 3 Project Summary

iSEC Partners was engaged by GlobaLeaks, to perform a security review of the upcoming v0.2 open whistleblowing platform. This was performed as part of a wider effort funded by Radio Free Asia (RFA) and delivered by iSEC Partners. Work was primarily performed by two iSEC consultants over the period of a single week, although two additional consultants joined in for additional review without cost to GlobaLeaks or RFA. iSEC was initially provided with a design and threat model documents for review, in addition to a basic system diagram. The iSEC team then formulated some initial questions and potential suggestions which were sent over to GlobaLeaks early in the week. Responses to these questions were provided by the GlobaLeaks team, helping to clarify the understanding of the product and gauging the acceptability of proposed suggestions; a very helpful contribution in the overall effort. Further information about the scope of this project can be found in section 1.2 on page 7.

It should be noted that several core GlobaLeaks architecture technologies were not adequately defined or yet developed, such as the GlobaLeaks client (GL Client) deployment, Receiver notification process and Node administration. Several core issues powering the GlobaLeaks design were also out of scope for this assessment. This includes technologies such as Tor and Tor Hidden Services (Tor HS) or scenarios which require out-of-band authentication of Receivers and out-of-band communication of essential information such as the location (onion address) of an unpromoted GlobaLeaks Node.

## 3.1 Core Findings

Overall, the security design is robust and threat model is accurate. Strong input validation, authentication and access control related considerations are also present within the architecture. iSEC security engineer Tom Ritter has developed a "cheatsheet" for auditing technologies[1] such as GlobaLeaks and was helpful in this assessment.

Tor and Tor location-hidden services provide for Anonymity, a key concern in the platform. These technologies,[2] while still considered experimental, offer a widely deployed and well developed anonymity platform. Several risks were discovered within the design the GlobaLeaks, although they cannot be easily fixed due to stated goals. In an ideal design, no single actor within the architecture can subvert the overall security nor should the anonymity of all actors rely upon the anonymity of any *single* actor within a GlobaLeaks node.

GlobaLeaks defines several scenarios which modify the security and anonymity requirements of the overall technologies and for each of the actors involved. This ranges from simple municipality related reporting to human rights violations whistleblowing. The risks and threats involved vary from none at all to severe forms of violence. While these configuration scenarios afford GlobaLeaks greater overall use and make the GlobaLeaks platform more recognizable, the various scenarios may detract from the trust Whistleblowers place in GlobaLeaks.

Whistleblowers and Receivers may not understand the difference in protections provided by TLS versus unpublished Tor HS onion addresses which provide anonymity. GlobaLeaks may consider alternative branding for low or high security scenarios. This alternative branding could also force configuration options to be in place for given and selected scenario during install time, helping to prevent from misconfiguration. Such misconfiguration is a significant risk in a high security scenario.

[1] https://github.com/iSECPartners/LibTech-Auditing-Cheatsheet/

[2] It should be noted that Tor does *not* claim to protect from an adversary which has the capability of end-to-end correlation.

# 4 Architecture and Design Overview

## 4.1 Actors

Actors represent the various direct participants in the GL system. Each has different requirements for anonymity depending on the circumstances of the Tip/Leak and the scenario in which the GlobaLeaks node is deployed.

**Whistleblower:** Has information (A "Tip") that needs to be shared. This could be a corporate or political leak, a street pot-hole notification or a human rights violation video. In many cases, the Whistleblower will remain anonymous.

**Receiver:** Someone who receives tips from Whistleblowers and acts upon them in some way. This could be a media reporter, government transparency advocate or other interested party. The Receiver may or may not reveal their identity to the Whistleblower depending on the scenario or configuration.

**Node Admin:** Manages the technical operation of a GlobaLeaks Node. This actor may or may not have a pre-established relationship with the Receiver, but is required to perform Node setup, authenticate receivers out of band (via unknown methods) and setup Receiver notifications. Per documentation, the GlobaLeaks node admin is trusted with all Tip data.

## 4.2 Assets

The various types of data contained in the GlobaLeaks backend. Data is documented as being stored in plaintext in addition to possible PGP encrypted data depending on the configuration or keys provided by Receivers.

**Tip:** Information to be shared by a Whistleblower to a Receiver. This may include text, images and other uploaded files such as Zip archives and PDF documents. Tips are stored in plaintext.

**Receipt:** An access token given to a Whistleblower to view tip status after submission. Receipts are hashed and salted form using scrypt.

**Authentication Data:** Usernames and passwords are used as credentials to establish the identity (or pseudonymity) of a Receiver or GlobaLeaks Node Admin, no other forms of authentication (other than Receipt ID for the Whistleblower) is supported. Whistleblowers use Receipts as a weak form of re-authentication, with a default being a random 10 digit number. No authentication is required for Tip submission. Passwords are stored as salted hashes using scrypt.

**Notification Method:** An email address or SMS message used to contact a Receiver and inform them of a new Tip. There are no documented plans for using this notification method as a form of two-factor authentication. Note the notification is a weakness in itself for anonymous Receivers attempting to avoid identification.

**Encryption Keys:** PGP public keys belonging to a particular Receiver can be used to encrypt communications to the selected Receiver. Ideally after a PGP key is specified, all communication, including notifications is encrypted with the provided key.

## 4.3 Activities

Typical activities on a GlobaLeaks node from the perspective of Actors and Assets. Activities are listed in a typical chronological order, but are not required to occur in this order.

**Setup:** GlobaLeaks Node Admin configures the site, which must be done at least once and may be repeated later. No other activities are possible until this is performed and possibly until at least one Receiver is configured.

**Add Receiver:** GlobaLeaks Node Admin adds Authentication Data and Contact Information for a new Receiver.

**Submit a Tip:** A Whistleblower submits a Tip for viewing by a selected set of Receivers or simply a context where the Receiver is hidden, depending on the configuration. The Whistleblower receives a Receipt in a format defined by the GlobaLeaks node admin.

**Check Tip Status:** A Whistleblower authenticates using the Receipt. The GL Node displays data about how many times the Tip corresponding to the Receipt has been accessed.

**New Tip Notification:** The GlobaLeaks Node sends a notification to one or more Receivers. The notification is delivered to the Receiver's method of choice (currently only email address at the time of this writing) which is configured by the GlobaLeaks node admin. This data is *optionally* encrypted with the Receiver's PGP Key if present.

**Receiver Authentication:** A Receiver signs in to the GlobaLeaks Node by providing correct Authentication Data (username and password).

**View Tip:** After Receiver Authentication, the Receiver can view Tips assigned to them.

**Extend Tip Deadline:** After Receiver Authentication, the Receiver can optionally extend the expiration date of a Tip (depending on configuration).

**Delete Tip:** After Receiver Authentication, the Receiver can optionally remove a Tip from the system (depending on configuration).

## 4.4 Scenarios

Several types of use are anticipated for GlobaLeaks Nodes. Each type requires different levels of security and anonymity for the Assets and Actors in the system. See GlobaLeaks documentation for further information.

**Media Outlet:** A public media organization allows anonymous Whistleblowers to share Tips with known journalists.

**Corporate Compliance:** A corporation internally allows insiders to share Tips with an internal audit organization. The insider Whistleblowers may be anonymous or not at their own discretion.

**Government Tax Whistleblowing:** A known government agency solicits information about tax evasion, etc. Receivers are not public (officials or auditors at the government agency), but Whistleblowers must identify themselves to claim a reward.

**Human Rights Activism Initiative:** A human rights organization in a hostile situation sets up a GlobaLeaks Node to allow reporting of human rights abuses. The Receivers and the Whistleblowers are both anonymous (Receivers may be pseudonymous).

**Citizen Media Initiative:** A lower-risk version of the Human Rights Activism Initiative, where the group operating the GlobaLeaks Node is less concerned with retaliation against Whistleblowers and Receivers. Receivers are pseudonymous and Whistleblowers are optionally anonymous

**Local Municipality "Street Hole" Reporting Service:** A local government wants an easy way to allow reporting of issues by citizens. Neither the Receiver nor the Whistleblower are required to be anonymous and their identities may be fully disclosed.

## 4.5 Core GlobaLeaks Components

Various software modules in a given GlobaLeaks Node. These may or may not be running on different physical or virtual hardware.

**GlobaLeaks Client (GL client):** A JavaScript front-end to allow interaction with the GL Node. This may be embedded into another site (For example, a Media Outlet puts the GL Client into their public website). The GL Client communicates to the GL Backend via REST calls which may or may not be performed via Tor HS depending on the scenario. A future GL client may consist of a browser plugin.

**GlobaLeaks Backend (GL backend):** Performs storage, authentication, notification and other communications between the Actors via the GL client. Actions are performed via JSON calls to a RESTful API provided by a python application server. Access is primarily over Tor HS although some configurations allow the use of tor2web or possibly SSL for confidentiality (without anonymity).

## 4.6 GlobaLeaks Privacy and Anonymity

The chief anonymity provider for a GlobaLeaks Node is Tor. Tor allows for a user of the Tor Network to connect anonymously to sites on the internet, and is chiefly designed to prevent the possibility of data sender or receiver identification. A related technology, Tor *Location*-Hidden Services (Tor HS), is used to allow encryption and authentication of the GlobaLeaks Node to the user without relying on a central certification authority. This also prevents the disclosure, identification or location of either the Tor HS or GlobaLeaks actor. A full description of Tor and Tor Hidden Services is available at:
`https://svn.torproject.org/svn/projects/design-paper/tor-design.html`

Transport Level Security (TLS) connections are also available for scenarios requiring confidentiality, but not anonymity. This can be paired with Tor2Web, a technology that allows users not on the Tor network to access Tor Hidden Services. Use of a Tor2Web connection is *not* anonymous and cleartext data could be observed by the Tor2Web node (especially if the SSL is terminated at this point).

To help protect anonymity, random delays between user requests and system actions will be added. These will help combat time and data correlation attacks which Tor cannot protect against and provide defense in depth against data and timing correlation.

A final aid to users for understanding their anonymity is provided via a "Anonymity Badge" this color and symbol coded indicator will show the user their current anonymity level with respect to GlobaLeaks. This requires a to-be-developed feature of Tor, and for the time being requires looking up a GlobaLeaks node or making a connection to a known good Tor HS.

## 4.7 GlobaLeaks Security and Authentication

The GlobaLeaks project has several security and authentication plans in place. The project will be following the OWASP guidelines for REST Web Services:
https://www.owasp.org/index.php/REST_Security_Cheat_Sheet

There is a plan in place to provide limited filtering of uploaded files to attempt to prevent transmission of malicious files to Receivers. Such a plan must highlight the risks involved with the receipt of possibly malicious files (urging Actors to take the appropriate action) and encourage simple data formats over complex ones (PDF, docx).

Receivers are to be authenticated with a username and password. To attempt to prevent password brute-force, a GlobaLeaks Node will begin adding a response delay and a CAPTCHA when 20 failed login attempts on any account occur within 120 seconds. An account is locked out if 5 failed authentication attempts occur. When this happens, the Receiver is contacted via the entered Contact Method and sent a URL to reset the password. Note this contact email should be sent after a random delay to avoid possible identification via data and time correlation.

A password strength validator, safe password practice information and minimum complexity and length requirements will be used to ensure selected passwords are strong. Passwords are hashed with scrypt[3] and a per-user 128 bit salt.

[3] https://www.tarsnap.com/scrypt.html

# 5 Scenario Diagrams

## 5.1 Whistleblower

Whistleblowers are expected to access a GlobaLeaks server node either as a Tor Hidden Service (Anonymous) or through a Tor2Web proxy (Confidential). Access is performed through a GlobaLeaks Client, here represented as a plug-in though this is considered a future development scenario. Communications between the Whistleblower and the GlobaLeaks server node are performed over JSON.

Figure 1 shows the flow for a Whistleblower accessing the GlobaLeaks server node as a Tor Hidden Service. This provides strong confidentiality and anonymity. Figure 2 shows the flow for a Whistleblower access the server via a Tor2Web proxy. This scenario doesn't provide effective anonymity and introduces a trusted intermediary. The Tor2Web server is able to see all traffic unencrypted.

1. **Submit a new tip.** Whistleblowers are able to submit new tips to the GlobaLeaks server, optionally with the ability to choose individual receivers.
2. **Retrieve status of a previously submitted tip.** Whistleblowers are able to retrieve the status of a previously submitted tip by supplying the receipt. They should not be able to retrieve or modify the tip itself.
3. **View and add comments on a submitted tip.** Whistleblowers must be able to review and add additional comments on a submitted tip. Access is again granted by supplying the receipt.

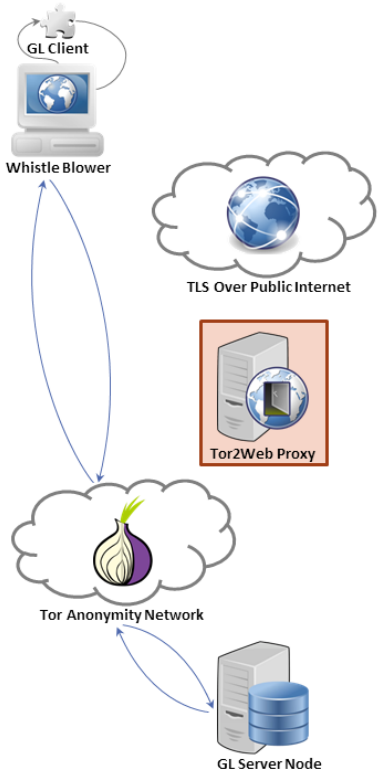



Figure 1: Anonymous Whistleblower

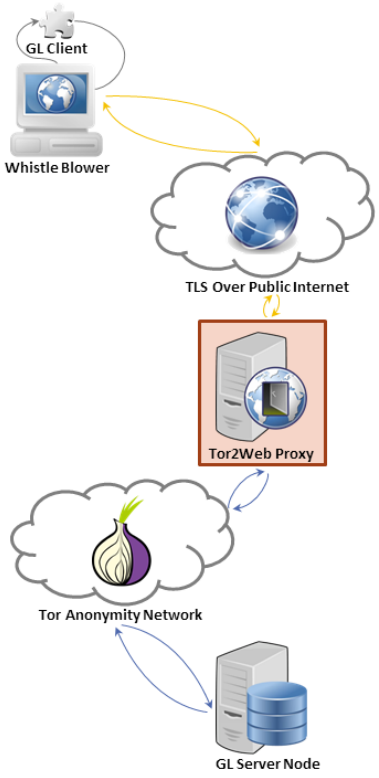



Figure 2: Confidential Whistleblower

## 5.2 Receiver

Receivers are expected to access a GlobaLeaks server node either as a Tor Hidden Service (Anonymous) or through a Tor2Web proxy (Confidential). They also receive notifications from the GlobaLeaks server via GPG encrypted email. Receivers also access the server node via a GlobaLeaks Client. They must authenticate themselves to the server before they are allowed to view the tips which they have been assigned.

Figure 3 shows the flow for a Receiver accessing the GlobaLeaks server node as a Tor Hidden Service. This provides strong confidentiality and anonymity. Figure 4 shows the flow for a Receiver access the server via a Tor2Web proxy. This scenario doesn't provide effective anonymity and introduces a trusted intermediary. The Tor2Web server is able to see all traffic unencrypted. In both cases, GPG encrypted notification emails are assumed to traverse the Tor network up until the mail server after which they will be delivered over the Internet.

1. **Receive a notification that they've been assigned a tip.** Such notifications will be sent encrypted to a pre-configured address. Notifications will contain neither the tip nor associated metadata.
2. **Retrieve a tip for which they've been assigned.** After authentication, receivers should be able to retrieve a list of assigned tips and then retrieve the contents of a specific tip.
3. **View and add comments on a submitted tip.** Authenticated receivers must be able to review and add additional comments on a submitted tip.

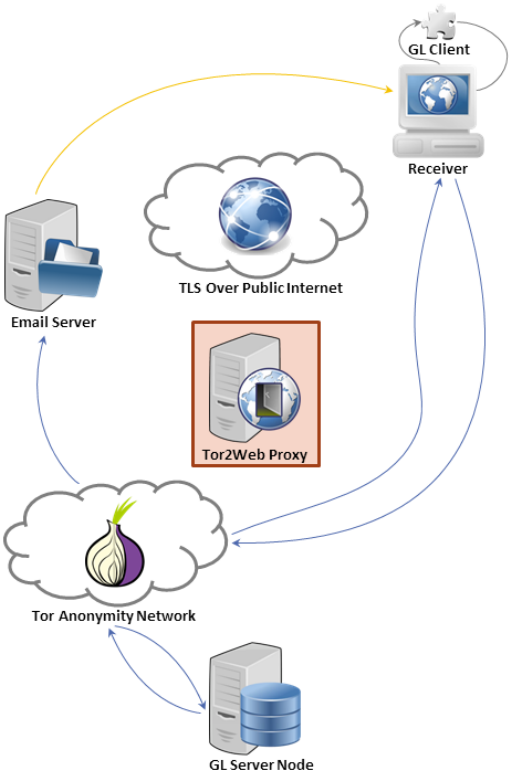



Figure 3: Anonymous Receiver

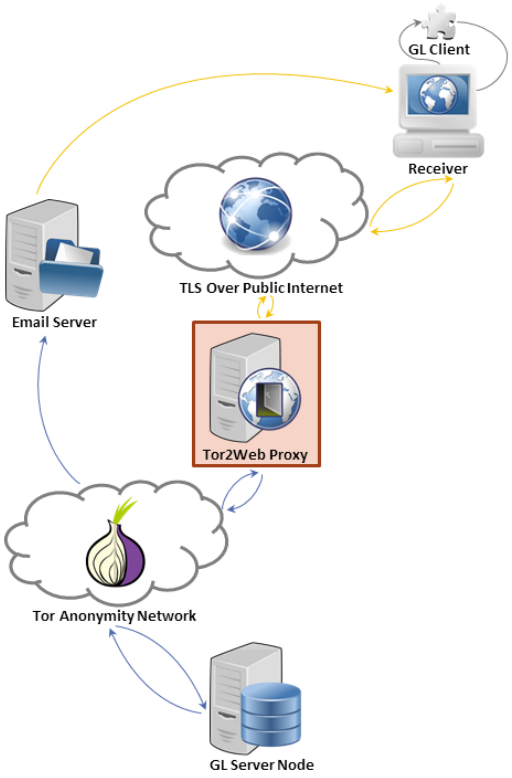



Figure 4: Confidential Receiver

## 5.3 Admin

Admins should *only* access their GlobaLeaks server node as a Tor Hidden Service (Anonymous) in deployment scenarios where there is concern regarding seizure of the server. As before, Admins also access the server node via a GlobaLeaks client instance. They must authenticate themselves to the server before they are allowed to undertake administration tasks including configuration and user provisioning.

Figure 5 shows the flow for a Receiver accessing the GlobaLeaks server node as a Tor Hidden Service. This provides strong confidentiality and anonymity.

1. **Manage Admin and Receiver accounts.** An Admin user is responsible for provisioning Receiver and Admin accounts in the system. They should perform some degree of out-of-band verification to prevent the introduction of malicious actors.
2. **Change configuration settings.** Such settings include tip expiration.
3. **Remove tips from the server.** In some instances this capability may be delegated to trusted Receivers.

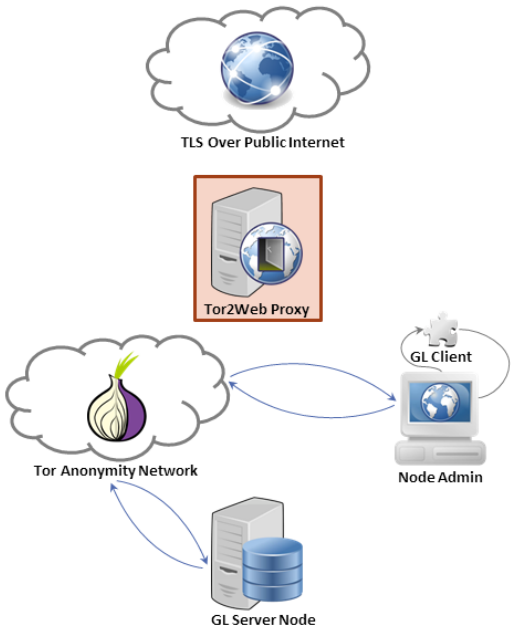



Figure 5: Node Administrator

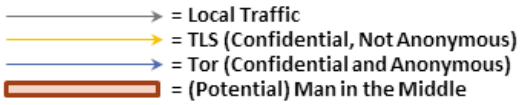



Figure 6: Legend

# 6 Design Risks and Concerns

These concerns address privacy, security and anonymity for actors with various GlobaLeaks scenarios and components. Recommendations are provided within each concern, in addition to more general recommendations discussed in section 7 on page 20.

## 6.1 Actors

Malicious GL administrators can subvert the system and possibly discover identities of system actors. While the GL node admin is only trusted with data security, in several cases they may be able to discover Whistleblower and Receiver identities by means of browser attacks or documented Tor weaknesses. This risk must be an accepted and documented for unestablished relationships. One example is through known Tor weaknesses[4] .[5]

Anonymity is available for the Whistleblower, while the Receiver is provided little to none in at least two scenarios. The GL node admin must setup Receiver accounts, which requires the Receiver to use a number of other methods to remain anonymous. Additionally, concerning as iSEC understands the system, is the fact that the Receiver list is exposed to unauthenticated and anonymous parties. Consider adding a "Receiver ID" nickname which is anonymous but provides some indication of identity (*e.g.* NYTimesReporter####) to the Whistleblower, assuming such information can be provided out of band or prior to submitting the Tip.

Node administrators are considered trusted for all Tip data. There are possible use cases where a technically competent administrator may want to set up and manage a GL node, but does not want to have access to the tip data. Consider an optional system where the receivers can publish public keys via an out of band channel and use them to receive encrypted tips. This provides plausible deniability for the GL node admin, data protection and protection from any malicious GL admin. An additional method could be to use JavaScript and OpenPGP to encrypt data within the GL client to aid non-technical Whistleblowers or Receivers.

## 6.2 GL Client

Remove username autofill. Automatic input of the username suggests storage of identifying information on the client side (possibly as part of the notification). This may be used to associate GL Receivers with Tip data or discover usernames if the Notification method is subverted. Any forensic analysis of a Whistleblower computer could reveal information that can link an individual to their GL node username and activity. *Note* after a discussion with GlobaLeaks develoepers, iSEC learned the username will be a random identifier and Receivers' anonymity will not be reduced by such a disclosure (according to statements by GlobaLeaks).

Privacy Badge fails open. As confirmed by GlobaLeaks, if the Whistleblower is mistakenly not using Tor when the badge is viewed (This is undefined) it will leak the address of the possibly undisclosed hidden service in addition to leaking the Whistleblower's location and intent. *Note* after a discussion with GlobaLeaks developers, the Privacy Badge will be more widely deployed and only perform a lookup

[4] https://en.wikipedia.org/wiki/Tor_(anonymity_network)#Weaknesses
[5] http://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-gregory_fleischer-attacking_tor.pdf

to an innocuous onion address (such as check.torproject.onion) rather than the intended GlobaLeaks backend.

GL client may be deployed or embedded into insecure Web Server. Until the GL client is designed as a browser extension, ideally building on the Tor Browser Bundle, it must be provided as a web application. This application may be embedded into other web applications or provided from the GL node itself. Simply put, this reduces the overall security of the GlobaLeaks architecture to the security of the web application or server in which it is deployed. If it is not provided directly from the GL node via a Tor HS, it may be modified by a compromised server. One example could be a news agency providing an embedded tip interface, but an adversary or coercive government hijacks such a GL node pointer or GL client code to point to a malicious GL node or otherwise attempts to identify Whistleblowers. Finally, the parent site may be subject to Denial of Service (DoS) attacks in order to prevent Whistleblowers from accessing the embedded GL client.

## 6.3 GL Backend

GL backend should only be accessible via a Tor Hidden Service. Other methods of access could leak identifying information about the Whistleblower or Receiver. A potential user of GL should be able to have strong assurances of anonymity that are not reliant on the configuration settings of a particular GlobalLeaks node.

Setting CORS `Access-Control-Allow-Origin` to * is incompatible with cookie based session management. The CORS headers are proposed to be set to * to allow for the GL Client to be deployed separately from the GL Server backend. Since there is no anonymous content hosted at a GL Server, this would seem to necessitate the use of the `Access-Control-Allow-Credentials` header if cookies were used for session management. Unfortunately, this would let malicious sites make requests to the GL Server and extract confidential data. Either sessions must be managed separately from browser provided mechanisms and manually inserted in every request *or* the `Access-Control-Allow-Origin` header must not be set to * and cookies can be used for session management. One proposal would be to copy the user's provided Origin into their Session when they provide their credentials. All further requests with a given session ID would set `Access-Control-Allow-Origin` to this stored Origin value.

The CSRF prevention mechanism is subject to minor weaknesses. The described solution for CSRF prevention requires that a submitted cookie and a HTTP header have identical values before state changing actions are performed. These items are only compared to each other; not against anything stored on the server. Additionally, since the GL Server node will be deployed with a CORS policy of *, any site can submit XHR requests to the GL Server with headers of their choice. The end result is that the protection relies upon an inability of attackers to control the value of the CSRF prevention cookie. This is provided when the service is accessed over Tor or via the HSTS header. If the HSTS header was misconfigured, an active network attacker would be able to set arbitrary cookies. To prevent CSRF, even in this case, bind the CSRF value to the session. For example, `HMAC-SHA256(<server-key>, <SessionID>)` although this is unnecessary if sessions are not managed through cookies.

Receiver notifications may allow traffic or time correlation attacks. Random delays should be required for any such notifications. If notifications are sent over Tor but to a monitored email account or mobile phone, an attacker may be able to discover a link in communication between the GL admin and their GL node or the GL Receiver and their GL node.

Default receipt settings are weak and may not provide plausible deniability. The current default for Whistleblower receipts is a 10 digit number. 10 digits does not provide sufficient key space to prevent dedicated adversaries from quickly enumerating all possible receipts. This could be used to enumerate all existing tips in the system, gather usage metrics, and discover receipt numbers in use (to then search for in other mediums). An attacker could also attempt to actively deny use of the node by submitting false tips until the receipt key space is exhausted.

Adversary discovery of something that appears to be a US phone number might raise suspicions on its own in some contexts. These could also easily be tested for plausibility, as many 10-digit numbers are not valid phone numbers.

iSEC recommends a default receipt system where the node selects four random words from a large dictionary file. English is a plausible default, but allowing node administrators to select a dictionary file would allow this approach to be easily applied to other languages. If GL includes internationalization options in its install/configuration process, an appropriate default language file could be selected by the system at installation.

# 7 Design Recommendations

## 7.1 Anonymity and Confidentiality

All HTTP requests should be performed via HTTP POST. GET requests should be avoided because sensitive data in the requests may be stored in browser history, web server logs, and proxy logs.

Allow the Whistleblower or GL Node Administrator to disable receipt functionality. Whistleblowers may not want any identifying information returned and should be able to perform a single POST with their tip information.

Warn actors about the threat that tor2web and similar Tor proxies pose to anonymity and confidentiality. Suggest using Tor for communication with GL Nodes, except in low security scenarios where convenience is more important.

Consider scenarios in which Node Administrators cannot be trusted with Tip data or want plausible deniability from such data (although this may be impossible until out of band encryption is established). Whistleblowers and Receivers may need to keep information secret from Node Administrators. Additionally, Node Administrators may not want the ability to view Tip data so they have plausible deniability in the event that their Node is seized or compromised, or to reduce the threat of physical violence. Encourage Whistleblowers and Recipients to encrypt messages with PGP and to share public keys out-of-band. Consider functionality for the CL client to enable easy encryption and decryption, or provide simple instructions for setting up PGP clients and encrypting messages.

Clearly warn actors that Tips, follow-ups, and other data can be viewed by the Node Administrator. Strongly encourage actors to encrypt their submissions if this is not an acceptable risk.

## 7.2 GL Client

Document that the GL client browser plugin does not perform logging. While the threat model does not intend to protect against attackers that have physical access to the client, logs would add an extra and unnecessary risk. Logging should not be a function for end-users of the GL client.

Provide a means to wipe all traces of the GL client when it is deployed as a browser extension. Whistleblowers may need to remove all traces of the GL client if they anticipate a malicious party seizing their device. Implement a mechanism to securely wipe the client within the browser extension, or provide instructions for doing it with a third-party application.

Redesign the session management and XSRF protection scheme. The GL architecture requires that the client can be hosted on a different domain than the GL backend. To allow this, the CORS policy must permit read and write access from any domain. This means that arbitrary domains could access content in the context of authenticated sessions if the session token is automatically appended as a cookie. Session management can be simplified and made more secure by not relying on the browser to automatically send such token cookies. Instead, manually append the session token as a custom HTTP header in each request. Consequently, this would remove the need for XSRF tokens because an attacker's domain would not be able to coerce the browser into automatically submitting the token cookie.

Only use safe JSON parsing methods and attempt to avoid any content from third-party domains within

the GL client. Malicious JSON could allow for XSS if it is not safely handled by the client. Avoid using `eval()` to parse JSON. Use safe alternatives such as `JSON.parse()` or `jQuery.parseJSON()`.

Ensure that output is encoded in a context-aware fashion. Cross-Site Scripting can be difficult to protect against due to different contexts in which malicious code can be executed. Ensure that proper encoding is applied on untrusted input in the context of HTML, JavaScript, URLs, and other contexts. Leverage the use of a vetted third-party library for XSS protection.

## 7.3 GL Backend

Accept and clearly document the risk that malicious documents may be sent to Receivers. No reasonable technical solutions can be deployed which will prevent malicious documents (such as those which contain "0day" exploits) from being transferred to Receivers by malicious Whistleblowers. As the system is open, any dedicated attacker can simply test and modify their exploit until it bypasses all security measures such as attempts at anti-virus and safe file extensions. Documentation should encourage opening the file, of any type, within a virtual machine without network access using the latest version of the associated reader or viewer. Attempts to parse, sanitize or otherwise secure the content all increase the attack surface of the GL node and may allow attacks.

Tip information is encouraged to be plaintext or in simple data formats. This has a dual purpose of both preventing metadata leaks and reducing the attack surface for targeted document readers and image viewers.

Ensure that that usernanes are not displayed to other Receivers or Whistleblowers. If a malicious party learns someone's username, they may abuse flaws in the system to brute-force their password or trigger an account lockout. The backend can avoid displaying usernames by allowing Receivers to create a pseudonym or generating a random identifier associated with a user that is displayed to other actors when they post information. Alternatively, a Context may be displayed instead of a username, as documented for Receivers although this provides the Whistleblower with little knowledge of the end recipients.

Guarantee the authentication flow does not allow attackers to guess valid usernames. Proper precautions should be taken to make sure that the backend does not reveal the validity of a guessed username. Implement the authentication process to be done in constant time, and return generic error messages when an invalid username or password is entered.

Consider deploying public key authentication. Passwords have several flaws such as being guessable, having to be remembered, and they have to be stored in some form on the server. Authentication is more secure with asymmetric keys than passwords because it requires a combination of something you have (private key) and something you know (password to decrypt private key). In this scenario the secret is never actually exchanged; the client just has to prove knowledge of some secret (private key) similar to how SSH key authentication is performed. Finally, if public key authentication is added, use vetted libraries to deploy it and require Receivers to password protect their keys. *Note:* while this requires significant development effort, passwords are the bane of the security community and iSEC encourages GlobaLeaks to consider alternate scenarios.

Update design documentation, mandating that all write operations should be append only. Such actions include submitting tips, adding notes, and uploading files. Actors should be able to trust that malicious parties have not modified data after the fact, assuming that the Node Administrator can be

fully trusted with the data.

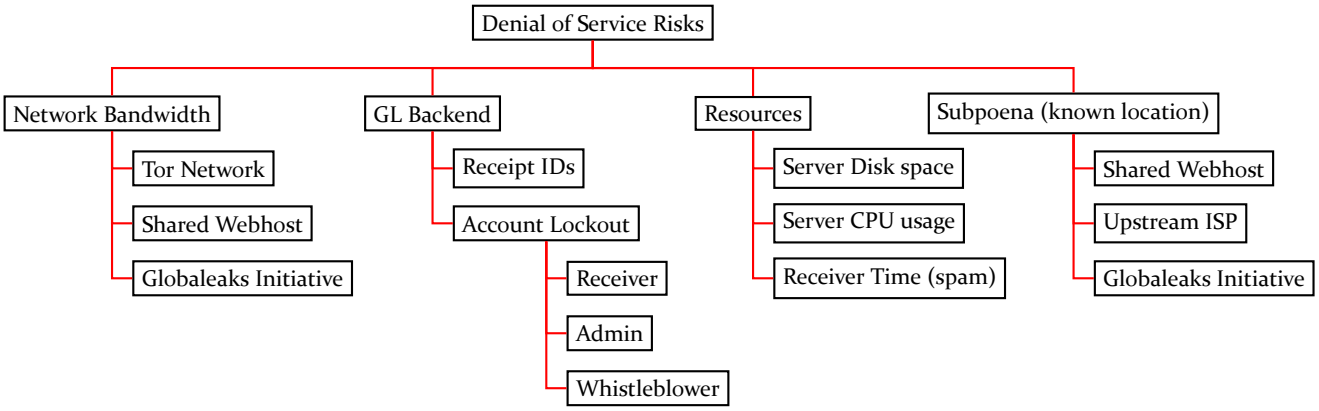
Ensure input validation is performed for Tip comments. As unauthenticated and anonymous Whistleblowers can leave comments, in addition to data, these inputs should be validated to prevent from SQL injection or other web attacks such as Cross-Site Scripting (XSS).

# 8 Attack Trees

These attack trees should be used by developers to aid in measuring security protections and gauging threat model accuracy.

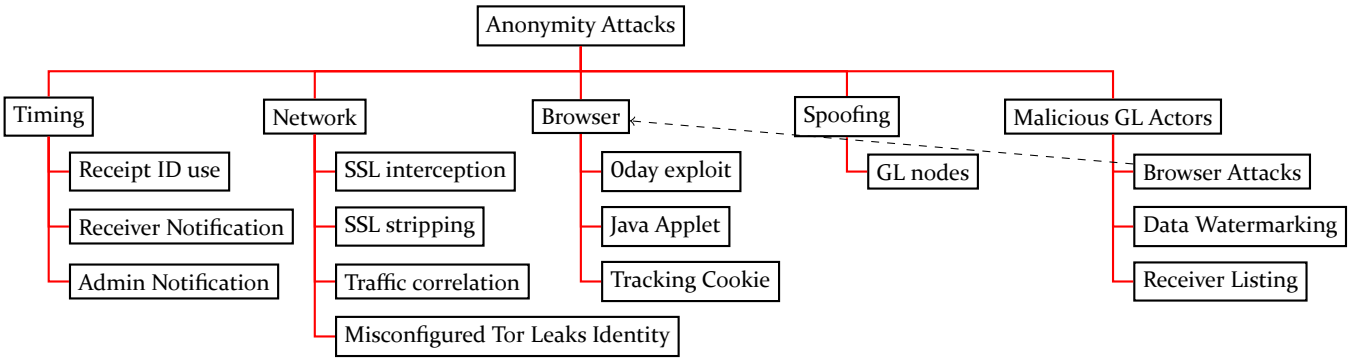
## 8.1 Denial of Service Attack Tree

Denial of Service (DoS) represents a significant risk for the platform on several fronts. Services with similar goals that hosted content themselves faced Distributed Denial of Service (DDoS) from a number of different adversaries. While Globaleaks differentiates itself from others by being decentralized, not publishing Tip information and using Tor Hidden Services (Tor HS) by default, servers still may face these threats.



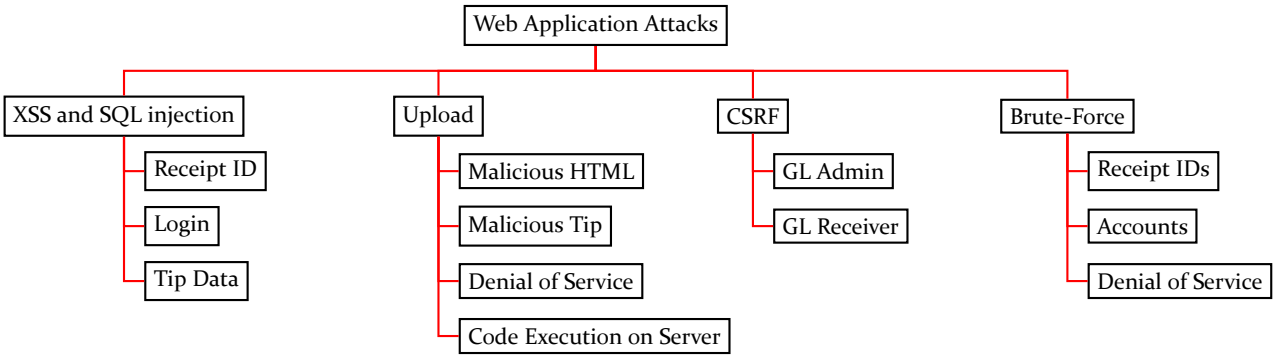

## 8.2 Anonymity Attack Tree

Attempts by state actors, malicious parties and others who wish to discover the identities of the Actors involved. While the use of Tor mitigates a large number of the attacks below, those by a malicious GlobaLeaks node admin remain possible and must be an accepted risk, even though this actor (the GlobaLeaks node admin) is anonymous.



## 8.3 Web Application Attack Tree

Web application attacks against the GlobaLeaks Client (GL Client) and GlobaLeaks Backend (GL Backend). While Globaleaks has a number of technical solutions in place, the list of public Globaleaks initiatives may come under attack from state actors, political rivals or other adversaries. Attacks will likely take the form of brute-forcing accounts, attempting to exhaust resources and injecting or tampering with content via SQL injection or stored Cross-Site Scripting (XSS).




## 8.4 Network Attack Tree

Configuration defaults for most scenarios specify the GL backend should only be accessed via a Tor Hidden Service (Tor HS). This "high security" default provides the anonymity and confidentiality protections of the Tor network and allows authentication of the server via the onion identifier. This identifier is provided out of band, and may be promoted publicly or privately. The concerns over promoting the GL client location and secure anonymous access is primarily outside the scope of this document.